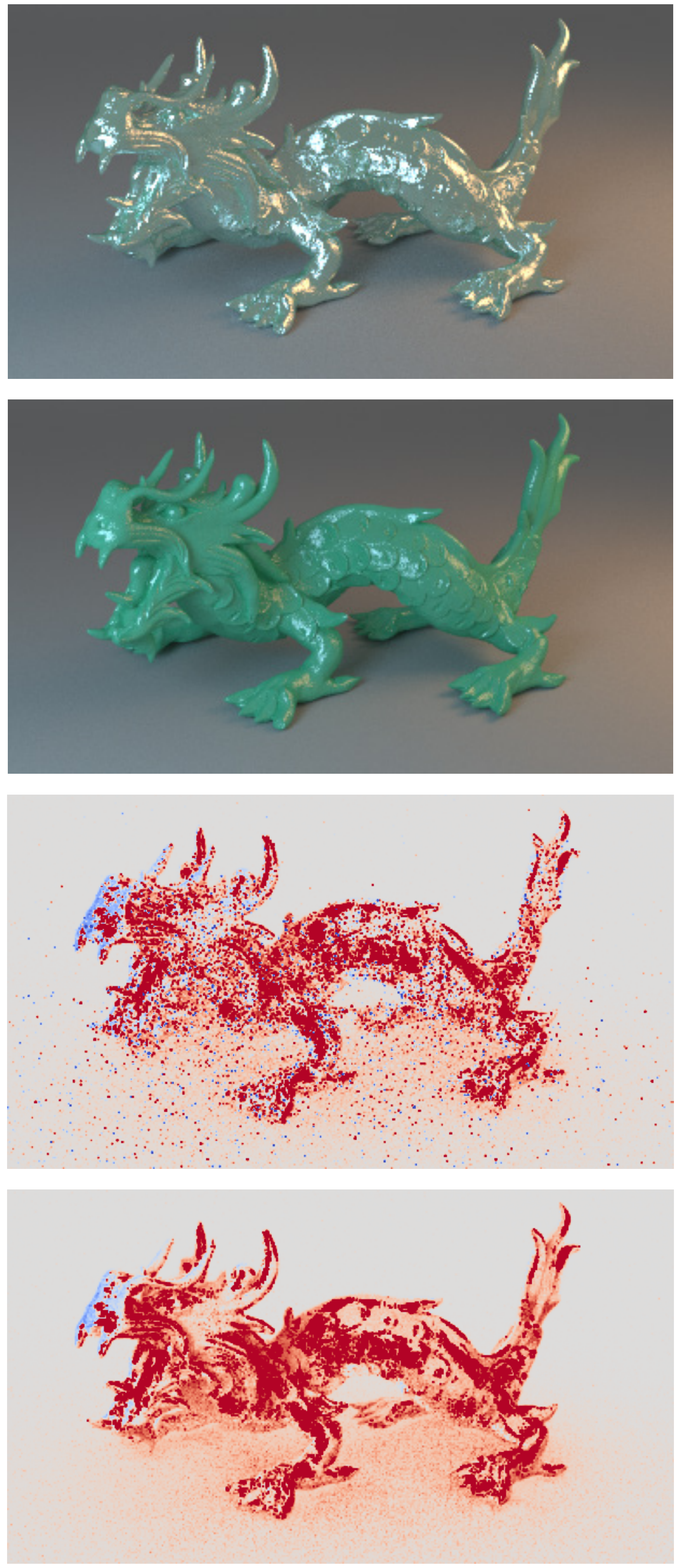
Path sampling methods for differentiable rendering

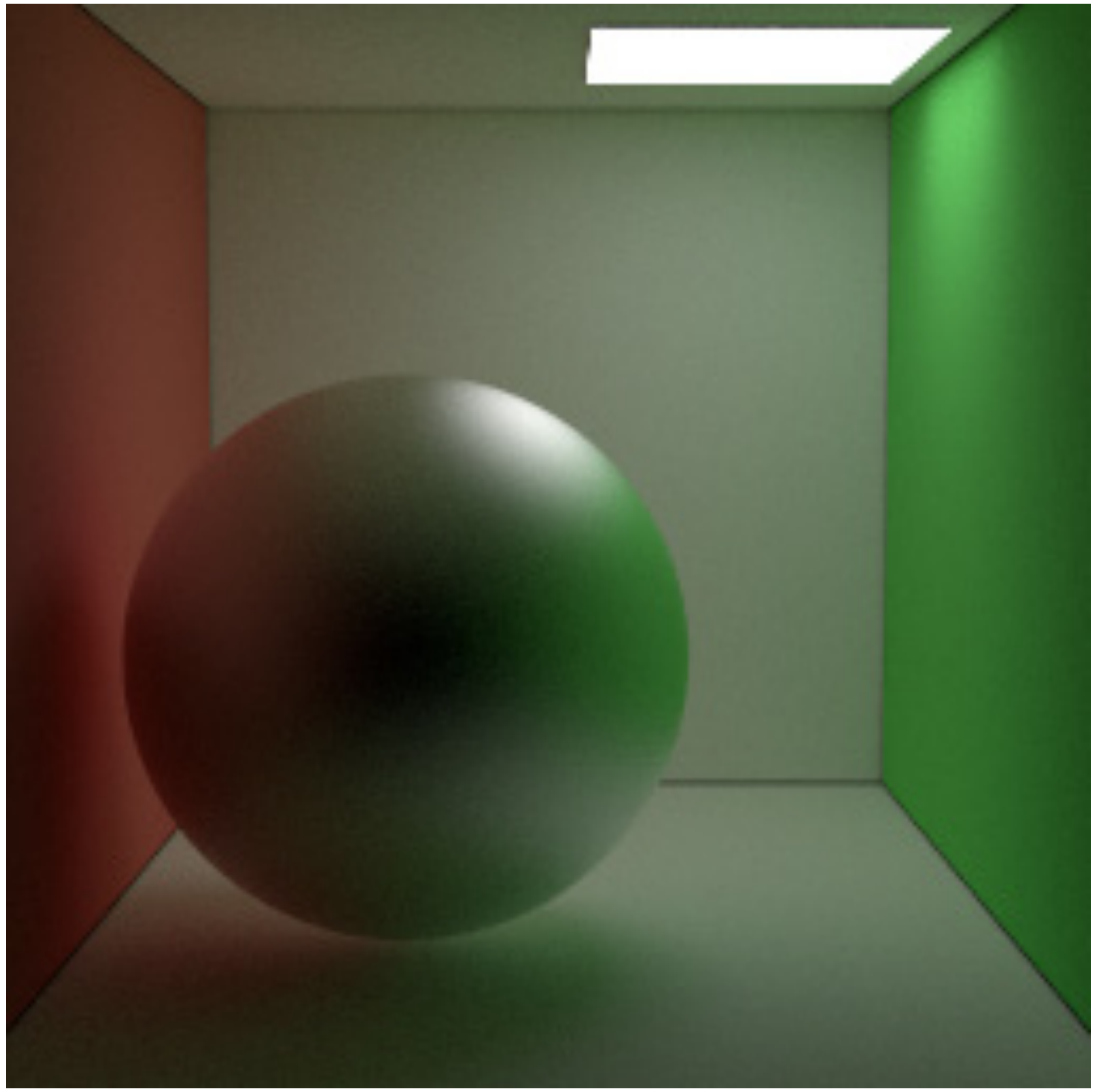
EGSR 2024

Tanli Su, Ioannis Gkioulekas
Carnegie Mellon University



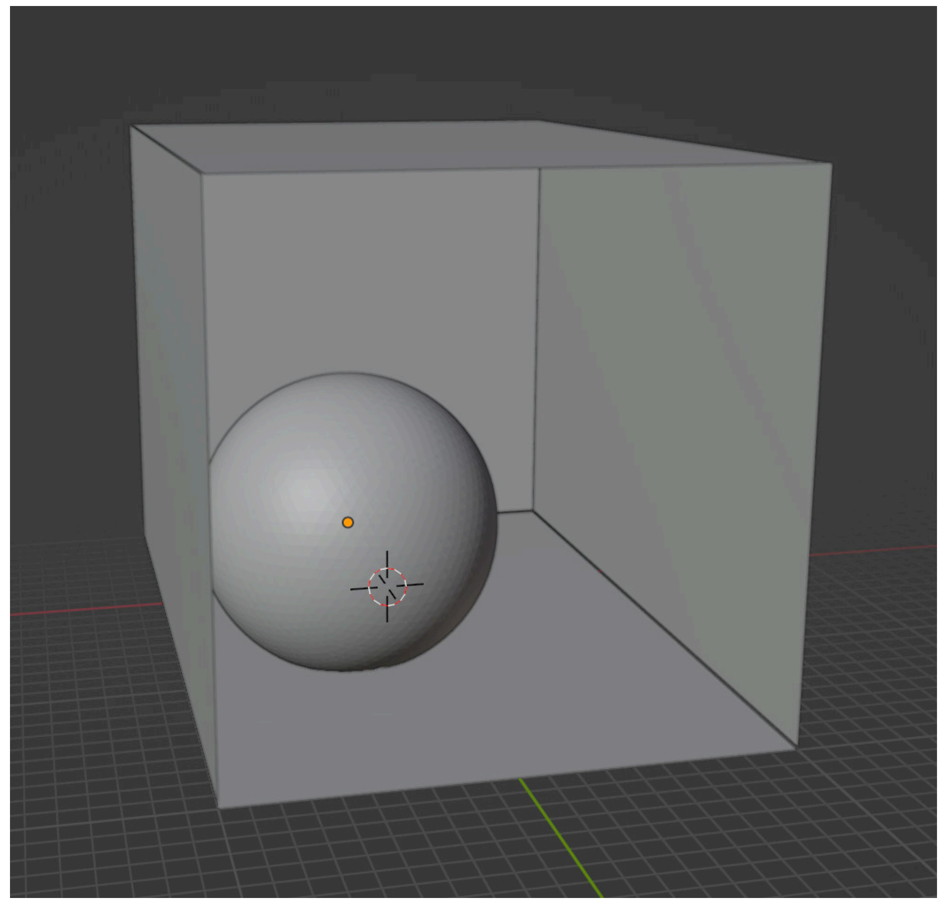
Introduction

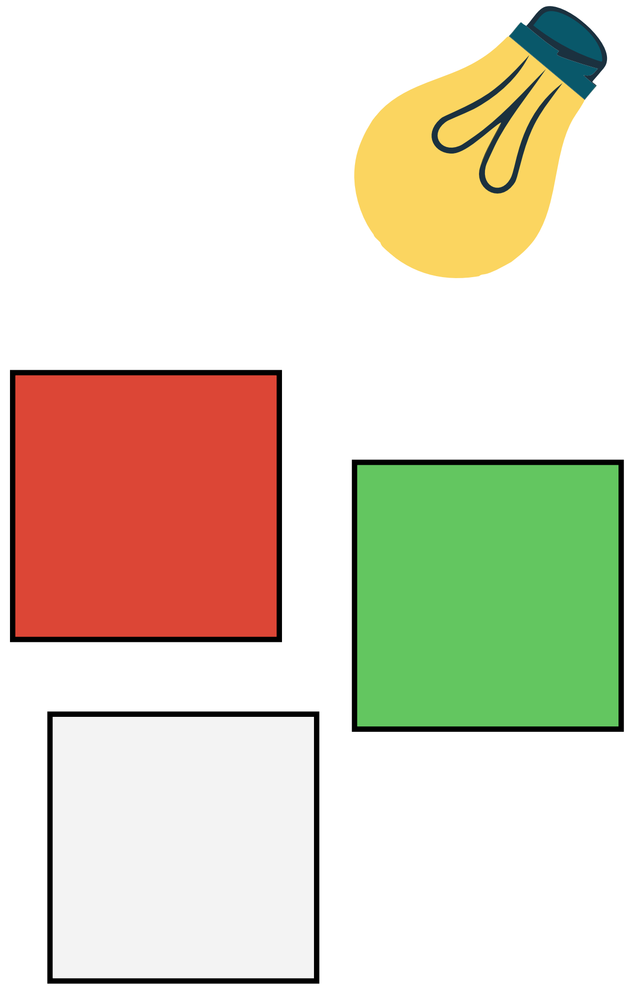
Realistic rendered image



Scene parameters

(geometry, materials, lighting, etc.)



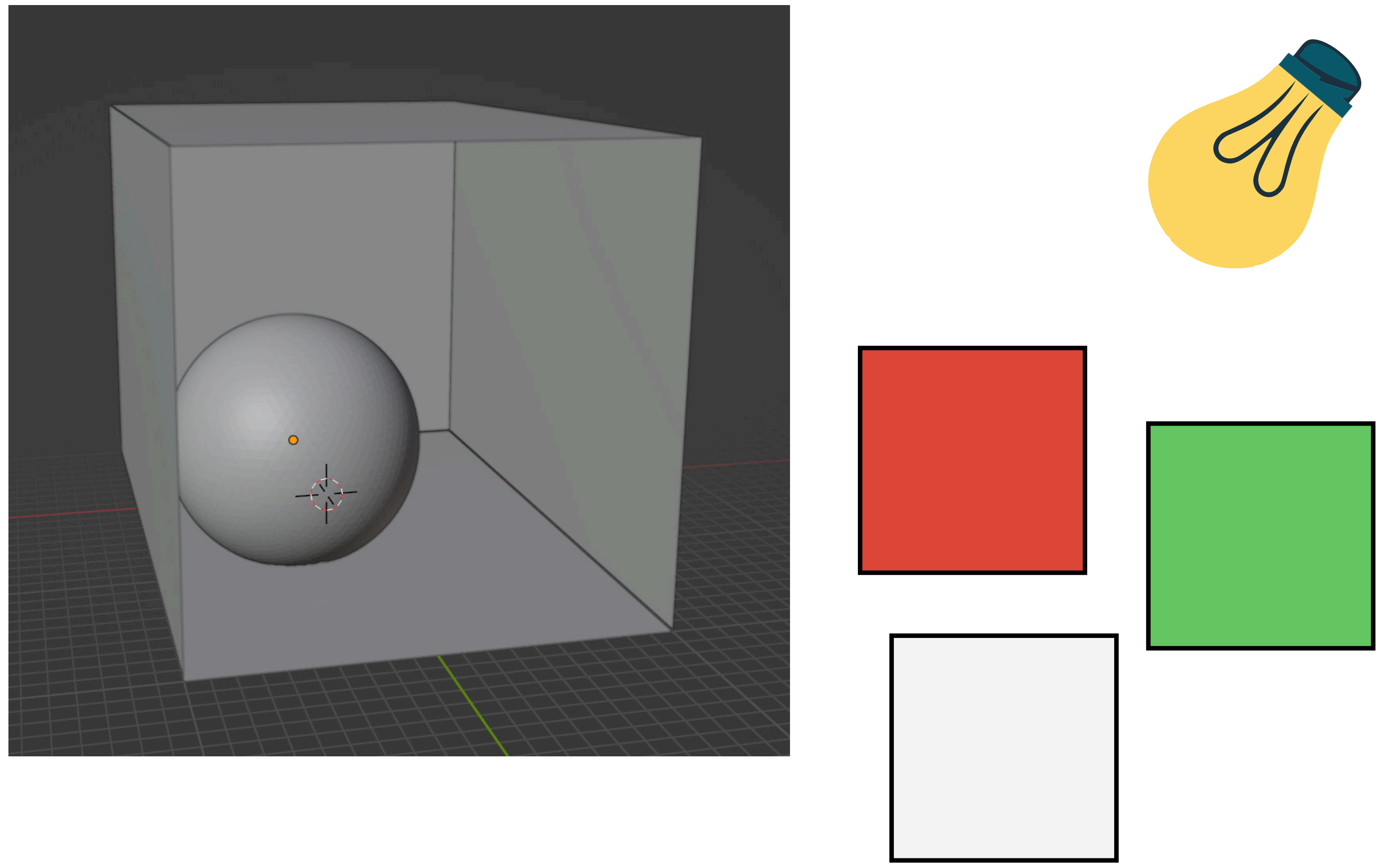


Forward rendering

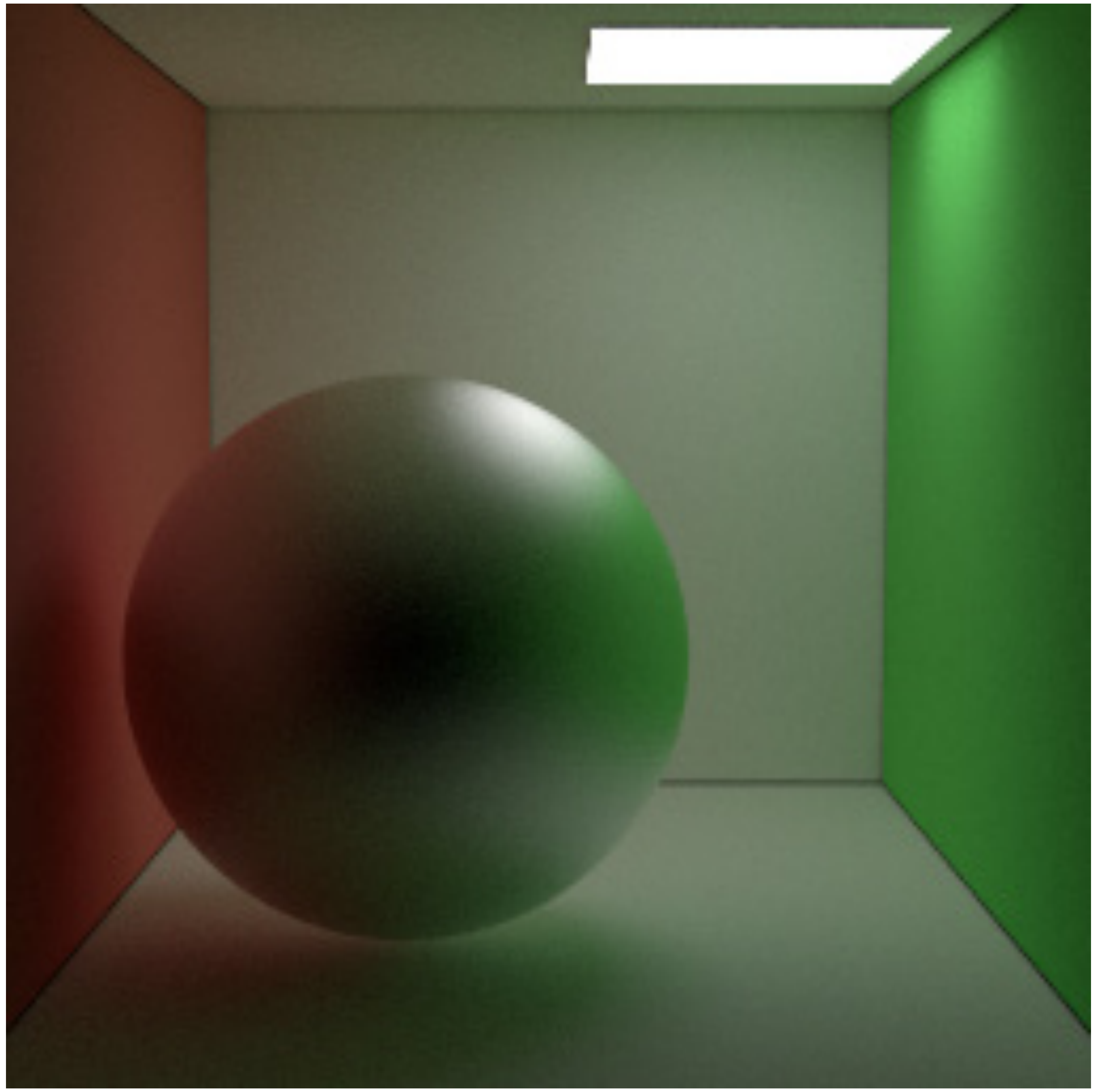
Introduction

Scene parameters

(geometry, materials, lighting, etc.)



Reference image

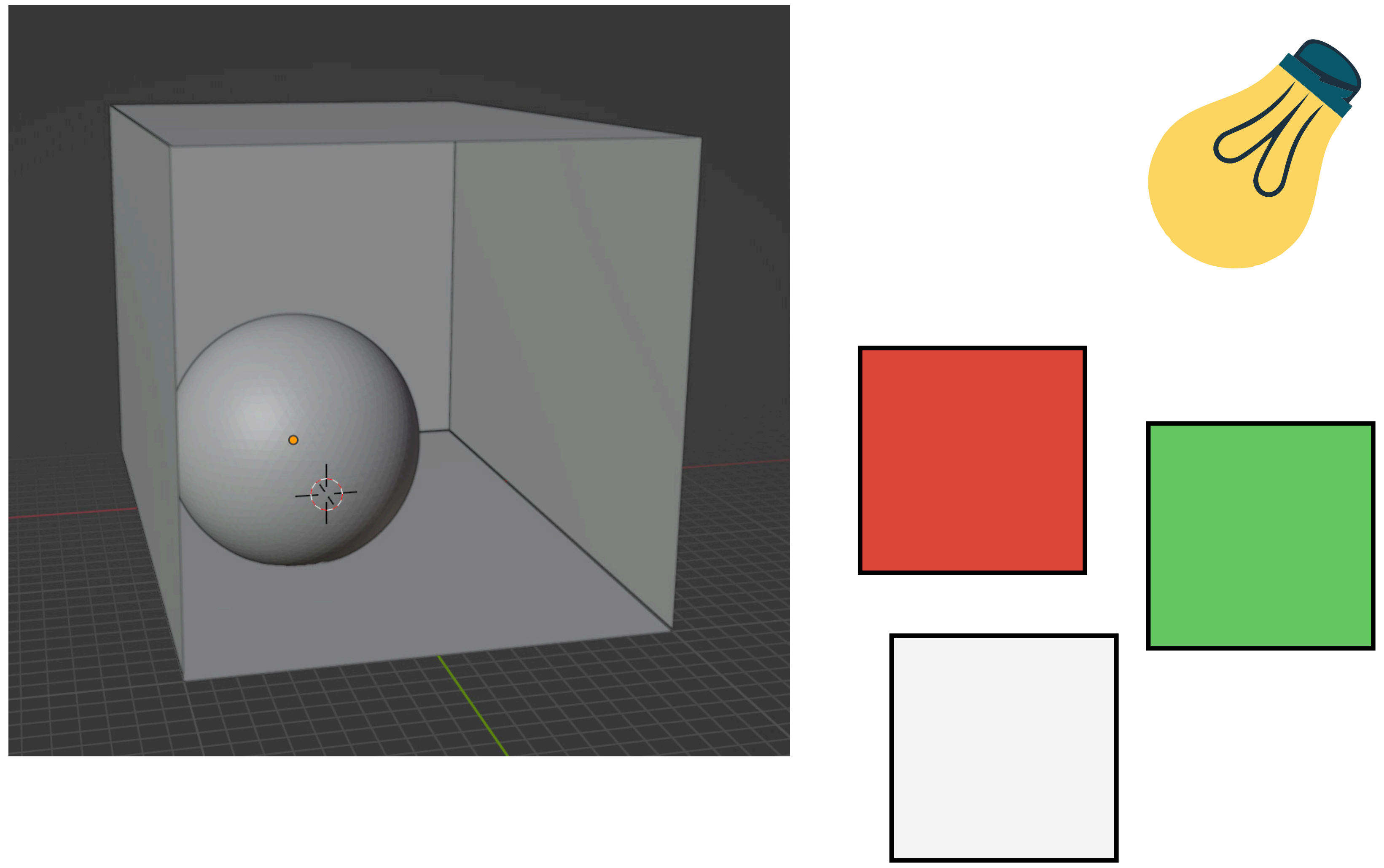


Inverse rendering

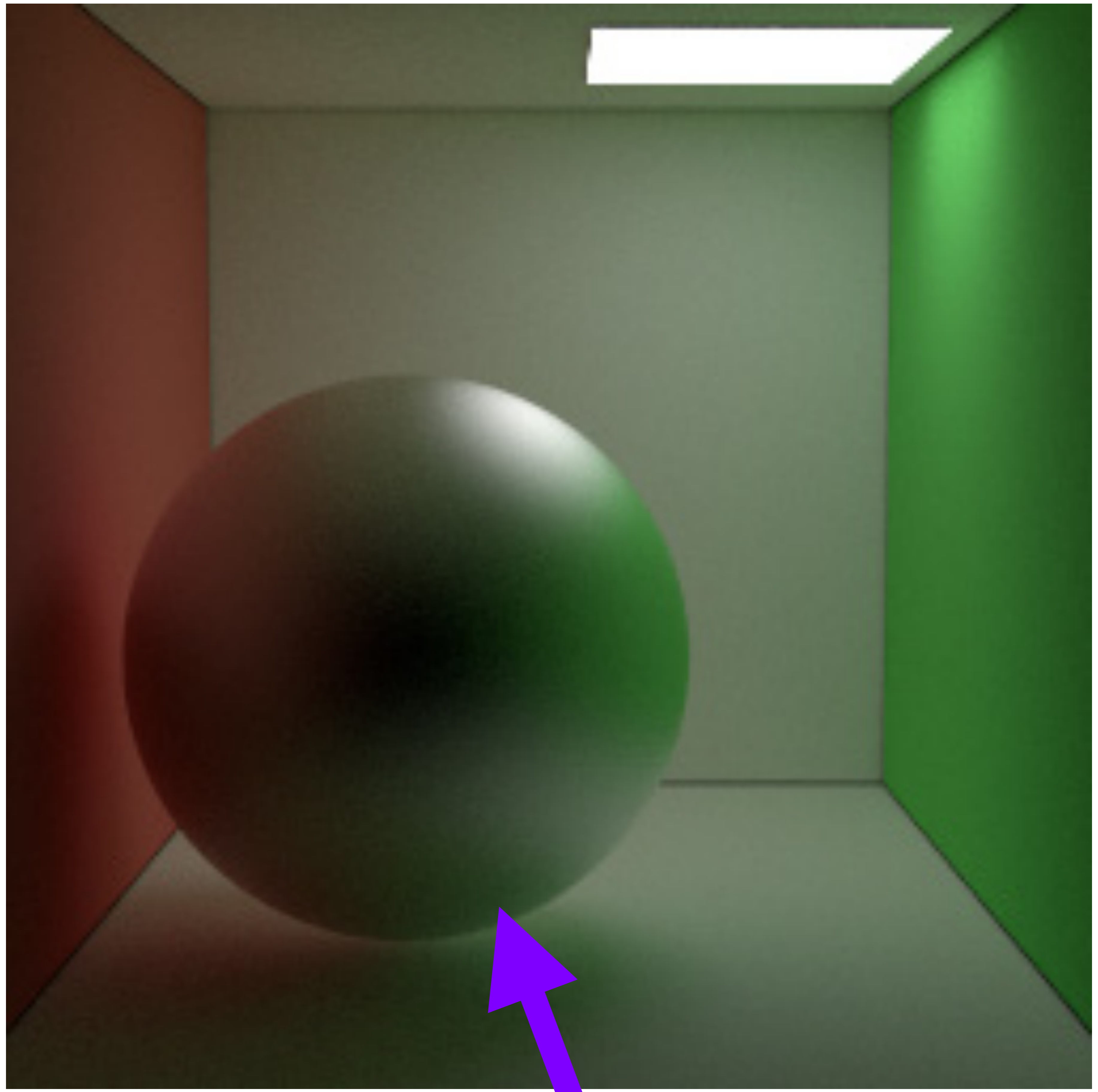
Introduction

Scene parameters

(geometry, materials, lighting, etc.)



Reference image



roughness = ?

Inverse rendering

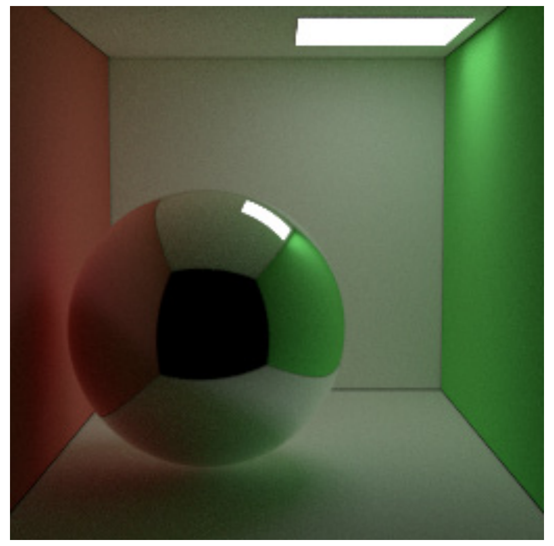
Inverse rendering

$$\min_{\theta} \mathcal{L}\left(\mathbf{I}(\theta), \tilde{\mathbf{I}}\right)$$

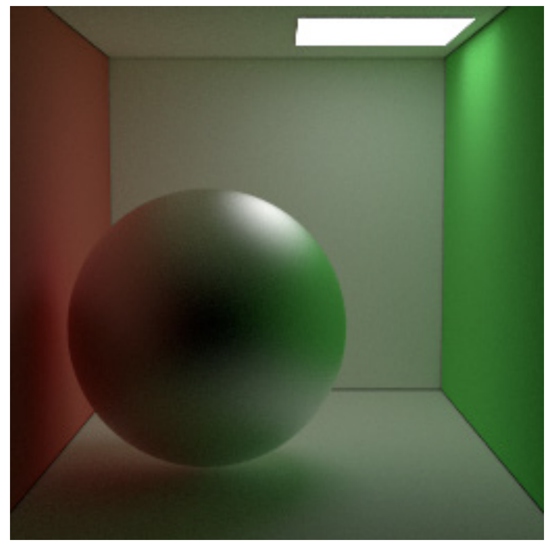
scene parameter → θ

loss function

rendered image



reference image



Inverse rendering

$$\min_{\theta} \mathcal{L}\left(\mathbf{I}(\theta),\, \tilde{\mathbf{I}}\right)$$

scene parameter → θ

loss function → $\mathcal{L}$

rendered image → $\mathbf{I}(\theta)$

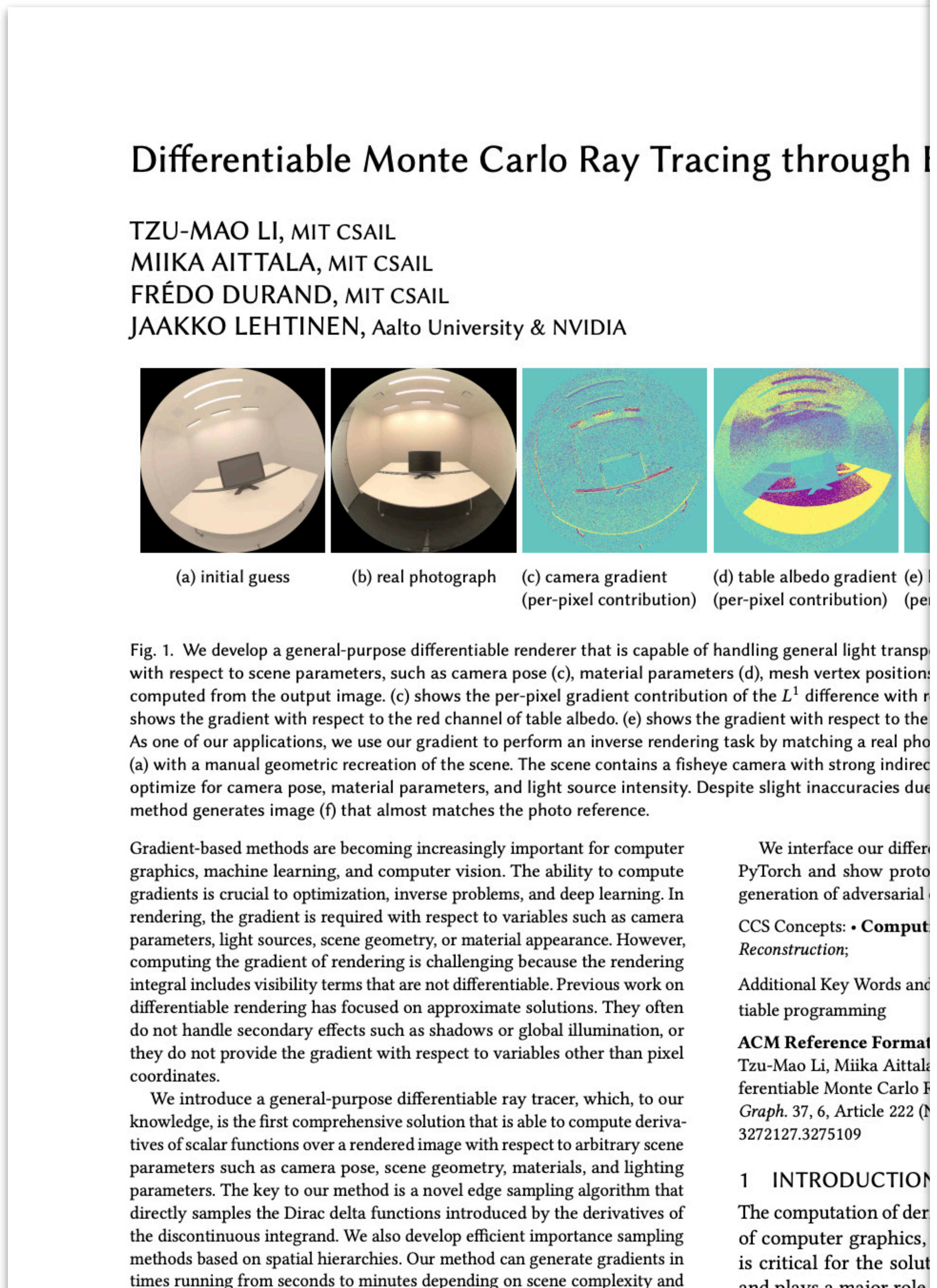
reference image → $\tilde{\mathbf{I}}$

★ **Use differentiable rendering to estimate $\partial_\theta \mathcal{L}$ to do gradient descent**

Prior work: Differentiable rendering

Prior work: Handling discontinuities

[Li et al. 2018]


Differentiable Monte Carlo Ray Tracing through E

TZU-MAO LI, MIT CSAIL
MIIKA AITTALA, MIT CSAIL
FRÉDO DURAND, MIT CSAIL
JAAKKO LEHTINEN, Aalto University & NVIDIA

(a) initial guess (b) real photograph (c) camera gradient (per-pixel contribution) (d) table albedo gradient (per-pixel contribution) (e) l (pe

Fig. 1. We develop a general-purpose differentiable renderer that is capable of handling general light transp with respect to scene parameters, such as camera pose (c), material parameters (d), mesh vertex position computed from the output image. (c) shows the per-pixel gradient contribution of the L^1 difference with r shows the gradient with respect to the red channel of table albedo. (e) shows the gradient with respect to the As one of our applications, we use our gradient to perform an inverse rendering task by matching a real pho (a) with a manual geometric recreation of the scene. The scene contains a fisheye camera with strong indirec optimize for camera pose, material parameters, and light source intensity. Despite slight inaccuracies due method generates image (f) that almost matches the photo reference.

Gradient-based methods are becoming increasingly important for computer graphics, machine learning, and computer vision. The ability to compute gradients is crucial to optimization, inverse problems, and deep learning. In rendering, the gradient is required with respect to variables such as camera parameters, light sources, scene geometry, or material appearance. However, computing the gradient of rendering is challenging because the rendering integral includes visibility terms that are not differentiable. Previous work on differentiable rendering has focused on approximate solutions. They often do not handle secondary effects such as shadows or global illumination, or they do not provide the gradient with respect to variables other than pixel coordinates.

We introduce a general-purpose differentiable ray tracer, which, to our knowledge, is the first comprehensive solution that is able to compute derivatives of scalar functions over a rendered image with respect to arbitrary scene parameters such as camera pose, scene geometry, materials, and lighting parameters. The key to our method is a novel edge sampling algorithm that directly samples the Dirac delta functions introduced by the derivatives of the discontinuous integrand. We also develop efficient importance sampling methods based on spatial hierarchies. Our method can generate gradients in times running from seconds to minutes depending on scene complexity and

We interface our differ PyTorch and show proto generation of adversarial

CCS Concepts: • Comput Reconstruction;

Additional Key Words and tiable programming

ACM Reference Format
Tzu-Mao Li, Miika Aittala ferentiable Monte Carlo R *Graph.* 37, 6, Article 222 (N 3272127.3275109

1 INTRODUCTION
The computation of der of computer graphics, is critical for the solut and plays a major role

[Zhang et al. 2020]


Path-Space Differentiable Rendering

CHENG ZHANG, University of California, Irvine
BAILEY MILLER, Carnegie Mellon University
KAI YAN, University of California, Irvine
IOANNIS GKIOULEKAS, Carnegie Mellon University
SHUANG ZHAO, University of California, Irvine

Original Derivative with re

Fig. 1. We introduce **path-space differentiable rendering**, a new theoretical framework to estimate deriv to arbitrary scene parameters (e.g., material properties and object geometries). By directly differentiating f *integral* framework, enabling the design of new unbiased Monte Carlo methods capable of efficiently estim geometry and light transport effects. This example shows a dinning room scene lit by the sun from outside the derivative image with respect to the vertical location of the sun. (Please use Adobe Acrobat to view the teas

Physics-based differentiable rendering, the estimation of derivatives of radiometric measures with respect to arbitrary scene parameters, has a diverse array of applications from solving analysis-by-synthesis problems to training machine learning pipelines incorporating forward rendering processes. Unfortunately, general-purpose differentiable rendering remains challenging due to the lack of efficient estimators as well as the need to identify and handle complex discontinuities such as visibility boundaries.

In this paper, we show how path integrals can be differentiated with respect to arbitrary differentiable changes of a scene. We provide a detailed theoretical analysis of this process and establish new differentiable rendering formulations based on the resulting differential path integrals. Our path-space differentiable rendering formulation allows the design of new Monte Carlo estimators that offer significantly better efficiency than state-of-the-art methods in handling complex geometric discontinuities and light transport phenomena such as caustics.

We validate our metho generated using the finite ness of our technique, we few state-of-the-art differ

CCS Concepts: • Compu

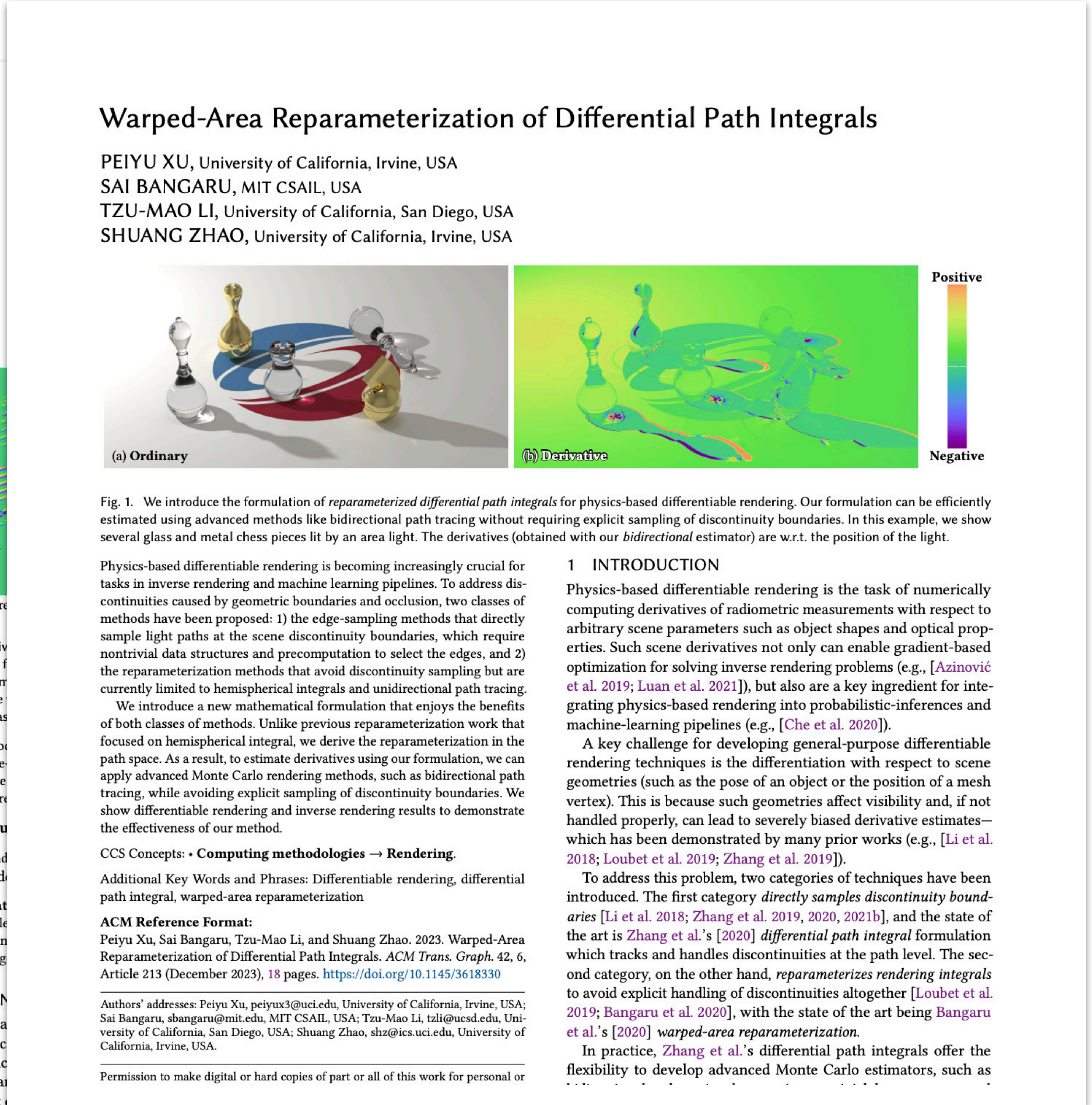
Additional Key Words and gral, Monte Carlo rend

ACM Reference Format
Cheng Zhang, Bailey Mille 2020. Path-Space Differen cle 143 (July 2020), 19 pag

1 INTRODUCTION
Physics-based light tra computer graphics sinc estimating radiometric scenes. Previous resear dering algorithms that

Authors' addresses: Cheng Zhang, University of California, Irvine, chengz20@uci.edu; Bailey Miller, Carnegie Mellon University, baileymark.miller@gmail.com; Kai Yan, University of California, Irvine, kyan8@uci.edu; Ioannis Gkioulekas, Carnegie Mellon University, igkioule@andrew.cmu.edu; Shuang Zhao, University of California, Irvine, shz@ics.uci.edu.

[Xu et al. 2023]


Warped-Area Reparameterization of Differential Path Integrals

PEIYU XU, University of California, Irvine, USA
SAI BANGARU, MIT CSAIL, USA
TZU-MAO LI, University of California, San Diego, USA
SHUANG ZHAO, University of California, Irvine, USA

(a) Ordinary (b) Derivative Positive Negative

Fig. 1. We introduce the formulation of *reparameterized differential path integrals* for physics-based differentiable rendering. Our formulation can be efficiently estimated using advanced methods like bidirectional path tracing without requiring explicit sampling of discontinuity boundaries. In this example, we show several glass and metal chess pieces lit by an area light. The derivatives (obtained with our *bidirectional* estimator) are w.r.t. the position of the light.

Physics-based differentiable rendering is becoming increasingly crucial for tasks in inverse rendering and machine learning pipelines. To address discontinuities caused by geometric boundaries and occlusion, two classes of methods have been proposed: 1) the edge-sampling methods that directly sample light paths at the scene discontinuity boundaries, which require nontrivial data structures and precomputation to select the edges, and 2) the reparameterization methods that avoid discontinuity sampling but are currently limited to hemispherical integrals and unidirectional path tracing.

We introduce a new mathematical formulation that enjoys the benefits of both classes of methods. Unlike previous reparameterization work that focused on hemispherical integral, we derive the reparameterization in the path space. As a result, to estimate derivatives using our formulation, we can apply advanced Monte Carlo rendering methods, such as bidirectional path tracing, while avoiding explicit sampling of discontinuity boundaries. We show differentiable rendering and inverse rendering results to demonstrate the effectiveness of our method.

CCS Concepts: • **Computing methodologies → Rendering.**

Additional Key Words and Phrases: Differentiable rendering, differential path integral, warped-area reparameterization

ACM Reference Format:
Peiyu Xu, Sai Bangaru, Tzu-Mao Li, and Shuang Zhao. 2023. Warped-Area Reparameterization of Differential Path Integrals. *ACM Trans. Graph.* 42, 6, Article 213 (December 2023), 18 pages. https://doi.org/10.1145/3618330

Authors' addresses: Peiyu Xu, peiyux3@uci.edu, University of California, Irvine, USA; Sai Bangaru, sbangaru@mit.edu, MIT CSAIL, USA; Tzu-Mao Li, tzli@ucsd.edu, University of California, San Diego, USA; Shuang Zhao, shz@ics.uci.edu, University of California, Irvine, USA.

Permission to make digital or hard copies of part or all of this work for personal or

1 INTRODUCTION
Physics-based differentiable rendering is the task of numerically computing derivatives of radiometric measurements with respect to arbitrary scene parameters such as object shapes and optical properties. Such scene derivatives not only can enable gradient-based optimization for solving inverse rendering problems (e.g., [Azinović et al. 2019; Luan et al. 2021]), but also are a key ingredient for integrating physics-based rendering into probabilistic-inferences and machine-learning pipelines (e.g., [Che et al. 2020]).

A key challenge for developing general-purpose differentiable rendering techniques is the differentiation with respect to scene geometries (such as the pose of an object or the position of a mesh vertex). This is because such geometries affect visibility and, if not handled properly, can lead to severely biased derivative estimates—which has been demonstrated by many prior works (e.g., [Li et al. 2018; Loubet et al. 2019; Zhang et al. 2019]).

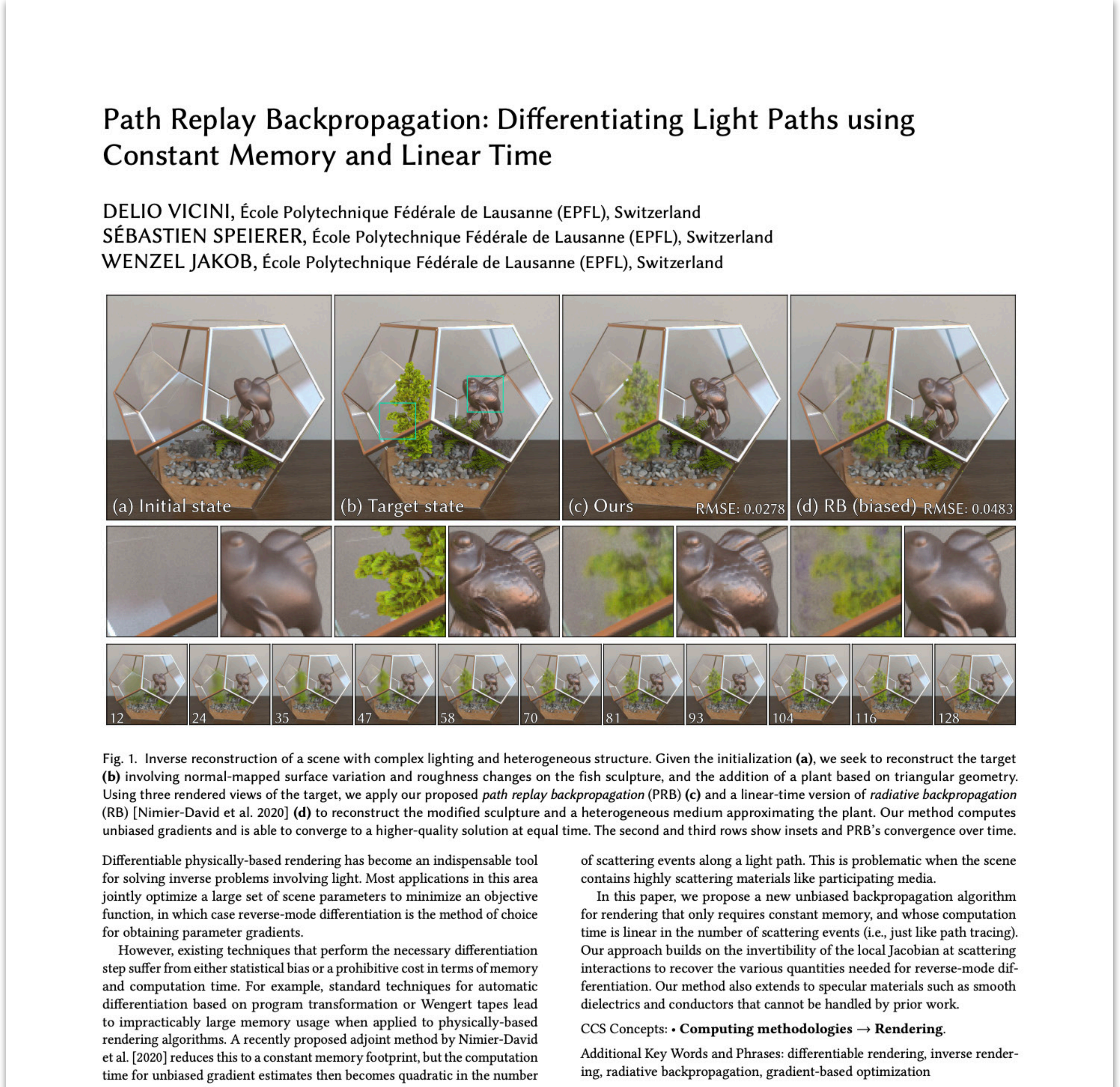
To address this problem, two categories of techniques have been introduced. The first category *directly samples discontinuity boundaries* [Li et al. 2018; Zhang et al. 2019, 2020, 2021b], and the state of the art is Zhang et al.'s [2020] *differential path integral* formulation which tracks and handles discontinuities at the path level. The second category, on the other hand, *reparameterizes rendering integrals* to avoid explicit handling of discontinuities altogether [Loubet et al. 2019; Bangaru et al. 2020], with the state of the art being Bangaru et al.'s [2020] *warped-area reparameterization*.

In practice, Zhang et al.'s differential path integrals offer the flexibility to develop advanced Monte Carlo estimators, such as

. . .

Prior work: Reducing complexity

[Vicini et al. 2021]



Path Replay Backpropagation: Differentiating Light Paths using Constant Memory and Linear Time

DELIO VICINI, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
SÉBASTIEN SPEIERER, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
WENZEL JAKOB, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Fig. 1. Inverse reconstruction of a scene with complex lighting and heterogeneous structure. Given the initialization **(a)**, we seek to reconstruct the target **(b)** involving normal-mapped surface variation and roughness changes on the fish sculpture, and the addition of a plant based on triangular geometry. Using three rendered views of the target, we apply our proposed *path replay backpropagation* (PRB) **(c)** and a linear-time version of *radiative backpropagation* (RB) [Nimier-David et al. 2020] **(d)** to reconstruct the modified sculpture and a heterogeneous medium approximating the plant. Our method computes unbiased gradients and is able to converge to a higher-quality solution at equal time. The second and third rows show insets and PRB's convergence over time.

Differentiable physically-based rendering has become an indispensable tool for solving inverse problems involving light. Most applications in this area jointly optimize a large set of scene parameters to minimize an objective function, in which case reverse-mode differentiation is the method of choice for obtaining parameter gradients.

However, existing techniques that perform the necessary differentiation step suffer from either statistical bias or a prohibitive cost in terms of memory and computation time. For example, standard techniques for automatic differentiation based on program transformation or Wengert tapes lead to impracticably large memory usage when applied to physically-based rendering algorithms. A recently proposed adjoint method by Nimier-David et al. [2020] reduces this to a constant memory footprint, but the computation time for unbiased gradient estimates then becomes quadratic in the number of scattering events along a light path. This is problematic when the scene contains highly scattering materials like participating media.

In this paper, we propose a new unbiased backpropagation algorithm for rendering that only requires constant memory, and whose computation time is linear in the number of scattering events (i.e., just like path tracing). Our approach builds on the invertibility of the local Jacobian at scattering interactions to recover the various quantities needed for reverse-mode differentiation. Our method also extends to specular materials such as smooth dielectrics and conductors that cannot be handled by prior work.

CCS Concepts: • **Computing methodologies** → **Rendering**.

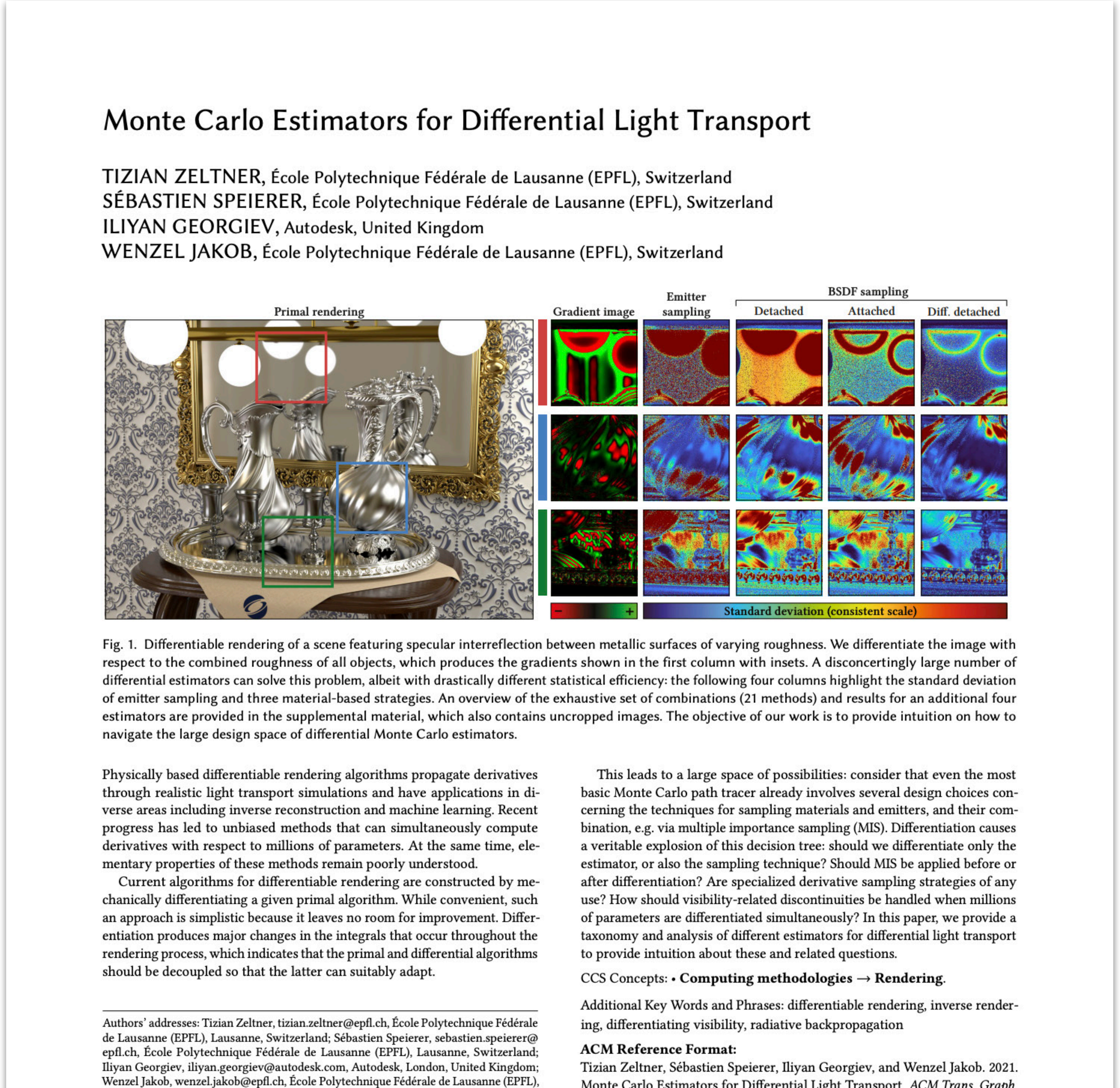
Additional Key Words and Phrases: differentiable rendering, inverse rendering, radiative backpropagation, gradient-based optimization

Path Replay Backpropagation (PRB):

- Differentiable rendering algorithm with constant memory, **linear time complexity**

Prior work: Path sampling methods

[Zeltner et al. 2021]



Monte Carlo Estimators for Differential Light Transport

TIZIAN ZELTNER, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
SÉBASTIEN SPEIERER, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
ILIYAN GEORGIEV, Autodesk, United Kingdom
WENZEL JAKOB, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Primal rendering | Gradient image | Emitter sampling | BSDF sampling: Detached, Attached, Diff. detached

Standard deviation (consistent scale)

Fig. 1. Differentiable rendering of a scene featuring specular interreflection between metallic surfaces of varying roughness. We differentiate the image with respect to the combined roughness of all objects, which produces the gradients shown in the first column with insets. A disconcertingly large number of differential estimators can solve this problem, albeit with drastically different statistical efficiency: the following four columns highlight the standard deviation of emitter sampling and three material-based strategies. An overview of the exhaustive set of combinations (21 methods) and results for an additional four estimators are provided in the supplemental material, which also contains uncropped images. The objective of our work is to provide intuition on how to navigate the large design space of differential Monte Carlo estimators.

Physically based differentiable rendering algorithms propagate derivatives through realistic light transport simulations and have applications in diverse areas including inverse reconstruction and machine learning. Recent progress has led to unbiased methods that can simultaneously compute derivatives with respect to millions of parameters. At the same time, elementary properties of these methods remain poorly understood.

Current algorithms for differentiable rendering are constructed by mechanically differentiating a given primal algorithm. While convenient, such an approach is simplistic because it leaves no room for improvement. Differentiation produces major changes in the integrals that occur throughout the rendering process, which indicates that the primal and differential algorithms should be decoupled so that the latter can suitably adapt.

This leads to a large space of possibilities: consider that even the most basic Monte Carlo path tracer already involves several design choices concerning the techniques for sampling materials and emitters, and their combination, e.g. via multiple importance sampling (MIS). Differentiation causes a veritable explosion of this decision tree: should we differentiate only the estimator, or also the sampling technique? Should MIS be applied before or after differentiation? Are specialized derivative sampling strategies of any use? How should visibility-related discontinuities be handled when millions of parameters are differentiated simultaneously? In this paper, we provide a taxonomy and analysis of different estimators for differential light transport to provide intuition about these and related questions.

CCS Concepts: • **Computing methodologies** → **Rendering**.

Additional Key Words and Phrases: differentiable rendering, inverse rendering, differentiating visibility, radiative backpropagation

ACM Reference Format:
Tizian Zeltner, Sébastien Speierer, Iliyan Georgiev, and Wenzel Jakob. 2021. Monte Carlo Estimators for Differential Light Transport. *ACM Trans. Graph.*

Authors' addresses: Tizian Zeltner, tizian.zeltner@epfl.ch, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland; Sébastien Speierer, sebastien.speierer@epfl.ch, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland; Iliyan Georgiev, iliyan.georgiev@autodesk.com, Autodesk, London, United Kingdom; Wenzel Jakob, wenzel.jakob@epfl.ch, École Polytechnique Fédérale de Lausanne (EPFL),

- Formulations for forward and differentiable rendering are similar, but not the same

Prior work: Path sampling methods

[Zeltner et al. 2021]

Monte Carlo Estimators for Differential Light Transport

TIZIAN ZELTNER, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
SÉBASTIEN SPEIERER, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
ILIYAN GEORGIEV, Autodesk, United Kingdom
WENZEL JAKOB, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

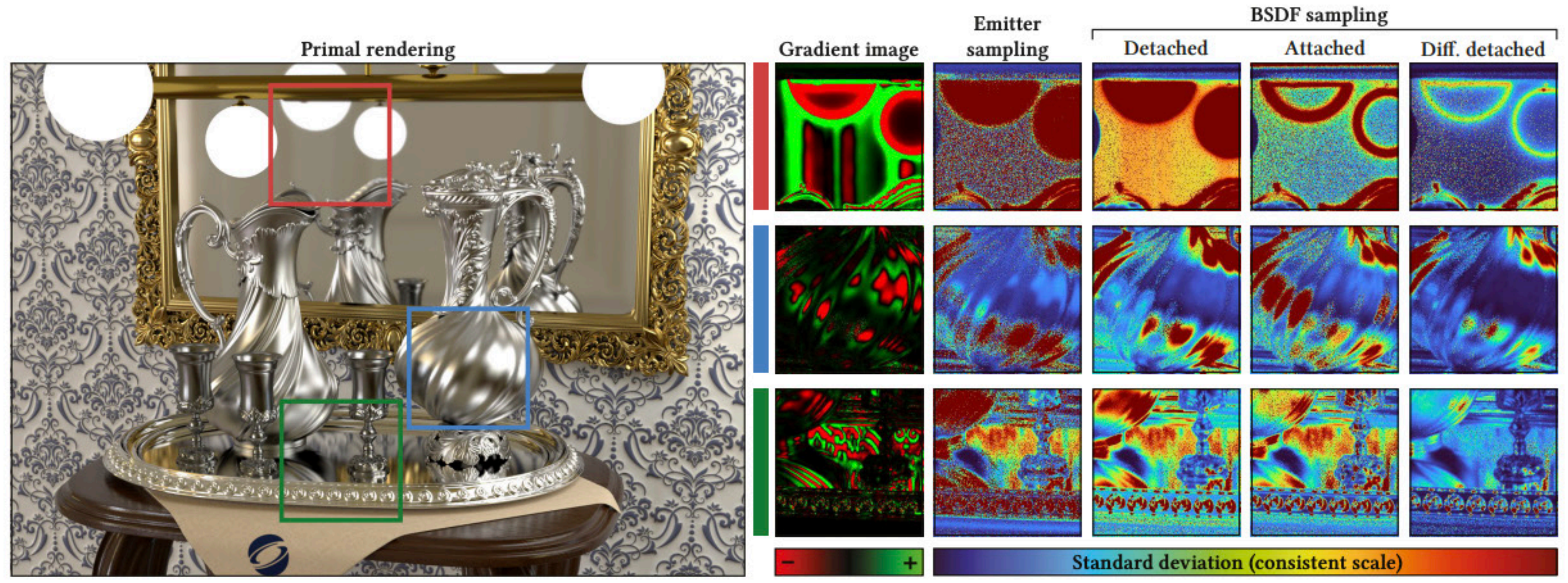


Fig. 1. Differentiable rendering of a scene featuring specular interreflection between metallic surfaces of varying roughness. We differentiate the image with respect to the combined roughness of all objects, which produces the gradients shown in the first column with insets. A disconcertingly large number of differential estimators can solve this problem, albeit with drastically different statistical efficiency: the following four columns highlight the standard deviation of emitter sampling and three material-based strategies. An overview of the exhaustive set of combinations (21 methods) and results for an additional four estimators are provided in the supplemental material, which also contains uncropped images. The objective of our work is to provide intuition on how to navigate the large design space of differential Monte Carlo estimators.

Physically based differentiable rendering algorithms propagate derivatives through realistic light transport simulations and have applications in diverse areas including inverse reconstruction and machine learning. Recent progress has led to unbiased methods that can simultaneously compute derivatives with respect to millions of parameters. At the same time, elementary properties of these methods remain poorly understood.

Current algorithms for differentiable rendering are constructed by mechanically differentiating a given primal algorithm. While convenient, such an approach is simplistic because it leaves no room for improvement. Differentiation produces major changes in the integrals that occur throughout the rendering process, which indicates that the primal and differential algorithms should be decoupled so that the latter can suitably adapt.

Authors' addresses: Tizian Zeltner, tizian.zeltner@epfl.ch, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland; Sébastien Speierer, sebastien.speierer@epfl.ch, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland; Iliyan Georgiev, iliyan.georgiev@autodesk.com, Autodesk, London, United Kingdom; Wenzel Jakob, wenzel.jakob@epfl.ch, École Polytechnique Fédérale de Lausanne (EPFL),

This leads to a large space of possibilities: consider that even the most basic Monte Carlo path tracer already involves several design choices concerning the techniques for sampling materials and emitters, and their combination, e.g. via multiple importance sampling (MIS). Differentiation causes a veritable explosion of this decision tree: should we differentiate only the estimator, or also the sampling technique? Should MIS be applied before or after differentiation? Are specialized derivative sampling strategies of any use? How should visibility-related discontinuities be handled when millions of parameters are differentiated simultaneously? In this paper, we provide a taxonomy and analysis of different estimators for differential light transport to provide intuition about these and related questions.

CCS Concepts: • **Computing methodologies → Rendering**.

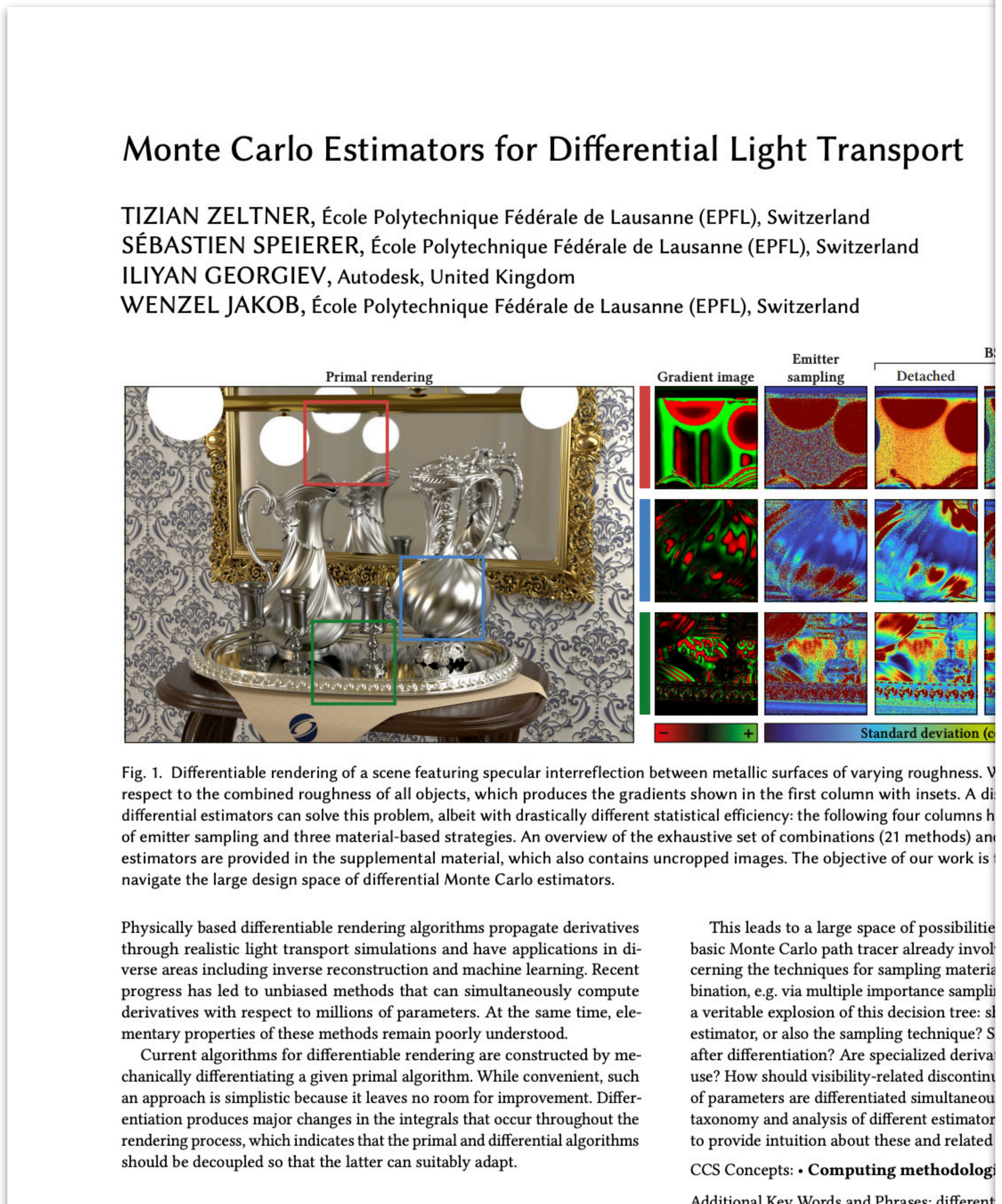
Additional Key Words and Phrases: differentiable rendering, inverse rendering, differentiating visibility, radiative backpropagation

ACM Reference Format:
Tizian Zeltner, Sébastien Speierer, Iliyan Georgiev, and Wenzel Jakob. 2021. Monte Carlo Estimators for Differential Light Transport. *ACM Trans. Graph.*

- Formulations for forward and differentiable rendering are similar, but not the same
- Sampling methods tailored for differentiable rendering can greatly reduce variance

Prior work: Differential BRDF sampling methods

[Zeltner et al. 2021]



[Zhang et al. 2021]

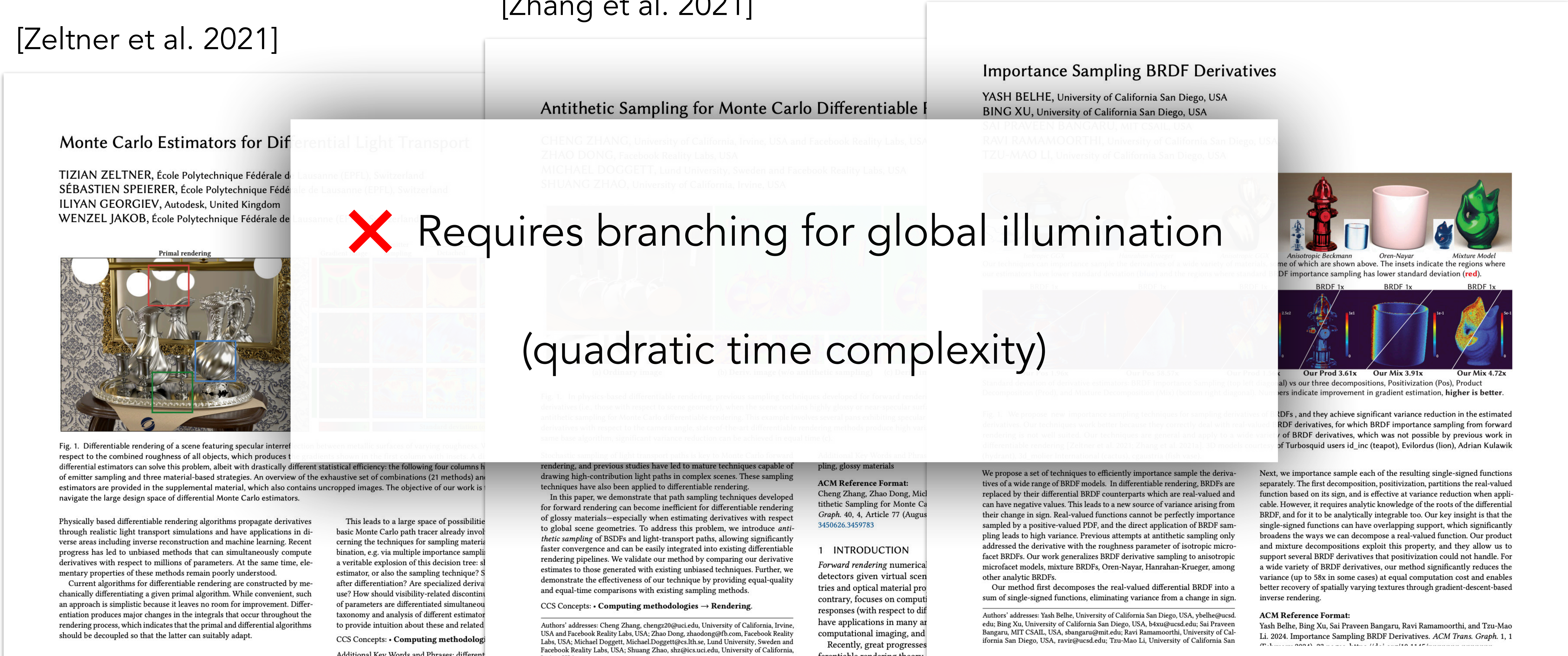
[Belhe et al. 2024]

Prior work: Differential BRDF sampling methods

[Zeltner et al. 2021]

[Zhang et al. 2021]

[Belhe et al. 2024]

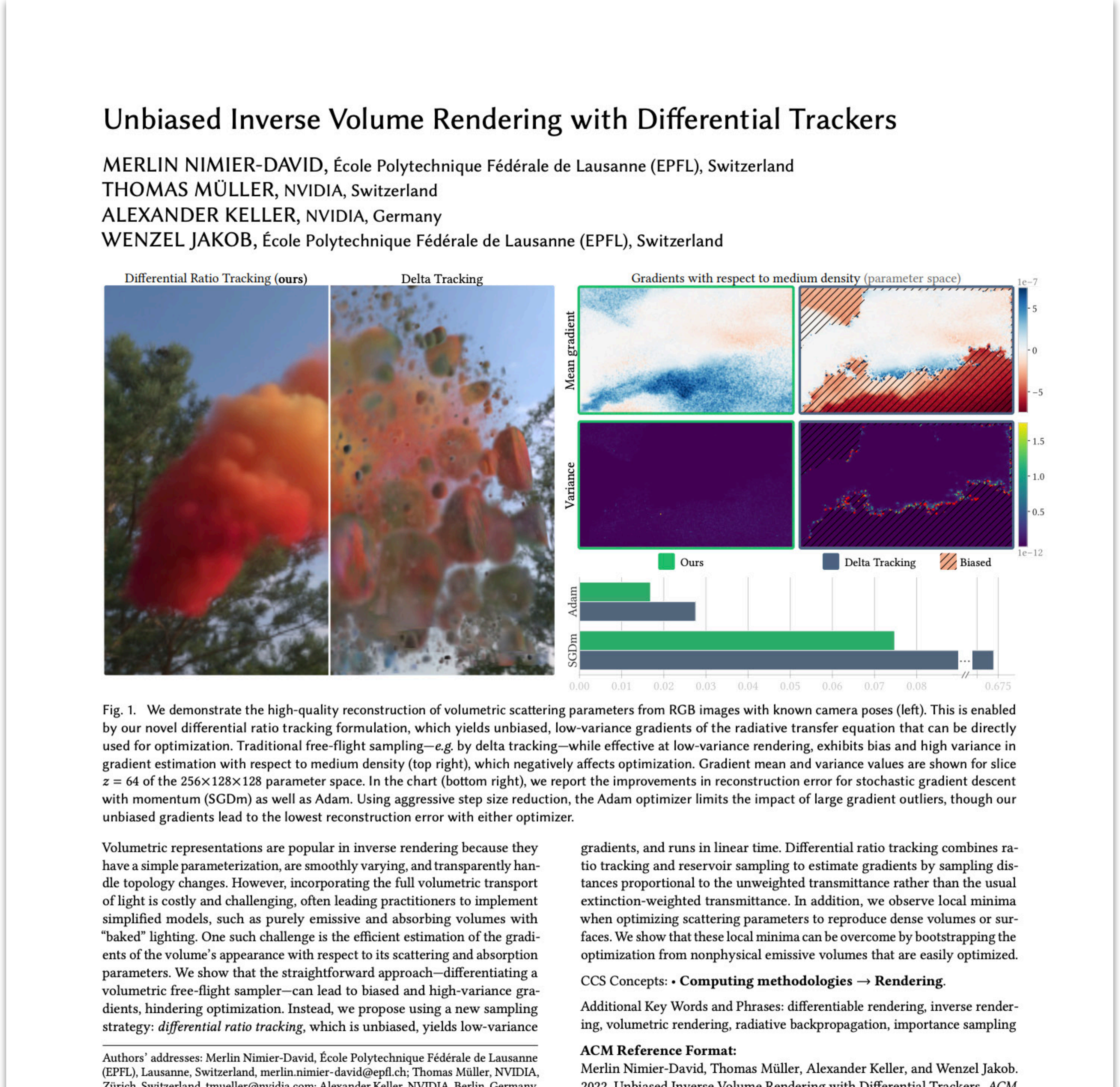


❌ Requires branching for global illumination

(quadratic time complexity)

Prior work: Differentiable volume rendering

[Nimier-David et al. 2022]



Unbiased Inverse Volume Rendering with Differential Trackers

MERLIN NIMIER-DAVID, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
THOMAS MÜLLER, NVIDIA, Switzerland
ALEXANDER KELLER, NVIDIA, Germany
WENZEL JAKOB, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Fig. 1. We demonstrate the high-quality reconstruction of volumetric scattering parameters from RGB images with known camera poses (left). This is enabled by our novel differential ratio tracking formulation, which yields unbiased, low-variance gradients of the radiative transfer equation that can be directly used for optimization. Traditional free-flight sampling—*e.g.* by delta tracking—while effective at low-variance rendering, exhibits bias and high variance in gradient estimation with respect to medium density (top right), which negatively affects optimization. Gradient mean and variance values are shown for slice $z = 64$ of the 256×128×128 parameter space. In the chart (bottom right), we report the improvements in reconstruction error for stochastic gradient descent with momentum (SGDm) as well as Adam. Using aggressive step size reduction, the Adam optimizer limits the impact of large gradient outliers, though our unbiased gradients lead to the lowest reconstruction error with either optimizer.

Volumetric representations are popular in inverse rendering because they have a simple parameterization, are smoothly varying, and transparently handle topology changes. However, incorporating the full volumetric transport of light is costly and challenging, often leading practitioners to implement simplified models, such as purely emissive and absorbing volumes with "baked" lighting. One such challenge is the efficient estimation of the gradients of the volume's appearance with respect to its scattering and absorption parameters. We show that the straightforward approach—differentiating a volumetric free-flight sampler—can lead to biased and high-variance gradients, hindering optimization. Instead, we propose using a new sampling strategy: *differential ratio tracking*, which is unbiased, yields low-variance gradients, and runs in linear time. Differential ratio tracking combines ratio tracking and reservoir sampling to estimate gradients by sampling distances proportional to the unweighted transmittance rather than the usual extinction-weighted transmittance. In addition, we observe local minima when optimizing scattering parameters to reproduce dense volumes or surfaces. We show that these local minima can be overcome by bootstrapping the optimization from nonphysical emissive volumes that are easily optimized.

CCS Concepts: • **Computing methodologies** → **Rendering**.

Additional Key Words and Phrases: differentiable rendering, inverse rendering, volumetric rendering, radiative backpropagation, importance sampling

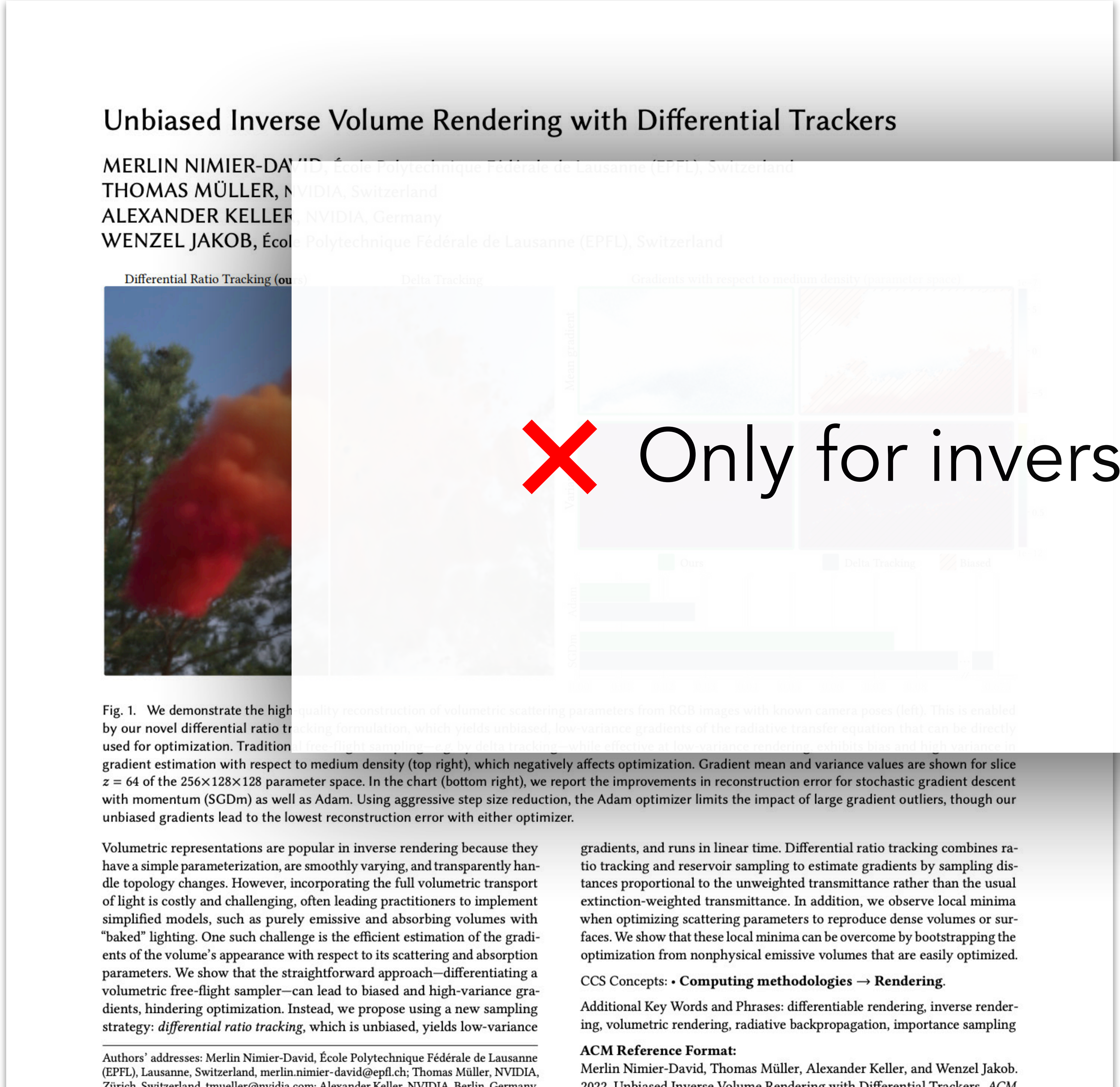
Authors' addresses: Merlin Nimier-David, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, merlin.nimier-david@epfl.ch; Thomas Müller, NVIDIA,

ACM Reference Format:
Merlin Nimier-David, Thomas Müller, Alexander Keller, and Wenzel Jakob.

- Sampling methods tailored for differentiable rendering of **volumes**
- Works with PRB (linear time complexity)

Prior work: Differentiable volume rendering

[Nimier-David et al. 2022]



Unbiased Inverse Volume Rendering with Differential Trackers

MERLIN NIMIER-DAVID, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
THOMAS MÜLLER, NVIDIA, Switzerland
ALEXANDER KELLER, NVIDIA, Germany
WENZEL JAKOB, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Fig. 1. We demonstrate the high-quality reconstruction of volumetric scattering parameters from RGB images with known camera poses (left). This is enabled by our novel differential ratio tracking formulation, which yields unbiased, low-variance gradients of the radiative transfer equation that can be directly used for optimization. Traditional free-flight sampling—as by delta tracking—while effective at importance rendering, exhibits bias and high variance in gradient estimation with respect to medium density (top right), which negatively affects optimization. Gradient mean and variance values are shown for slice z = 64 of the 256×128×128 parameter space. In the chart (bottom right), we report the improvements in reconstruction error for stochastic gradient descent with momentum (SGDm) as well as Adam. Using aggressive step size reduction, the Adam optimizer limits the impact of large gradient outliers, though our unbiased gradients lead to the lowest reconstruction error with either optimizer.

Volumetric representations are popular in inverse rendering because they have a simple parameterization, are smoothly varying, and transparently handle topology changes. However, incorporating the full volumetric transport of light is costly and challenging, often leading practitioners to implement simplified models, such as purely emissive and absorbing volumes with "baked" lighting. One such challenge is the efficient estimation of the gradients of the volume's appearance with respect to its scattering and absorption parameters. We show that the straightforward approach—differentiating a volumetric free-flight sampler—can lead to biased and high-variance gradients, hindering optimization. Instead, we propose using a new sampling strategy: *differential ratio tracking*, which is unbiased, yields low-variance gradients, and runs in linear time. Differential ratio tracking combines ratio tracking and reservoir sampling to estimate gradients by sampling distances proportional to the unweighted transmittance rather than the usual extinction-weighted transmittance. In addition, we observe local minima when optimizing scattering parameters to reproduce dense volumes or surfaces. We show that these local minima can be overcome by bootstrapping the optimization from nonphysical emissive volumes that are easily optimized.

CCS Concepts: • **Computing methodologies** → **Rendering**.

Additional Key Words and Phrases: differentiable rendering, inverse rendering, volumetric rendering, radiative backpropagation, importance sampling

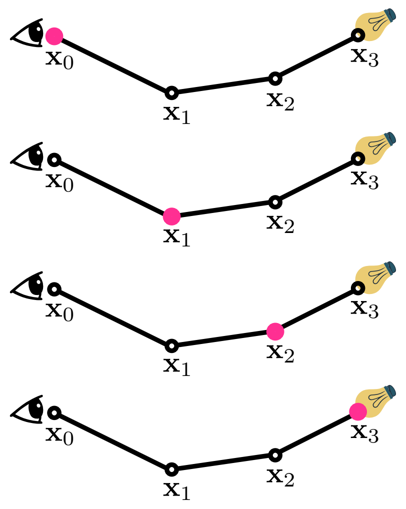
ACM Reference Format:
Merlin Nimier-David, Thomas Müller, Alexander Keller, and Wenzel Jakob. 2022. Unbiased Inverse Volume Rendering with Differential Trackers. ACM

Authors' addresses: Merlin Nimier-David, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, merlin.nimier-david@epfl.ch; Thomas Müller, NVIDIA, Zürich, Switzerland, tmueller@nvidia.com; Alexander Keller, NVIDIA, Berlin, Germany

- Sampling methods tailored for differentiable rendering of **volumes**
- Works with PRB (linear time complexity)

❌ Only for inverse volume rendering

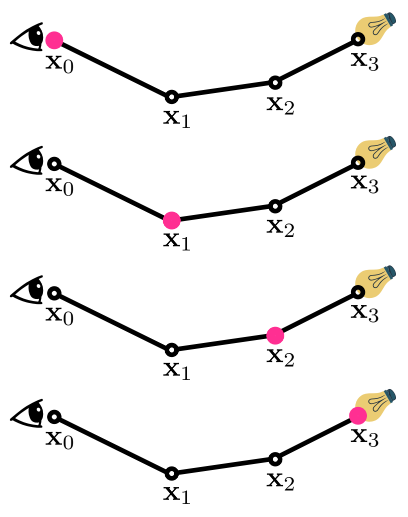
Our contributions



Differential path space integral

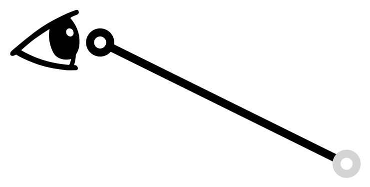
New theoretical formulation

Our contributions



Differential path space integral

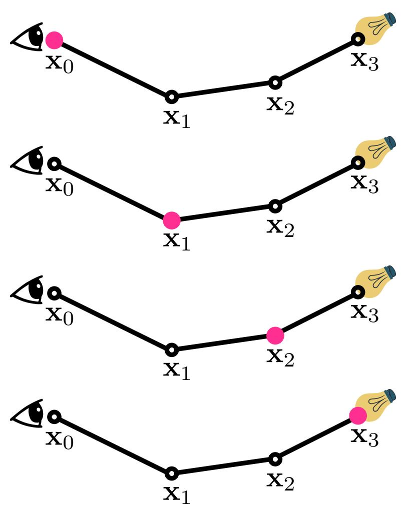
New theoretical formulation



Differential sampling method

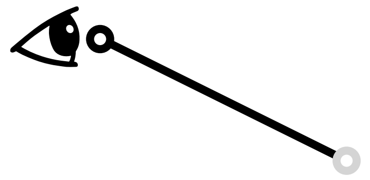
Importance sample paths using the new formulation (linear time complexity)

Our contributions



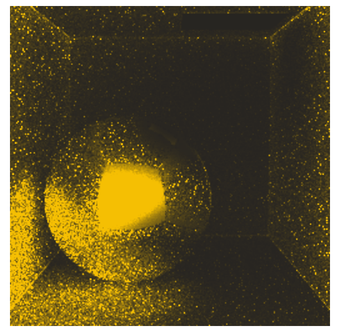
Differential path space integral

New theoretical formulation



Differential sampling method

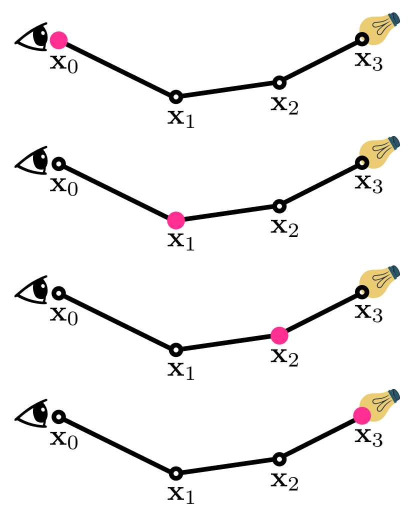
Importance sample paths using the new formulation (linear time complexity)



Adaptive pixel sampling method

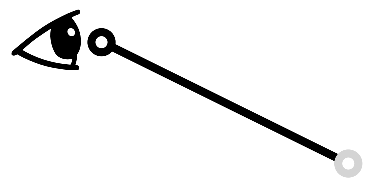
Importance sample pixels during inverse rendering optimization

Our contributions



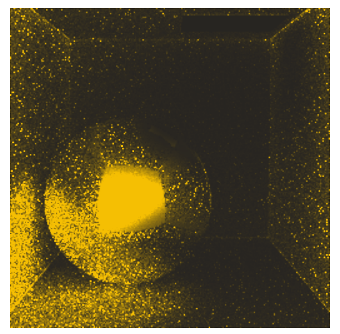
Differential path space integral

New theoretical formulation



Differential sampling method

Importance sample paths using the new formulation (linear time complexity)



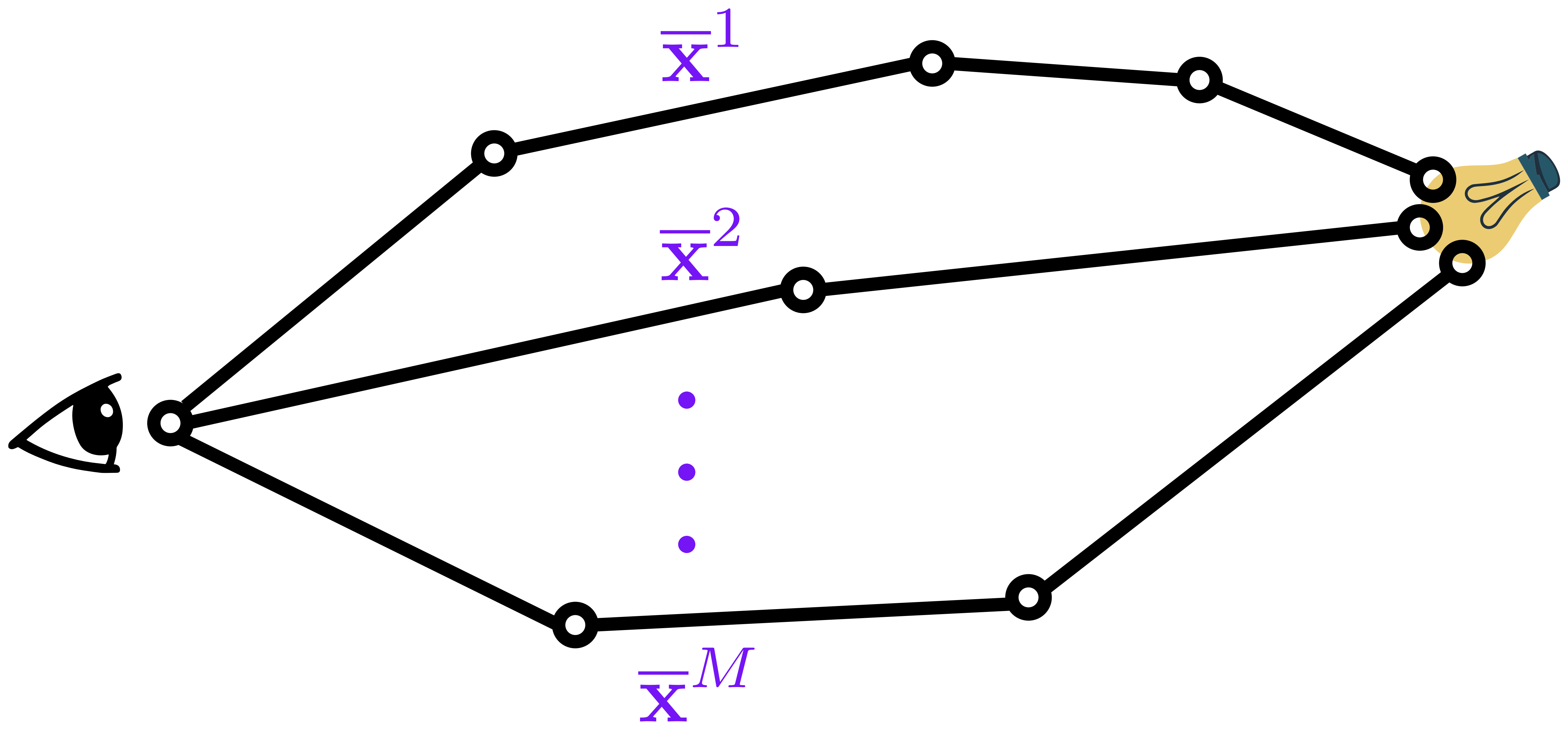
Adaptive pixel sampling method

Importance sample pixels during inverse rendering optimization

Lower gradient variance + improved inverse rendering performance

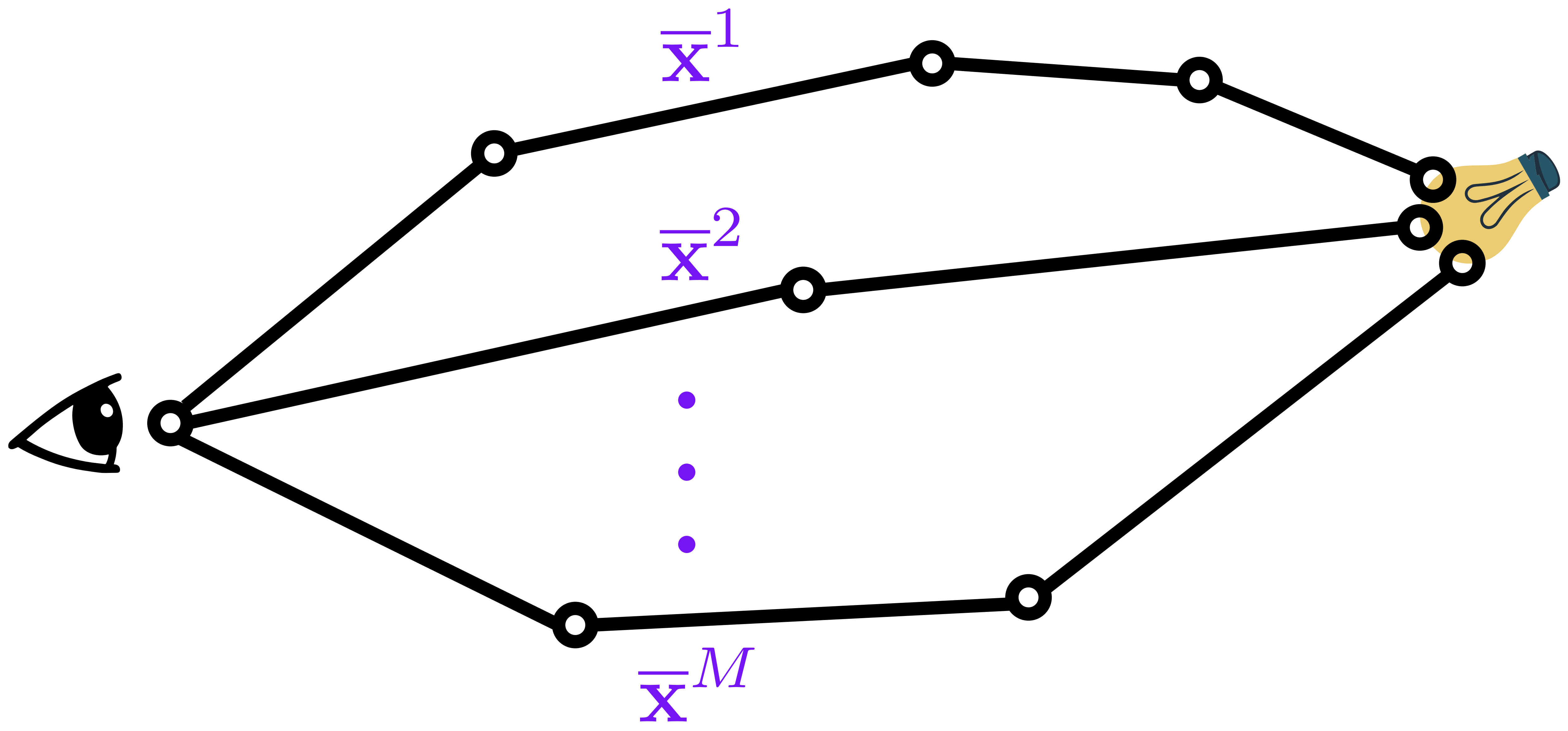
Differential path space integral

Path integral



$$I = \int_{\mathcal{P}} f(\overline{\mathbf{x}}) \mathrm{d}\overline{\mathbf{x}}$$

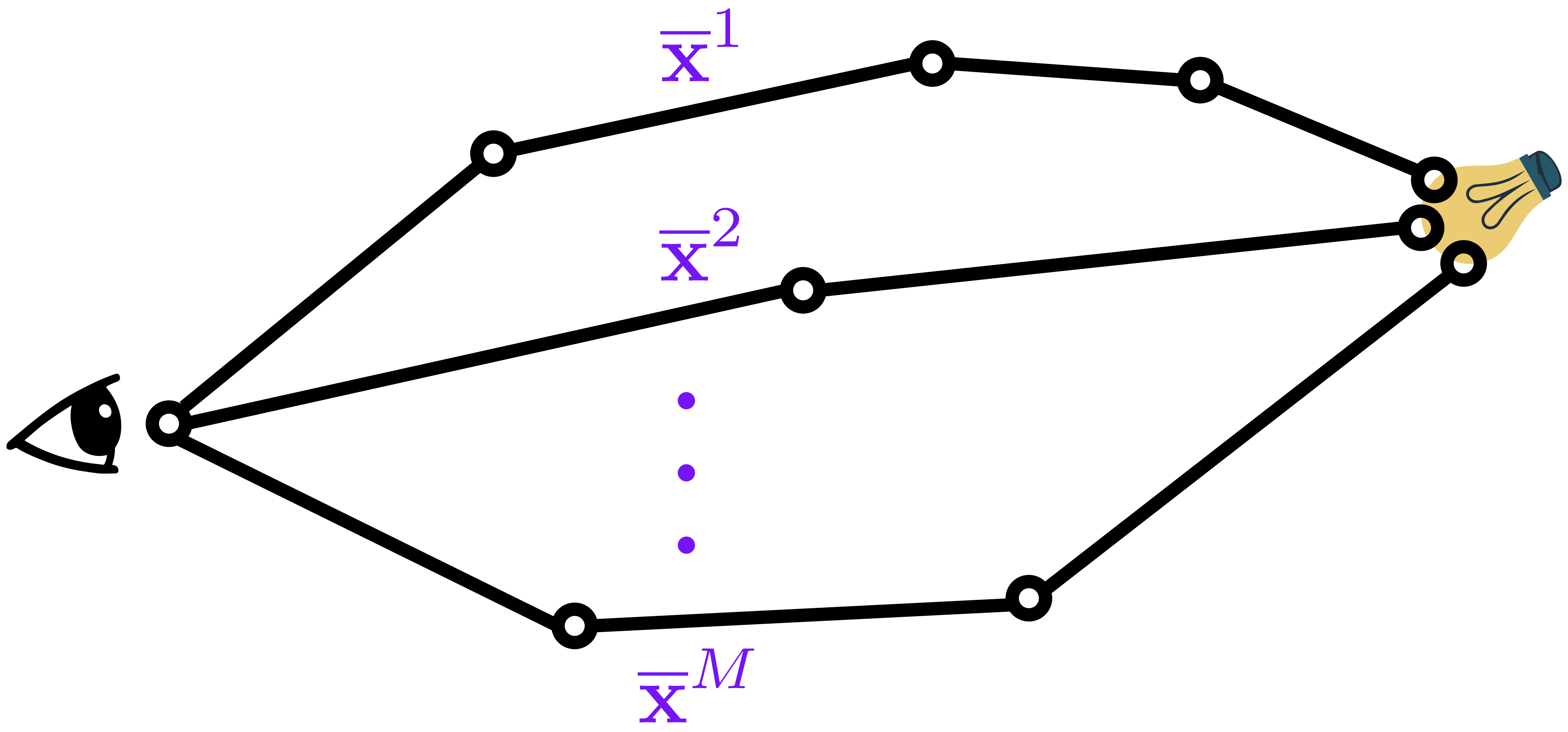
Path integral



$$I = \int_{\mathcal{P}} f(\overline{\mathbf{x}})\mathrm{d}\overline{\mathbf{x}}$$

value of a pixel

Path integral

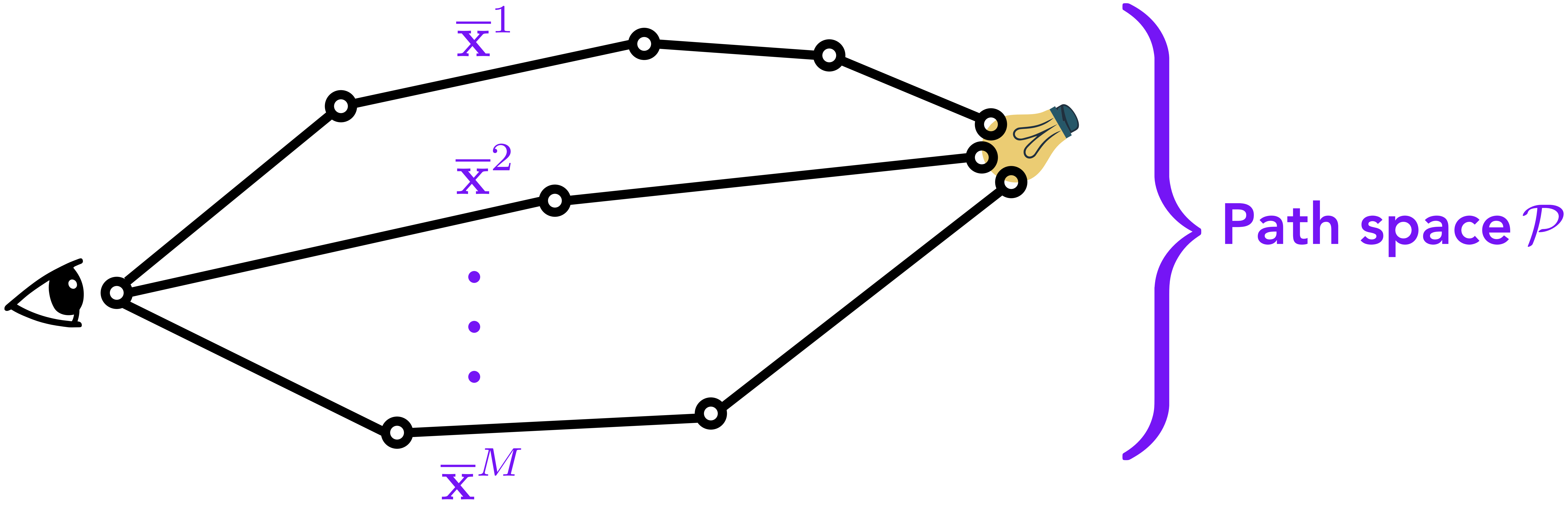


$$I = \int_{\mathcal{P}} f(\overline{\mathbf{x}}) \mathrm{d}\overline{\mathbf{x}}$$

value of a pixel

contribution of a path

Path integral



$$I = \int_{\mathcal{P}} f(\overline{\mathbf{x}}) \mathrm{d}\overline{\mathbf{x}}$$

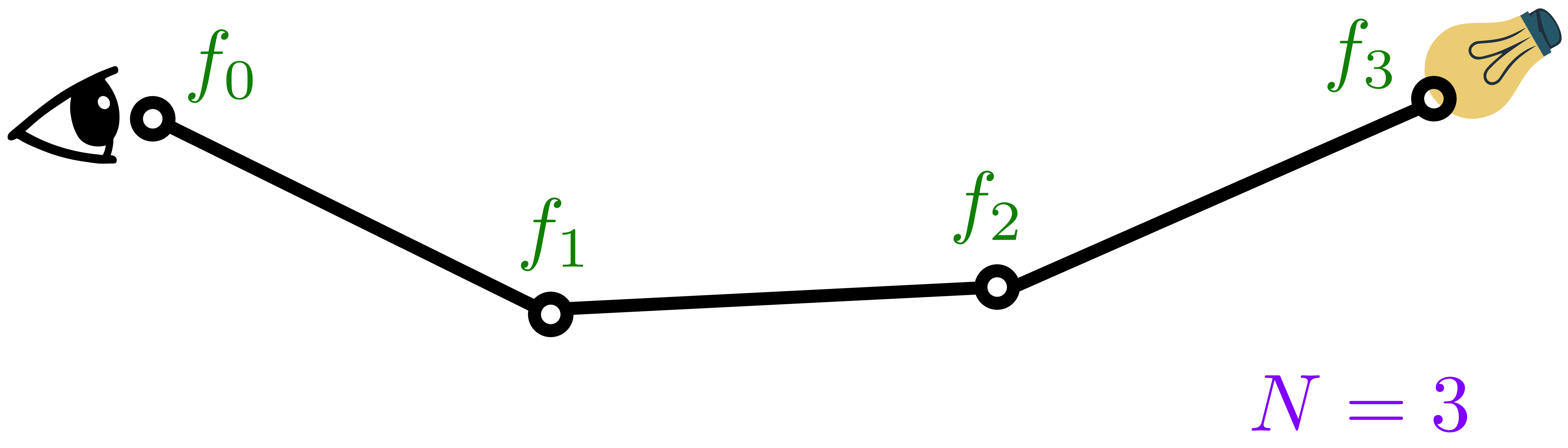
I: value of a pixel

$f(\overline{\mathbf{x}})$: contribution of a path

Path integral

$$I = \int_{\mathcal{P}} f(\overline{\mathbf{x}}) \mathrm{d}\overline{\mathbf{x}}$$

Path integral



$$I = \int_{\mathcal{P}} f_0 f_1 \cdots f_N \, \mathrm{d}\overline{\mathbf{x}}$$

product of contributions

Estimating the path integral

$$I = \int_{\mathcal{P}} f(\overline{\mathbf{x}}) \mathrm{d}\overline{\mathbf{x}}$$

$$\approx \frac{1}{M} \sum_{m=1}^{M} \frac{f(\overline{\mathbf{x}}^m)}{p(\overline{\mathbf{x}}^m)}$$

path contribution

probability of sampling the path $\overline{\mathbf{x}}^m$

Estimating the path integral

$$I = \int_{\mathcal{P}} f(\overline{\mathbf{x}}) \mathrm{d}\overline{\mathbf{x}}$$

$$\approx \frac{1}{M} \sum_{m=1}^{M} \frac{f(\overline{\mathbf{x}}^m)}{p(\overline{\mathbf{x}}^m)}$$

path contribution

probability of sampling the path $\overline{\mathbf{x}}^m$

★ p **should be a good approximation of** f

Estimating the path integral

$$I = \int_{\mathcal{P}} f(\overline{\mathbf{x}})\mathrm{d}\overline{\mathbf{x}}$$

$$\approx \frac{1}{M} \sum_{m=1}^{M} \frac{f(\overline{\mathbf{x}}^m)}{p(\overline{\mathbf{x}}^m)}$$

path contribution

probability of sampling the path $\overline{\mathbf{x}}^m$

★ p **should be a good approximation of** f

✓ **Forward rendering:** Use BRDF sampling at every vertex

✗ **Differentiable rendering:** Need a better strategy

Derivative of path integral

$$\partial_\theta I = \int_{\mathcal{P}} \partial_\theta \big[f_0 f_1 \cdots f_N \big] \mathrm{d}\overline{\mathbf{x}}$$

Derivative of path integral

$$\partial_\theta I = \int_{\mathcal{P}} \partial_\theta \big[f_0 f_1 \cdots f_N \big] \mathrm{d}\overline{\mathbf{x}}$$

$$= \int_{\mathcal{P}} \Big[(\partial_\theta f_0) f_1 \cdots f_N + f_0 (\partial_\theta f_1) f_2 \cdots f_N + \ldots + f_0 \cdots f_{N-1} (\partial_\theta f_N) \Big] \, \mathrm{d}\overline{\mathbf{x}}$$

Derivative of path integral

$$\partial_\theta I = \int_{\mathcal{P}} \partial_\theta \big[f_0 f_1 \cdots f_N\big] \mathrm{d}\overline{\mathbf{x}}$$

$$= \int_{\mathcal{P}} \Big[(\partial_\theta f_0) f_1 \cdots f_N + f_0 (\partial_\theta f_1) f_2 \cdots f_N + \ldots + f_0 \cdots f_{N-1} (\partial_\theta f_N)\Big] \, \mathrm{d}\overline{\mathbf{x}}$$

Derivative of path integral

$$\partial_\theta I = \int_{\mathcal{P}} \partial_\theta \left[f_0 f_1 \cdots f_N \right] \mathrm{d}\overline{\mathbf{x}}$$

$$= \int_{\mathcal{P}} \left[(\partial_\theta f_0) f_1 \cdots f_N + f_0 (\partial_\theta f_1) f_2 \cdots f_N + \ldots + f_0 \cdots f_{N-1} (\partial_\theta f_N) \right] \mathrm{d}\overline{\mathbf{x}}$$

$g_0(\overline{\mathbf{x}})$ $g_1(\overline{\mathbf{x}})$ $g_N(\overline{\mathbf{x}})$

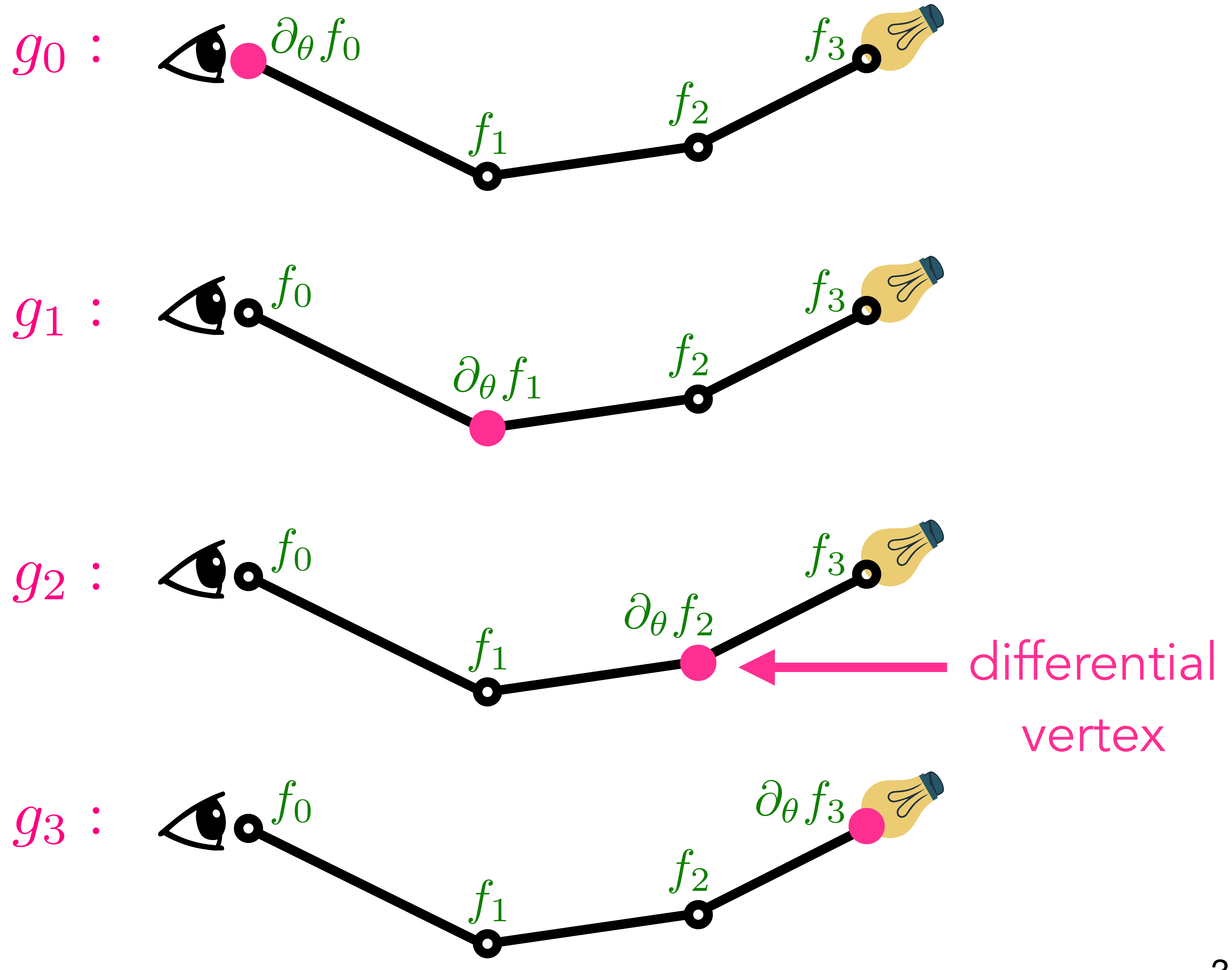
differential contributions

Derivative of path integral

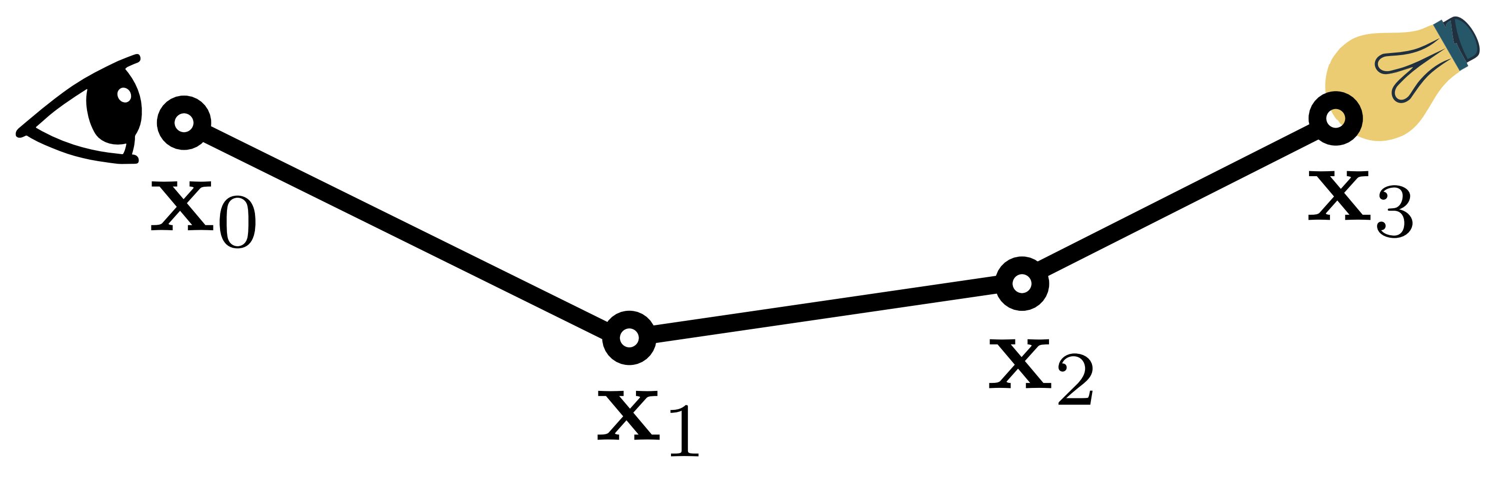
$$
\begin{aligned}
\partial_\theta I &= \int_{\mathcal{P}} \partial_\theta \big[f_0 f_1 \cdots f_N \big] \mathrm{d}\overline{\mathbf{x}} \\
&= \int_{\mathcal{P}} \Big[(\partial_\theta f_0) f_1 \cdots f_N + f_0 (\partial_\theta f_1) f_2 \cdots f_N + \ldots + f_0 \cdots f_{N-1} (\partial_\theta f_N) \Big] \, \mathrm{d}\overline{\mathbf{x}} \\
&= \int_{\mathcal{P}} \sum_{n=0}^{N} g_n(\overline{\mathbf{x}}) \, \mathrm{d}\overline{\mathbf{x}}
\end{aligned}
$$

Differential contributions

$$\partial_\theta I = \int_{\mathcal{P}} \sum_{n=0}^{N} g_n(\overline{\mathbf{x}})\, \mathrm{d}\overline{\mathbf{x}}$$

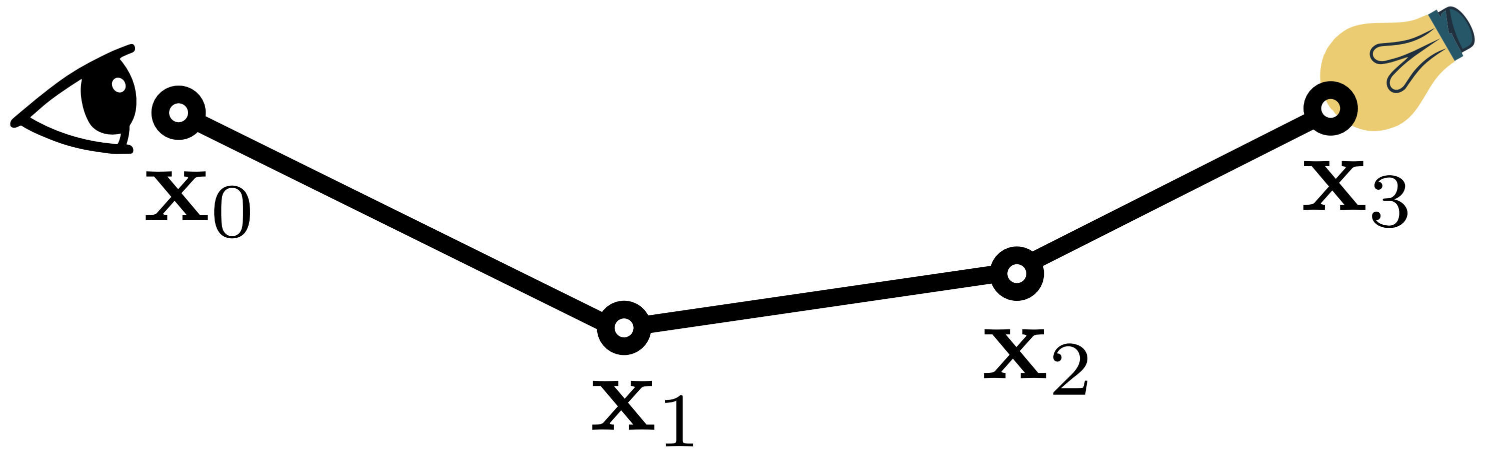


Differential path space

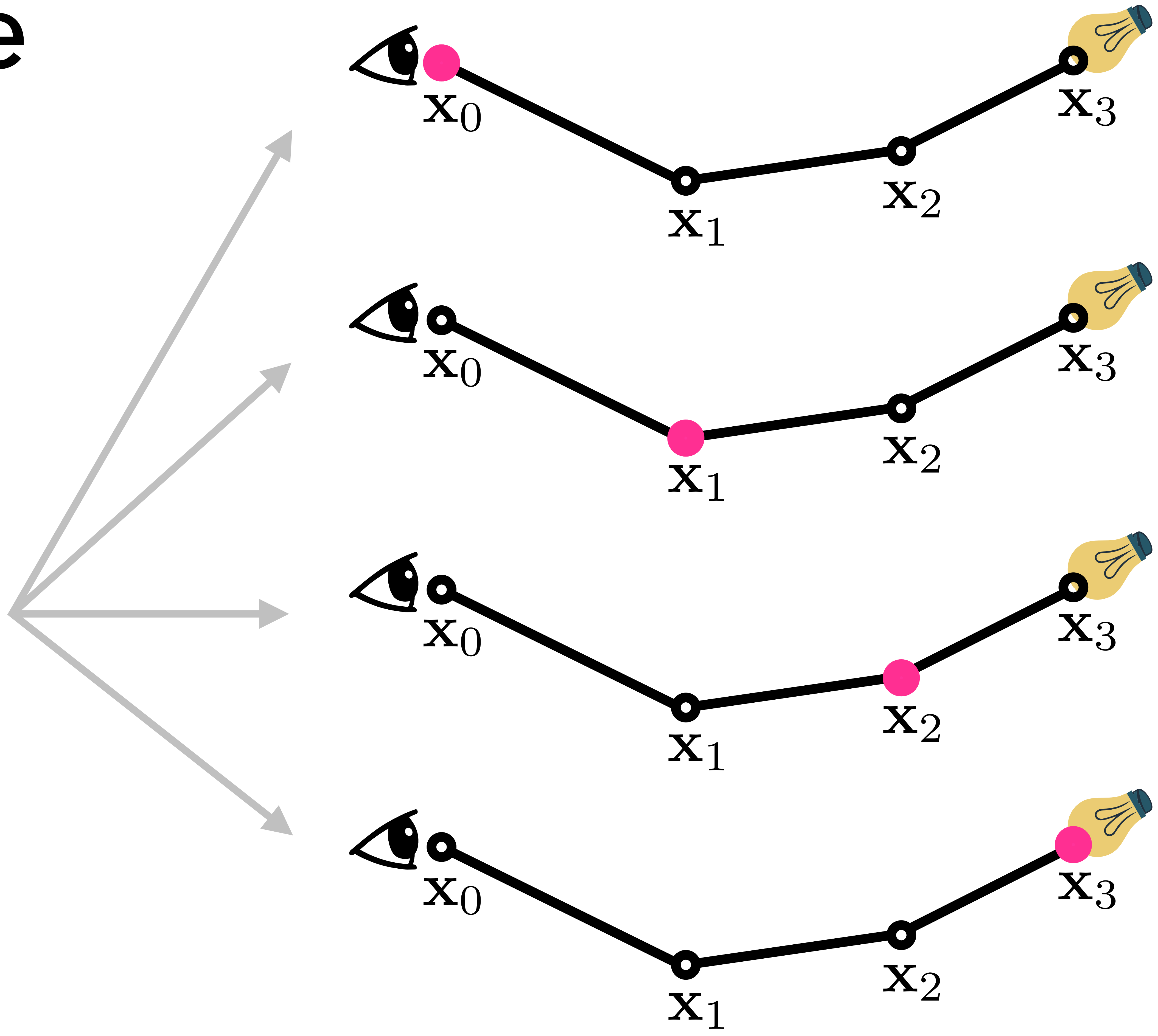


Path space $\mathcal{P}$

Differential path space



Path space $\mathcal{P}$



Differential path space $\partial\mathcal{P}$

Differential path space integral

$$\partial_\theta I = \int_{\mathcal{P}} \sum_{n=0}^{N} g_n(\overline{\mathbf{x}}) \, \mathrm{d}\overline{\mathbf{x}}$$

Differential path space integral

Our formulation

$$\partial_\theta I = \int_{\mathcal{P}} \sum_{n=0}^{N} g_n(\overline{\mathbf{x}})\,\mathrm{d}\overline{\mathbf{x}} = \int_{\partial\mathcal{P}} g_n(\overline{\mathbf{x}})\mathrm{d}\overline{\mathbf{x}}$$

($\overline{\mathbf{x}}$ has differential vertex at $\mathbf{x}_n$)

Differential path space integral

Our formulation

$$\partial_\theta I = \int_{\mathcal{P}} \sum_{n=0}^{N} g_n(\overline{\mathbf{x}}) \, \mathrm{d}\overline{\mathbf{x}} = \int_{\partial\mathcal{P}} g_n(\overline{\mathbf{x}}) \mathrm{d}\overline{\mathbf{x}}$$

Path space integral

Differential path space integral

($\overline{\mathbf{x}}$ has differential vertex at $\mathbf{x}_n$)

Our method

Forward rendering

$$I = \int_{\mathcal{P}} f(\overline{\mathbf{x}}) \mathrm{d}\overline{\mathbf{x}}$$

Sample paths from $\mathcal{P}$ proportionally to f

Differentiable rendering

$$\partial_\theta I = \int_{\partial\mathcal{P}} g_n(\overline{\mathbf{x}}) \mathrm{d}\overline{\mathbf{x}}$$

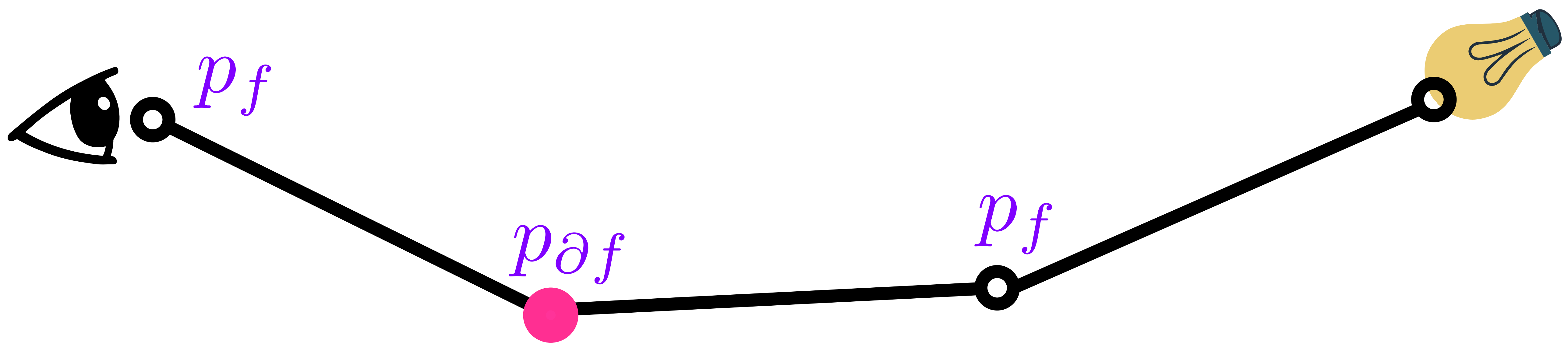
Sample paths from $\partial\mathcal{P}$ proportionally to g

Sampling paths from $\partial\mathcal{P}$

Sampling paths from $\partial\mathcal{P}$

Need a path with one differential vertex

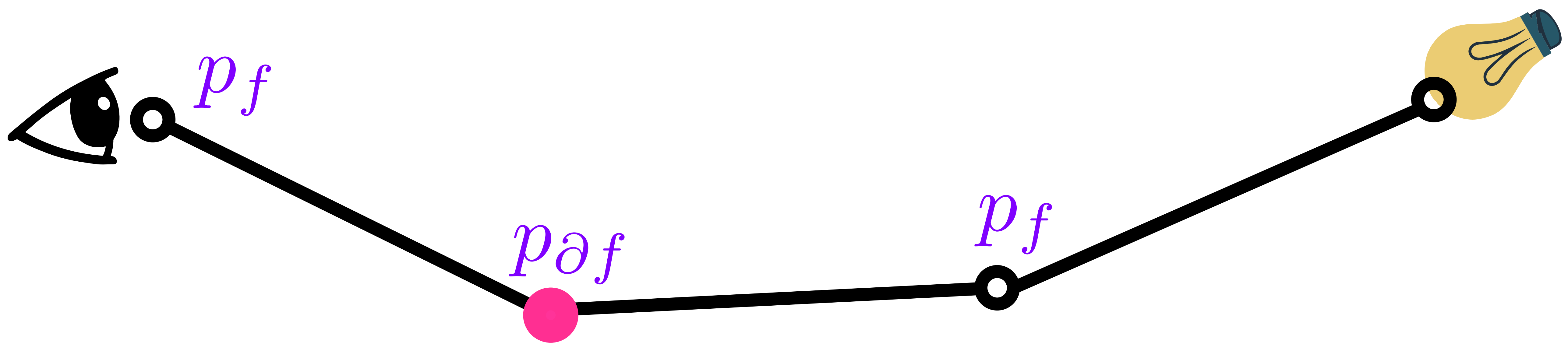
Sampling paths from $\partial\mathcal{P}$



p_f = standard pdf

$p_{\partial f}$ = differential pdf

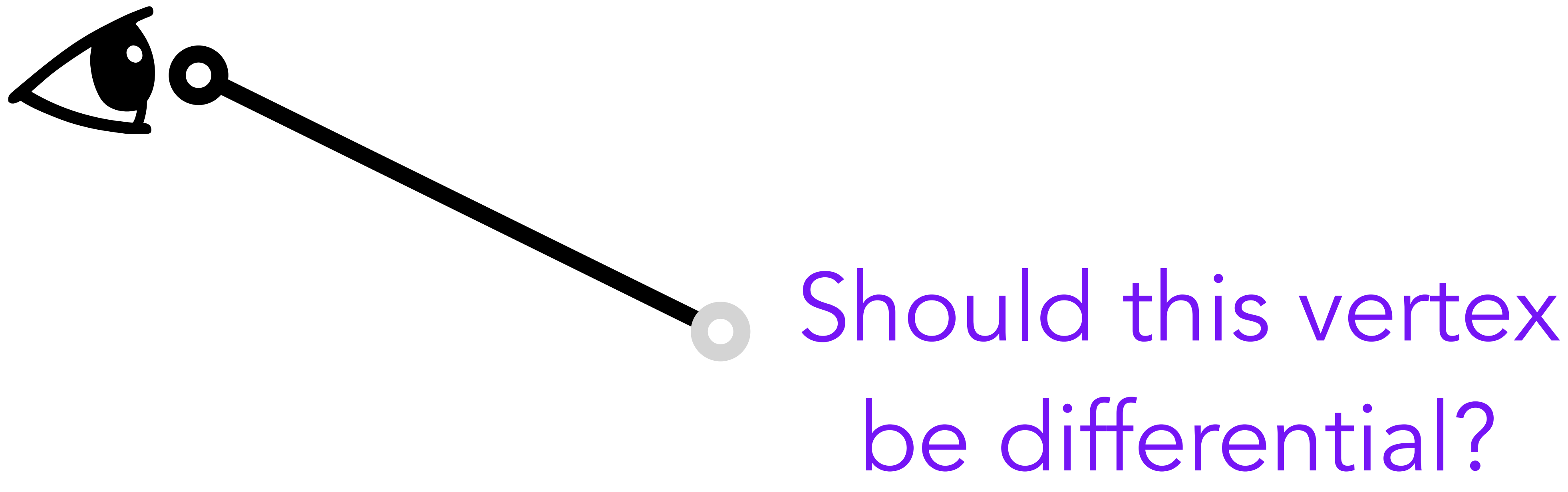
Sampling paths from $\partial\mathcal{P}$



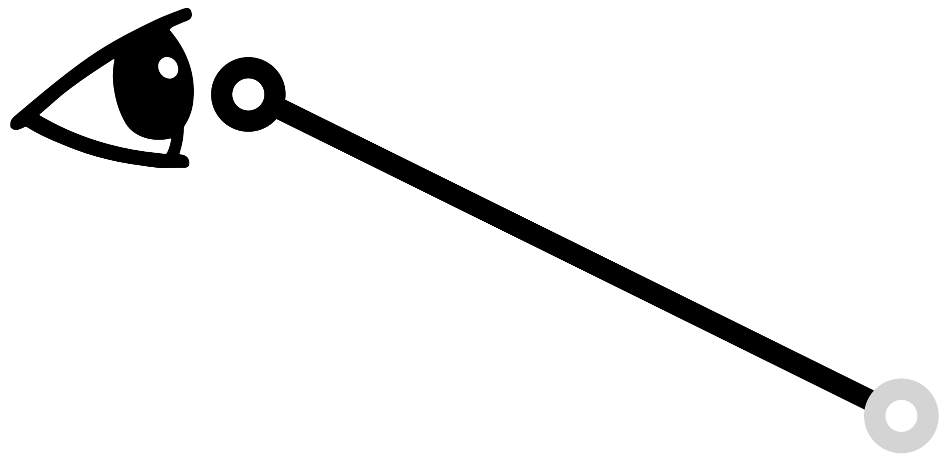
p_f = standard pdf, proportional to the BRDF

$p_{\partial f}$ = differential pdf, proportional to the BRDF's derivative

Choosing the differential vertex

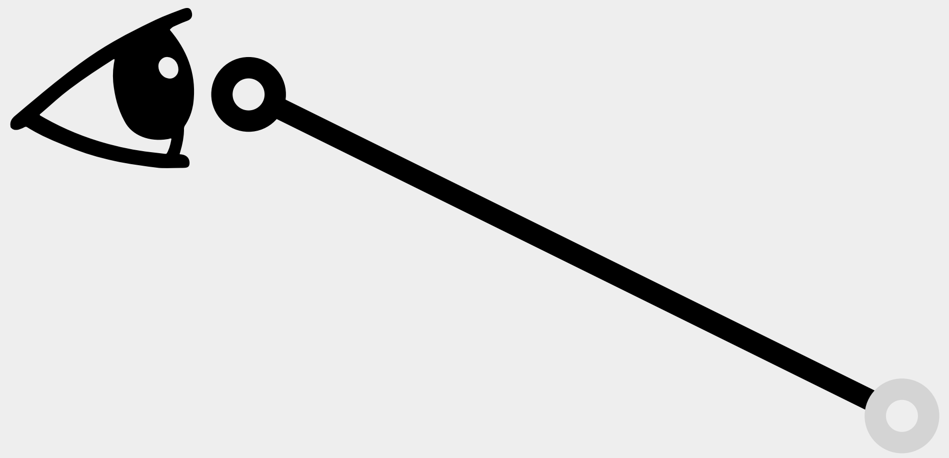


Choosing the differential vertex



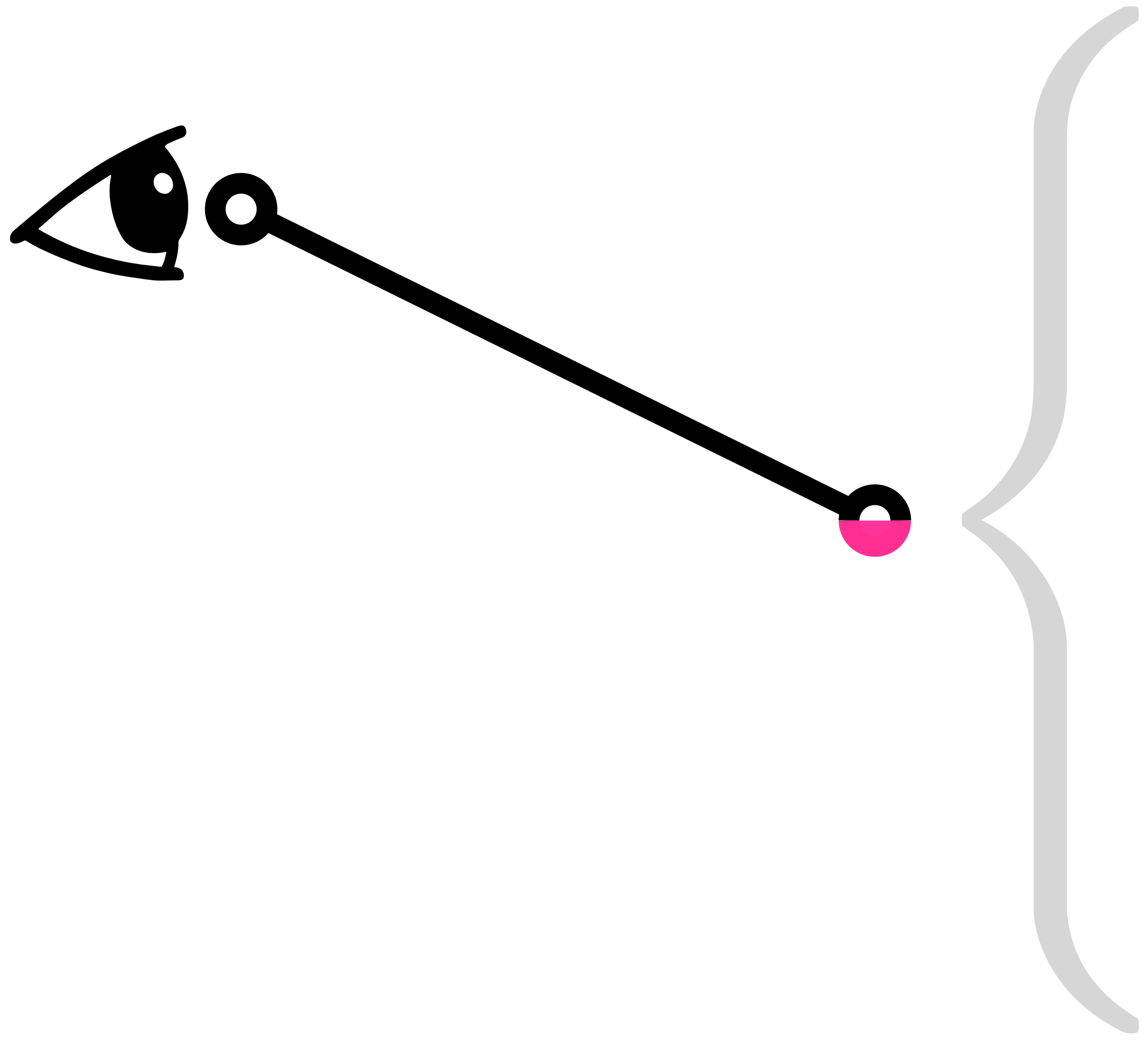
Should this vertex
be differential?

Russian roulette



Terminate path at
this vertex?

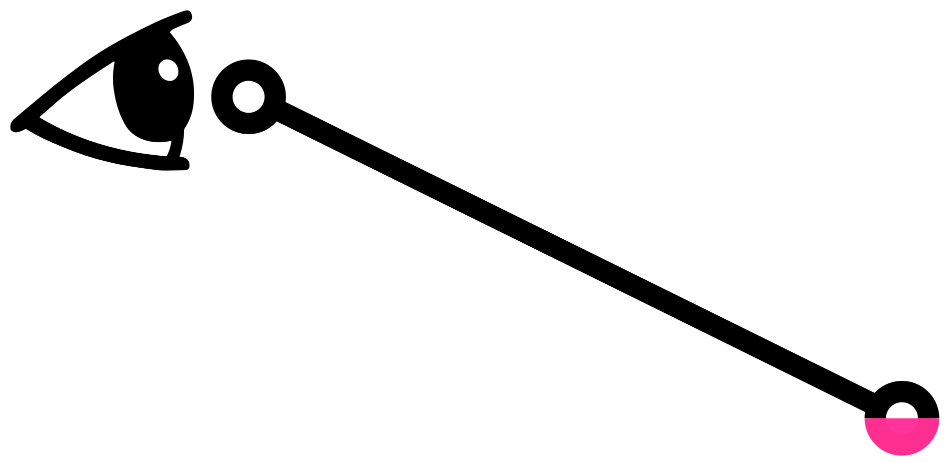
Choosing the differential vertex



probability $1 - q$

probability q

Choosing the differential vertex

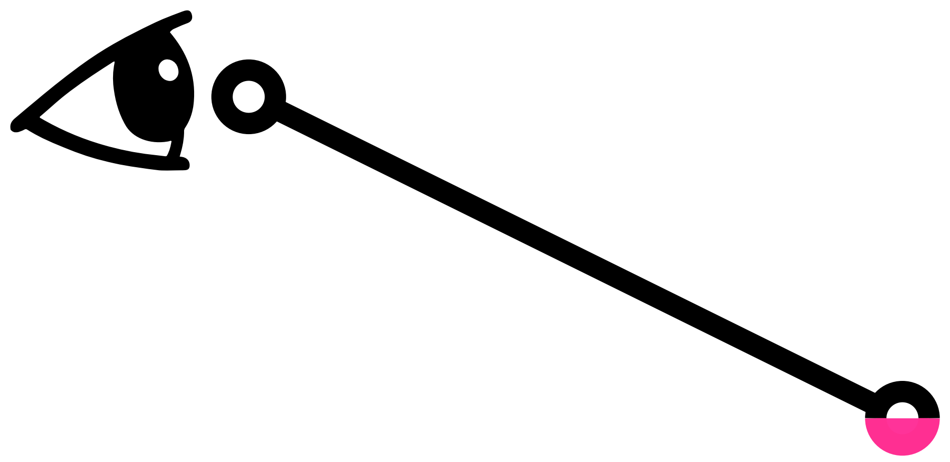


probability $1 - q$ → **normal vertex**

sample p_f, weight $\frac{1}{p_f \cdot (1-q)}$

probability q

Choosing the differential vertex



probability $1 - q$ → **normal vertex**

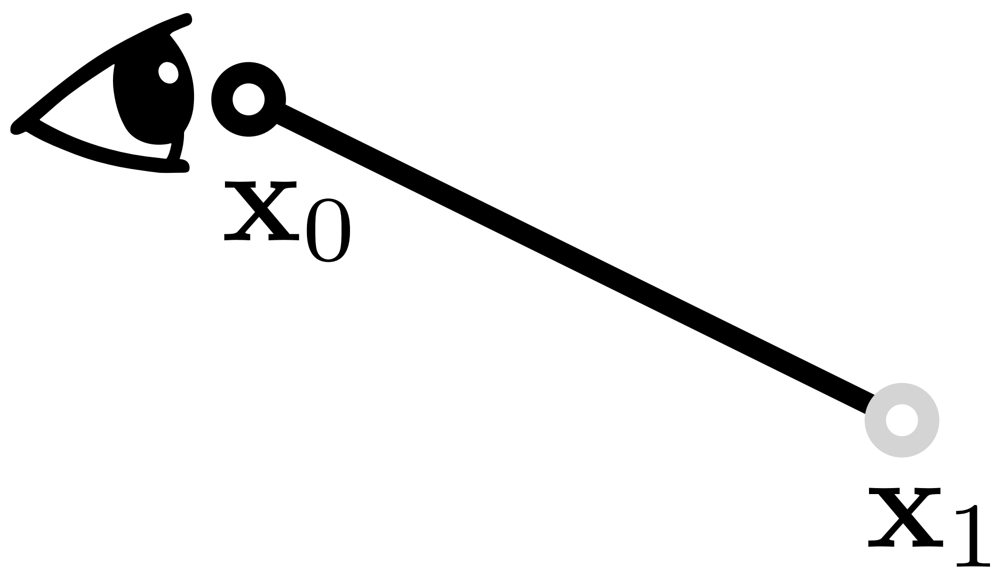
sample p_f, weight $\frac{1}{p_f \cdot (1-q)}$

probability q → **differential vertex**

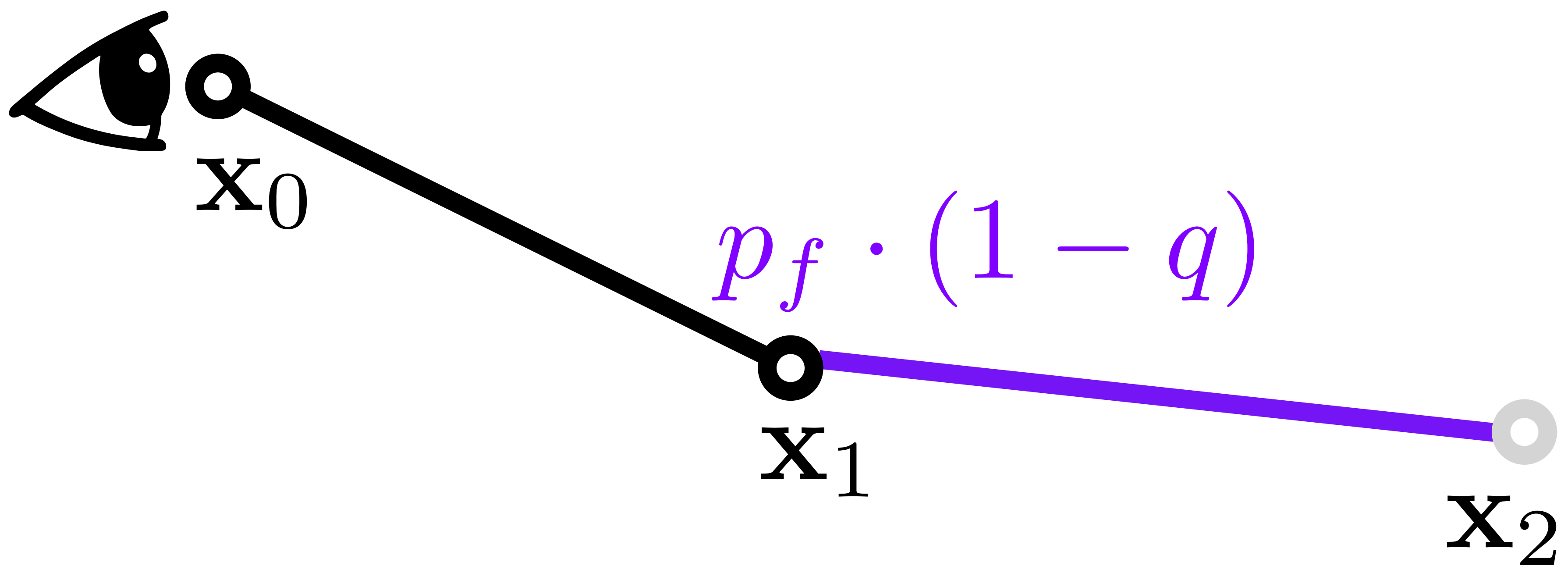
sample $p_{\partial f}$, weight $\frac{1}{p_{\partial f} \cdot q}$

set $q = 0$ for later vertices

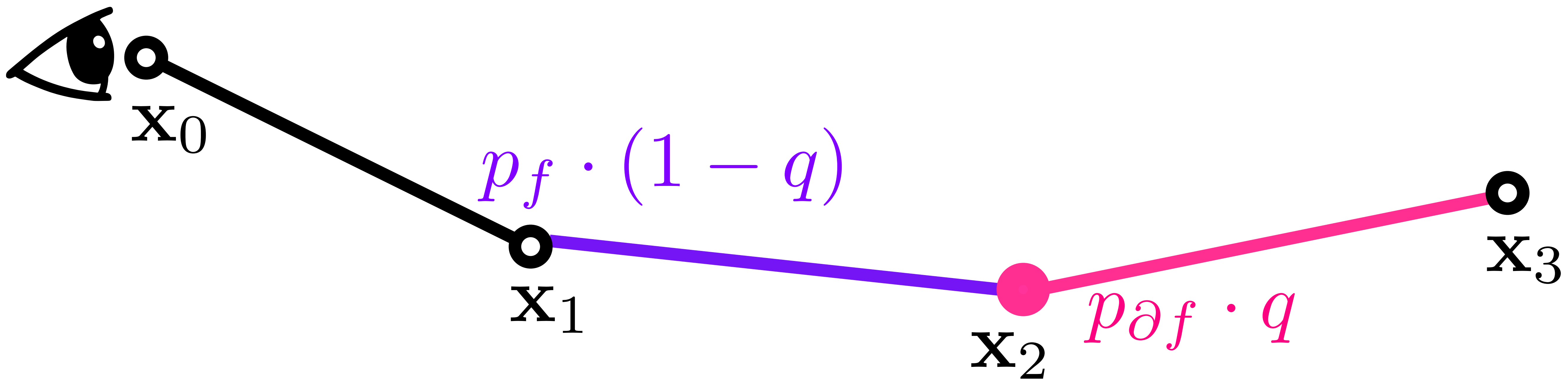
Sampling probability for a full path



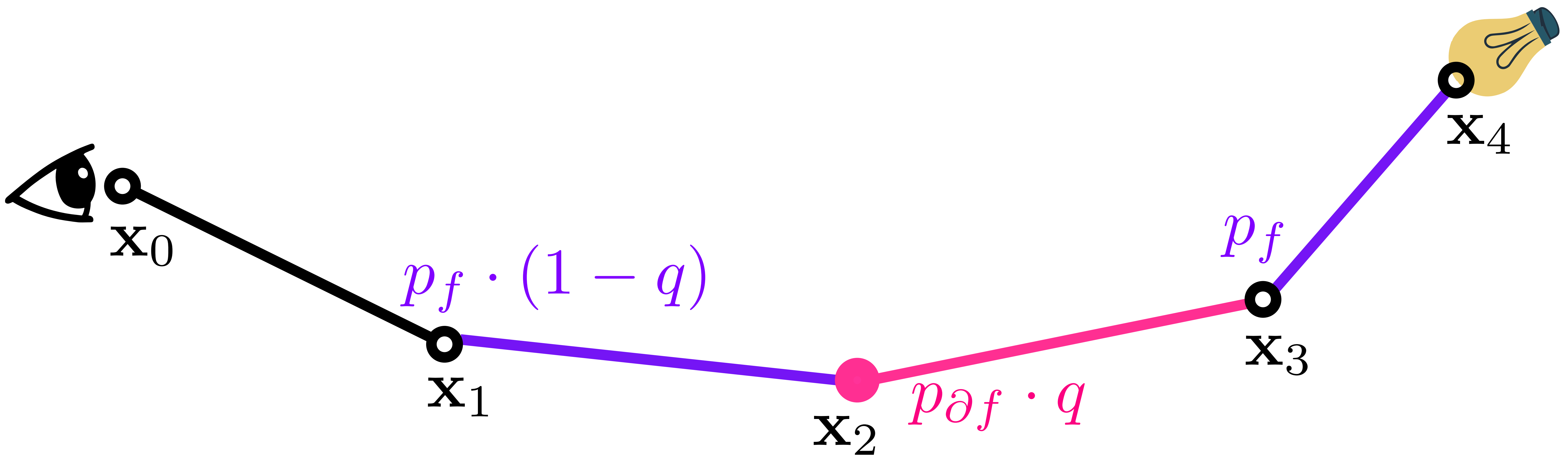
Sampling probability for a full path



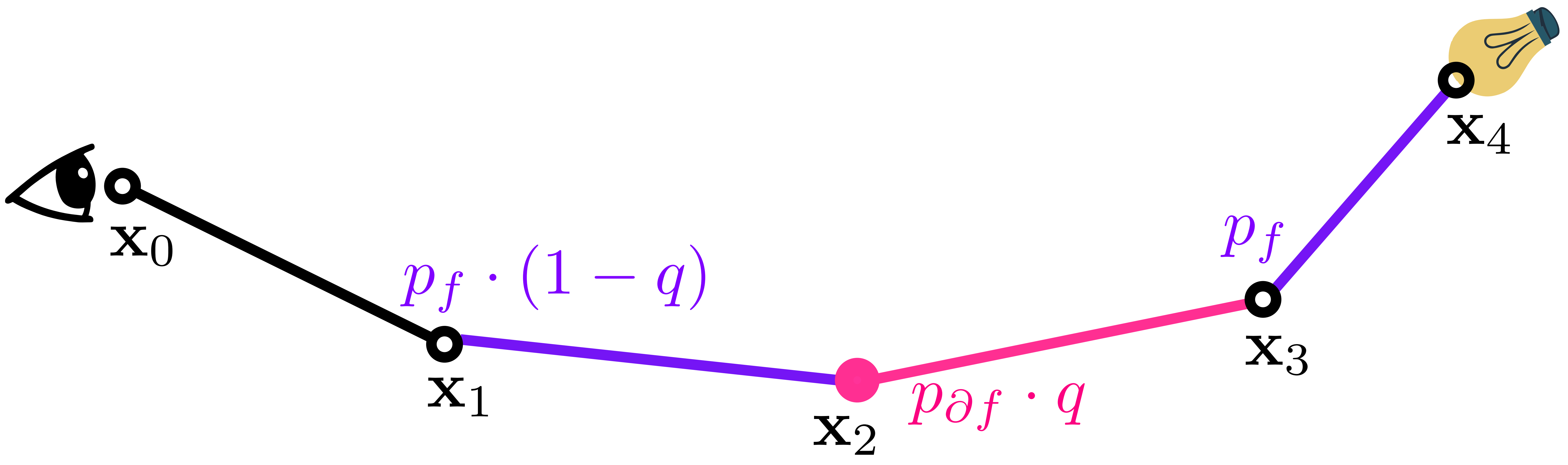
Sampling probability for a full path



Sampling probability for a full path



Sampling probability for a full path

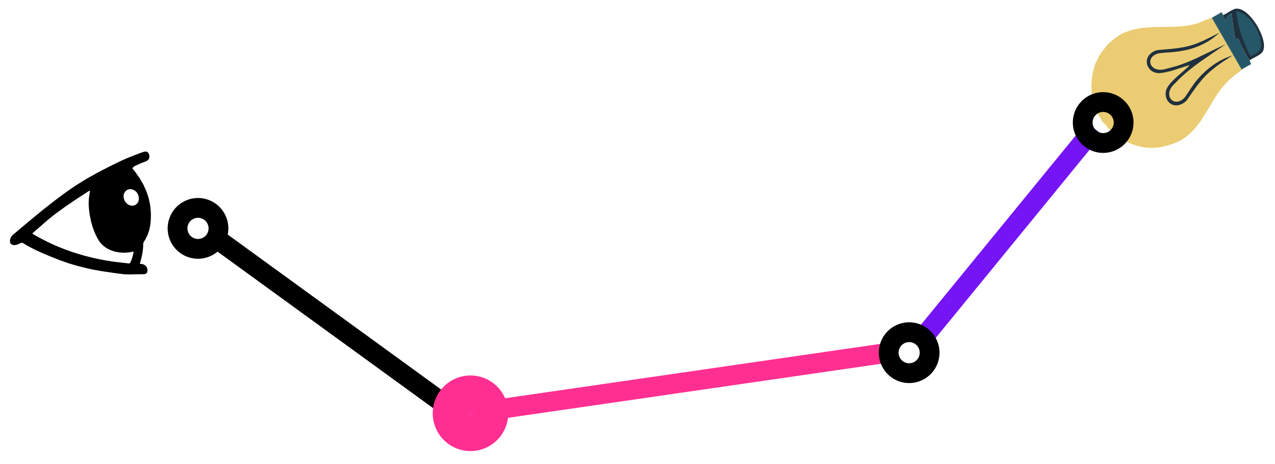


$$p_{\text{path}} = p_f(\mathbf{x}_1) \cdot (1-q) \cdot p_{\partial f}(\mathbf{x}_2) \cdot q \cdot p_f(\mathbf{x}_3)$$

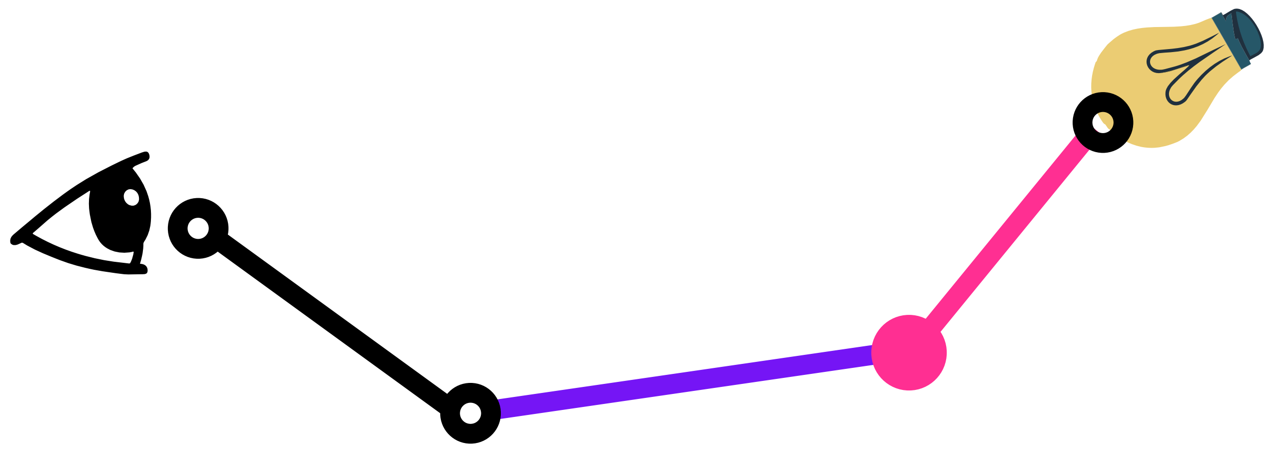
weight path contribution by $\dfrac{1}{p_{\text{path}}}$

Multiple importance sampling (MIS)

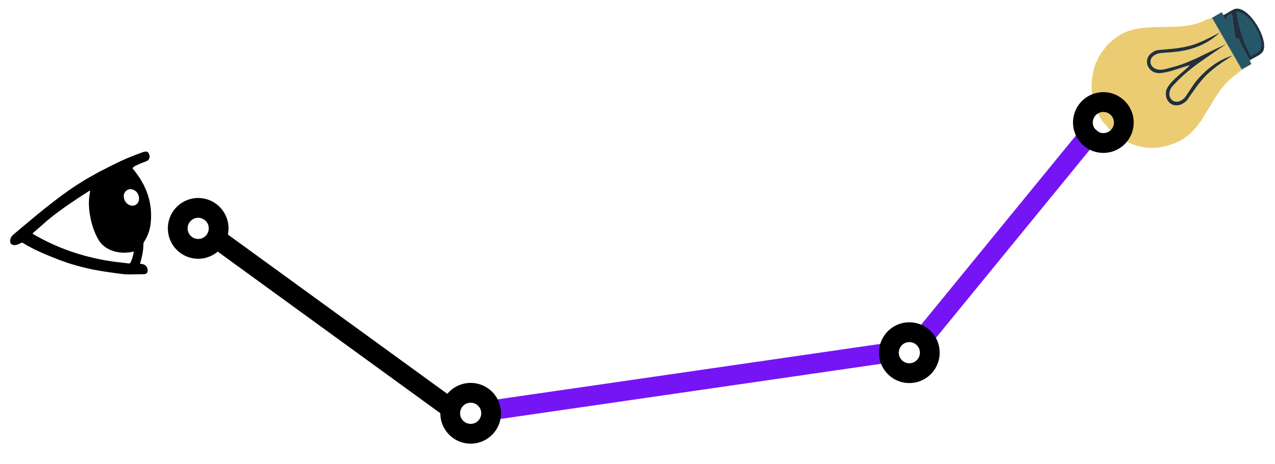
Multiple importance sampling (MIS)



Differential vertex at $\mathbf{x}_1$

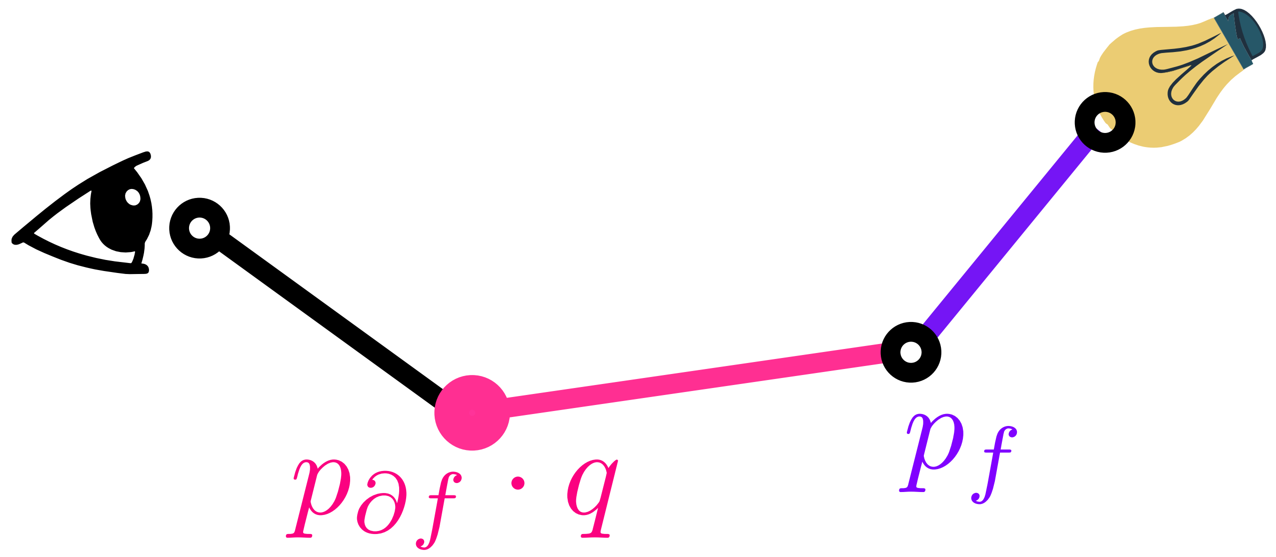


Differential vertex at $\mathbf{x}_2$



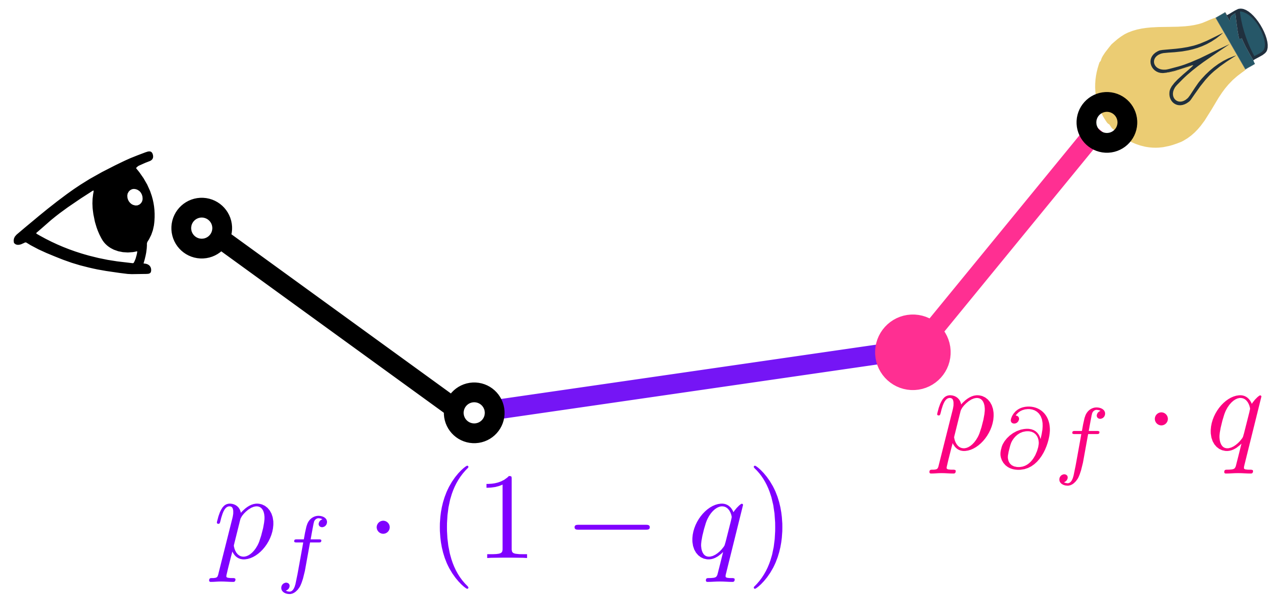
No differential vertex

Multiple importance sampling (MIS)



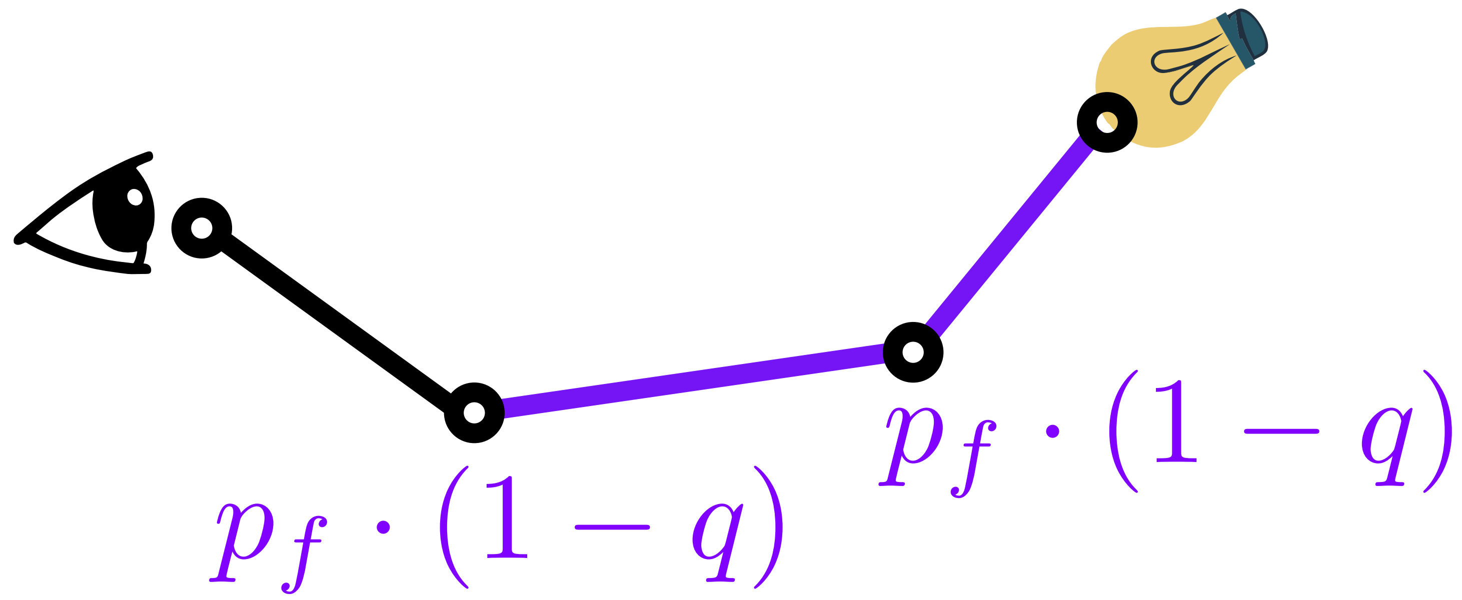
Differential vertex at $\mathbf{x}_1$

$p_{\text{path}} = p_{\partial f}(\mathbf{x}_1) \cdot q \cdot p_f(\mathbf{x}_2)$



Differential vertex at $\mathbf{x}_2$

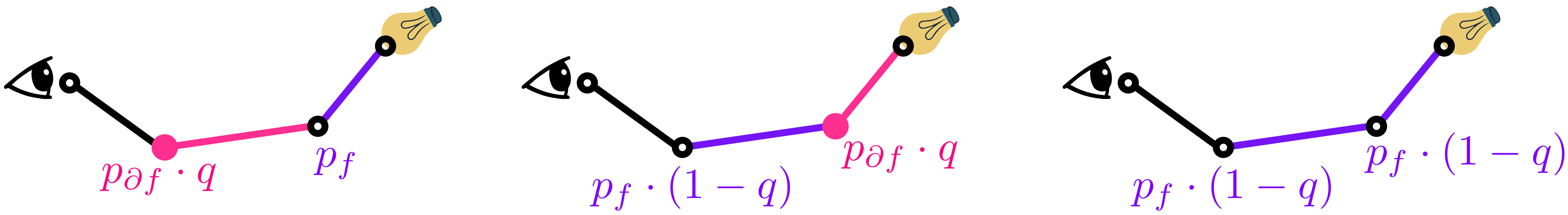
$p_{\text{path}} = p_f(\mathbf{x}_1) \cdot (1-q) \cdot p_{\partial f}(\mathbf{x}_2) \cdot q$



No differential vertex

$p_{\text{path}} = p_f(\mathbf{x}_1) \cdot (1-q) \cdot p_f(\mathbf{x}_2) \cdot (1-q)$

Multiple importance sampling (MIS)

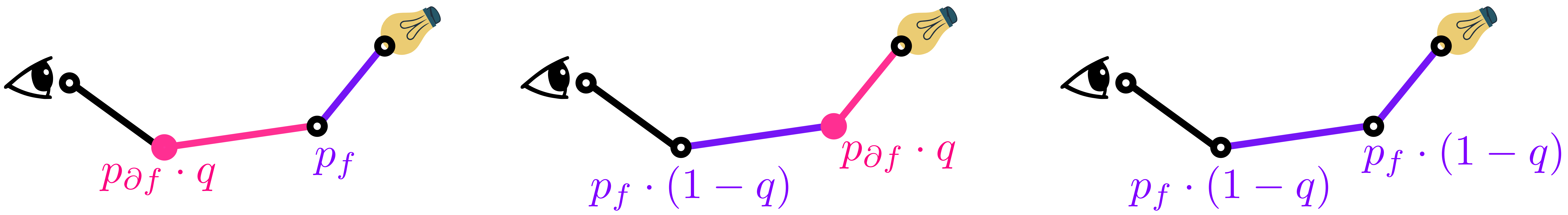


Differential vertex at $\mathbf{x}_1$ | Differential vertex at $\mathbf{x}_2$ | No differential vertex

$$p_{\text{mixture}} = p_{\partial f}(\mathbf{x}_1) \cdot q \cdot p_f(\mathbf{x}_2) + p_f(\mathbf{x}_1) \cdot (1-q) \cdot p_{\partial f}(\mathbf{x}_2) \cdot q + p_f(\mathbf{x}_1) \cdot (1-q) \cdot p_f(\mathbf{x}_2) \cdot (1-q)$$

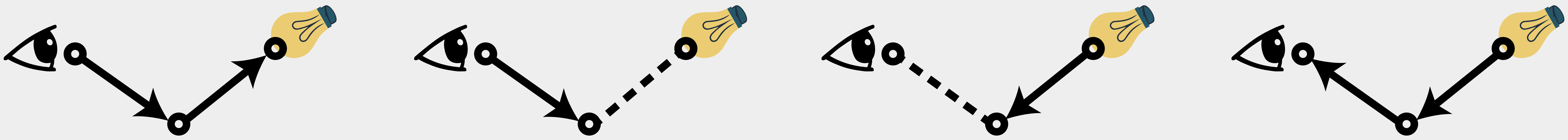
weight path contribution by $\dfrac{1}{p_{\text{mixture}}}$

Multiple importance sampling (MIS)

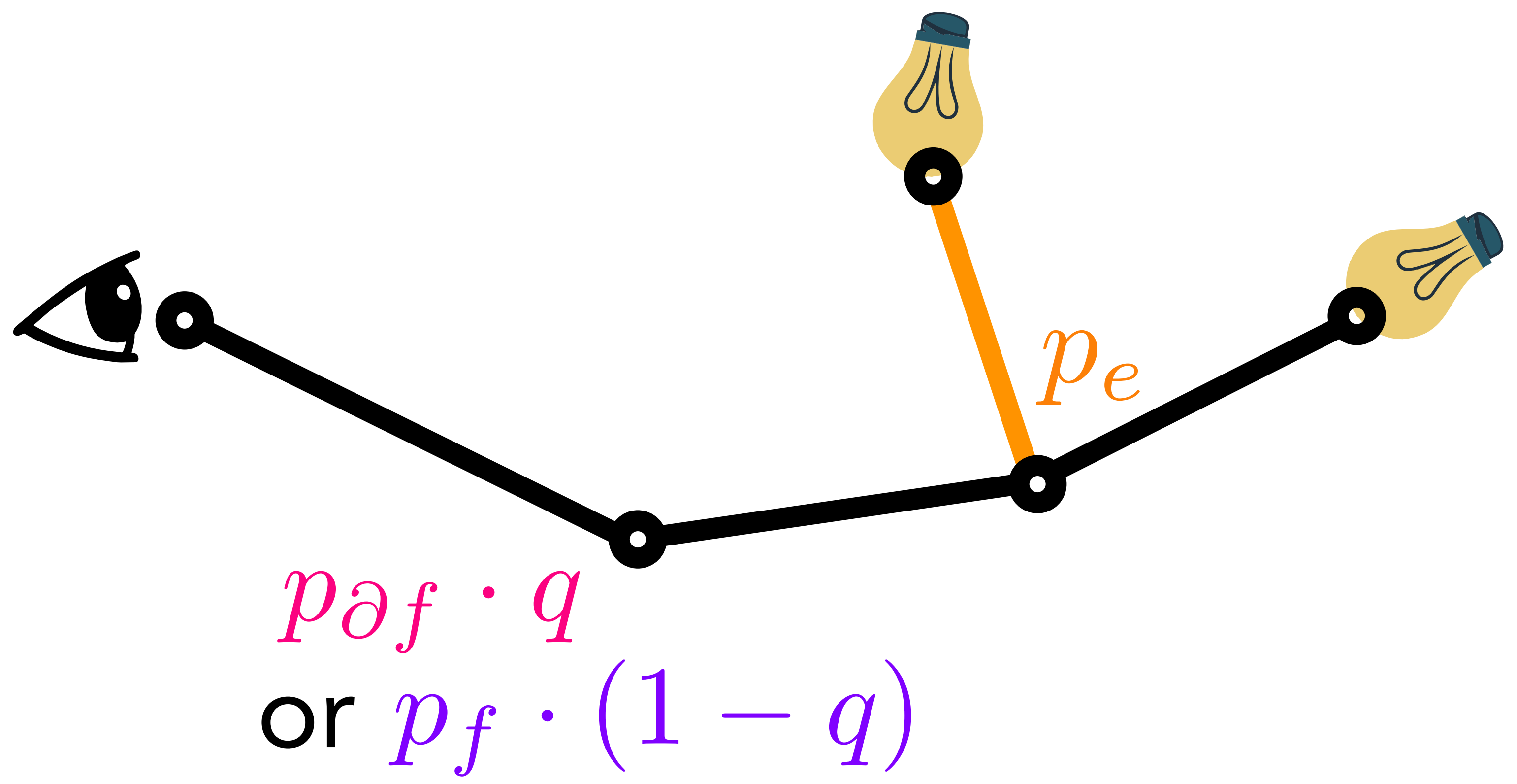


Differential vertex at $\mathbf{x}_1$ | Differential vertex at $\mathbf{x}_2$ | No differential vertex

Bidirectional path tracing



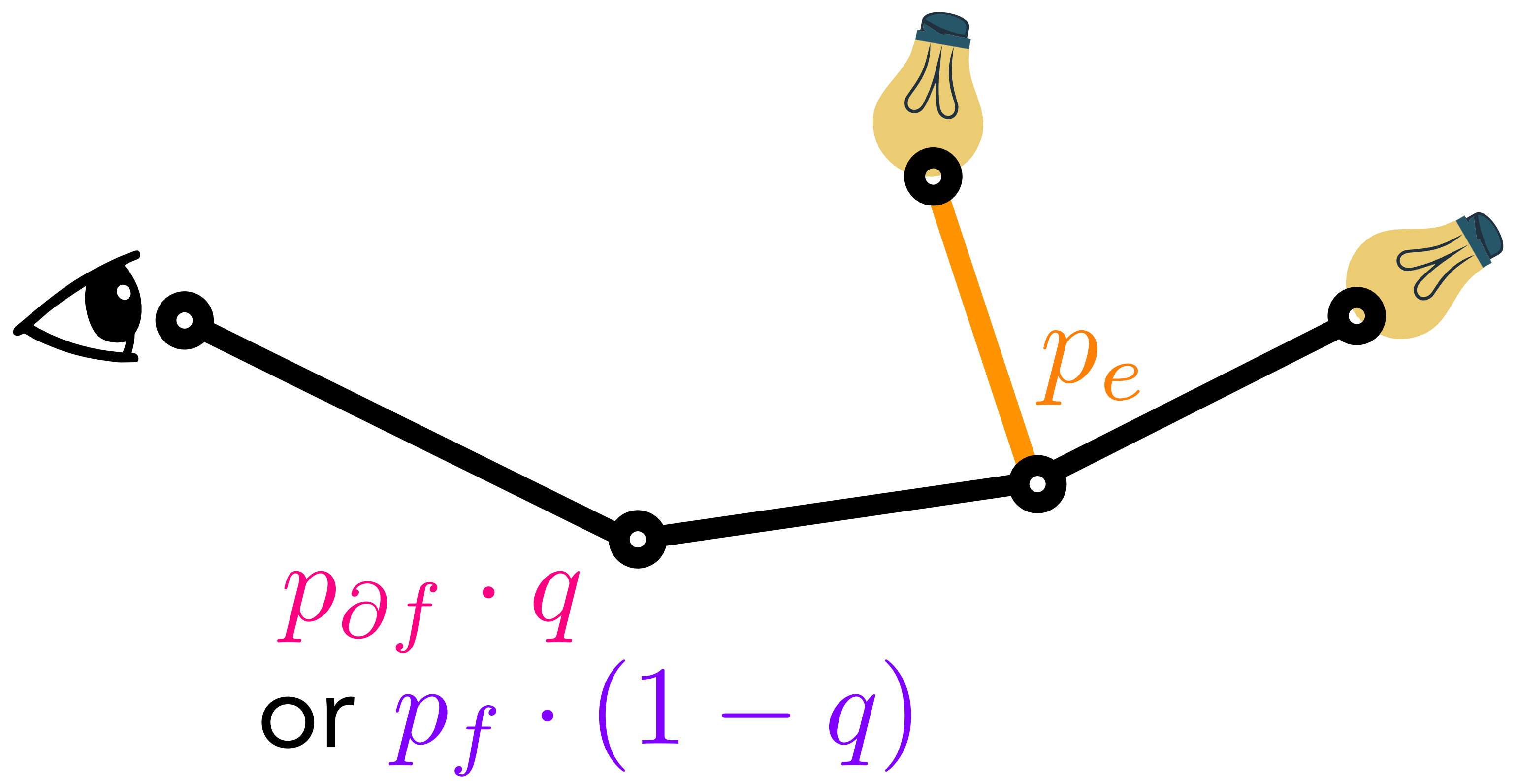
Next event estimation (NEE)



NEE connection

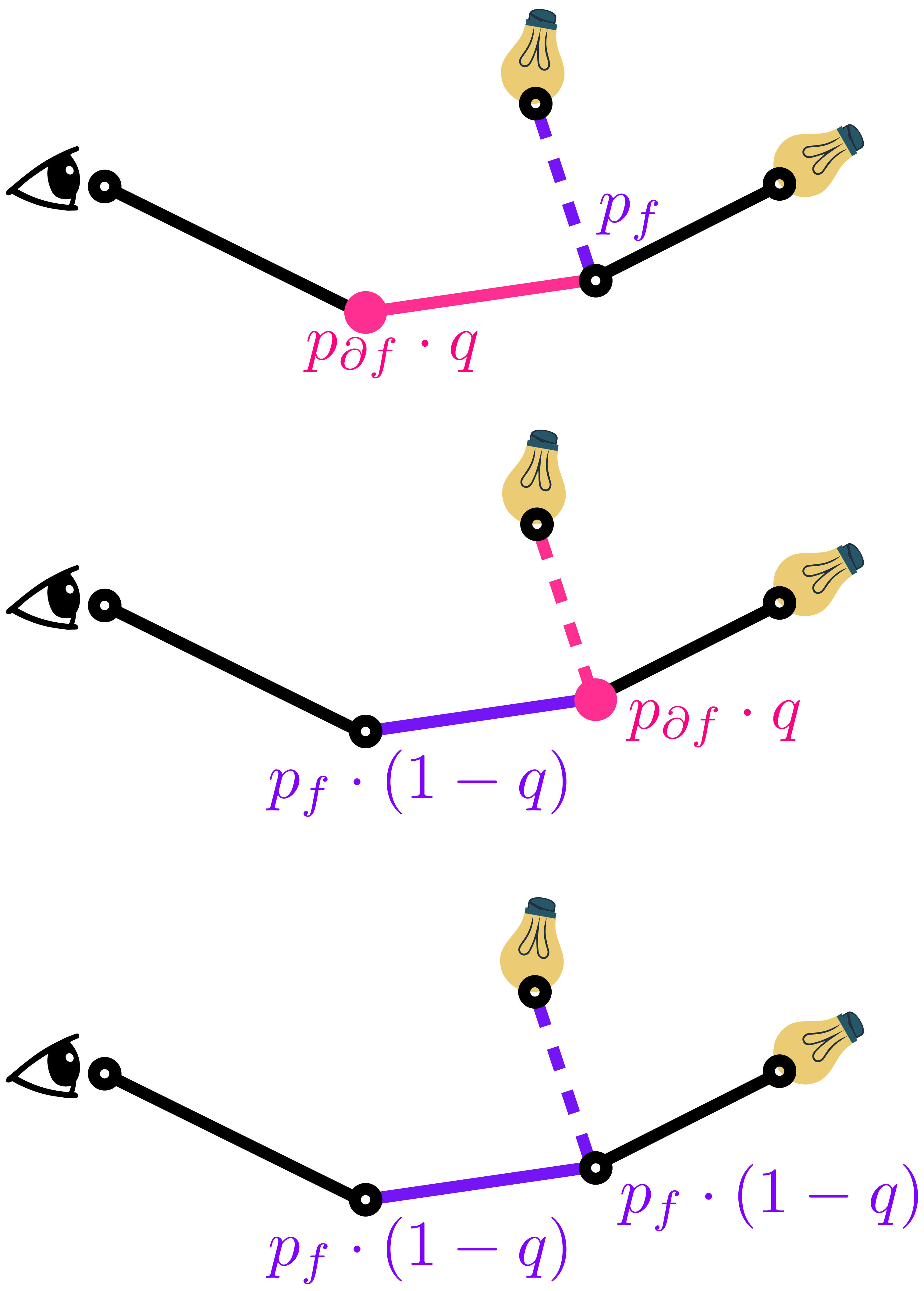
p_e = emitter pdf

Next event estimation (NEE)



NEE connection

p_e = emitter pdf



Other possible sampling methods

★ **Details are in the paper**

Computing path contributions

Path space integral

Differential path space integral

$$\partial_\theta I = \int_{\mathcal{P}} \sum_{n=0}^{N} g_n(\overline{\mathbf{x}})\, \mathrm{d}\overline{\mathbf{x}} = \int_{\partial\mathcal{P}} g_n(\overline{\mathbf{x}})\mathrm{d}\overline{\mathbf{x}}$$

Sample paths from here

Computing path contributions

Path space integral

Differential path space integral

$$\partial_\theta I = \int_{\mathcal{P}} \sum_{n=0}^{N} g_n(\overline{\mathbf{x}})\, \mathrm{d}\overline{\mathbf{x}} = \int_{\partial\mathcal{P}} g_n(\overline{\mathbf{x}}) \mathrm{d}\overline{\mathbf{x}}$$

Computing path contributions

Path space integral

Differential path space integral

$$\partial_\theta I = \int_{\mathcal{P}} \sum_{n=0}^{N} g_n(\overline{\mathbf{x}})\, \mathrm{d}\overline{\mathbf{x}} = \int_{\partial\mathcal{P}} g_n(\overline{\mathbf{x}})\mathrm{d}\overline{\mathbf{x}}$$

Computing path contributions

Path space integral

Differential path space integral

$$\partial_\theta I = \int_{\mathcal{P}} \sum_{n=0}^{N} g_n(\overline{\mathbf{x}})\,\mathrm{d}\overline{\mathbf{x}} = \int_{\partial\mathcal{P}} g_n(\overline{\mathbf{x}})\mathrm{d}\overline{\mathbf{x}}$$

Sample paths from here

Computing path contributions

Path space integral

Differential path space integral

$$\partial_\theta I = \int_{\mathcal{P}} \sum_{n=0}^{N} g_n(\overline{\mathbf{x}})\, \mathrm{d}\overline{\mathbf{x}} = \int_{\partial \mathcal{P}} g_n(\overline{\mathbf{x}}) \mathrm{d}\overline{\mathbf{x}}$$

Compute this integrand

Sample paths from here

Computing path contributions

```
function SamplePath(ray)
    L = 0, β = 1
    for i = 0 to N − 1 do
        L += β · L_e(...)
        ω′, f = Sample_BRDF(...)
        β *= f / p_f(ω′, ...)
    return L

function SamplePathAdjoint(ray, L, δL)
    β = 1
    for i = 0 to N − 1 do
        L -= β · L_e(...)
        ω′, f = Sample_BRDF(...)
        δ_θ += BackwardGrad(f, δL · L/f)
        β *= f / p_f(ω′, ...)
    return δ_θ
```

```
function SamplePath(ray)
    L = 0, β = 1, w_1 = 0, w_2 = 1, sampled_∂x = FALSE
    for i = 0 to N − 1 do
        L += β · L_e(...) / (w_1 + w_2)
        if !sampled_∂x and Rand() < q then
            ω′, f = Sample_∂BRDF(...)
            sampled_∂x = TRUE
        else
            ω′, f = Sample_BRDF(...)
        β *= f / p_f(ω′, ...)
        w_1 += w_2 · q · p_∂f(ω′, ...) / p_f(ω′, ...)
        w_2 *= 1 − q
    return L

function SamplePathAdjoint(ray, L, δL)
    β = 1, w_1 = 0, w_2 = 1, sampled_∂x = FALSE
    for i = 0 to N − 1 do
        L -= β · L_e(...) / (w_1 + w_2)
        if !sampled_∂x and Rand() < q then
            ω′, f = Sample_∂BRDF(...)
            sampled_∂x = TRUE
        else
            ω′, f = Sample_BRDF(...)
        δ_θ += BackwardGrad(f, δL · L/f)
        β *= f / p_f(ω′, ...)
        w_1 += w_2 · q · p_∂f(ω′, ...) / p_f(ω′, ...)
        w_2 *= 1 − q
    return δ_θ
```

★ **Implement with simple modifications to Path Replay Backpropagation**

Adaptive pixel sampling

Inverse rendering optimization

$$\min_{\theta} \mathcal{L}\left(\mathbf{I}(\theta), \tilde{\mathbf{I}}\right)$$

scene parameter → θ

loss function → $\mathcal{L}$

rendered image → $\mathbf{I}(\theta)$

reference image → $\tilde{\mathbf{I}}$

★ **Need to estimate** $\partial_\theta \mathcal{L}$ **to do gradient descent**

Loss gradient integral

$$\partial_\theta \mathcal{L} = \partial_{\mathbf{I}} \mathcal{L} \cdot \partial_\theta \mathbf{I}$$

Loss gradient integral

$$\partial_\theta \mathcal{L} = \partial_{\mathbf{I}} \mathcal{L} \cdot \partial_\theta \mathbf{I}$$
$$= \partial_{\mathbf{I}} \mathcal{L} \cdot \int_{\partial \mathcal{P}} g_n(\overline{\mathbf{x}}) \mathrm{d}\overline{\mathbf{x}}$$

Loss gradient integral

$$\begin{aligned}\partial_\theta \mathcal{L} &= \partial_{\mathbf{I}} \mathcal{L} \cdot \partial_\theta \mathbf{I} \\ &= \partial_{\mathbf{I}} \mathcal{L} \cdot \int_{\partial \mathcal{P}} g_n(\overline{\mathbf{x}}) \mathrm{d}\overline{\mathbf{x}} \\ &= \int_{\partial \mathcal{P}} \partial_{\mathbf{I}} \mathcal{L} \cdot g_n(\overline{\mathbf{x}}) \mathrm{d}\overline{\mathbf{x}}\end{aligned}$$

adjoint radiance

Loss gradient integral

$$
\begin{aligned}
\partial_\theta \mathcal{L} &= \partial_{\mathbf{I}} \mathcal{L} \cdot \partial_\theta \mathbf{I} \\
&= \partial_{\mathbf{I}} \mathcal{L} \cdot \int_{\partial \mathcal{P}} g_n(\overline{\mathbf{x}}) \mathrm{d}\overline{\mathbf{x}} \\
&= \int_{\partial \mathcal{P}} \partial_{\mathbf{I}} \mathcal{L} \cdot g_n(\overline{\mathbf{x}}) \mathrm{d}\overline{\mathbf{x}}
\end{aligned}
$$

Separate rendering passes

Adaptive pixel sampling

$$\partial_\theta \mathcal{L} = \int_{\partial \mathcal{P}} \partial_{\mathbf{I}} \mathcal{L} \cdot g_n(\overline{\mathbf{x}}) \, \mathrm{d}\overline{\mathbf{x}}$$

adjoint radiance

Forward rendering

$$I = \int_{\mathcal{P}} W_\mathrm{e} \, f(\overline{\mathbf{x}}) \, \mathrm{d}\overline{\mathbf{x}}$$

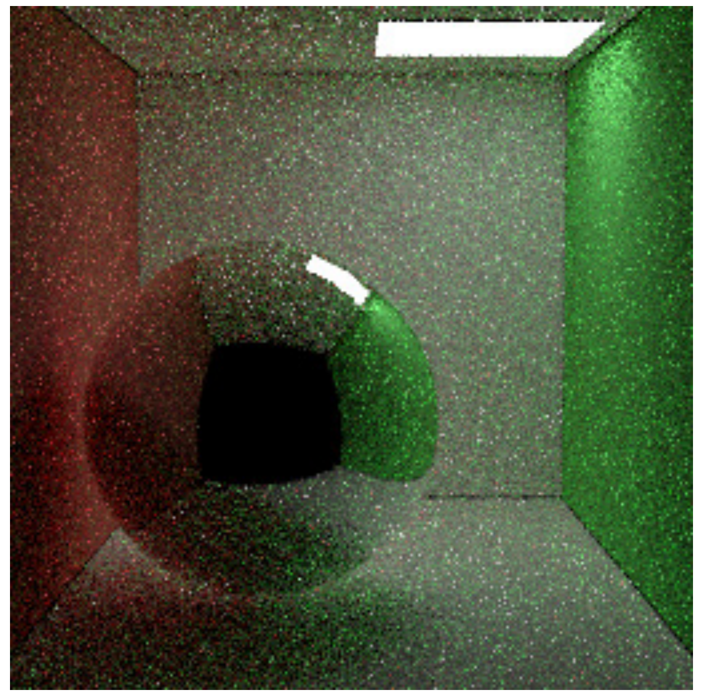
sensor importance

Adaptive pixel sampling

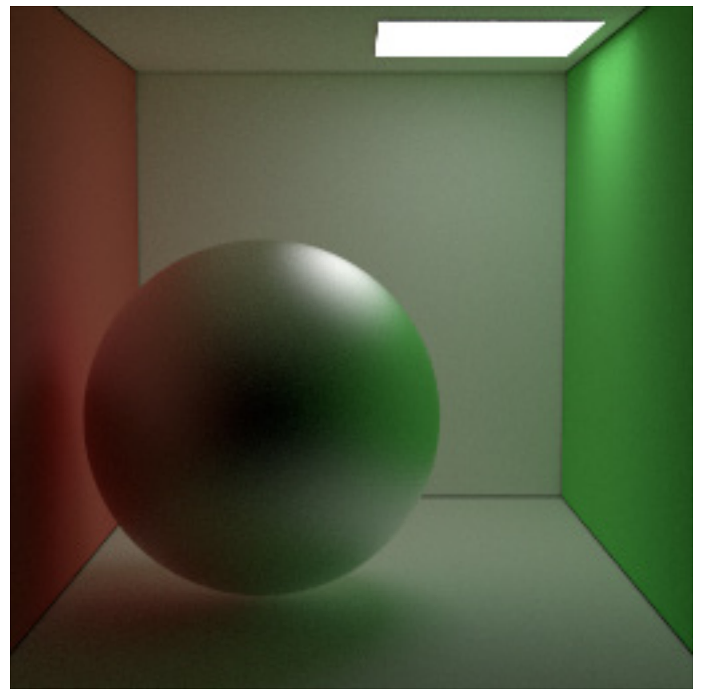
$$\partial_\theta \mathcal{L} = \int_{\partial \mathcal{P}} \partial_{\mathbf{I}} \mathcal{L} \cdot g_n(\overline{\mathbf{x}}) \, \mathrm{d}\overline{\mathbf{x}}$$

Our method: Use to importance sample pixels to start paths from

Adaptive pixel sampling



Rendered $\mathbf{I}(\theta)$



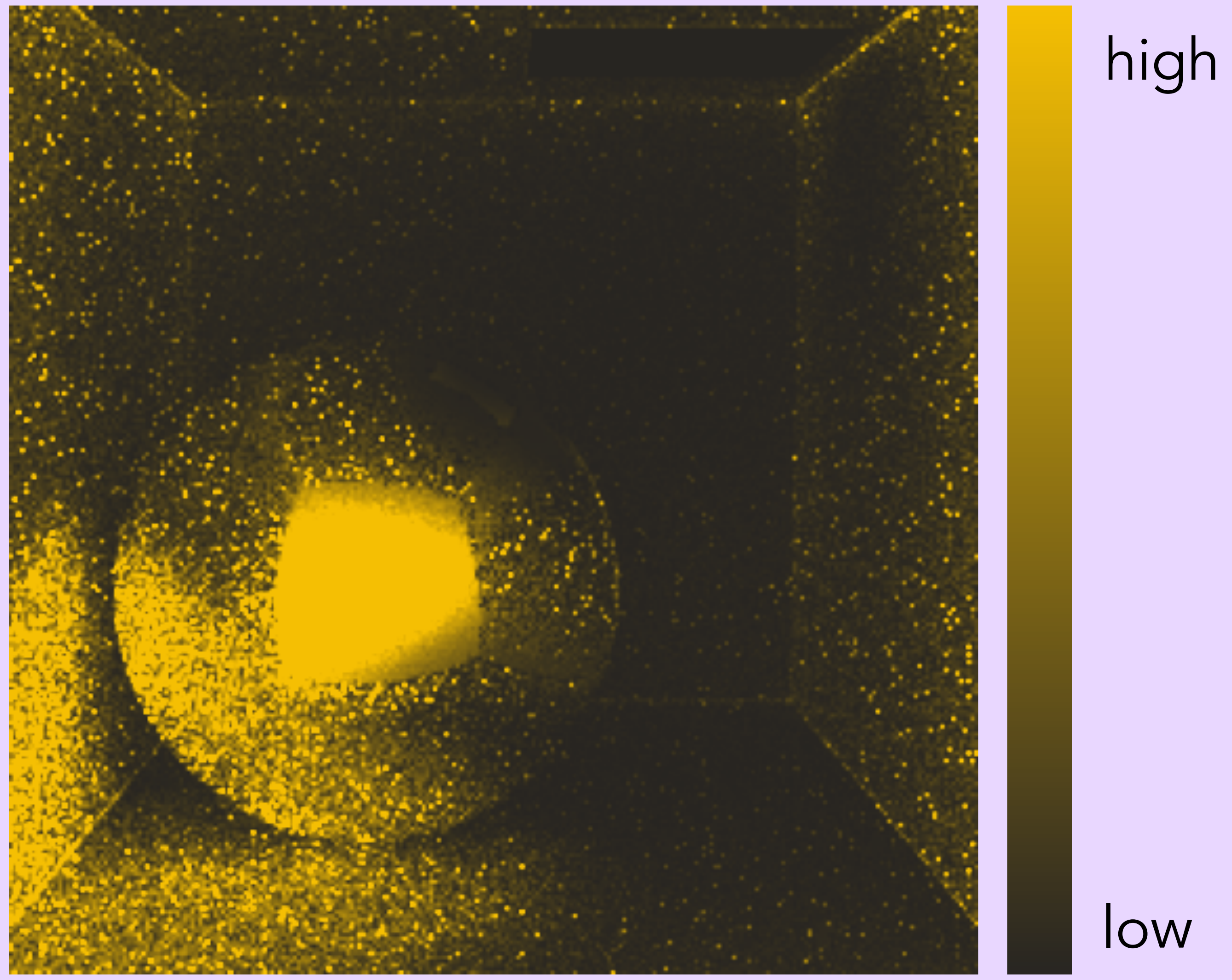
Reference $\tilde{\mathbf{I}}$

$\mathcal{L}$

Loss

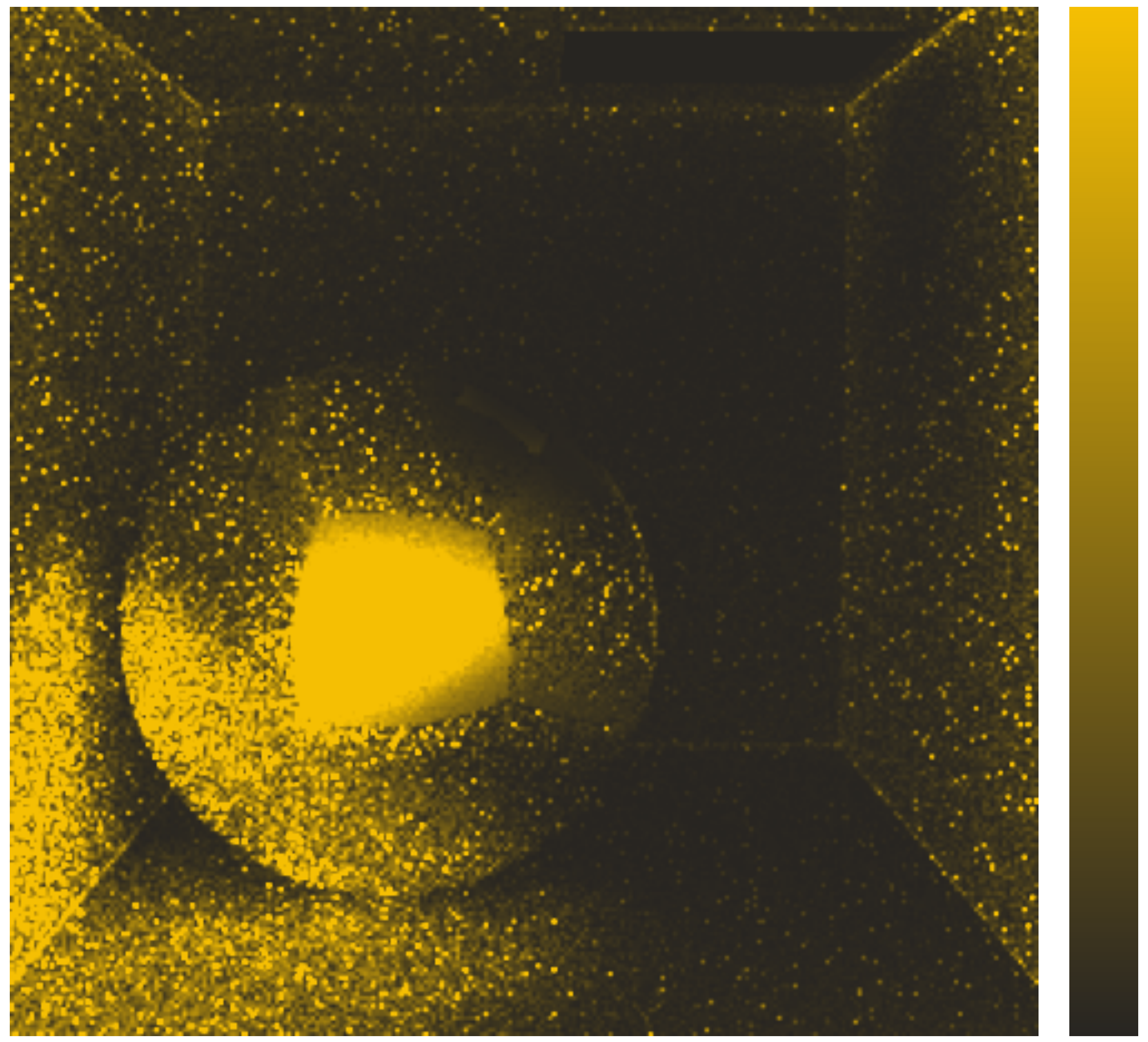
$\partial_{\mathbf{I}}\mathcal{L}$

Adjoint radiance



Sampling weights proportional to $|\partial_{\mathbf{I}}\mathcal{L}|$

Adaptive pixel sampling



Sampling weights proportional to $|\partial_{\mathbf{I}} \mathcal{L}|$

Sample pixels to start paths from → $\partial_\theta \mathcal{L}$ → Gradient step

↓

Next iteration of optimization…

Experiments

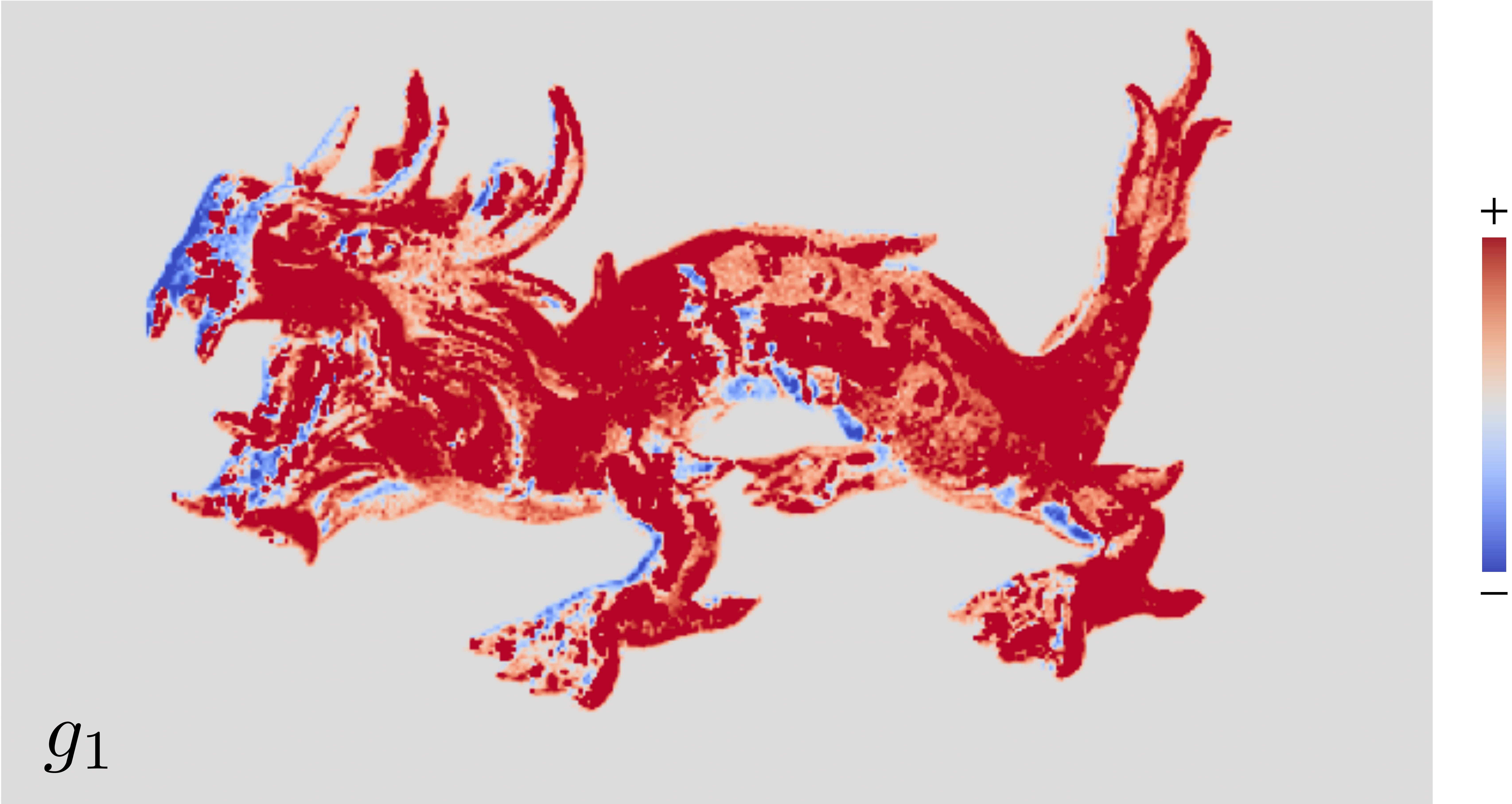
Visualizing gradients

Forward render

Visualizing gradients

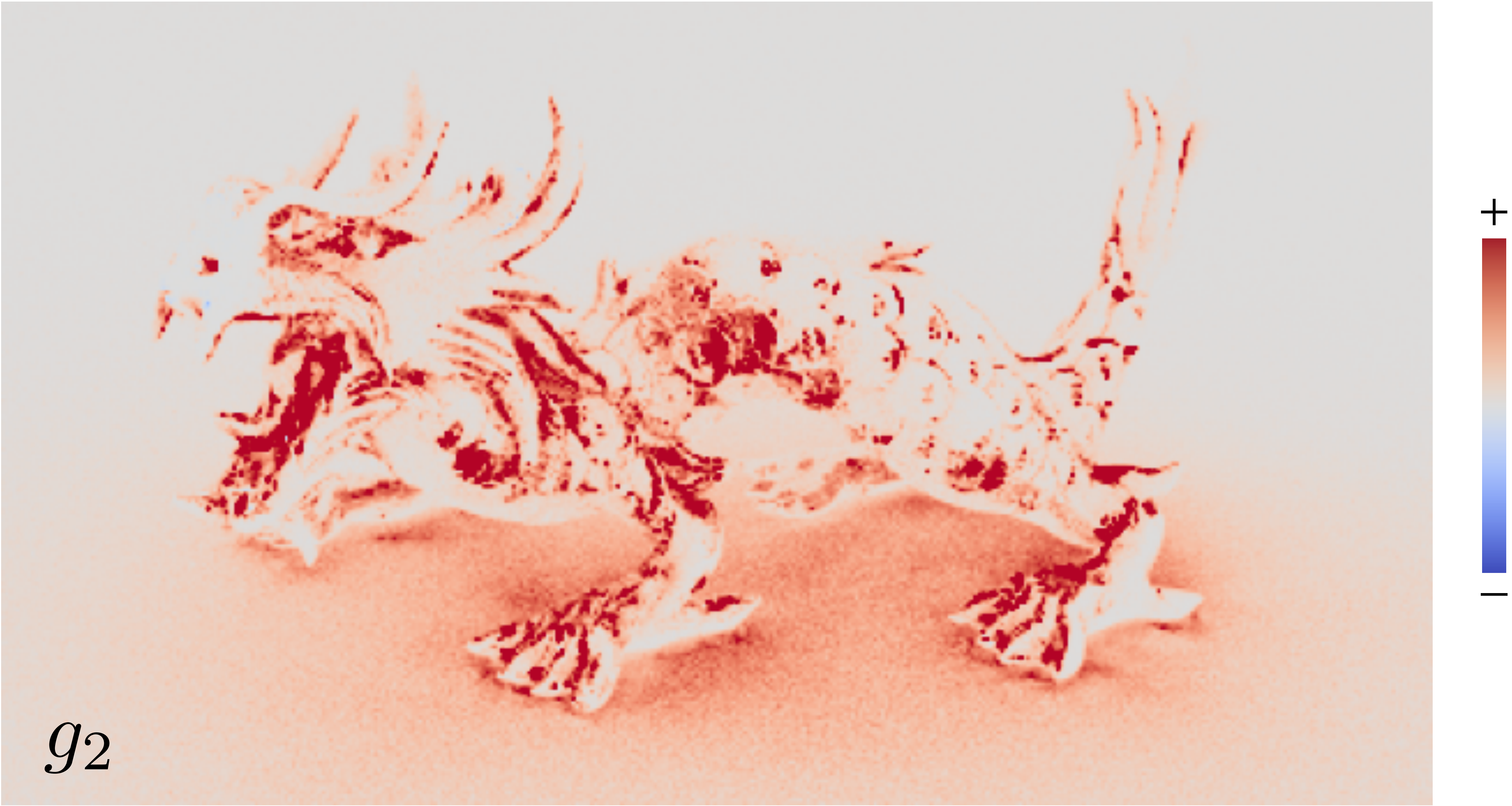
Full gradient

Visualizing gradients



Fixed differential vertex at $\mathbf{x}_1$

Visualizing gradients

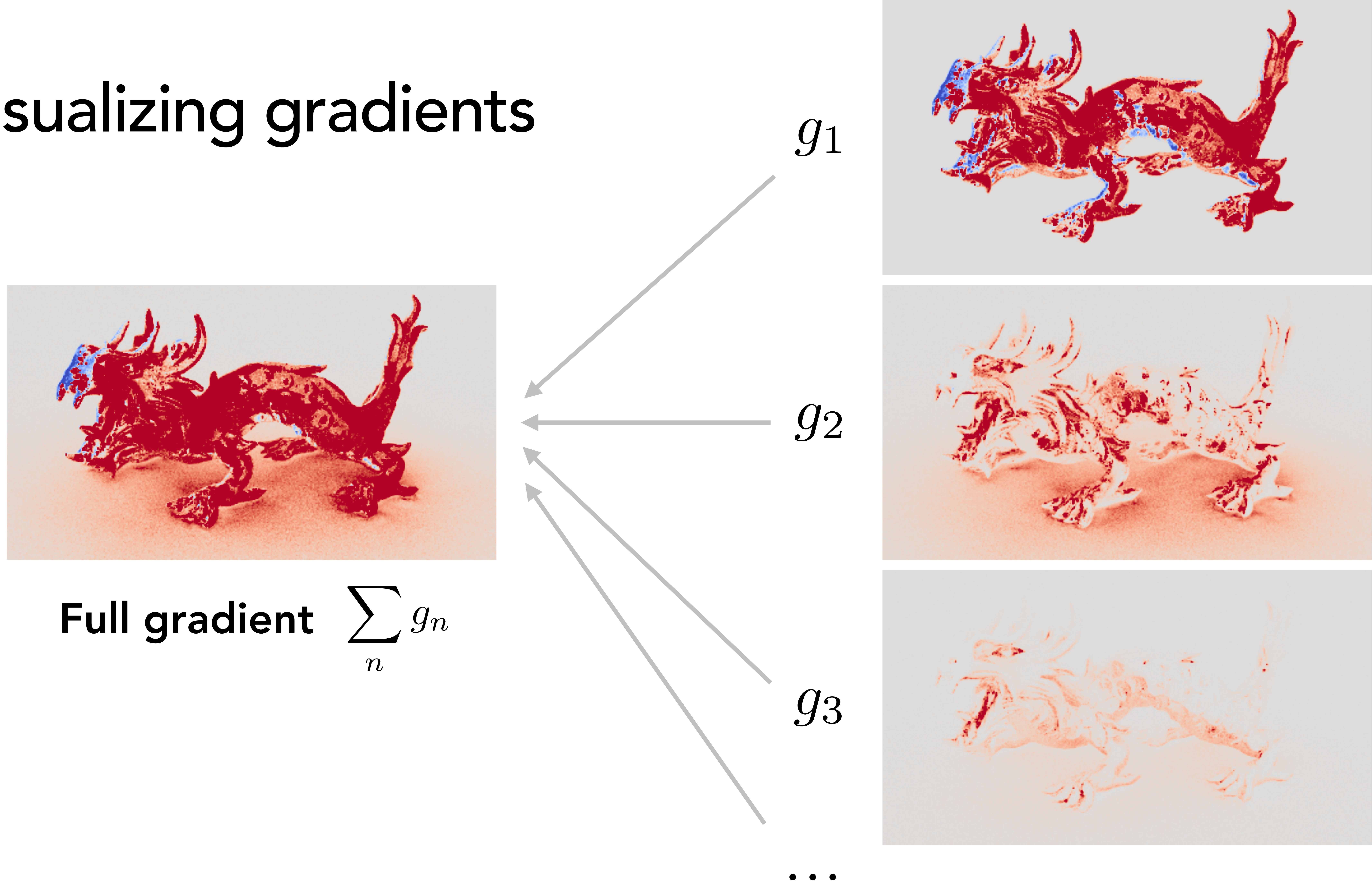


Fixed differential vertex at $\mathbf{x}_2$

Visualizing gradients

Fixed differential vertex at $\mathbf{x}_3$

Visualizing gradients



Visualizing gradients

Forward render

Full gradient $\sum_n g_n$

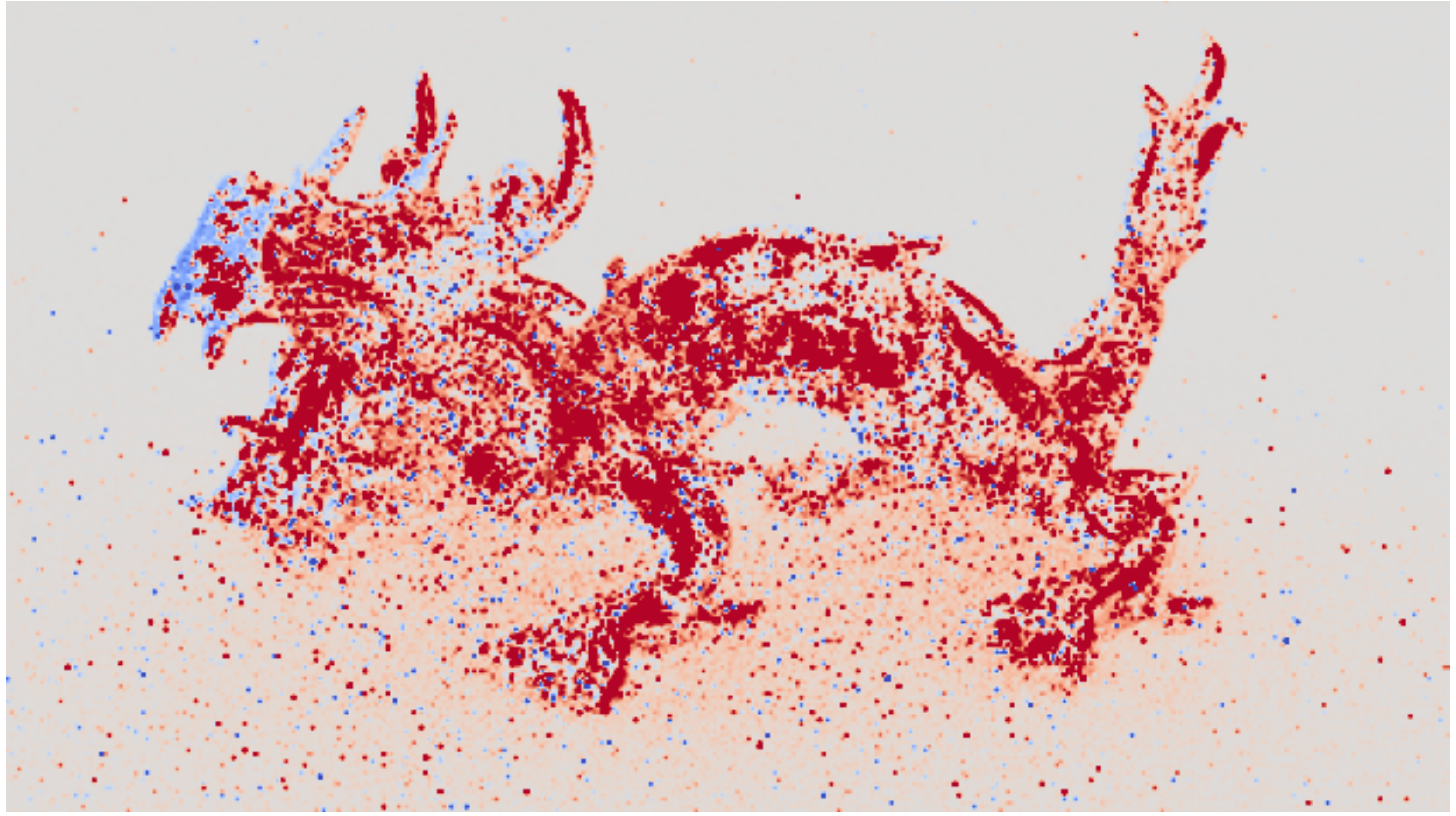
g_1

g_2

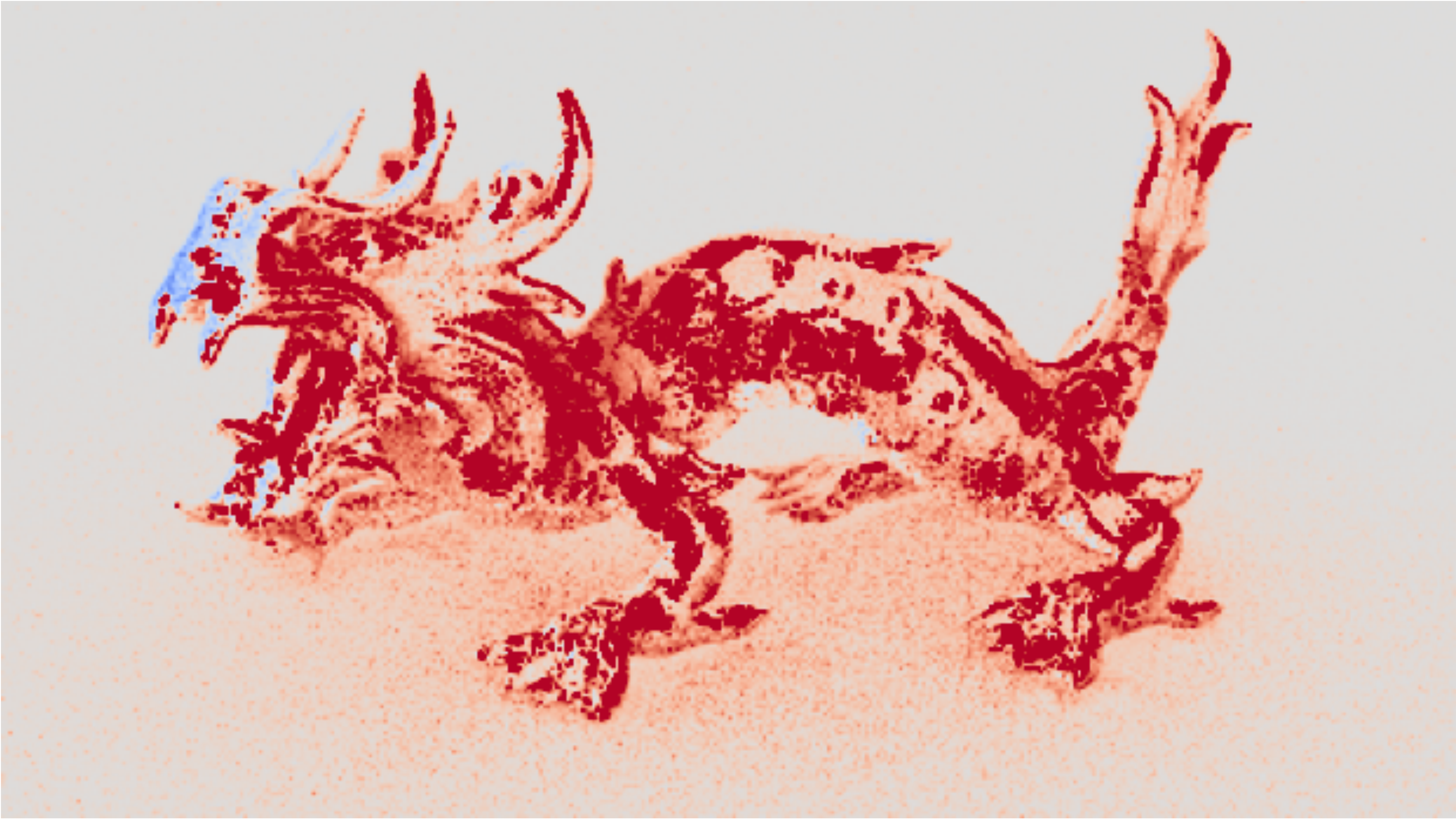
g_3

Image gradients (equal-time comparison)

BRDF sampling



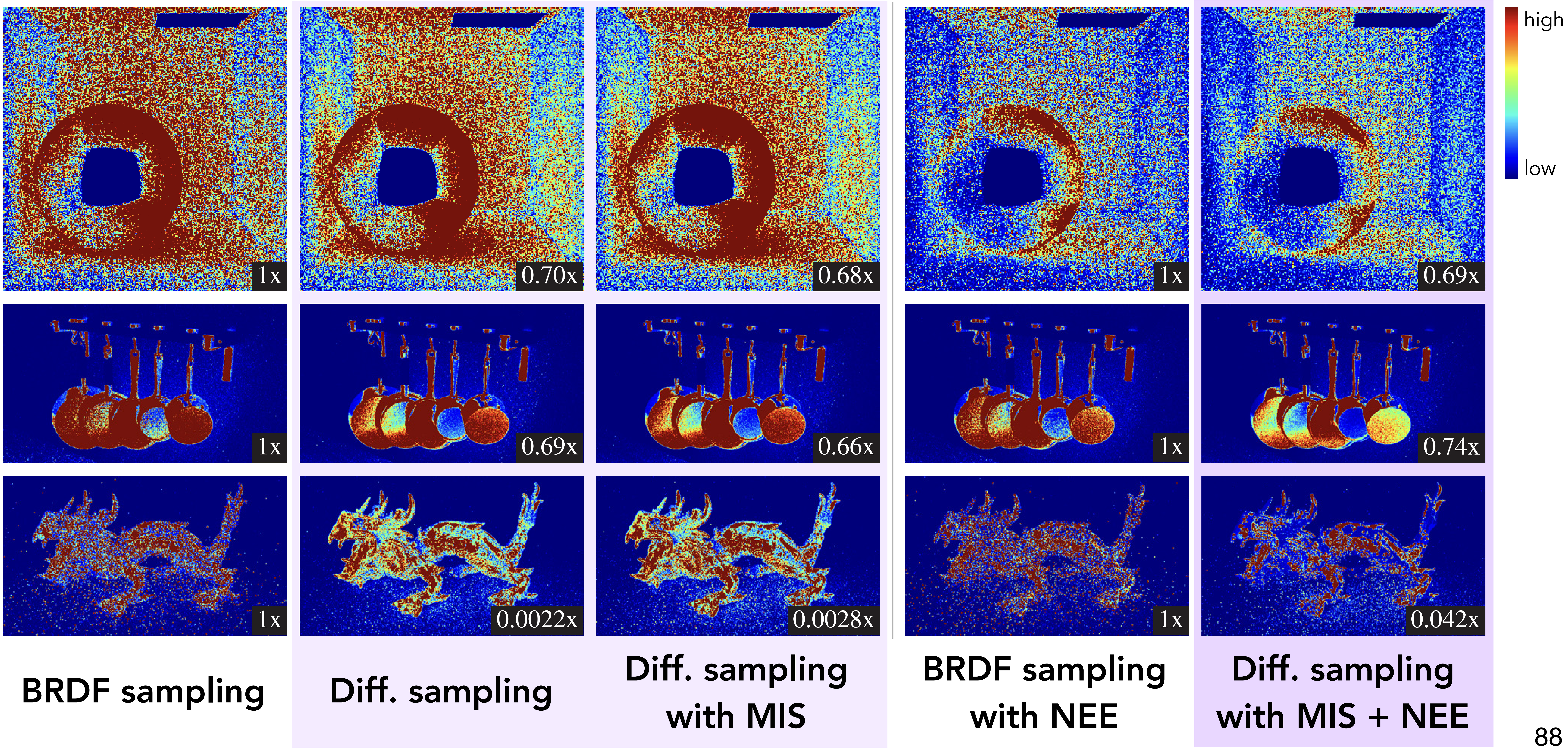
Differential sampling (ours)



2 orders of magnitude less noisy

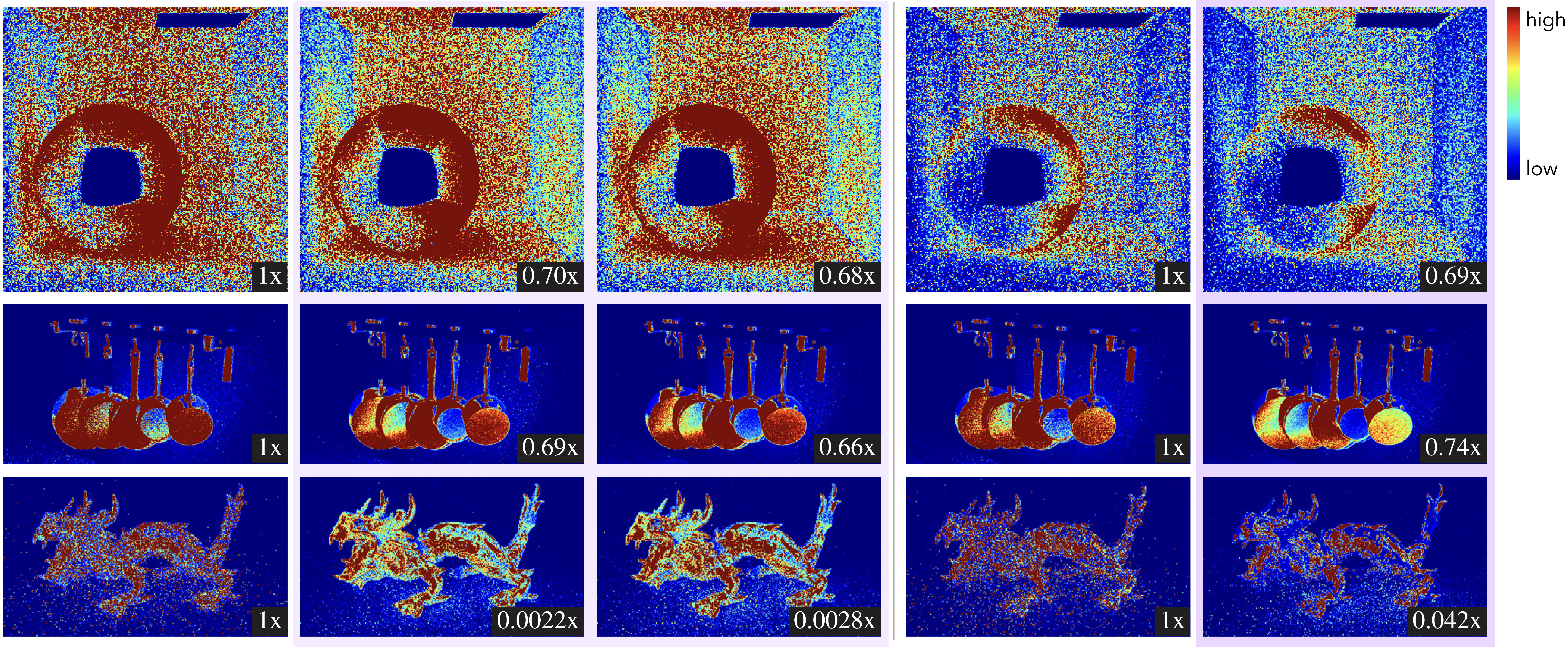
Numbers = ratios of mean variance with the baseline (lower is better)

Variance of image gradients $\partial_\theta \mathbf{I}$



Variance of image gradients $\partial_\theta \mathbf{I}$

Numbers = ratios of mean variance with the baseline (lower is better)



Lowest variance: our differential sampling method with MIS + NEE

Variance of loss gradients $\partial_\theta \mathcal{L}$

Variance of loss gradients $\partial_\theta \mathcal{L}$

Scene	**BRDF**	**BRDF + adaptive**	**Differential + adaptive**
BOWL	13.6	4.34	3.49
SPHERE	172	51.7	8.62
PANS	4.32	0.487	0.412
DRAGON	4.76	0.633	0.0109
VASES	0.00164	0.000193	1.31×10^{-5}

Variance of loss gradients $\partial_\theta \mathcal{L}$

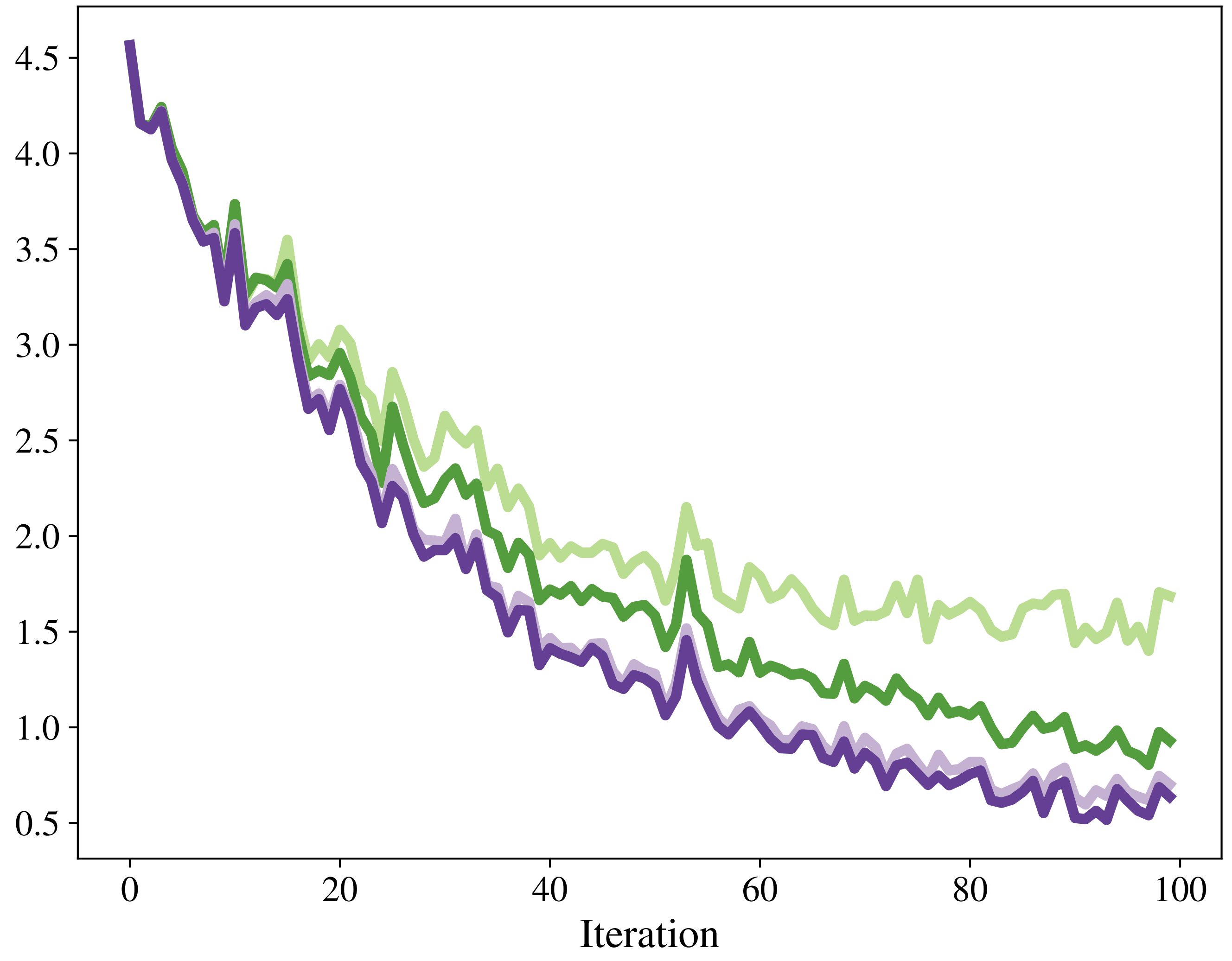
Scene	**BRDF**	**BRDF + adaptive**	**Differential + adaptive**
Bowl	13.6	4.34	3.49
Sphere	172	51.7	8.62
Pans	4.32	0.487	0.412
Dragon	4.76	0.633	0.0109
Vases	0.00164	0.000193	1.31×10^{-5}

Lowest variance: our combined method

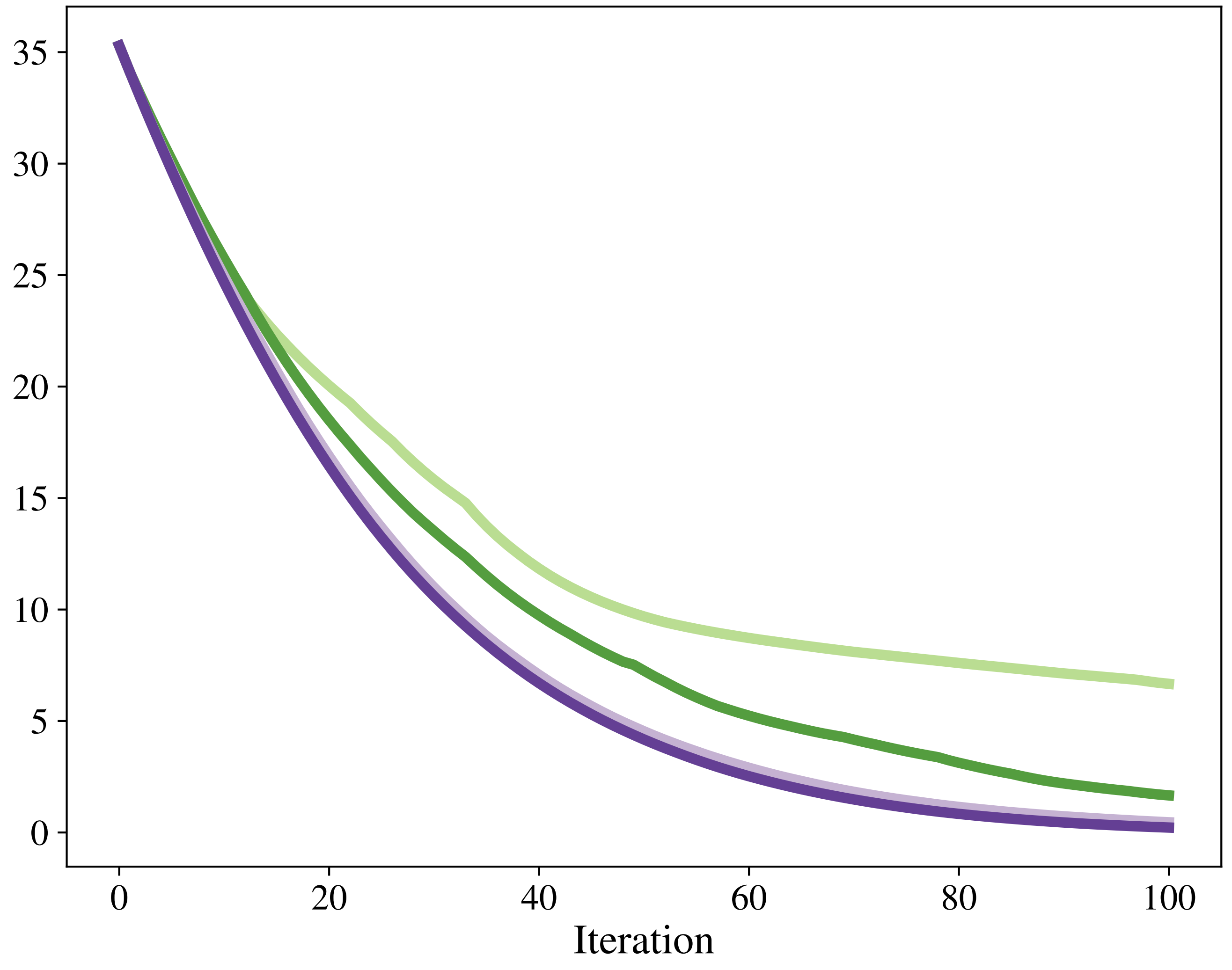
(1-2 orders of magnitude better)

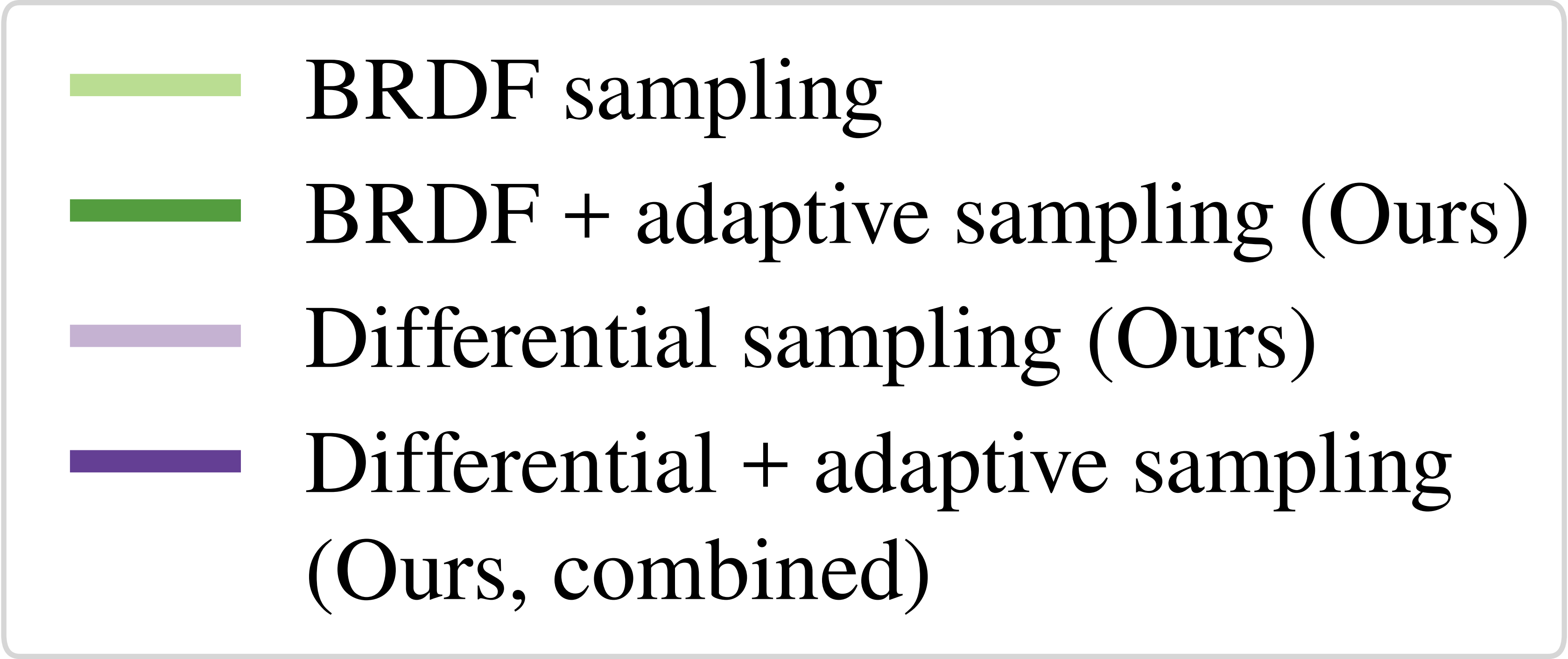
Inverse rendering performance

Relative MSE of image



Relative MSE of parameter





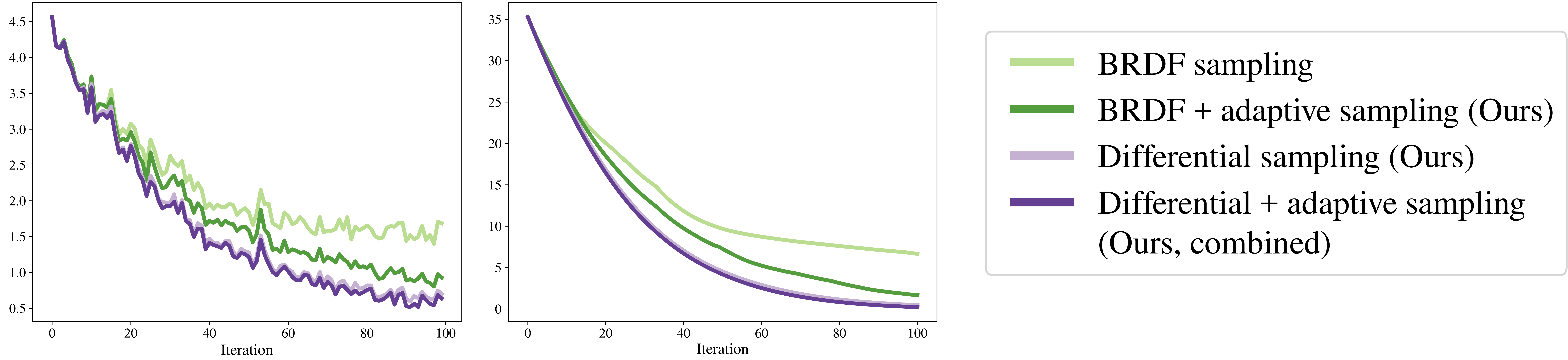
Initialization

Target

Inverse rendering performance

Relative MSE of image

Relative MSE of parameter



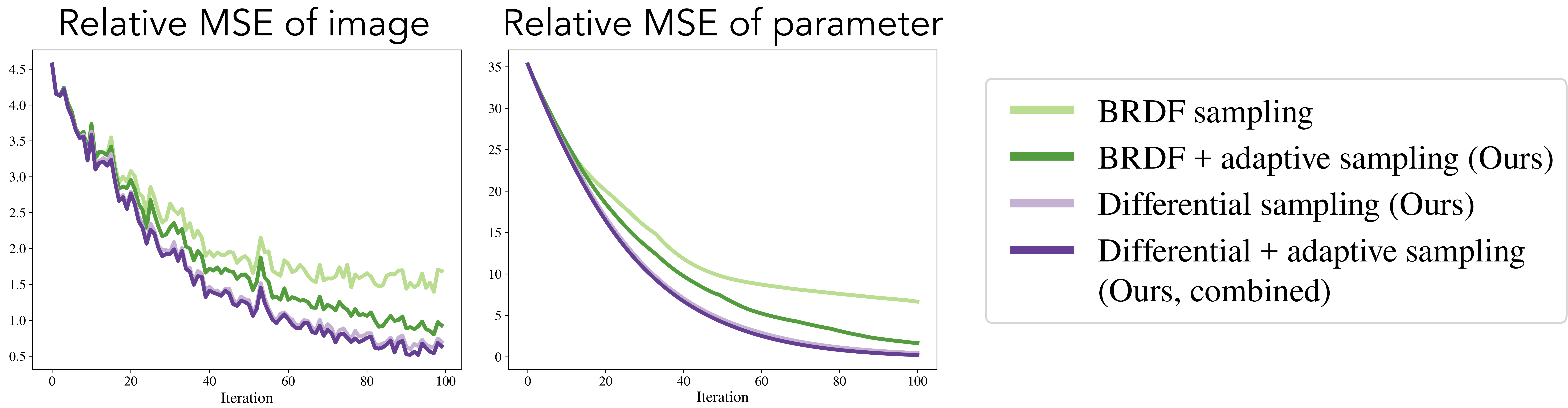
Initialization

BRDF sampling

Ours, combined

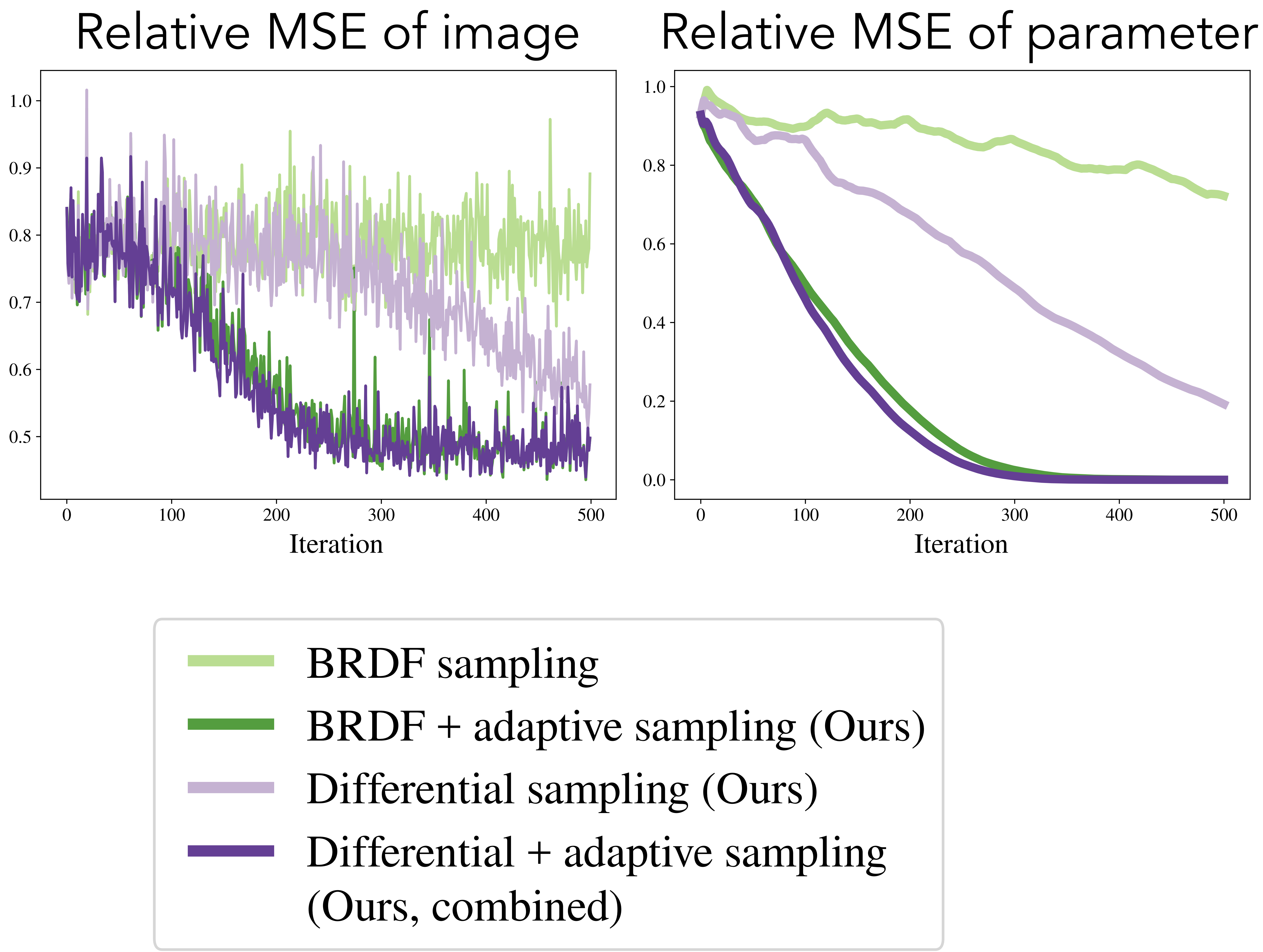
Target

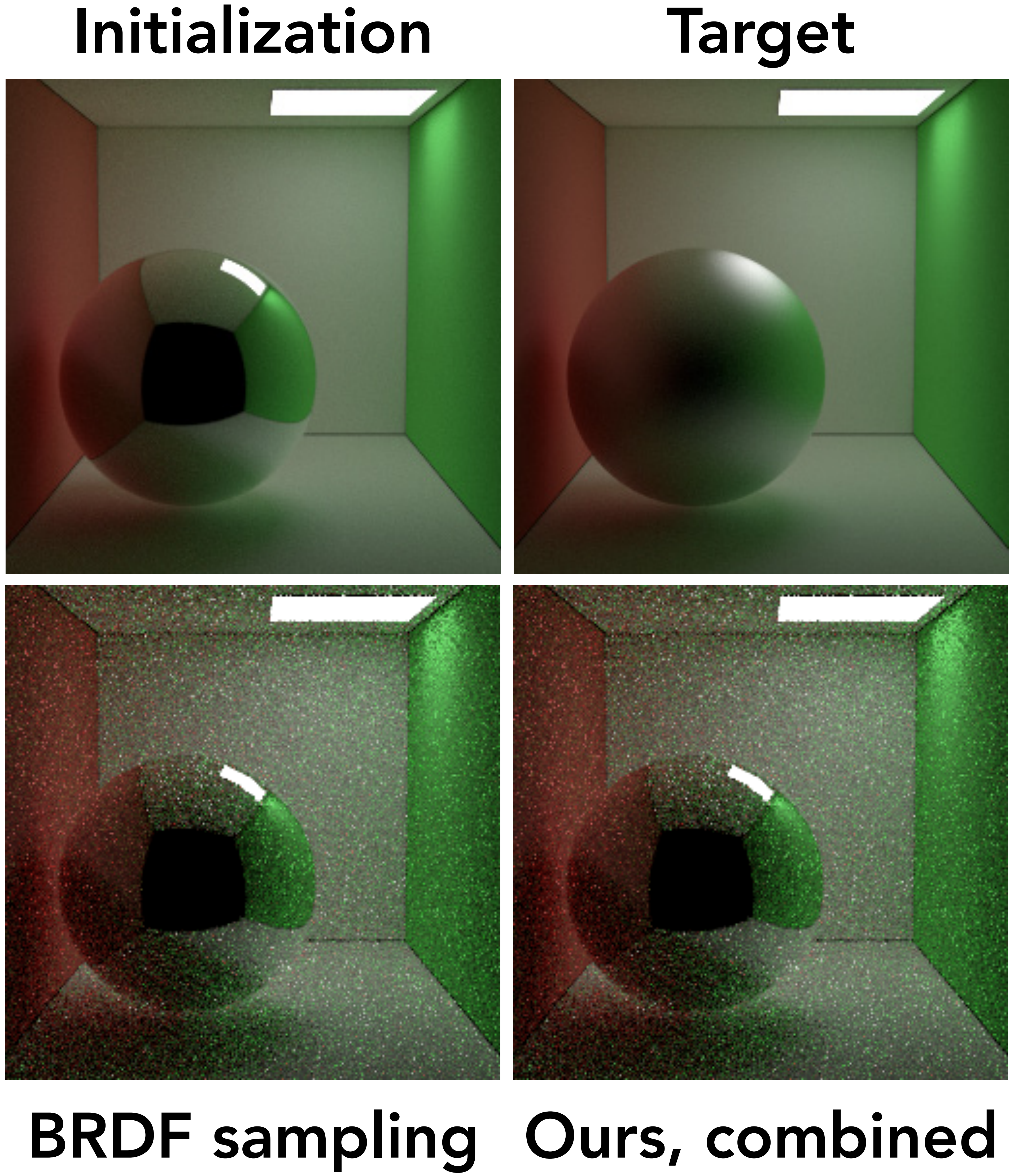
Inverse rendering performance



Initialization | **BRDF sampling** | **Ours, combined** | **Target**

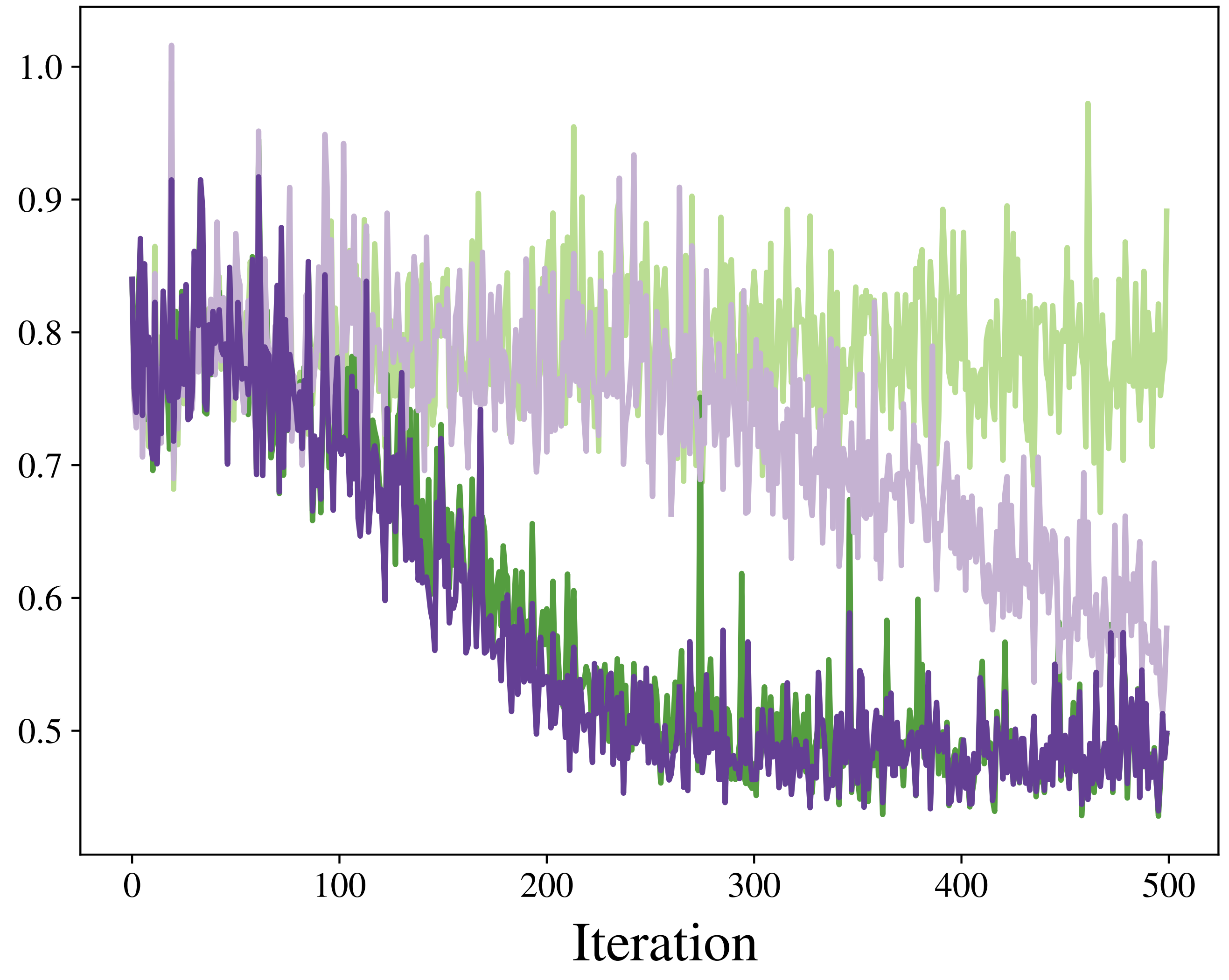
Inverse rendering performance



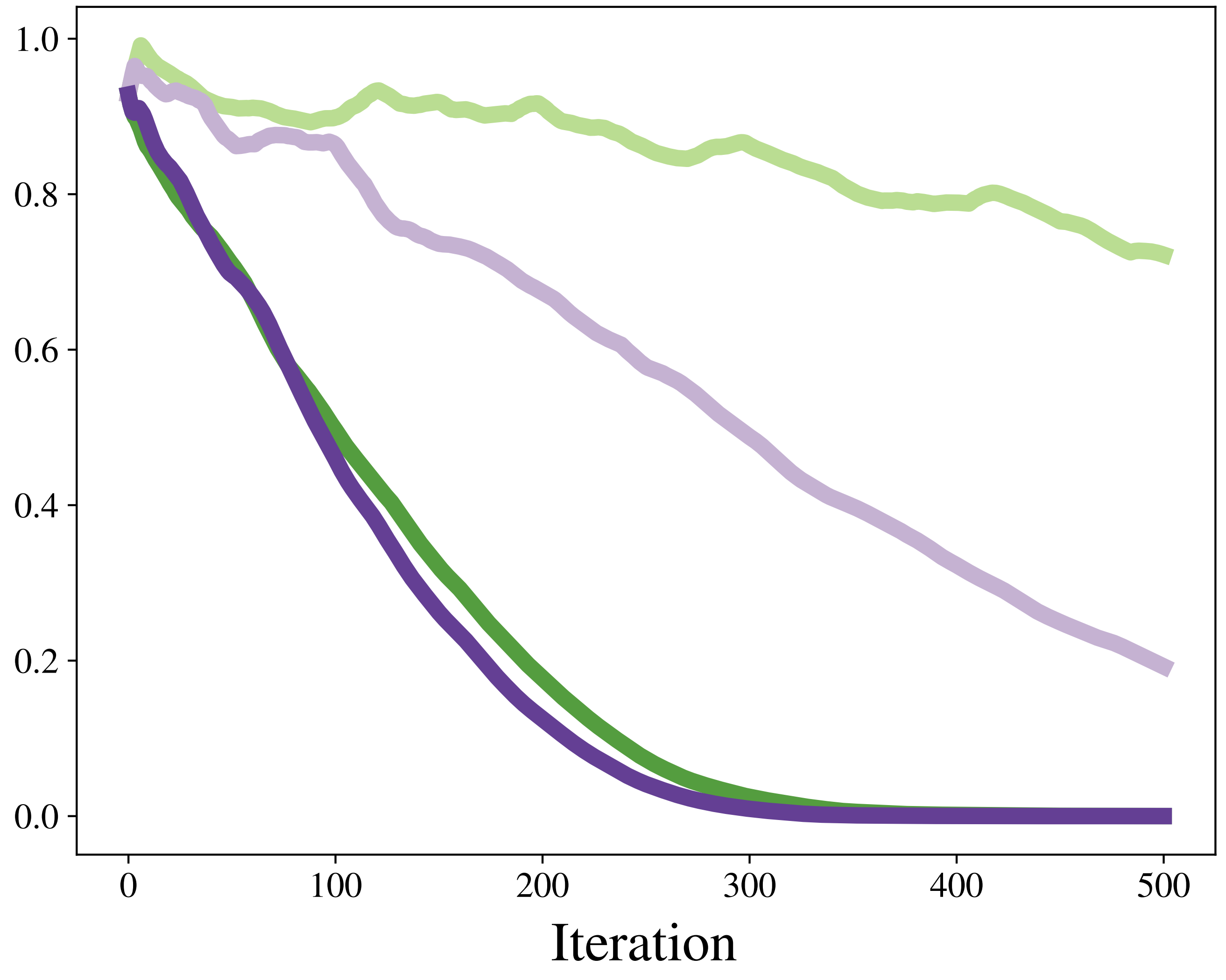


Inverse rendering performance

Relative MSE of image

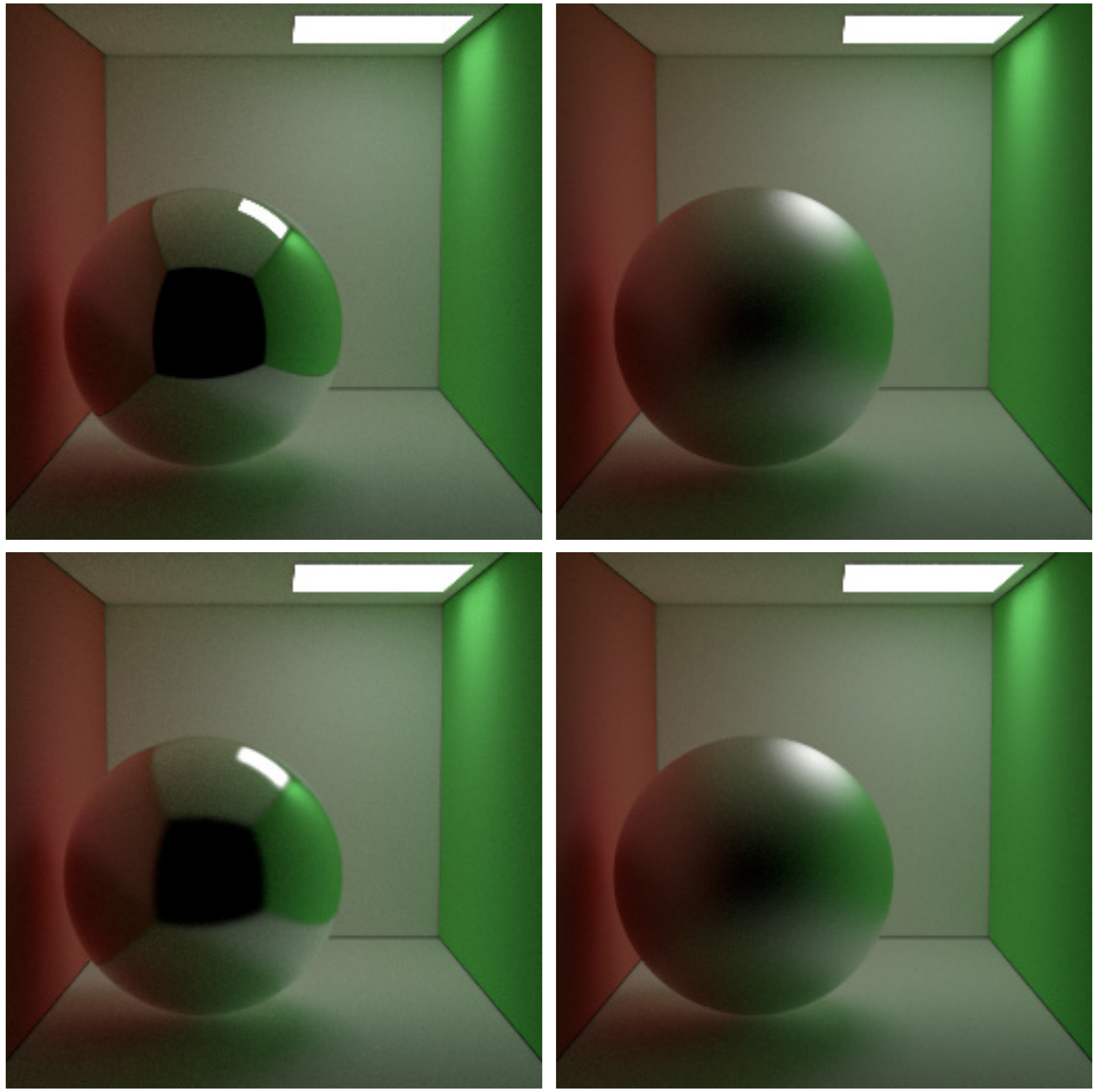


Relative MSE of parameter



- BRDF sampling
- BRDF + adaptive sampling (Ours)
- Differential sampling (Ours)
- Differential + adaptive sampling (Ours, combined)

Initialization | Target



BRDF sampling | Ours, combined

Conclusion

Conclusion

- **Differential path sampling method** tailored to the integral computed in differentiable rendering

Conclusion

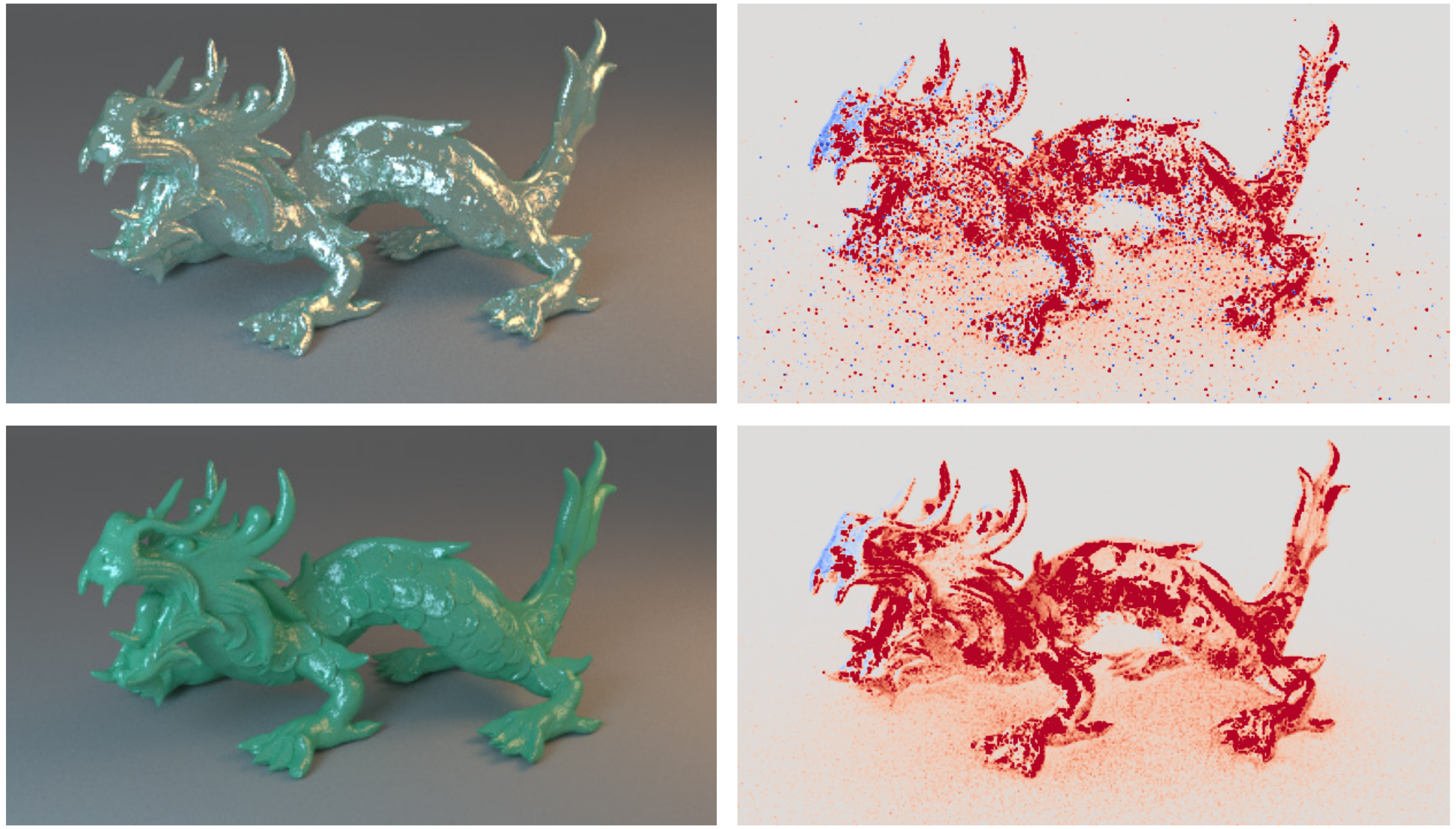
- **Differential path sampling method** tailored to the integral computed in differentiable rendering
- **Adaptive pixel sampling method** for path launching in each inverse rendering optimization step

Conclusion

- **Differential path sampling method** tailored to the integral computed in differentiable rendering
- **Adaptive pixel sampling method** for path launching in each inverse rendering optimization step

Future work: extending differential sampling to scenes that optimize more than one scene parameter

Thanks for listening :-)



Project website

NSF

ALFRED P. SLOAN
FOUNDATION

Project website: https://imaging.cs.cmu.edu/path_sampling_differentiable_rendering/

Code: https://github.com/cmu-ci-lab/path_sampling_differentiable_rendering/