FROM PARTICLES TO FIELDS: REFRAMING PHOTON MAPPING WITH CONTINUOUS GAUSSIAN PHOTON FIELDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Accurately modeling light transport is essential for realistic image synthesis. Photon mapping captures complex global illumination effects such as caustics and specular-diffuse interactions, but dense photon storage and repeated surface gathering remain expensive when rendering multiple views, even when the photon map is reused. To accelerate multi-view rendering, we reformulate surface-lighting estimation as a continuous and reusable field. We introduce the **Gaussian Photon Field (GPF)**, a learnable representation of surface irradiance using anisotropic 3D Gaussian primitives, enabling efficient radiance evaluation across viewpoints. GPF is initialized with photon-guided surface coverage and optimized against sparse multi-view stochastic progressive photon mapping (SPPM) radiance references. Differentiable Gaussian interpolation allows the irradiance coefficients and their spatial support to be fitted jointly through the rendered image predictions. At rendering time, the field is combined with the terminal Lambertian response and explicitly traced camera-side transport, preserving view-dependent reflection and refraction without further photon tracing or progressive refinement. Experiments on scenes with caustics and specular-diffuse interactions demonstrate high-fidelity multi-view rendering with reduced per-view computation, connecting physically-based light transport with the efficiency of reusable learned representations.

1 INTRODUCTION

Accurately modeling light transport (Kajiya, 1986) is fundamental to realistic image synthesis, with applications in interactive graphics (Keller, 1997; Zhang et al., 2019; Bitterli et al., 2020), medical imaging simulation (Wang et al., 1995; Fang & Boas, 2009; Sarrut et al., 2022), scene understanding via inverse graphics (Barron & Malik, 2014; Azinovic et al., 2019; Zhang et al., 2022), *etc*. Physically-based rendering techniques (Goral et al., 1984; Veach, 1998), such as photon mapping (Jensen, 1996) and its variants (Hachisuka et al., 2008; Hachisuka & Jensen, 2009; Knaus & Zwicker, 2011), have long served as reliable estimators for complex global illumination, effectively capturing challenging phenomena such as caustics, indirect reflections, and refractions. Despite their accuracy, these methods remain computationally expensive when rendering multiple views of the same scene.

Prior work on photon mapping (Jensen, 1996) addresses complex global illumination by decoupling light transport into two stochastic processes: photon tracing and density estimation. During photon tracing, photons emitted from light sources are stored upon surface and volumetric interactions, forming a global photon map. In the subsequent density estimation stage, the radiance at visible points is reconstructed by gathering nearby photons using spatial averaging based on stochastic kernel density estimation (KDE) (Silverman, 2018; Jensen, 2001). This formulation enables accurate estimation of caustics and diffuse inter-reflections, making photon mapping a robust estimator for a wide range of lighting phenomena.

Photon tracing is view-independent, and a photon map can be reused across cameras (Jensen, 1996). Reuse alone, however, does not eliminate the cost of storing dense photon deposits and gathering them at newly visible surface points. Progressive methods additionally maintain camera-dependent gathering statistics (Hachisuka et al., 2008; Hachisuka & Jensen, 2009). This motivates a compact, optimized field that amortizes surface-lighting estimation across views.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

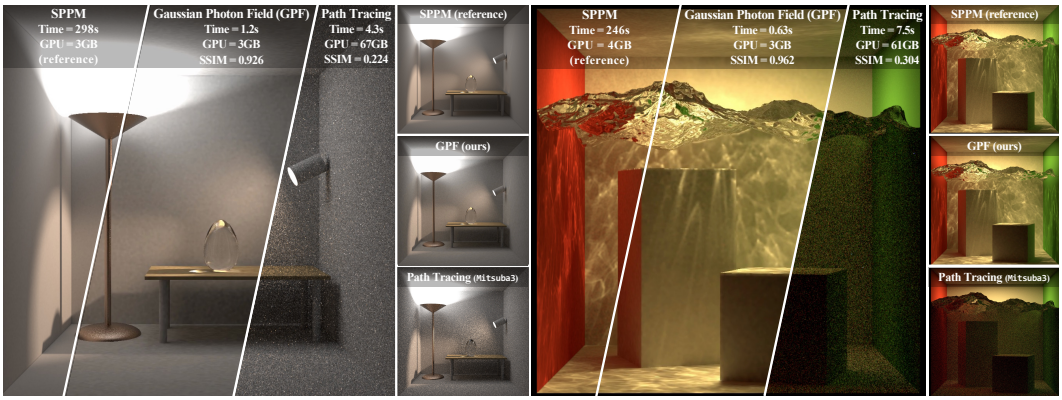


Figure 1: **Gaussian Photon Field (GPF)** provides view-reusable global illumination for indirect lighting (*left*) and water caustics (*right*). Compared with SPPM and path tracing, GPF preserves complex lighting at a lower per-view rendering cost.

In contrast, modern neural scene representations, *e.g.*, neural radiance fields (NeRF) (Mildenhall et al., 2021) and 3D Gaussian splatting (3DGS) (Kerbl et al., 2023), represent scene appearance and geometry as continuous functions that can be queried from arbitrary viewpoints. By learning such a shared representation, these methods amortize rendering costs across views. This motivates a fundamental question: **can photon-based light transport be organized into a compact, learnable field that supports efficient evaluation across viewpoints?** Such a formulation would combine light-transport simulation with the efficiency of a shared scene representation.

Toward this goal, we introduce the **Gaussian Photon Field (GPF)**, a learnable surface-illumination representation that unifies photon mapping and neural field paradigms. GPF represents surface illumination with nonnegative RGB irradiance coefficients and learnable anisotropic spatial support, parameterized by Gaussian positions, rotations, and scales. At rendering time, the field is combined with the terminal Lambertian response and explicitly traced camera-side transport to produce view-dependent images. Unlike traditional 3D Gaussian splatting, which represents view-dependent surface appearance, GPF stores illumination arriving at diffuse surfaces, including caustics and indirect lighting. Once optimized, GPF renders novel views without further photon tracing or progressive refinement, replacing repeated surface KDE with queries to a persistent Gaussian illumination field.

We construct and optimize our Gaussian Photon Field in three stages: (1) **Initialization:** photon deposits seed the Gaussian centers, supplemented by surface samples from the training views. We initialize surface-aligned spatial support and infer initial irradiance coefficients from stochastic progressive photon mapping (SPPM) (Hachisuka & Jensen, 2009) reference images using the traced camera-path factors; (2) **Radiance Query:** camera rays trace specular, dielectric, and glossy interactions until reaching a Lambertian terminal surface. Gaussian interpolation supplies irradiance, which is converted to outgoing radiance by the terminal material response; (3) **Supervision:** we jointly optimize the irradiance coefficients and Gaussian spatial parameters through the recomposed image predictions, using sparse multi-view SPPM references. This process encodes light-side transport in a reusable irradiance field while retaining explicitly traced, view-dependent camera transport.

To validate the effectiveness of our approach, we conduct extensive experiments across multiple physically-challenging scenes, including complex caustic and specular-diffuse interactions. We compare GPF against classical physically-based integrators (*e.g.*, path tracing (Jakob et al., 2022b) and SPPM (Hachisuka & Jensen, 2009)), Gaussian mixture fitting (EM-GMM (Jakob et al., 2011)), neural scene representations (*e.g.*, NeRF (Mildenhall et al., 2021), 3DGS (Kerbl et al., 2023), and AnySplat (Jiang et al., 2025)), and feed-forward rendering (RenderFormer (Zeng et al., 2025)). Measuring visual quality, render time, and storage cost, these experiments evaluate how a reusable Gaussian illumination field amortizes photon-based light transport across views.

2 Related Work

Physically-Based Global Illumination. Simulating global illumination (GI) (Kajiya, 1986) is a central problem in computer graphics. Unbiased Monte Carlo methods, *e.g.*, path tracing (PT) (Kajiya,

1986) and its bidirectional variants (BDPT) (Veach, 1998), are considered the gold standard for accuracy. However, they struggle to efficiently sample complex, low-probability light paths, such as those that form caustics, often resulting in high variance and noise.

To address this, biased techniques like photon mapping (Jensen, 1996) and its stochastic progressive variant SPPM (Hachisuka & Jensen, 2009) were developed. As detailed in Sec. 3, these methods excel at capturing complex GI phenomena by decoupling light transport into a two-pass process. Photon maps can be reused across views, but their storage and the gathering of photons at newly visible surface points remain substantial costs; SPPM also maintains camera-dependent progressive statistics. Our work targets this repeated surface-lighting estimation. As illustrated in Figure 2, rather than proposing a new sampling strategy, GPF organizes photon-guided surface illumination into a persistent irradiance field, queried during camera ray tracing and combined with the terminal material response to produce radiance.

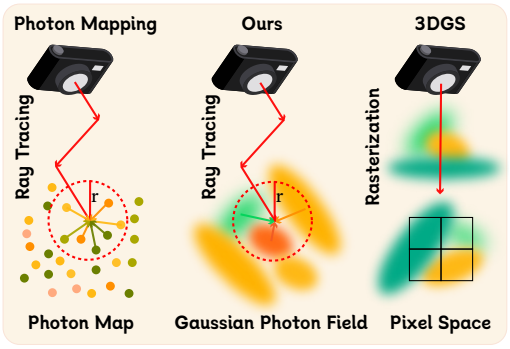


Figure 2: **Rendering paradigm comparison.** Classical photon mapping gathers discrete photons to estimate surface illumination. GPF queries a reusable Gaussian irradiance field during camera ray tracing and applies the terminal material response to obtain radiance. In contrast, 3DGS rasterizes primitives to reproduce scene appearance.

Neural and Learned Scene Representations. The concept of a continuous, learnable scene representation was popularized by neural radiance fields (NeRF) (Mildenhall et al., 2021). NeRF and its many successors (Chen et al., 2024; Guo et al., 2023; Li et al., 2021; Liu et al., 2023; Ma et al., 2024; Park et al., 2021a;b; Pumarola et al., 2021; Tretschk et al., 2021; Wu et al., 2025; Xian et al., 2021) achieve state-of-the-art novel view synthesis. However, these methods are essentially "black-box" appearance models: optimized for view interpolation, they lack physical grounding and cannot guarantee accurate light transport for complex effects such as caustics or relighting.

More recently, 3D Gaussian splatting (3DGS) (Kerbl et al., 2023; Huang et al., 2024; Yang et al., 2024) has achieved remarkable success, combining the efficiency of explicit primitives with the representational quality of neural scene models for real-time reconstruction. We draw inspiration from 3DGS in its use of Gaussians as compact, expressive primitives. However, our approach differs fundamentally in both objective and rendering process. While 3DGS models scene appearance and geometry through differentiable rasterization (splatting), GPF represents irradiance arriving at Lambertian surfaces. As shown in Figure 2, instead of rasterizing Gaussians to reproduce appearance, we query the field after explicitly tracing camera-side transport and apply the terminal material response. Differentiable Gaussian interpolation enables field fitting with respect to its continuous parameters, while scene geometry and training-camera paths remain fixed. GPF thus extends Gaussian primitives to reusable surface illumination, coupling a learned representation to physically traced camera transport.

Density Learning with Gaussian Mixtures. Gaussian mixture models (GMM) (Condor et al., 2025; Xu et al., 2013; Green et al., 2006; Vorba et al., 2014; Yan et al., 2016; Jakob et al., 2011; Zhou et al., 2024) have also been used to approximate light transport. Jakob *et al.* (Jakob et al., 2011) proposed using Expectation-Maximization (EM) to fit Gaussian mixtures to photon distributions in *volumetric* participating media. Their derivation relies on a key property: a 3D Gaussian integrates to unity over 3D space, *i.e.*, $\int G(\mathbf{x})dV = 1$, enabling closed-form density estimation. However, this formulation does not transfer to *surface* photon mapping. For photons deposited on 2D manifolds, kernel density estimation requires $\int K(\mathbf{x})dS = 1$ over the surface. A 3D Gaussian does not satisfy this constraint, its integral over an arbitrary surface S depends on the position, orientation, and local geometry of S relative to the Gaussian, with no closed-form solution. Fitting 2D Gaussians in a global UV parameterization is also impractical for complex geometry with discontinuous or overlapping parameterizations. Our method differs in 2 aspects: (1) **Learning target**: EM-GMM fits photon *density* and analytically derives radiance; GPF learns surface *irradiance* coefficients, supervised through the recomposed pixel radiance. (2) **Optimization**: EM-GMM uses unsupervised EM; GPF jointly optimizes illumination coefficients and Gaussian spatial support using gradients through the image predictions.

3 PRELIMINARIES

Our work bridges classical light transport simulation with learnable representations. We first review the rendering equation and stochastic progressive photon mapping, then build on their separation of light-side transport and camera-side evaluation.

3.1 RENDERING EQUATION

Physically-based rendering computes the outgoing radiance from a surface point $\mathbf{x}$ in direction ω_o (Kajiya, 1986):

$$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_\Omega f_r(\mathbf{x}, \omega_i, \omega_o) L_i(\mathbf{x}, \omega_i)(\mathbf{n}^\top \omega_i)\, d\omega_i, \tag{1}$$

where L_e is emitted radiance, L_i is incident radiance, f_r is the BRDF (Nicodemus, 1965), and Ω is the incoming hemisphere around the surface normal $\mathbf{n}$. For a Lambertian surface, $f_r = \boldsymbol{\rho}/\pi$, with RGB reflectance $\boldsymbol{\rho}$, so the reflected radiance is $\boldsymbol{\rho}E/\pi$, where E is incident irradiance integrated over incoming directions. This separates the terminal material response from the illumination arriving at the surface.

The difficulty lies in computing that illumination: incident radiance depends recursively on transport elsewhere in the scene. Concentrated light paths, such as refractive caustics, can be sampled inefficiently by unidirectional path tracing, producing high variance at practical sample budgets.

3.2 STOCHASTIC PROGRESSIVE PHOTON MAPPING

Photon mapping (Jensen, 1996) traces light paths toward receiving surfaces; progressive variants (Hachisuka et al., 2008; Hachisuka & Jensen, 2009) refine their estimates over repeated passes. In stochastic progressive photon mapping (SPPM), each iteration samples camera paths to obtain visible surface points and traces fresh light paths whose deposits carry position, incoming direction, and flux $\Delta\boldsymbol{\Phi}_j$. For example, a surface irradiance estimate from one pass of N_{emit} emitted paths is

$$\widehat{E}(\mathbf{x}) = \frac{1}{N_{\text{emit}}\pi r^2} \sum_{j \in \mathcal{B}_r(\mathbf{x})} \Delta\boldsymbol{\Phi}_j, \tag{2}$$

where $\mathcal{B}_r$ contains compatible deposits within radius r, and the stored flux has not yet been divided by the emitted-path count. Multiplication by $\boldsymbol{\rho}/\pi$ converts this quantity to reflected radiance at a Lambertian receiver.

SPPM does not simply average independent estimates with a prescribed shrinking radius. It maintains a photon count, accumulated flux, and radius *per pixel*, updating all three from the number of photons gathered in each iteration. Algorithm S1 gives the recurrence used for our reference renderer, including separately estimated direct lighting. Light-side deposits can be reused across views, but the visible points and progressive gathering statistics are camera-dependent. GPF replaces this repeated surface-lighting estimation with a learned field shared by all views.

4 METHOD

4.1 GAUSSIAN PHOTON FIELD REPRESENTATION

We transform photon-guided surface samples into a reusable Gaussian Photon Field (GPF). Each primitive combines a nonnegative RGB irradiance coefficient with a learnable position, orientation, and scale, allowing illumination values and their spatial support to be optimized jointly. We write $\mathcal{G}_i = (\boldsymbol{\mu}_i, \mathbf{q}_i, \mathbf{s}_i, \boldsymbol{\Phi}_i)$, where $\boldsymbol{\mu}_i \in \mathbb{R}^3$, $\mathbf{q}_i$ is a unit quaternion, $\mathbf{s}_i \in \mathbb{R}^3_+$, and $\boldsymbol{\Phi}_i \in \mathbb{R}^3_+$. Each primitive also has a fixed surface normal $\mathbf{n}_i$ and an initial orthonormal surface frame $\mathbf{B}_i$. Its covariance is

$$\boldsymbol{\Sigma}_i = \mathbf{B}_i \mathbf{R}(\mathbf{q}_i)\, \mathrm{diag}(\mathbf{s}_i^2) \mathbf{R}(\mathbf{q}_i)^\top \mathbf{B}_i^\top. \tag{3}$$

The quaternion rotates the Gaussian support relative to its initial surface frame, while the scene geometry and surface normals remain fixed.

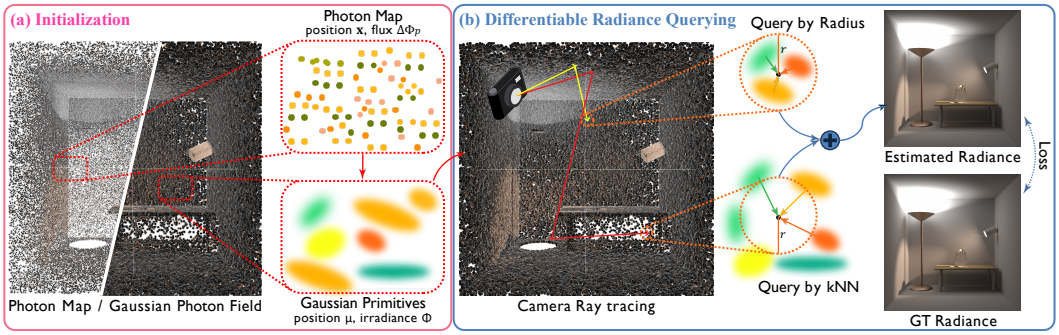


Figure 3: **Overview of the Gaussian Photon Field (GPF) pipeline.** **(a) Initialization.** Photon deposits and supplementary training-surface samples seed surface-aligned Gaussian primitives. Their RGB irradiance coefficients are initialized from the reference images and traced camera paths. **(b) Differentiable radiance querying.** Camera paths explicitly trace specular and glossy transport to Lambertian terminal surfaces. Surface-aware Gaussian interpolation estimates irradiance, which is multiplied by the terminal reflectance divided by π and accumulated into the pixel prediction. Image-space supervision jointly optimizes Gaussian coefficients, positions, rotations, and scales. The fitted field is reused across novel views.

Photon-guided coverage. A light-tracing pass seeds Gaussian centers at photon deposits, allocating support according to light transport rather than image appearance alone. We supplement these centers with voxel-subsampled terminal surface points from the training views to cover supervised regions with sparse photon support. The resulting field combines light-guided placement with coverage of the surfaces needed to explain the reference images.

Surface-aligned support. We initialize each covariance in its local surface frame, with two equal tangential scales derived from neighboring-center distances and a smaller normal scale. Starting from these surface-aligned kernels, optimization adjusts positions, rotations, and scales to fit the spatial structure of illumination.

Radiometric initialization. The coefficients $\mathbf{\Phi}_i$ represent irradiance, in the scene's linear radiometric scale, rather than conserved per-photon flux. We initialize them from SPPM reference images after removing the traced camera-prefix contribution and accounting for the terminal reflectance and prefix throughput. Surface-compatible observations from multiple training views stabilize these initial values. The complete pixel-rendering loss subsequently refines both coefficients and support; Appendix A.2 details the initialization.

4.2 Differentiable Radiance Querying

GPF combines explicitly traced camera transport with a reusable surface illumination field. At each Lambertian terminal interaction, a field query replaces further light-transport estimation; specular, dielectric, and glossy interactions preceding that terminal remain explicitly traced for every view.

Camera transport and terminal response. Let $E_{\mathrm{GPF}}(\mathbf{x}, \mathbf{n})$ denote the queried irradiance at terminal position $\mathbf{x}$ with normal $\mathbf{n}$. Its reflected outgoing radiance is

$$L_{\mathrm{GPF}}(\mathbf{x}, \mathbf{n}) = \frac{\boldsymbol{\rho}(\mathbf{x})}{\pi} E_{\mathrm{GPF}}(\mathbf{x}, \mathbf{n}). \tag{4}$$

For pixel p, we average S camera-path samples:

$$\widehat{L}_p = L_{\mathrm{prefix},p} + \frac{1}{S} \sum_{s \in \mathcal{T}_p} \boldsymbol{\beta}_{p,s} L_{\mathrm{GPF}}(\mathbf{x}_{p,s}, \mathbf{n}_{p,s}), \tag{5}$$

where $\mathcal{T}_p$ indexes samples reaching a valid Lambertian terminal, $\boldsymbol{\beta}_{p,s}$ is their accumulated camera-prefix throughput, and $L_{\mathrm{prefix},p}$ is the already sample-averaged emitter contribution along the traced prefixes. Products of RGB quantities are componentwise.

Each nonterminal BSDF sample contributes its importance weight, including the cosine and sampling-PDF ratio, to $\boldsymbol{\beta}$. At the terminal, the material response is the deterministic factor $\boldsymbol{\rho}/\pi$ in Eq. 4, applied

exactly once. This decomposition retains view-dependent camera-side reflection and refraction while caching the light-side transport arriving at diffuse surfaces. A single RGB coefficient per primitive suffices because Lambertian reflected radiance is independent of outgoing direction. Algorithm S2 specifies the camera integration.

Surface-aware Gaussian interpolation. For each query, we retrieve K_c nearest centers and evaluate their spatial and normal compatibility:

$$a_i(\mathbf{x}, \mathbf{n}) = -\tfrac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1}(\mathbf{x} - \boldsymbol{\mu}_i) + \eta \log\big(\max(|\mathbf{n}^\top \mathbf{n}_i|, \epsilon_n)\big)\,. \tag{6}$$

The Gaussian term adapts to the learned anisotropic support; the normal term downweights differently oriented surfaces. We retain the K candidates with the largest log weights as $\mathcal{N}(\mathbf{x}, \mathbf{n})$ and compute

$$E_{\mathrm{GPF}}(\mathbf{x}, \mathbf{n}) = \sum_{i \in \mathcal{N}(\mathbf{x},\mathbf{n})} \frac{\exp(a_i - a_{\max})}{\sum_{j \in \mathcal{N}(\mathbf{x},\mathbf{n})} \exp(a_j - a_{\max})} \boldsymbol{\Phi}_i, \tag{7}$$

where $a_{\max}$ is the largest retained log weight. This stable normalization interpolates irradiance-valued coefficients with dimensionless weights: the coefficients carry the absolute illumination scale, while their positions and covariances control its spatial variation. We use $K_c = 64$ candidates and $K = 32$ retained primitives. Algorithm S3 gives the query.

4.3 Supervision and Training

Sparse multi-view supervision. We generate linear-RGB reference images $L_{\mathrm{ref},p}$ with SPPM from a small set of training cameras. These are pixel-radiance targets: supervision compares the fully recomposed prediction in Eq. 5 with the reference, rather than assigning a pixel's radiance directly to one surface point. The one-time reference generation and field fitting produce a representation that is then frozen and reused for novel views.

Reusable training paths. Since scene geometry, materials, lighting, and training cameras are fixed, we precompute the camera prefixes, terminal positions and normals, and their throughput–reflectance factors. Each optimization step samples a pixel minibatch, queries the current field at all terminal records associated with those pixels, and accumulates their contributions into complete predictions. This preserves the image-formation model while avoiding repeated camera tracing during field fitting.

Joint optimization. We optimize all four parameter groups $(\boldsymbol{\Phi}_i, \boldsymbol{\mu}_i, \mathbf{q}_i, \mathbf{s}_i)$ with

$$\mathcal{L} = \mathcal{L}_{\mathrm{image}} + \lambda_a \mathcal{L}_{\mathrm{anchor}} + \lambda_s \mathcal{L}_{\mathrm{smooth}} + \lambda_g \mathcal{L}_{\mathrm{geometry}}. \tag{8}$$

The image objective combines display-space, compressed-HDR, and clipped-linear errors between $\widehat{L}$ and L_{ref}. An initialization anchor and local surface smoothness regularize the field values, while geometric penalties control center displacement, scale changes, and covariance tilt. Positive coefficients and scales are parameterized through softplus and exponentiation, respectively.

After a short RGB-only warmup, all parameter groups are optimized jointly and the neighbor index is refreshed as centers move. Gradients flow through the selected Gaussian weights, coefficients, and pixel accumulation; nearest-neighbor retrieval and top-K selection are discrete, giving a piecewise-differentiable fitting pipeline. Appendix A.2 provides the parameterization, losses, and optimization settings.

5 Experiments

We validate our method on our newly-proposed challenging benchmark, comparing to both classical physically-based integrators and standard neural rendering approaches and justifying design choices via multiple ablation studies.

5.1 Experimental Setup

Datasets. Since no standard or well-established datasets exist for this task, we propose a new one consisting of five physically-based rendering scenes that cover a wide range of light-transport phenomena, including specular–diffuse interactions, caustics, and glossy reflections:

Table 1: **Comparison across 5 complex scenes**, reporting PSNR, SSIM, LPIPS, and storage cost (in MB). Best and 2nd best results are highlighted.

	Veach-Bidir				*Veach-Ajar*				*Water-Caustic*			
Method	PSNR↑	SSIM↑	LPIPS↓	Storage↓	PSNR↑	SSIM↑	LPIPS↓	Storage↓	PSNR↑	SSIM↑	LPIPS↓	Storage↓
Path Tracer 10 spp	16.01	0.134	0.778	–	9.276	0.098	0.651	–	12.09	0.221	0.663	–
Path Tracer 50 spp	19.90	0.224	0.716	–	15.21	0.242	0.531	–	12.86	0.319	0.573	–
Particle Tracer 64 spp	25.29	0.587	0.365	–	14.19	0.240	0.750	–	12.31	0.542	0.476	–
SPPM - 3 it	18.01	0.146	0.696	26.49	14.58	0.314	0.516	405.3	26.53	0.664	0.413	14.16
EM-GMM	13.52	0.095	0.823	–	8.14	0.072	0.708	–	12.39	0.185	0.702	–
RenderFormer	14.23	0.108	0.812	–	8.56	0.081	0.692	–	11.34	0.198	0.695	–
GPF (Ours)	37.61	0.926	0.338	18.33	25.84	0.691	0.324	267.2	27.12	0.936	0.158	9.724

	Water-Caustic 2				*Cornell Box*				**Mean**			
Method	PSNR↑	SSIM↑	LPIPS↓	Storage↓	PSNR↑	SSIM↑	LPIPS↓	Storage↓	PSNR↑	SSIM↑	LPIPS↓	Storage↓
Path Tracer 10 spp	11.88	0.248	0.637	–	30.49	0.654	0.497	–	15.95	0.271	0.645	–
Path Tracer 50 spp	12.24	0.304	0.579	–	34.59	0.850	0.403	–	18.96	0.388	0.560	–
Particle Tracer 64 spp	12.08	0.528	0.489	–	33.89	0.820	0.370	–	19.55	0.543	0.490	–
SPPM - 3 it	25.64	0.632	0.412	16.26	30.47	0.754	0.399	21.63	23.05	0.502	0.487	96.77
EM-GMM	10.47	0.192	0.688	–	13.26	0.205	0.558	–	11.56	0.150	0.696	–
RenderFormer	10.92	0.210	0.672	–	32.14	0.738	0.452	–	15.44	0.267	0.665	–
GPF (Ours)	28.72	0.962	0.196	12.34	39.73	0.936	0.183	15.46	31.80	0.890	0.240	64.61

- *Veach-Bidir* and *Veach-Ajar*: classic Mitsuba-3-compatible test scenes from Bitterli's rendering resources (Bitterli, 2016), featuring multi-bounce specular–diffuse paths and complex indirect illumination.
- *Water-Caustic* and *Water-Caustic 2*: scenes inspired by Bitterli (2016) and reconfigured for Mitsuba 3, with challenging refractive and specular-diffuse caustics.
- *Cornell Box* (Jakob et al., 2022b): canonical diffuse reference scene to evaluate radiometric accuracy and convergence behavior.

5.2 Quantitative Comparisons

Comparison with Baselines. Table 1 compares GPF against physically-based integrators (path/particle tracing (Jakob et al., 2022b), SPPM (Hachisuka & Jensen, 2009)), Gaussian mixture modeling (EM-GMM (Jakob et al., 2011)), and feed-forward neural rendering (RenderFormer (Zeng et al., 2025)). For EM-GMM, we reproduce the method in Mitsuba 3 as no public code is available. For RenderFormer, we use the official code and convert our scenes to their format.

The results show that GPF achieves the highest overall performance across all metrics (PSNR (Eskicioglu & Fisher, 1995), SSIM (Wang et al., 2004), LPIPS (Zhang et al., 2018)). In scenes dominated by indirect lighting (*Veach-Bidir*, *Veach-Ajar*), our method significantly surpasses all baselines in both SSIM and LPIPS, demonstrating its ability to capture smooth, high-frequency radiance variations while maintaining global consistency. For caustic-dominated scenes (*Water-Caustic*, *Water–Caustic 2*), GPF achieves the best PSNR and substantially outperforms others in SSIM and LPIPS, highlighting its capacity to accurately model concentrated photon energy distributions through the learned continuous field. While PSNR is a pixel-wise metric sensitive to small local deviations, our method preserves perceptual and structural fidelity much better, as reflected by large gains in SSIM and LPIPS. Even on the simple *Cornell Box*, GPF consistently outperforms all baselines across metrics, confirming its robustness and general applicability to diverse light transport phenomena.

We further compare against two methods that represent alternative paradigms for accelerating radiance computation, but both fail on our scenes. EM-GMM (Jakob et al., 2011), the only prior method applying Gaussian mixtures to photon-based light transport, serves as the most directly relevant baseline. However, its volumetric kernel normalization ($\int G(\mathbf{x})\, dV = 1$) does not hold for photons deposited on surfaces, resulting in incorrect radiance estimates (see Sec. 2 for a detailed discussion). RenderFormer (Zeng et al., 2025) represents the feed-forward neural rendering paradigm, sharing GPF's motivation of replacing repeated computation with a learned mapping. However, it is trained exclusively on opaque materials and cannot handle transparency or refraction, precluding caustics. These failures underscore that complex light transport demands physics-grounded representations rather than purely statistical or data-driven alternatives.

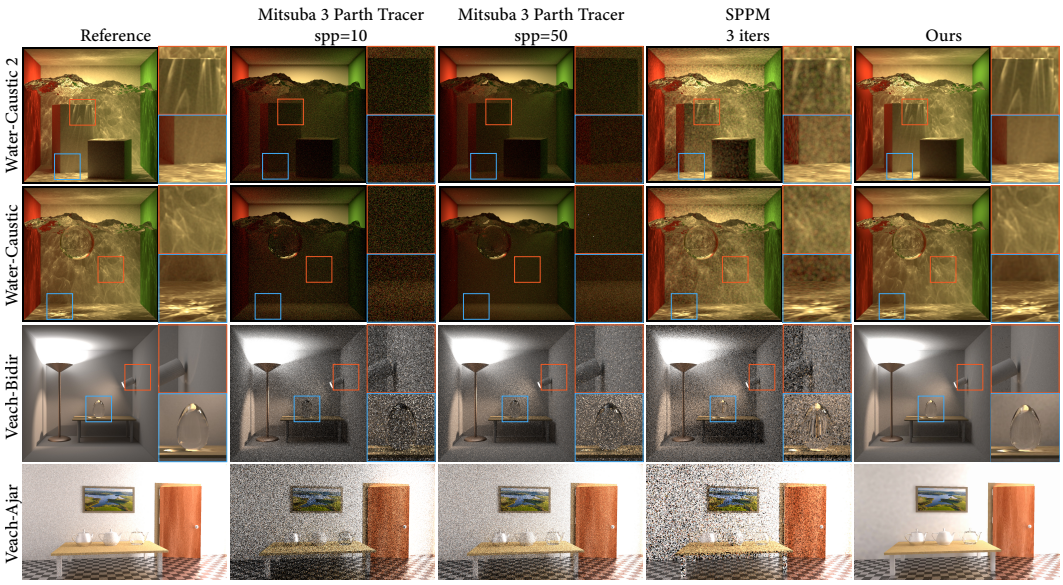


Figure 4: **Qualitative comparison results.** We compare GPF with path tracing and *SPPM* on scenes with complex indirect illumination and caustics. Our method yields cleaner results with fewer noise artifacts and more accurate light transport, especially in challenging caustic and multi-bounce diffuse regions. Please 🔍 zoom in for details.

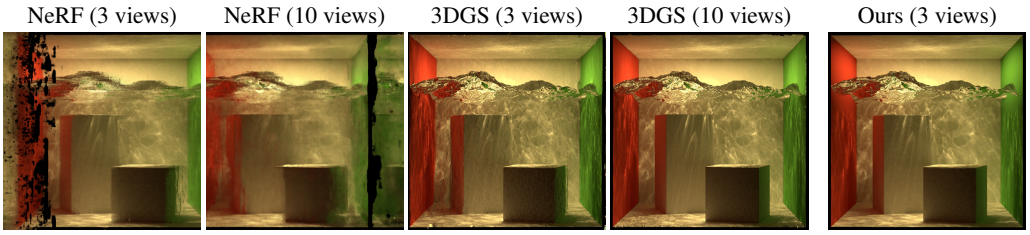
Table 2: **Comparison with Instant-NGP, 3DGS, AnySplat** on caustic scenes, with varying training view numbers. Best and 2nd best results are highlighted.

	Water-Caustic			*Water-Caustic 2*		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Instant-NGP (Müller et al., 2022) - 3 views	13.42	0.466	0.748	13.09	0.460	0.734
Instant-NGP (Müller et al., 2022) - 10 views	12.59	0.473	0.618	12.44	0.480	0.620
3DGS (Kerbl et al., 2023) - 3 views	16.76	0.508	0.493	17.80	0.550	0.485
3DGS (Kerbl et al., 2023) - 10 views	26.58	0.812	0.246	26.44	0.843	0.244
AnySplat (Jiang et al., 2025) - 3 views	17.24	0.522	0.476	18.35	0.572	0.468
AnySplat (Jiang et al., 2025) - 10 views	26.83	0.825	0.238	26.71	0.858	0.235
GPF (Ours) - 3 views	27.12	0.936	0.158	28.72	0.962	0.196

Comparison with Neural Scene Representations. To contextualize GPF within the broader paradigm of continuous scene representations, we compare it with neural radiance field methods such as Instant-NGP (Müller et al., 2022), 3DGS (Kerbl et al., 2023), and AnySplat (Jiang et al., 2025) in Table 2. We emphasize that this comparison is not intended as an equal-input benchmark. Neural baselines operate under fundamentally different assumptions: they rely solely on RGB images and camera poses, and primarily optimize for appearance reconstruction. In contrast, GPF assumes access to scene geometry, materials, and lighting, and explicitly models physics-informed light transport. Our goal in this comparison is conceptual rather than competitive: to evaluate whether purely appearance-driven representations can faithfully reconstruct complex global illumination phenomena, *e.g.*, caustics and specular–diffuse transport. Even when given significantly more input views than GPF, these methods struggle to recover high-frequency and physically-consistent illumination effects under sparse supervision. This suggests that physics-grounded modeling remains essential for accurate and view-consistent reconstruction of complex light transport.

5.3 Qualitative Evaluation

Figure 4 presents qualitative comparisons across three representative scenes, with more exhaustive results in supplementary material. For the *Water-Caustic* scene, GPF produces sharp, accurate caustics that capture the subtle refractive patterns, while *SPPM - 3 iters* yields blurred illumination due to its



To view the animation, please use compatible software (*e.g.*, *Adobe Acrobat*, KDE).

Figure 5: **Multi-view comparison on *Water-Caustic 2* with NeRF and 3DGS.** Unlike NeRF and 3DGS, which exhibit view-dependent artifacts under sparse supervision, GPF preserves consistent caustics and global illumination even with 3 views.

limited photon budget and path tracing (Jakob et al., 2022b) (PT) fails to recover detailed structures even at 50 spp. In the *Veach–Bidir* scene, which features complex multi-bounce diffuse transport and glossy interreflections, GPF achieves a clean, noise-free reconstruction with well-defined highlights, unlike *SPPM - 3 iters* and path tracing, which exhibit noise and structural artifacts. Similarly, in the *Veach-Ajar* scene, GPF provides smooth, stable illumination with accurate indirect lighting, whereas other integrators suffer from high-frequency noise and incomplete light transport. Overall, GPF combines the physical accuracy of photon-based rendering with the smoothness of a continuous learned radiance field, delivering superior results for both caustic and indirect illumination.

Furthermore, as shown in Figure 5, our photon field enables strong consistency across novel viewpoints. Despite large changes in camera pose and viewing direction, both direct and indirect lighting remain coherent, with stable caustics and diffuse shading. This highlights the robustness and view-reusability of our learned field, effectively amortizing photon-based light transport across views. Additional visualizations for all scenes are provided in the appendix.

5.4 Efficiency and Amortization

GPF moves the estimation of terminal-surface illumination into a one-time fitting stage and reuses the resulting field across viewpoints. Once fitting is complete, the Gaussian parameters and spatial index remain fixed. Each novel view retraces camera-side transport, queries the cached illumination, and accumulates the pixel contributions. On Water-Caustic at 1024×1024, this workflow averages 0.92 seconds per view on an A100 GPU over the 60-view trajectory. The timing includes camera tracing, terminal-state compaction, field evaluation, pixel accumulation, and CPU transfer.

The recorded one-time cost is 85.1 minutes, comprising 20.5 minutes for three training references, 63.7 minutes for the full 30k-step optimization, and approximately 0.9 minutes for field preparation and inference initialization. Each SPPM training reference uses 2000 iterations with 100k emitted paths per iteration and takes 6.84 minutes on average. Using these timings, we estimate a crossover at the 13th additional view. Under the same accounting, rendering 100 additional views would cost approximately 86.7 minutes with GPF versus 11.4 hours for repeated SPPM reference generation. The amortized GPF cost is then approximately 52 seconds per rendered view. This illustrates how the initial fitting cost is amortized over a camera sequence while the per-view work remains small. Appendix C provides the cost accounting and five-scene timings.

6 Conclusion

We introduce the Gaussian Photon Field (GPF), a learnable surface-irradiance representation that connects photon mapping with reusable Gaussian fields. Photon-guided placement and reference-based irradiance initialization provide the starting point for jointly optimizing illumination coefficients and spatial support. At rendering time, the field is combined with the terminal Lambertian response and explicitly traced camera-side transport, enabling efficient multi-view global illumination without repeated photon tracing or progressive refinement. This separation of reusable light-side illumination and view-dependent camera transport brings the efficiency of learned scene representations to photon-based rendering. Future work will explore extending differentiable field fitting toward inverse caustics.

AI USE STATEMENT

We used generative AI tools to assist with code implementation and debugging, experimental design and result interpretation, literature review, translation, and manuscript editing. The authors reviewed the AI-assisted content and take full responsibility for the final manuscript, implementation, and reported results.

REPRODUCIBILITY STATEMENT

We ensure full reproducibility through comprehensive documentation of our formulations, algorithms, and evaluation protocols. Section 4 defines the representation, radiance query, and training objective of GPF. Appendix A provides the remaining implementation details: software versions, the SPPM reference integrator and its parameters, and the GPF support construction, coefficient initialization, parameterization, loss weights, learning rates, step budget, and checkpoint-selection protocol. Algorithm S1, Algorithm S2, and Algorithm S3 give pseudocode for the reference integrator, camera-prefix tracing, and field query. Section 5.1 describes the five evaluation scenes, and Appendix A.2 specifies the training and validation views and image resolutions. Section 5.2 states how the baselines were run, including our reimplementation of EM-GMM. For the efficiency results, Appendix C specifies the hardware, which operations are included in the reported timings, and the stage-based cost accounting behind the crossover estimate (Eq. 14, Tables S2 and S3).

ETHICS STATEMENT

All authors have read and adhere to the ICLR Code of Ethics. This work concerns the efficient simulation of light transport for physically-based rendering. It involves no human subjects, crowdsourcing, or personally identifiable information. All experiments use synthetic scenes: Veach-Bidir and Veach-Ajar are taken from publicly available rendering resources (Bitterli, 2016), the Cornell Box from Mitsuba 3 (Jakob et al., 2022b), and the two Water-Caustic scenes are inspired by Bitterli (2016) and reconfigured for Mitsuba 3. These scenes, the Mitsuba 3 and Dr.Jit (Jakob et al., 2022a) frameworks (BSD-3-Clause), and the baseline implementations are credited and used under the terms of their licenses. GPF is not a generative model: it is fitted per scene to reproduce the light transport of a fully specified scene description (geometry, materials, and emitters), and it neither learns from nor reconstructs real-world photographs, people, or environments. We therefore do not foresee misuse risks beyond those of physically-based rendering software in general.

REFERENCES

Dejan Azinovic, Tzu-Mao Li, Anton Kaplanyan, and Matthias Nießner. Inverse path tracing for joint material and lighting estimation. In *CVPR*, 2019.

Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. In *TPAMI*, 2014.

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

Benedikt Bitterli, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. In *TOG*, 2020.

Ting-Hsuan Chen, Jie Wen Chan, Hau-Shiang Shiu, Shih-Han Yen, Changhan Yeh, and Yu-Lun Liu. Narcan: Natural refined canonical image with integration of diffusion prior for video editing. In *NeurIPS*, 2024.

Jorge Condor, Sebastien Speierer, Lukas Bode, Aljaz Bozic, Simon Green, Piotr Didyk, and Adrian Jarabo. Don't splat your gaussians: Volumetric ray-traced primitives for modeling and rendering scattering and emissive media. *ACM Transactions on Graphics (TOG)*, 44(1):1–17, February 2025. ISSN 1557-7368. doi: 10.1145/3711853. URL http://dx.doi.org/10.1145/3711853.

A. M. Eskicioglu and P. S. Fisher. Image quality measures and their performance. *IEEE Transactions on Communications*, 43(12):2959–2965, 1995. doi: 10.1109/26.477498.

Qianqian Fang and David A Boas. Monte carlo simulation of photon migration in 3d turbid media accelerated by graphics processing units. *Optics express*, 17(22):20178–20190, 2009.

Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modeling the interaction of light between diffuse surfaces. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '84, pp. 213–222, New York, NY, USA, 1984. Association for Computing Machinery. ISBN 0897911385. doi: 10.1145/800031.808601. URL https://doi.org/10.1145/800031.808601.

Paul Green, Jan Kautz, Wojciech Matusik, and Frédo Durand. View-dependent precomputed light transport using nonlinear gaussian function approximations. In *Proceedings of I3D*, 2006.

Xiang Guo, Jiadai Sun, Yuchao Dai, Guanying Chen, Xiaoqing Ye, Xiao Tan, Errui Ding, Yumeng Zhang, and Jingdong Wang. Forward flow for novel view synthesis of dynamic scenes, 2023.

Toshiya Hachisuka and Henrik Wann Jensen. Stochastic progressive photon mapping. *ACM Transactions on Graphics (TOG)*, 28(5):1–8, December 2009. ISSN 0730-0301. doi: 10.1145/1618452.1618487. URL https://doi.org/10.1145/1618452.1618487.

Toshiya Hachisuka, Shinji Ogaki, and Henrik Wann Jensen. Progressive photon mapping. *ACM Transactions on Graphics (TOG)*, 27(5), December 2008. ISSN 0730-0301. doi: 10.1145/1409060.1409083. URL https://doi.org/10.1145/1409060.1409083.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24*, SIGGRAPH '24, pp. 1–11. ACM, July 2024. doi: 10.1145/3641519.3657428. URL http://dx.doi.org/10.1145/3641519.3657428.

Wenzel Jakob, Christian Regg, and Wojciech Jarosz. Progressive Expectation–Maximization for hierarchical volumetric photon mapping. *Computer Graphics Forum*, 30(4), 2011.

Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, and Delio Vicini. Dr.jit: A just-in-time compiler for differentiable rendering. *Transactions on Graphics (Proceedings of SIGGRAPH)*, 41(4), July 2022a. doi: 10.1145/3528223.3530099.

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

Henrik Wann Jensen. Global illumination using photon maps. In *Proceedings of the Eurographics Workshop on Rendering Techniques '96*, pp. 21–30, Berlin, Heidelberg, 1996. Springer-Verlag. ISBN 3211828834.

Henrik Wann Jensen. *Realistic image synthesis using photon mapping*. AK Peters/crc Press, 2001.

Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, et al. Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. *ACM Transactions on Graphics (TOG)*, 44(6):1–16, 2025.

James T. Kajiya. The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '86, pp. 143–150, New York, NY, USA, 1986. Association for Computing Machinery. ISBN 0897911962. doi: 10.1145/15922.15902. URL https://doi.org/10.1145/15922.15902.

Alexander Keller. Instant radiosity. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 49–56, 1997.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4), July 2023. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

Claude Knaus and Matthias Zwicker. Progressive photon mapping: A probabilistic approach. *ACM Transactions on Graphics (TOG)*, 30(3), May 2011. ISSN 0730-0301. doi: 10.1145/1966394.1966404. URL https://doi.org/10.1145/1966394.1966404.

Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. Robust dynamic radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Caoyuan Ma, Yu-Lun Liu, Zhixiang Wang, Wu Liu, Xinchen Liu, and Zheng Wang. Humannerf-se: A simple yet effective approach to animate humannerf with diverse poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106, 2021.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (TOG)*, 41(4):102:1–102:15, July 2022. doi: 10.1145/3528223.3530127. URL https://doi.org/10.1145/3528223.3530127.

Fred E Nicodemus. Directional reflectance and emissivity of an opaque surface. *Applied optics*, 4(7):767–775, 1965.

Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5865–5874, October 2021a. URL https://openaccess.thecvf.com/content/ICCV2021/html/Park_Nerfies_Deformable_Neural_Radiance_Fields_ICCV_2021_paper.html.

Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021b.

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10318–10327, 2021.

David Sarrut, Thomas Baudier, Damian Borys, Ane Etxebeste, Hermann Fuchs, Jan Gajewski, Loïc Grevillot, Sébastien Jan, George C Kagadis, Han Gyu Kang, et al. The opengate ecosystem for monte carlo simulation in medical physics. *Physics in Medicine & Biology*, 67(18):184001, 2022.

Bernard W Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.

Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Nonrigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Eric Veach. *Robust monte carlo methods for light transport simulation*. PhD thesis, Stanford, CA, USA, 1998. AAI9837162.

Jiří Vorba, Ondřej Karlík, Martin Šik, Tobias Ritschel, and Jaroslav Křivánek. On-line learning of parametric mixture models for light transport simulation. *ACM Transactions on Graphics (TOG)*, 33(4), 2014.

Lihong Wang, Steven L Jacques, and Liqiong Zheng. Mcml—monte carlo modeling of light transport in multi-layered tissues. *Computer methods and programs in biomedicine*, 47(2):131–146, 1995.

Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing (TIP)*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

Chun-Hung Wu, Shih-Hong Chen, Chih-Yao Hu, Hsin-Yu Wu, Kai-Hsin Chen, Yu-You Chen, Chih-Hai Su, Chih-Kuo Lee, and Yu-Lun Liu. Denver: Deformable neural vessel representations for unsupervised video vessel segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9421–9431, 2021.

Kun Xu, Wei-Lun Sun, Zhao Dong, Dan-Yong Zhao, Run-Dong Wu, and Shi-Min Hu. Anisotropic spherical gaussians. *ACM Transactions on Graphics (TOG)*, 32(6), 2013.

Ling-Qi Yan, Miloš Hašan, Steve Marschner, and Ravi Ramamoorthi. Position-normal distributions for efficient rendering of specular microstructure. *ACM Transactions on Graphics (TOG)*, 35(4), 2016.

Ziyi Yang, Xinyu Gao, Yangtian Sun, Yihua Huang, Xiaoyang Lyu, Wen Zhou, Shaohui Jiao, Xiaojuan Qi, and Xiaogang Jin. Spec-gaussian: Anisotropic view-dependent appearance for 3d gaussian splatting, 2024. Preprint.

Chong Zeng, Yue Dong, Pieter Peers, Hongzhi Wu, and Xin Tong. Renderformer: Transformer-based neural rendering of triangle meshes with global illumination. In *ACM SIGGRAPH 2025 Conference Papers*, 2025.

Aijia Zhang, Yan Zhao, and Shigang Wang. Illumination estimation for augmented reality based on a global illumination model. *Multimedia tools and applications*, 78(23):33487–33503, 2019.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18643–18652, 2022.

Yang Zhou, Songyin Wu, and Ling-Qi Yan. Unified gaussian primitives for scene representation and rendering. *arXiv preprint arXiv:2406.09733*, 2024.

In this supplementary material, we provide implementation details and algorithms (Section A), ablations (Section B), an efficiency analysis (Section C), additional results (Section D), and limitations (Section E).

A Implementation Details

We use Mitsuba 3.6.0 (Jakob et al., 2022b) with Dr.Jit 1.0.1 (Jakob et al., 2022a) for camera and photon transport, and PyTorch for Gaussian-field optimization and pixel accumulation. During joint fitting, Open3D CUDA nearest-neighbor search indexes the updated Gaussian centers. For inference, a static exact CUDA KD-tree indexes the frozen centers and is reused across views (Appendix C). Both the reference renderer and GPF use the same scene geometry, materials, emitters, and camera definitions. Algorithms Algorithm S1–Algorithm S3 describe the surface-rendering pipeline.

A.1 SPPM Reference Integrator

Each iteration traces jittered camera samples to Lambertian visible points and emits N_{emit} light paths in importance transport mode. Photon records retain position, flux, incoming direction, and shading normal; transport includes the shading-normal adjoint correction. The camera pass estimates direct illumination by next-event estimation at smooth interactions. Explicit emitter hits contribute at the primary intersection or after a delta event, avoiding double counting with next-event estimation. Accordingly, the indirect photon gather excludes deposits at the first light-path surface interaction.

The GPU gather builds an index over the current visible points, finds candidate photon–point pairs within the largest active radius, and then filters each pair by its pixel's own radius. We require a positive incoming-direction dot product with the receiving normal and an absolute photon/receiver normal dot product of at least 0.9. For each pixel, the stored count N, radius r, and flux accumulator $\boldsymbol{\tau}$ are updated using the iteration's gathered count M and flux sum $\mathbf{F}$:

$$N' = N + \alpha M, \qquad a = \frac{N'}{N+M}, \qquad r' = r\sqrt{a}, \tag{9}$$

$$\boldsymbol{\tau}' = a\left(\boldsymbol{\tau} + \boldsymbol{\beta}\frac{\boldsymbol{\rho}}{\pi}\mathbf{F}\right). \tag{10}$$

The update is skipped when $M = 0$. With $\mathbf{D}$ the accumulated camera-pass contribution, the final image is $\mathbf{D}/T + \boldsymbol{\tau}/(TN_{\text{emit}}\pi r^2)$. We use $\alpha = 0.7$, initial radius 0.02 in scene units, $100{,}000$ emitted paths per iteration, maximum path depth 10, and photon Russian roulette starting at depth 4. Training references use $2{,}000$ iterations for Water-Caustic 2, Water+sphere, and Veach-Ajar, and $5{,}000$ for Veach-Bidir. These are sampling budgets; reference convergence is evaluated separately.

Listing Algorithm S1: **Surface SPPM reference rendering.** Each pixel maintains its own progressive photon count, radius, and flux. The camera pass estimates direct lighting separately; the photon pass estimates the remaining surface transport.

```
def surface_sppm(scene, camera, T, N_emit, r0=0.02, alpha=0.7):
    N, tau, D = zeros_pixels(), zeros_rgb(), zeros_rgb()
    radius = full_pixels(r0)
    for t in range(T):
        # Jitter camera samples; stop at Lambertian visible points.
        # d includes camera-side emission and next-event estimation.
        vp, d = trace_sppm_camera_pass(scene, camera, t)
        D += d
        # Importance transport; omit first-surface deposits because
        # direct illumination is estimated by the camera pass.
        photons = trace_light_paths(
            scene, N_emit, mode="importance", omit_direct=True)
        index = build_visible_point_index(vp.position)
        M, F = zeros_visible_points(vp), zeros_visible_rgb(vp)
        for photon in photons:
            candidates = radius_query(index, photon.x, max(radius))
            for v in candidates:
```

```
                p = vp[v].pixel
                if norm(photon.x - vp[v].x) > radius[p]:
                    continue
                if dot(vp[v].n, photon.wi) <= 0:
                    continue
                if abs(dot(vp[v].n, photon.n)) < 0.9:
                    continue
                M[v] += 1
                F[v] += photon.flux    # Not divided by N_emit yet.
        for v in vp:
            if M[v] == 0:
                continue
            p = vp[v].pixel
            N_new = N[p] + alpha * M[v]
            area_ratio = N_new / (N[p] + M[v])
            factor = vp[v].beta * vp[v].rho / pi
            tau[p] = (tau[p] + factor * F[v]) * area_ratio
            radius[p] *= sqrt(area_ratio)
            N[p] = N_new
    return D / T + tau / (T * N_emit * pi * radius**2)
```

A.2 GPF Initialization, Rendering, and Optimization

Support construction. A separate light-tracing pass emits 100,000 paths and supplies the photon-deposit centers. A path may deposit at multiple surfaces, so the number of centers is not the emitted-path count. We append one representative supervised terminal point per occupied voxel, using voxel size $h = d/512$, where d is the diagonal length of the supervised-point bounding box. At photon centers, initial normals, reflectance metadata, and irradiance values are copied from the nearest supervised terminal observation; added support centers retain their own metadata. The surface normals remain fixed during fitting and are not rotated by the learned covariance.

The initial frame $\mathbf{B}_i$ contains two tangents and the surface normal. Its tangential scale is $\frac{1}{2}\max(m\, d_{i,8}, 10^{-6}d)$, where $d_{i,8}$ is the distance to the eighth other center (the ninth entry of the self-inclusive neighbor query). We use $m = 1.5$ for the Water scenes and $m = 1.8$ for the Veach scenes; the normal scale is one quarter of the tangential scale. All local quaternions start at the identity.

Coefficient initialization. Write the terminal record's RGB factor as $\mathbf{A}_{p,s} = \boldsymbol{\beta}_{p,s}\boldsymbol{\rho}(\mathbf{x}_{p,s})/(\pi S)$. For initialization, we form an effective irradiance observation

$$\widetilde{\mathbf{E}}_p = \frac{\max(L_{\mathrm{ref},p} - L_{\mathrm{prefix},p}, 0)}{\max(\sum_{s\in\mathcal{T}_p} \mathbf{A}_{p,s}, 10^{-6})}, \tag{11}$$

with componentwise division. We attach this observation to the first camera-prefix sample's valid terminal point, retain finite observations whose summed factor has maximum RGB component above 10^{-5}, and clip initial RGB values between 10^{-5} and their per-channel 99.9th percentiles. When a pixel's samples reach different terminals, Eq. 11 is an effective initialization, not a separate irradiance measurement at each terminal. Training instead recomposes every recorded contribution through Eq. 5.

To stabilize isolated initialization outliers, we gather up to 12 neighboring observations from each training view, retain up to four with absolute normal agreement at least 0.995 and reflectance-vector distance at most 0.025, and take their componentwise median. A view is used when its nearest compatible sample lies within $\max(6h, 4d_{\min} + 2h)$, where $d_{\min}$ is the nearest compatible distance across views. With at least two valid views, their median supplies a consensus. We replace an initial value only when its luminance is below one quarter of the consensus with consensus luminance above 0.02, or when its luminance is greater than twice the consensus luminance and also exceeds the consensus luminance by more than 0.2. All other initial values are retained. This step uses training views only and does not alter the reference-image targets.

Camera-prefix records. Algorithm Algorithm S2 separates camera tracing from field evaluation. Each record stores the pixel index, terminal position and shading normal, and factor $\mathbf{A}_{p,s}$; emitter

contributions are accumulated separately in $L_{\text{prefix},p}$. The BSDF sample at a Lambertian terminal is used only for diffuse classification; its sampled direction and importance weight are discarded. The deterministic terminal response is $\boldsymbol{\rho}/\pi$. At nonterminal interactions, the returned BSDF sample weight already contains the sampling-PDF ratio and is multiplied into throughput without an additional division. Unlike the SPPM camera pass, this GPF prefix tracer performs no next-event estimation: terminal illumination, including direct illumination, is supplied by the field, and prefix emitter hits are accumulated directly.

The Water scenes use 8 prefix samples per pixel in both training and evaluation. Veach-Bidir and Veach-Ajar use 12 for training and 16 for evaluation, with maximum depth 10. These samples independently trace camera-prefix scattering from the configured primary ray; they are not additional subpixel antialiasing samples. Training reuses the precomputed records, whereas novel-view rendering traces prefixes for the new camera and then accumulates the frozen field's contributions.

Listing Algorithm S2: **Camera transport and GPF rendering.** Camera tracing records prefix radiance and weighted terminal queries. Training reuses these records; novel-view rendering traces new ones. The terminal Lambertian factor is included once in each record.

```
def trace_gpf_records(scene, camera, S, max_depth):
    prefix, records = zeros_rgb(), []
    for p in camera.pixels:
        primary_ray, sensor_weight =
        ↪ primary_ray_and_sensor_weight(camera, p)
        for s in range(S):
            ray, beta = copy(primary_ray), copy(sensor_weight)
            for depth in range(max_depth):
                hit = scene.intersect(ray)
                if not hit.valid:
                    break
                if hit.is_emitter:
                    prefix[p] += beta * emitter_radiance(hit) / S
                    break
                # weight already includes BSDF * cosine / PDF.
                sample, weight = hit.bsdf.sample(hit)
                if sample.pdf <= 0:
                    break
                if is_diffuse(sample.type):
                    rho = hit.bsdf.eval_diffuse_reflectance(hit)
                    records.append((p, hit.x, hit.n, beta * rho /
                    ↪ (pi*S)))
                    break  # Discard terminal direction and sample
                    ↪ weight.
                beta *= weight
                if not finite(beta) or max(beta) <= 0:
                    break
                ray = hit.spawn_ray(hit.to_world(sample.wo))
    return prefix, records

def compose_gpf_image(prefix, records, G, index):
    image = prefix.copy()
    for p, x, n, factor in records:
        image[p] += factor * query_gpf_irradiance(x, n, G, index)
    return image
```

Field query. The CUDA index retrieves the 64 closest centers. We compute Eq. 6 with $\eta = 8$ and $\epsilon_n = 10^{-4}$, select the 32 largest log weights, and apply a numerically stable softmax before RGB accumulation. There is no additional distance falloff or kernel-area factor in this interpolation. The fixed normals provide surface-orientation compatibility; the covariance controls spatial support.

Listing Algorithm S3: **Surface-aware Gaussian irradiance query.** We select the strongest anisotropic, normal-compatible weights from the nearest-center candidates and interpolate non-negative RGB irradiance coefficients. The neighbor index is refreshed from current centers during joint training.

```
def query_gpf_irradiance(x, n, G, index, Kc=64, K=32, eta=8):
    candidates = knn(index, x, min(Kc, len(G)))
    log_weight = []
    for i in candidates:
        frame = G[i].B @ rotation_matrix(normalize(G[i].q))
        local = frame.T @ (x - G[i].mu)
        spatial = -0.5 * sum((local / G[i].scale)**2)
        normal = eta * log(max(abs(dot(n, G[i].normal)), 1e-4))
        log_weight.append(spatial + normal)
    selected = topk_indices(log_weight, min(K, len(candidates)))
    a = log_weight[selected]
    w = exp(a - max(a))
    w /= sum(w)
    phi = nonnegative_rgb_coefficients(G, candidates[selected])
    return sum(w[:, None] * phi, axis=0)
```

Joint parameterization. Let $\boldsymbol{\mu}_i^0$, $\mathbf{s}_i^0$, and $\mathbf{B}_i$ denote initialization values. The optimized variables are RGB logits $\mathbf{u}_i$, normalized local offsets $\mathbf{o}_i$, log-scale changes $\boldsymbol{\delta}_i$, and quaternions $\mathbf{q}_i$:

$$\boldsymbol{\Phi}_i = \mathrm{softplus}(\mathbf{u}_i), \quad \boldsymbol{\mu}_i = \boldsymbol{\mu}_i^0 + \mathbf{B}_i(\mathbf{s}_i^0 \odot \mathbf{o}_i), \quad \mathbf{s}_i = \mathbf{s}_i^0 \odot \exp(\boldsymbol{\delta}_i). \tag{12}$$

Covariances follow Eq. 3. All photon-seeded and supplementary primitives participate in joint optimization.

Image objective. For a pixel minibatch, let MSE average over pixels and RGB channels, g denote the linear-to-sRGB transform after clamping to $[0, 1]$, and c be the 99.5th percentile of the batch's reference values. We use

$$\begin{aligned}\mathcal{L}_{\mathrm{image}} = {} & 0.55\ \mathrm{MSE}\Big(g(\widehat{L}), g(L_{\mathrm{ref}})\Big) \\ & + 0.30\ \mathrm{MSE}\Big(\mathrm{asinh}(\widehat{L}/0.1), \mathrm{asinh}(L_{\mathrm{ref}}/0.1)\Big) \\ & + \frac{0.15}{\max(c^2, 10^{-6})}\ \mathrm{MSE}\Big(\mathrm{clip}(\widehat{L}, 0, c), \mathrm{clip}(L_{\mathrm{ref}}, 0, c)\Big)\,.\end{aligned} \tag{13}$$

This combines display-space detail with sensitivity to bright, scene-linear illumination.

Regularization. The anchor compares $\log(1 + E_{\mathrm{GPF}}(\boldsymbol{\mu}_i^0, \mathbf{n}_i))$ to $\log(1 + \boldsymbol{\Phi}_i^0)$ with a mean absolute error. It regularizes the *queried field*, rather than independently fixing primitive coefficients. The smoothness term is the mean absolute difference of log-irradiance queries at neighboring initial centers, using edges to the four nearest other centers whose absolute normal agreement exceeds 0.9. We sample 512 centers and 512 edges per step for these terms. The geometric penalty is the sum of the mean squared tangential offset, four times the mean squared normal offset, the mean squared log-scale change, and the mean tilt penalty $1 - [\mathbf{R}(\mathbf{q}_i)]_{33}$. The weights in Eq. 8 are $\lambda_a = 0.05$, $\lambda_s = 2 \times 10^{-4}$, and $\lambda_g = 10^{-3}$.

Optimization settings. We use three training views per scene, at 1024×1024 for Water-Caustic 2, Water+sphere, and Veach-Bidir, and 1280×720 for Veach-Ajar. Each step cycles through the training cameras and samples 8192 complete pixels, including all their terminal records. Adam learning rates are 0.02 for RGB logits, 0.002 for offsets and log scales, and 0.001 for quaternions. The first 100 steps optimize RGB alone; subsequent steps optimize all four groups and rebuild the CUDA neighbor index from the current centers at every step. We clip the global gradient norm to 10, constrain each local offset component to $[-0.5, 0.5]$, constrain log-scale changes to $[-\log 2, \log 2]$, and renormalize quaternions after each update. The joint-training budget is $30{,}000$ steps. Every 500 steps, we evaluate fixed samples of 8192 pixels per validation camera and retain the checkpoint with the highest mean validation PSNR; the selected checkpoint can therefore precede the final step. Four separate trajectory views (indices $8, 18, 38, 48$) are used for validation, not for image-loss training.

Table S1: **Ablation study on the number of Gaussian primitives.**

# Gaussians	PSNR↑	SSIM↑	LPIPS↓	Time (s)↓
10K	24.32	0.8521	0.2847	0.691
50K	26.18	0.8892	0.2156	0.862
100K	27.12	0.9355	0.1579	0.921
200K	27.25	0.9403	0.1564	1.238

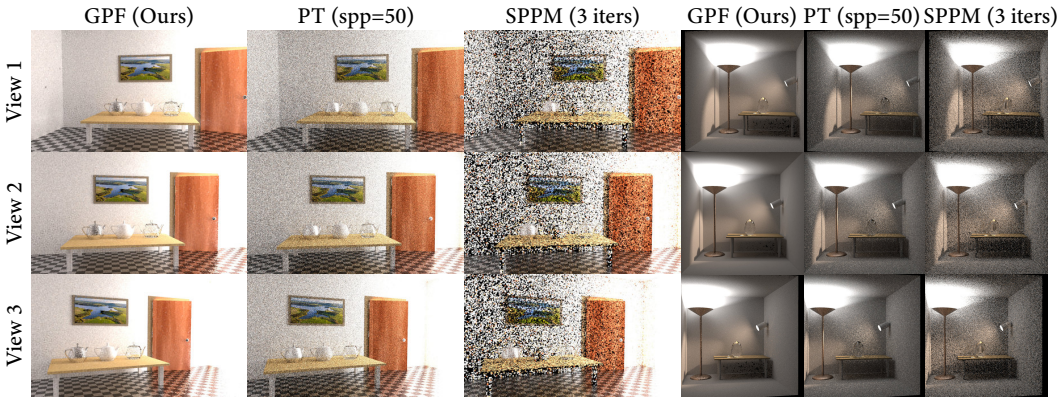


Figure S1: **Multi-view rendering results.** We compare our solution with *Mitsuba 3 Path Tracer* (Jakob et al., 2022b), *SPPM* (Hachisuka & Jensen, 2009), *3DGS* (Kerbl et al., 2023) and *Instant-NGP* (Müller et al., 2022) across water-caustic. Our method produces cleaner results with significantly fewer noise artifacts and more accurate light transport, especially in challenging caustic. Please 🔍 zoom in for more details.

Differentiation. PyTorch differentiates the selected Gaussian weights, irradiance coefficients, and pixel accumulation with respect to all continuous field parameters. The nearest-neighbor index and top-K membership are discrete. Their selected indices are held fixed for each backward pass, while subsequent forward passes use the updated field. The scene and precomputed training-camera paths are fixed; field fitting does not require differentiating ray–surface intersections.

B Ablation Study

Ablation on Different Number of Gaussian Primitives. Table S1 investigates the effect of varying the number of Gaussian primitives. Increasing the number of Gaussians consistently improves image quality, as reflected by higher PSNR/SSIM and lower LPIPS, owing to denser photon coverage and finer radiance approximation. However, beyond 100K primitives, the performance gain becomes marginal while rendering time increases notably. We therefore adopt 100K Gaussians as a balanced configuration between quality and efficiency.

C Efficiency Analysis

Rendering efficiency. GPF moves the estimation of terminal-surface illumination into a one-time field-fitting stage. Each novel view retraces camera-side transport and evaluates the frozen irradiance field, replacing repeated progressive photon gathering with local Gaussian interpolation. Table S2 reports current GPF timings on NVIDIA A100-SXM4-80GB GPUs. After one warmup view, each of the 60 trajectory cameras is rendered once; the eight-test-view means use the corresponding subset of this traversal. Timing includes camera tracing, terminal-state compaction, field evaluation, pixel accumulation, and transfer of valid-pixel RGB values to the CPU. Full-raster assembly, file output, and one-time setup are excluded. Water-Caustic 2 and Veach-Bidir average 0.63 and 1.23 seconds per view at 1024×1024, respectively.

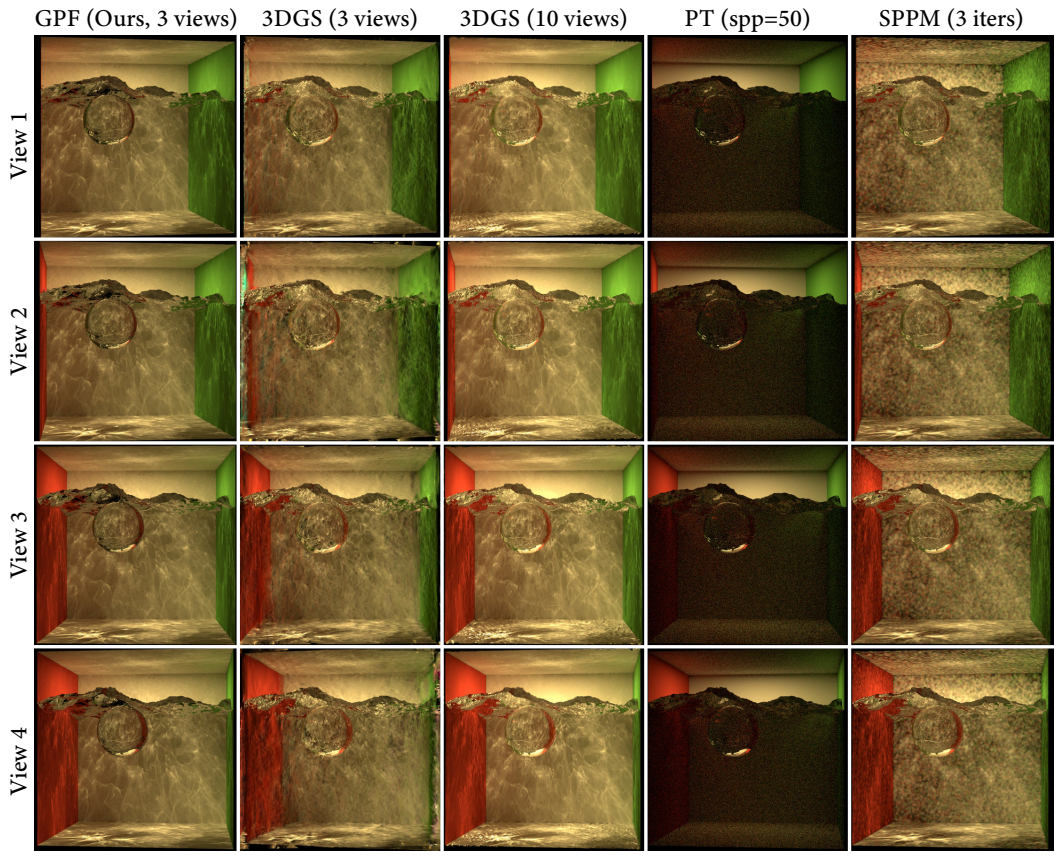


Figure S2: **Multi-view rendering results.** We compare our solution with *Mitsuba 3 Path Tracer* (Jakob et al., 2022b), *SPPM* (Hachisuka & Jensen, 2009), *3DGS* (Kerbl et al., 2023) across water-caustic. Our method produces cleaner results with significantly fewer noise artifacts and more accurate light transport, especially in challenging caustic. Please 🔍 zoom in for more details.

Table S2: **GPF rendering time (seconds per view).** Current frozen joint-optimization checkpoints, one full 60-view traversal per scene. The eight-view column is the held-out subset of that traversal. Camera and query columns are 60-view means; totals also include compaction and CPU result transfer.

Scene	Resolution	S	Camera	Query	8 views	60 views
Water-Caustic 2	1024^2	8	0.464	0.158	0.504	0.629
Water-Caustic	1024^2	8	0.759	0.155	0.848	0.921
Veach-Bidir	1024^2	16	1.009	0.219	1.075	1.231
Veach-Ajar	1280×720	16	1.811	0.596	2.355	2.419
Cornell Box	1024^2	8	0.087	0.164	0.233	0.255

Efficient field evaluation. Once fitting is complete, Gaussian centers are fixed and a static exact CUDA KD-tree is constructed for reuse across views. Camera-prefix processing, nearest-neighbor lookup, Gaussian interpolation, and accumulation operate on GPU-resident data, avoiding per-query CPU round trips. This implementation retains the same 64-candidate, 32-Gaussian query, field parameters, and camera-sampling settings. Over the Water-Caustic 2 and Veach-Bidir trajectories, query and accumulation take 0.16 and 0.22 seconds per view, while camera transport takes 0.46 and 1.01 seconds. Camera-side tracing is therefore the main remaining cost in these two scenes.

One-time preparation. Preparing GPF consists of reference generation, photon-guided initialization and training-path extraction, followed by joint optimization of the Gaussian parameters. Training

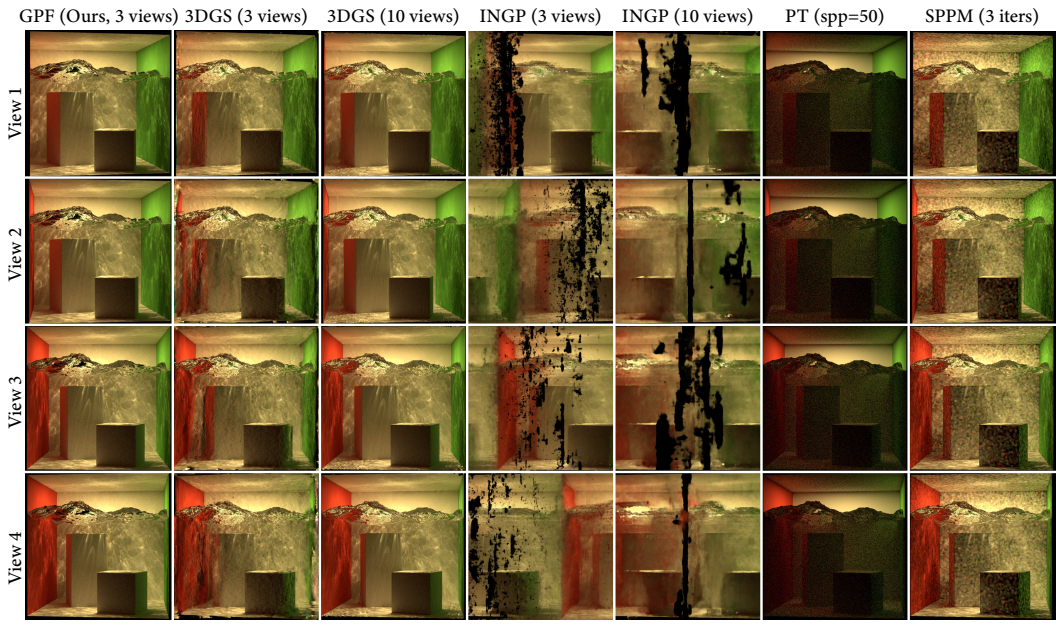


Figure S3: **Multi-view rendering results.** We compare our solution with *Mitsuba 3 Path Tracer* (Jakob et al., 2022b), *SPPM* (Hachisuka & Jensen, 2009), *3DGS* (Kerbl et al., 2023) and *Instant-NGP* (Müller et al., 2022) across water-caustic. Our method produces cleaner results with significantly fewer noise artifacts and more accurate light transport, especially in challenging caustic. Please 🔍 zoom in for more details.

Table S3: **Recorded preparation stages.** Reference compute is summed over all three training or four validation views, not the wall time of parallel jobs. Fitting records the full 30k-step schedule with validation. These measurements are individual stages, not an end-to-end offline total.

Scene	Train refs. (min)	Val. refs. (min)	Field prep. (s)	Fitting (min)	Setup (s)
Water-Caustic 2	21.5	13.9	48.8	68.4	3.0
Water-Caustic	20.5	14.3	51.4	63.7	3.1
Veach-Bidir	68.4	17.7	76.2	94.5	3.2
Veach-Ajar	35.4	24.2	71.9	161.0	3.7
Cornell Box	18.7	11.8	48.4	59.4	3.3

reuses the extracted paths, while inference traces new paths for each viewpoint. Table S3 separates the recorded stages, including the three training references and four validation references used for checkpoint selection. The fitting times cover the full 30,000-step budget with periodic validation, rather than stopping the clock at the retrospectively selected best checkpoint. Reference generation for final test scoring is an evaluation cost and is not charged to deployment.

Multi-view amortization. For N additional views of a fixed scene, GPF's cumulative cost is

$$T_{\mathrm{GPF}}(N) = T_{\mathrm{prepare}} + T_{\mathrm{setup}} + N\, t_{\mathrm{render}}, \tag{14}$$

where T_{prepare} includes training and validation reference generation, path extraction, initialization, fitting, and any scene-specific configuration selection; T_{setup} includes inference loading, index construction, and warmup. The small per-view term is the benefit of retaining one field across a camera trajectory. A per-view SPPM workflow pays for progressive estimation at every camera, whereas a reusable photon map pays its own construction and indexing costs followed by per-view gathering.

Amortization on Water-Caustic. Under the stage-based accounting specified in Table S3, the recorded one-time cost is 85.1 minutes: 20.5 minutes for three training references, 63.7 minutes for

the full 30k-step optimization, and approximately 0.9 minutes for field preparation and inference initialization, including warmup. GPF subsequently renders each view in 0.92 seconds, averaged over the 60-view trajectory, while SPPM reference generation averages 6.84 minutes over the three training views (2000 iterations with 100k emitted paths per iteration). Dividing the recorded one-time cost by the per-view time saving yields an estimated crossover at 12.47 additional views, or the 13th frame. This case study compares against repeated SPPM reference generation, not a reusable photon map or an equal-quality rendering budget.

D Additional Experiment Results

We also provide a supplementary video demonstrating the multi-view rendering consistency across all scenes.

Multi-View Rendering Results on Veach-Ajar and Veach-Bidir. We present multi-view rendering results on the *Veach-Ajar* and *Veach-Bidir* scenes from three viewpoints (Figure S1). Our method is compared with Path Tracing (50 spp) and SPPM (3 iterations). The results demonstrate that our approach produces cleaner images with reduced noise and more stable indirect illumination across views compared to both baselines.

Multi-View Rendering Results on Water-Caustic. We present multi-view rendering results on the *Water-Caustic* (Figure S2) scene using four viewpoints. We compare our method with 3DGS under both *3-view* and *10-view* supervision, as well as with Path Tracing (50 spp) and SPPM (3 iterations). The results indicate that our method achieves sharper caustic structures, reduced noise levels, and more stable multi-view radiance behavior compared to the baseline approaches.

Multi-View Rendering Results on Water-Caustic 2. We present multi-view rendering results on the *Water-Caustic 2* scene (Figure S3), using four viewpoints. We compare our method with 3DGS and Instant-NGP under both *3-view* and *10-view* supervision, as well as with Path Tracing (50 spp) and SPPM (3 iterations). The results show that our approach produces cleaner caustics, fewer noise artifacts, and more consistent light transport across views compared to all baselines.

Reusable Photon Map Control. SPPM accuracy comes from progressively shrinking per-hit-point radii while injecting fresh photons each pass, and these statistics are anchored to camera hit points. This is what prevents SPPM results from being reusable across views. To the best of our knowledge, no standard estimator exists for a pooled, camera-independent map at this budget, partly because pooling is memory-prohibitive. In the following experiment, we construct such a control solution, i.e., comparing GPF with a photon map reused across viewpoints and reporting the results on *Water-Caustic 2* at 256×256 resolution over 57 test views. The reusable map pools the photon deposits from 1,000 rounds of 100,000 emitted paths each, for a total of 100 million emitted paths. This produces approximately 206 million surface-photon records occupying 9.90 GB. The same stored map is queried across all test views without emitting new photons.

Table S4 reports the rendering quality for this control. GPF achieves higher PSNR and SSIM than the pooled photon map in this setting, showing that reusing this photon map does not by itself match the reconstruction quality of the learned field.

Table S4: **Reusable photon map control on Water-Caustic 2.** Evaluation uses 57 test views at 256×256 resolution.

Method	PSNR ↑	SSIM ↑
GPF	29.23	0.9675
Reusable photon map	23.16	0.7920

E Limitations and Future Work

GPF targets the offline, multi-view physically-based rendering setting and inherits the standard assumptions of this paradigm: scene geometry, materials, and light sources are provided in a path-tracing-compatible representation (*e.g.*, a Mitsuba 3 scene file), which makes our approach

complementary, rather than directly comparable, to image-only neural reconstruction methods such as NeRF or 3DGS. GPF amortizes a one-time reference-generation and field-fitting cost across many views, making repeated rendering of a static scene its target workload. Appendix C separates the measured rendering and preparation stages and defines the cumulative-cost accounting. We focus on static scenes in this work, and extending GPF to dynamic geometry, animated lighting, and participating media is a natural next step that is well aligned with the differentiability of the field. Finally, although we adopt anisotropic Gaussian primitives for their compactness and well-understood differentiable behavior, the radiance-distillation framework is representation-agnostic, and exploring alternative continuous parameterizations remains an interesting direction for future work.