

Multi-feature Radiance Baking Neural Networks for Instant Volumetric Rendering

JIAMING LIANG, Peking University, China
HONGLIANG YUAN, XiaoMi company, China
MENG GAI, Peking University, China
GUOPING WANG, Peking University, China
SHENG LI*, Peking University, China

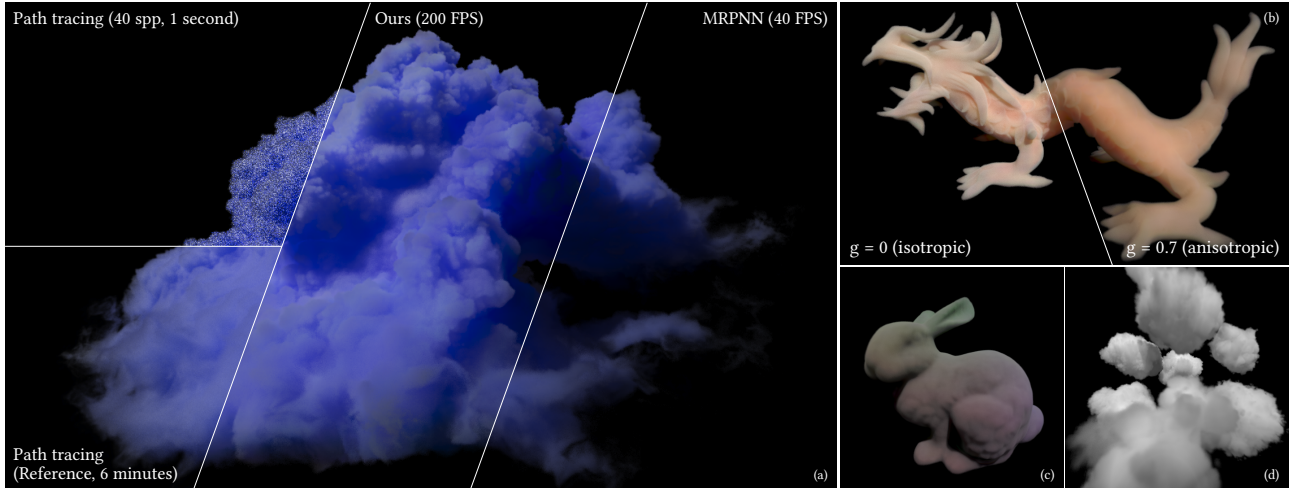


Fig. 1. MRBNN (Ours) renders complex multi-scattering effects of dense volumes in 200 FPS for 1024×1024 resolution with three-channel albedo, reaching $5\times$ speedup over the state-of-the-art (MRPN) [Hu et al. 2023] while preserving high visual quality (a) thanks to our compact feature sampling and network design. Moreover, our method supports not only modification of globally homogeneous albedo and phase (b), but also rendering volumes with spatially varying albedo (c) and heterogeneous phase parameters (d), which are beyond the capabilities of prior work.

Photorealistic volume rendering suffers from slow convergence with traditional path tracing algorithms and poses significant challenges for real-time applications. In this paper, we present Multi-feature Radiance Baking Neural Networks (MRBNN), a neural volumetric rendering method that achieves real-time performance by leveraging analytic decomposition of scattering with an efficient learned representation. Our method reformulates in-scattering integral in Volumetric Radiance Transfer Equation to diagonal scaling operator in spherical harmonics spectrum. Building on this formulation, we propose a compact factorized representation with low-rank compression in latent space that replaces costly integration. We then introduce efficient dual-pattern feature sampling and a lightweight neural decoder for fast radiance prediction. With these dedicated designs, MRBNN

achieves $4 \sim 5\times$ speedup over the state-of-the-art while maintaining high visual fidelity. Moreover, our method addresses key limitations of prior work, including over-illumination artifacts and inability to handle volumes with spatially varying albedo or heterogeneous phase parameters. Extensive experiments demonstrate that our neural baking method can synthesize photorealistic images for complex volumes in a few milliseconds.

CCS Concepts: • **Computing methodologies** → **Ray tracing; Neural networks.**

Additional Key Words and Phrases: Neural rendering, Volume rendering, Participating media, Real-time rendering

ACM Reference Format:

Jiaming Liang, Hongliang Yuan, Meng Gai, Guoping Wang, and Sheng Li. 2026. Multi-feature Radiance Baking Neural Networks for Instant Volumetric Rendering. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '26)*, July 19–23, 2026, Los Angeles, CA, USA. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3799902.3811179>

*Sheng Li is the corresponding author.

Authors' Contact Information: Jiaming Liang, Peking University, Beijing, China, liangjiaming@stu.pku.edu.cn; Hongliang Yuan, XiaoMi company, Beijing, China, 11488336@qq.com; Meng Gai, Peking University, Beijing, China, gaimeng@pku.edu.cn; Guoping Wang, Peking University, Beijing, China, wgp@pku.edu.cn; Sheng Li, Peking University, Beijing, China, lisheng@pku.edu.cn.



This work is licensed under a Creative Commons Attribution 4.0 International License. *SIGGRAPH Conference Papers '26, Los Angeles, CA, USA*
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2554-8/26/07
<https://doi.org/10.1145/3799902.3811179>

1 Introduction

Volumetric rendering has been thoroughly investigated to depict unique properties of non-surface materials such as clouds, fogs, and jade. Although certain materials like skin can be approximated

using subsurface scattering [Hanrahan and Krueger 1993], it is inadequate to model correct interaction of ray and particles in general participating media. Since Kajiya and Von Herzen [1984] introduced the Volumetric Radiance Transport Equation (VRTE) to computer graphics, path tracing with Monte Carlo integration method has been widely adopted. The estimation process is accelerated by generalizing traditional methods to volumetric scattering like better tracing method (e.g. bidirectional path tracing [Lafortune and Willems 1996]), more efficient sampling for path construction and transmittance estimation (e.g. volumetric scattering path guiding [Xu et al. 2024]), progressive null-tracking [Misso et al. 2023]), more sophisticated sample-reuse techniques (e.g. volumetric ReSTIR [Lin et al. 2021]) and so on. VRTE is also further derived into other practical forms [Miller et al. 2019]. However, complex volumetric path tracing is still too costly for real-time applications. Particularly, some media, notably clouds, have high single-scattering albedo, leading to visually perceptible radiance contribution even after hundreds of bounces in a single volume. This makes practical post-processing like real-time denoising with low samples per pixel (spp) [Chaitanya et al. 2017] inapplicable, as the input is too far from the converged result in some circumstances.

Recently, neural methods have been applied to volumetric rendering. Radiance-Predicting Neural Networks (RPNN) [Kallweit et al. 2017] opened up a new paradigm to estimate multi-scattering effects, achieving speedup of up to three orders of magnitude over CPU-based path tracing. However, RPNN still requires several minutes to produce a converged result. Multi-feature RPNN (MRPNN) [Hu et al. 2023] improves the algorithm further, reaching up to 40 FPS on an NVIDIA RTX 2080 GPU at 1024^2 resolution. MRPNN also supports globally configurable albedo and phase function parameters.

Despite these advances, MRPNN still has some key limitations. First, its performance remains insufficient in complex real-time applications for consuming tens of milliseconds in a single pass. Second, it assumes fragmented volumes to be continuous low-density volumes and thus systematically overestimates global illumination for such scenes, as shown in Fig. 2. Finally, it requires consistent albedo and phase function parameters throughout the scene, which significantly restricts its ability to represent more complex media.

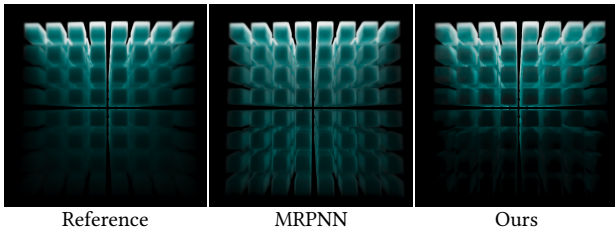


Fig. 2. Fragmented-volume scenario as a failure case for MRPNN [Hu et al. 2023]. Our latent-space baking operates without low-density volume assumption and thus alleviates over-illumination artifact.

To address these challenges, we propose *Multi-feature Radiance Baking Neural Networks* (MRBNN) that exploits the structure of volumetric scattering. Inspired by Neural Radiosity [Hadadan et al. 2021], we observe that evaluating the integral is more accurate but computationally expensive than direct prediction for in-scattering

field. By expanding the phase function in Legendre polynomials, we show that the in-scattering integral forms a diagonal scaling operator in spherical harmonics basis, yielding an equivalent reweighting that eliminates integration. While such spectral operator form is analytically compact, its coefficients remain high-dimensional field that is infeasible to store or evaluate explicitly under dynamic conditions. To bridge this gap, we introduce a compact latent-space radiance baking that provides a finite-rank, factorized parameterization with practical memory cost. After querying features with our dual-pattern sampling strategy, we employ a lightweight neural decoder for interpreting the learned representation to scattering radiance with substantially low latency. As shown in Fig. 1, MRBNN achieves significant acceleration over prior work while maintaining high visual fidelity. It also supports challenging configurations such as spatially varying albedo and heterogeneous phase parameters that cannot be handled previously.

In summary, our contributions include the following aspects:

- We reformulate in-scattering integral in VRTE as a diagonal scaling operator in the spherical harmonics spectrum, which eliminates explicit spherical integration in theory.
- We propose a compact latent-space neural baking that learns a low-rank, factorized approximation of such scattering operator with practical space complexity, which can be decoded by our dual-pattern feature query and lightweight network.
- We design a real-time volumetric renderer that achieves $4 \sim 5\times$ speedup over the state-of-the-art while supporting spatially varying albedo, heterogeneous phase functions and improved temporal stability.

2 Related Work

Stochastic Volumetric Ray Tracing. Photorealistic rendering of participating media typically relies on stochastic approaches like Monte Carlo integration method to solve VRTE numerically. Sampling inside volumes has drawn researchers’ attention as a distinct problem. To avoid explicit transmittance estimation, null-collision methods apply delta tracking [Woodcock et al. 1965] to sample the next position; the resulting path-space PDF was analytically formulated by Miller et al. [2019]. In other sampling methods that require transmittance calculation, Novák et al. [2018] provided a comprehensive survey of related estimation methods. More recently, power-series expansion [Georgiev et al. 2019] and unbiased ray marching [Ketunen et al. 2021] have been proposed to introduce a range of novel transmittance estimators and reduce estimation variance further.

Alternatively, volumetric path tracing can adapt techniques originally developed for surface rendering. For instance, bidirectional path tracing [Lafortune and Willems 1996] samples from both camera and light, and Metropolis methods [Pauly et al. 2000] focus on sampling high-contribution paths through mutation and acceptance. More recent advances include path guiding that performs product sampling over different integrand components [Herholz et al. 2019; Xu et al. 2024] or uses neural network to fit target distribution [Dong et al. 2023], reservoir resampling [Lin et al. 2021], and extension to kernel-based volumetric primitives [Condor et al. 2025].

Though capable of producing photorealistic images, complex volumetric path tracing is generally unable to converge to a satisfying

result in real time, even with advanced sampling or post-processing techniques. Our goal is to accelerate this process by leveraging light-weight network that predicts multi-scattering radiance directly.

Neural Rendering. Deep learning has been actively explored to model neural global illumination for its powerful approximation capability. Representative approaches include mesh-based rendering with homogeneous translucent materials using DiffusionNet and BSSRDF [An et al. 2025; Sharp et al. 2022], vertex-based feature learning with refinement [Su et al. 2025], object-oriented representations for modeling object-environment interactions [Zheng et al. 2024], and transformer-based architectures for end-to-end image synthesis [Zeng et al. 2025]. However, these methods are designed for surface-based light transport and rely on the surface RTE, making them ill-suited for real-time volumetric rendering governed by VRTE. Extending them to participating media either breaks their core assumptions (e.g. surface vertices and normals) or leads to unpromising results and cost.

Particularly, neural feature fields have played a key role in graphics in recent years, ranging from 3D reconstruction (e.g. neural radiance fields [Mildenhall et al. 2021]) to rendering acceleration (e.g. neural radiance cache [Müller et al. 2021], neural radiosity [Hadadan et al. 2021; Su et al. 2024]). An important representation for such field is feature grid, where features are spatially discretized and interpolated when used. To minimize memory usage, various strategies including octree [Takikawa et al. 2021], hashing [Müller et al. 2022] and k-planes decomposition [Cao and Johnson 2023] were proposed. In this paper, we also utilize feature grid as a baking approach to handle dynamic material and light conditions.

Precomputed Radiance Transfer (PRT). PRT [Sloan et al. 2002] approximates indirect illumination by computing transfer matrix under angular basis, notably spherical harmonics basis. Such representation allows the rendering equation to be evaluated as simple dot products at runtime, and spherical harmonics also enables handling of dynamic transport [Pan et al. 2007; Sloan et al. 2005]. Recent advances include using neural network to substitute transport in PRT [Rainer et al. 2022]. Building on this line of work, we exploit the zonal convolution properties of spherical harmonics to construct a latent-space feature grid for dynamic lighting modeling.

RPNN Series. Among existing methods for photorealistic volumetric rendering, the RPNN family stands out for its impressive acceleration. Kallweit et al. [2017] pioneered this line by predicting indirect illumination through layered density field to avoid explicit simulation of multi-bounce scattering. This process was further improved by Panin and Nikolenko [2019] through caching intermediate features that achieved $2 \sim 3\times$ speedup. Hu et al. [2023] later identified that direct mapping from density field to radiance field is overly complex, leading to a large model with millions of parameters and minutes of inference time. To address this problem, MRPNN introduces auxiliary hints on albedo, transmittance field and cumulative phase, substantially reducing model scale and enabling real-time performance with globally configurable albedo and phase. MRPNN also improves shadow boundaries prediction through additional high-frequency feature sampling.

Despite its compelling optimization, MRPNN still exhibits several limitations that hinder efficient and robust prediction. First, to capture surrounding volumetric context, MRPNN relies on multiple complex sampling strategies, leading to numerous irregular texture lookups and remaining artifacts such as over-illumination in fragmented or challenging scenes. Second, MRPNN does not model spatially varying albedo or heterogeneous phase configurations, and thus cannot handle volumes with scattering coefficient grids or multiple phase parameters. Finally, its network architecture remains relatively complex and leaves room for further simplification to better exploit GPU computational efficiency. In this paper, we introduce a tiny network with compact latent-space baking that targets to address these problems.

3 Neural Baking

We adopt a common and effective strategy for volumetric rendering by approximating indirect illumination beyond the first scattering event. As illustrated in Fig. 3, we first draw volumetric samples along the camera ray using stochastic sampling techniques (see Sec. 2). For each sampled point, we compute direct illumination explicitly and predict high-order scattering effects via our model by approximating the in-scattering radiance. The contributions from all samples are then accumulated to estimate the radiance received by the camera. Formally, the whole process can be summarized as:

$$\hat{L}(p, \omega) = \frac{1}{N} \sum_{i=1}^N \frac{T_r(p'_i \rightarrow p)}{f(p'_i)} \left(L_{\text{direct}}(p'_i, \omega) + S(\mathcal{F}(p'_i, \omega)) \right) \quad (1)$$

where $T_r(p_1 \rightarrow p_2)$ is transmittance from p_1 to p_2 , p'_i is position sampled from distribution f , starting from camera position p and view direction ω ; $\mathcal{F}$ denotes multi-scale feature extraction from the baking, and S denotes in-scattering prediction networks. Besides, as a basic assumption in RPNN series, we establish sampling pattern and predict contribution from a directional light. For ambient light environment, light direction can be drawn randomly to fulfill such assumption. Additionally, we emphasize that as a baking method, our latent-space representation needs a per-volume tuning to find suitable solution, while MRPNN operates as a general mapping with input extracted from volume directly. This is a trade-off in our method to achieve substantial acceleration.

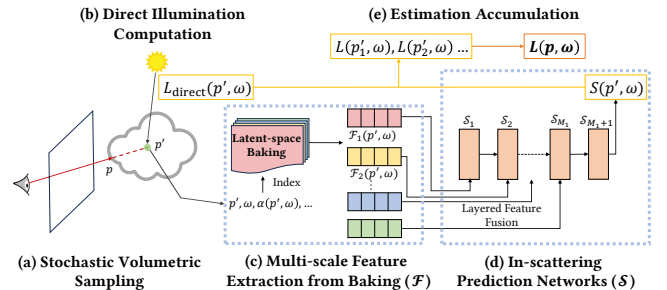


Fig. 3. Rendering pipeline overview.

In Sec. 3, we first briefly review VRTE and present theoretical derivation to express in-scattering integral in spherical harmonics basis using Legendre polynomials. Next, we will illustrate how such

formulation is practically integrated in our feature baking. In Sec. 4, we will propose sampling patterns to extract features from latent space and a lightweight network for decoding input to radiance.

3.1 In-scattering Integral Transformation

VRTE expresses the radiance $L(p, \omega)$ at position p and direction ω in the form of an integro-differential equation. By solving ordinary differential equation, VRTE can be written in the classic form below:

$$\begin{aligned} L_s(p, \omega) &= \sigma_a(p, \omega)L_e(p, \omega) + \sigma_s(p, \omega) \int_{S^2} \phi(p, \omega_i, \omega) L(p, \omega_i) d\omega_i, \\ L(p, \omega) &= \int_0^\infty T_r(p' \rightarrow p) L_s(p', \omega) dt \end{aligned} \quad (2)$$

where $\sigma_a(p, \omega)$ and $\sigma_s(p, \omega)$ are absorption coefficient and scattering coefficient respectively, and $\phi(p, \omega_1, \omega_2)$ is phase function between direction ω_1 and ω_2 at position p . Specifically, this equation can be interpreted as follows:

- $L_s(p, \omega)$ denotes radiance contribution from point p along direction ω , which contains self-emission (i.e. $L_e(p, \omega)$) and in-scattering from surrounding directions (i.e. the spherical integral). Phase function models radiance ratio scattered from one direction to another.
- To get $L(p, \omega)$, all radiance contribution at point p' along direction ω will be accumulated, weighted by transmittance to represent decay when travelling through media.

And finally let $\sigma_t(p, \omega) = \sigma_a(p, \omega) + \sigma_s(p, \omega)$ be extinction coefficient (or density), albedo $\alpha(p, \omega)$ is defined as $\sigma_s(p, \omega)/\sigma_t(p, \omega)$ and transmittance is formulated as $T_r(p_1 \rightarrow p_2) = e^{-\int_0^d \sigma_t(p, \omega) dt}$ where $p = p_1 + \omega t$, $d = \|p_1 - p_2\|_2$. For notation brevity, the in-scattering integral $\int_{S^2} \phi(p, \omega_i, \omega) L(p, \omega_i) d\omega_i$ is denoted by $S(p, \omega)$ in following formulas.

In practice, phase function $\phi(\omega_1, \omega_2)$ is rotationally symmetric and therefore a function of $\cos \theta = \omega_1 \cdot \omega_2$, even for the most complex form Lorenz-Mie. Thus essentially, the in-scattering integral in VRTE naturally implies an angular operator $\mathcal{K}$:

$$(\mathcal{K}h)(p, \omega) := \int_{S^2} \phi(p, (\omega_i \cdot \omega)) h(p, \omega_i) d\omega_i, \quad (3)$$

Applying such operator on the angular radiance distribution $L(p, \cdot)$ results in the in-scattering field $S(p, \cdot)$, i.e. $S = \mathcal{K}L$. Our primary goal is to estimate $S(p, \omega)$ so there are two approaches:

- (1) Direct: decompose and estimate $S(p, \omega)$ directly.
- (2) Indirect: decompose and estimate $L(p, \omega)$, then apply $\mathcal{K}$ to estimate $S(p, \omega)$ by evaluating integral.

Though case (2) is more computationally expensive, it is likely to yield less error as similar conclusion in surface rendering is experimentally verified by Neural Radiosity [Hadadan et al. 2021]. **However, we prove that $\mathcal{K}$ forms a diagonal scaling operator analytically under spherical harmonics (SH) spectrum, which allows it to be evaluated without explicit integration.** Formally:

THEOREM 3.1. *Let radiance field SH decomposition be $L(p, \omega) = \sum_{l=0}^\infty \sum_{m=-l}^l L_l^m(p) Y_l^m(\omega)$, the in-scattering field can be expressed*

as:

$$S(p, \omega) = \sum_{l=0}^\infty \eta_l(p) \sum_{m=-l}^l L_l^m(p) Y_l^m(\omega)$$

where $\eta_l(p) = \frac{4\pi f_l(p)}{2l+1}$ and $f_l(p)$ is the l -th order coefficients of Legendre polynomials P_l decomposed from phase function ϕ at position p , and Y_l^m is the SH of degree l and order m .

By expanding SH components in two matrices, $\mathcal{K}$ is essentially a diagonal matrix $\text{diag}(\eta_0, \eta_1, \dots)$ (see Supplementary Sec. D.1 for details). When the whole volume shares the same phase parameter as MRPNN assumes, this matrix is globally consistent.

PROOF. Substituting the SH expansion of $L(p, \omega)$ and the Legendre expansion of ϕ , then applying the addition theorem of SH to P_l , $S(p, \omega)$ is thus equal to:

$$\int_{S^2} \left(\sum_{l=0}^\infty \frac{4\pi f_l(p)}{2l+1} \sum_{m=-l}^l Y_l^m(\omega) \bar{Y}_l^m(\omega_i) \right) \cdot \left(\sum_{l', m'} L_{l'}^{m'}(p) Y_{l'}^{m'}(\omega_i) \right) d\omega_i \quad (4)$$

Utilizing the orthogonality property $\int_{S^2} Y_{l'}^{m'}(\omega_i) \bar{Y}_l^m(\omega_i) d\omega_i = \delta_{ll'} \delta_{mm'}$, the integral is reduced to:

$$S(p, \omega) = \sum_{l, m} \frac{4\pi f_l(p)}{2l+1} L_l^m(p) Y_l^m(\omega) \quad (5)$$

Identifying $\eta_l(p) = \frac{4\pi f_l(p)}{2l+1}$ yields the theorem. $\square$

Detailed derivation, including illustration of summation reordering, is provided in Supplementary Sec. D.2 and D.3. In particular, Henyey-Greenstein phase function (HG) has an elegant decomposition form [Sansone 1959] as stated in Lemma 3.2.

LEMMA 3.2. *Given $\phi(g, \omega_1, \omega_2) = \frac{1}{4\pi} \frac{1-g^2}{(1+g^2-2g(\omega_1 \cdot \omega_2))^{3/2}}$, $\eta_l = g^l$.*

It's also worth noting that decomposition of Lorenz-Mie function, as the most complex phase function practically used in rendering, is already derived by physicists [Fowler 1983].

3.2 Operator Approximation via Latent-space Baking

Theorem 3.1 reveals the fact that the in-scattering operator $\mathcal{K}$ essentially acts as a diagonal scaling in SH space, which is an ideal estimator that eliminates explicit integration in $L \rightarrow S$. However, such formulation assumes full knowledge of L decomposition, which is infeasible under dynamic lighting or materials. Therefore, we reinterpret neural baking as an approximation of $\mathcal{K}$ applied to a compressed latent representation of L .

For simplicity, now we might as well assume that only light direction ω_s varies, so both the total radiance field and in-scattering field become functions of ω_s :

$$\begin{aligned} L(p, \omega) &= \sum_{l=0}^\infty \sum_{m=-l}^l L_l^m(p, \omega_s) Y_l^m(\omega) \\ S(p, \omega) &= \sum_{l=0}^\infty \eta_l(p) \sum_{m=-l}^l L_l^m(p, \omega_s) Y_l^m(\omega) \end{aligned} \quad (6)$$

We align view and light direction (i.e. $\omega = \omega_s$) first and introduce factorized encoding $\mathcal{L}_l^m(p)Y_l^m(\omega_s)$ to express $L_l^m(p, \omega_s)Y_l^m(\omega_s)$ in latent space, which results in $\tilde{L}(p, \omega) = \sum_{l=0}^{\infty} \sum_{m=-l}^l \mathcal{L}_l^m(p)Y_l^m(\omega_s)$ as the latent-space representation of $L(p, \omega)$. Applying $\mathcal{K}$ with low-rank compression, we assume $\tilde{L}$ can be transformed into scattering field $\tilde{S}$ in latent space:

$$\tilde{S}(p, \omega) = \sum_{l=0}^K \eta_l(p) \sum_{m=-l}^l \mathcal{L}_l^m(p)Y_l^m(\omega_s) \quad (7)$$

where K denotes truncation rank in SH.

This expression allows us to convey effects of $\mathcal{K}$ by weighted sum of 3D feature grids $\mathcal{L}_l^m$, which avoids both explicit integration and $O(N^5)$ space complexity before factorization. To account for view directions ω that are distinct from ω_s , we introduce *view module* $\mathcal{V}$, another 2D feature grid indexed by ω , as condition in decoding. The neural networks are expected to decode $\tilde{S}(p, \omega)$ with conditions to physical in-scattering radiance, or formally,

$$\mathcal{S} : \left(\sum_{l=0}^K \eta_l(p) \sum_{m=-l}^l \mathcal{L}_l^m(p)Y_l^m(\omega_s), \mathcal{V}(\omega) \right) \rightarrow \sum_{l=0}^{\infty} \eta_l(p) \sum_{m=-l}^l L_l^m(p, \omega_s)Y_l^m(\omega) \quad (8)$$

Similarly, we also add corresponding modules for other parameters as network input (i.e. *phase module* $\mathcal{P}$ and *albedo module* $\mathcal{A}$); and to further eliminate effects of aligning ω and ω_s , both modules will condition on $\cos \theta = \omega \cdot \omega_s$ as a new dimension. Specifically, in configuration with HG function, we introduce the phase grid $\mathcal{G}$ indexed by $(g, \cos \theta)$.

4 Latent-space Field Decoding

4.1 Sampling Pattern Design

In order to capture both low-frequency and high-frequency details in the baking, we use dual-pattern sampling as displayed in Fig. 4. All feature grids adopt a multi-resolution representation as different levels of details (LoD) to model scattering environment across varying spatial scopes. Normally, given a position p , features are sampled at p exactly in different LoDs as local sampling. To depict shadow boundary more accurately, some features are also sampled towards light from p by cone tracing. Although our sampling strategy resembles two-part approach in MRPNN, our implicit baking enables a much simpler workload for avoiding extensive irregular texel access.

Naïvely, radiance field feature $\mathcal{L}_l^m$ could be sampled both locally and towards light directly. However, as a single sample of SH grid requires $(K+1)^2$ grids to be sampled, such design will amplify feature extraction cost. Instead, we introduce another 3D grid called base grid $\mathcal{B}$ indexed by position solely. Besides, a mipmapped transmittance map $\mathcal{T}$ with respect to incoming light direction is also introduced as input to assist learning. $\mathcal{B}$ and $\mathcal{T}$, instead of $(K+1)^2$ grid components $\mathcal{L}_l^m$, adopt towards-light sampling. Our experiments in Sec. 6 show that this does not necessarily affect the final quality, while it contributes to additional 10% performance gain.

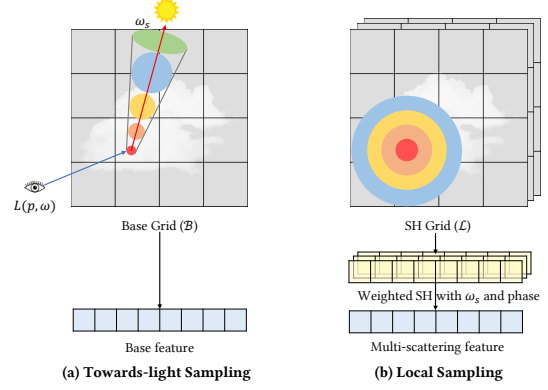


Fig. 4. Dual-pattern sampling illustration, projected on 2D plane. $\mathcal{B}$ and $\mathcal{T}$ use (a) towards-light sampling like cone tracing, which increases step size exponentially as well as raising feature grid level; other features use (b) local sampling, which queries only at shading point across levels of feature grid. Circles with different colors represent features extraction in different LoDs, which will be input into network gradually later.

4.2 Feature Extraction and Network Architecture

To progressively describe the scattering environment, features are extracted level by level across different LoDs and concatenated. For instance, for volumes with HG phase function, feature extraction at i -th level $\mathcal{F}_i$ is just concatenation of the following features:

$$\mathcal{F}_i(p, \omega) = \left(\mathcal{B}_i(p_i), \sum_{l=0}^K g(p)^l \sum_{m=-l}^l \mathcal{L}_{i,l}^m(p)Y_l^m(\omega_s), \mathcal{V}_i(\omega), \mathcal{G}_i(g(p), \cos \theta), \mathcal{A}_i(\alpha(p, \omega), \cos \theta), \left[\mathcal{T}_i(p_j, \omega_s) \right]_{j=1}^{M_2} \right) \quad (9)$$

where $i \in \{1, \dots, M_1\}$ and M_1 means both grid levels and total steps towards light to sample $\mathcal{B}$, so $\mathcal{B}_i$ samples at the i -th towards-light point p_i ; M_2 denotes steps to sample $\mathcal{T}$. Since $\mathcal{T}$ does not lie in latent space, each level $\mathcal{T}_i$ samples at all towards-light points p_j and concatenated as vector of M_2 dimensions to depict the environment.

To fuse multi-scale features to capture both local and global scattering effects, we employ hierarchical residual blocks, where each level additionally incorporates features from the next scale. Formally, let $\mathcal{F}_i(p, \omega) \in \mathbb{R}^D$ denote the feature at level i , the fusing process is:

$$\begin{aligned} \overline{O}_i(p, \omega) &= O_{i-1}(p, \omega) + \mathcal{F}_i(p, \omega), \\ O_i(p, \omega) &= \mathcal{S}_i(\overline{O}_i(p, \omega)) + \overline{O}_i(p, \omega), \end{aligned} \quad (10)$$

where O_i is the output latent feature, $\mathcal{S}_i$ is a multi-layer perceptron (MLP) with input and output both in $\mathbb{R}^D$. The initial latent feature O_0 is set to zero. After aggregating all features, a final MLP $\mathcal{S}_{M_1+1}$ decodes the latent representation O_{M_1} into radiance. An illustration of the network architecture is provided in the Supplementary (Sec. B).

5 Implementation

5.1 Feature Configuration

Practically, we choose $M_1 = M_2 = 4$ for Eq. 9 to reuse position samples. The feature grids are sampled within $[0, 1]^N$ in reality, so directions and positions are represented by normalized spherical coordinates and relative coordinate in volume bounding box respectively. $\cos \theta \in [-1, 1]$ is mapped by $(x+1)/2$. Considering that media with higher albedo scatter rays with increased frequency, finer grid resolution is allocated by a non-linear mapping $(e^x - 1)/(e - 1)$. And for volume used in production, scattering and extinction coefficients are stored as scalar voxels that are not affected by direction, therefore we omit ω in albedo. Finally, the transmittance map stores optical depth normalized by maximum density. Detailed grid configurations and error analysis on low-order compression are given in the Supplementary Sec. A.

5.2 Network Implementation

In practice, we use tiny-cuda-nn [Müller 2021] with PyTorch [Paszke et al. 2019] to implement the network prototype and do training for convenient network adjustment. On completion of the training phase, we export the parameters to perform inference with tailored C++ code. We implement our own feature grid module to fuse sampling within the rendering procedures for avoiding possible computation redundancy. Furthermore, to maximize GPU utilization, we use fully fused MLP [Müller et al. 2021] with just-in-time compilation to execute neural network in a single kernel. Practically, it is roughly sufficient to equip each MLP with two 64-neuron fully-connected layers. We also find it beneficial to use ReLU6 as the final layer activation in MLPs for numeric stability.

5.3 Training

Ground Truth Generation. For usual cases, we use the path-tracing renderer provided by MRPNN implementation to ensure a fair comparison. Mitsuba3 Renderer [Jakob et al. 2022] is used for scenes that cannot be handled by MRPNN. Normally, 9k images of resolution 512^2 with random configurations are used as ground truth. Each pixel contains sampled position p , direction ω and radiance $S(p, \omega)$ generated by path tracing with 1024 spp and 1024 maximum bounce depth. However, for hard scenes that suffer from high variance, increased spp and depth can be used to generate better result (Sec. 6.3).

Dataset Preference. To improve robustness in challenging configurations, we use a non-uniform sampling strategy over the parameter space to prefer high albedo and strongly anisotropic phase function. We also augment the training set in which only phase parameters are altered while albedo is fixed with constant near 1. Practically, the dataset is augmented in a 4:1 ratio.

Training Settings. We use Adam optimizer [Kingma and Ba 2017] with an initial learning rate of $1e-3$, exponentially decayed by 0.8 every epoch. All scenes are trained for 10 epochs using MSE as loss function. Moreover, considering that radiance values in volume rendering can span several orders of magnitude and thus harm the training process, we compress ground truth S by $\log(x + 1)$.

6 Results and Analysis

We train our model on an NVIDIA RTX 4090 GPU while testing the performance on an NVIDIA RTX 4080 SUPER GPU. Training completes within four hours in each scene. Our renderer uses delta tracking to sample the first bounce to cancel $T_r(p'_i \rightarrow p)/f(p'_i)$ in Eq. 1, which is aligned with implementation in MRPNN. All results for MRPNN are produced by authors' open-source implementation.

6.1 Rendering Result and Performance Comparison

Comparison with MRPNN. We first demonstrate our performance improvement quantitatively in Tab. 1. Compared with MRPNN, our sampling strategy yields $\sim 1.6\times$ speedup, and our network inference achieves above $20\times$ speedup by eliminating complex module designs and using five times fewer matrices. Combined, these optimizations result in $4 \sim 5\times$ acceleration for the entire render loop. It is worth noting that the performance boost in inference does not mainly attribute to partitioned WMMA used in tiny-cuda-nn, as MRPNN implementation also has corresponding option to enable such feature while no performance gain is observed. This makes it rather fair to illustrate the importance of our simplification on network structure.

We further compare our method with MRPNN under the same time budgets (30 FPS, as a common baseline for real time). As shown in Fig. 5, our method exhibits comparable numerical accuracy to MRPNN while producing less noisy results, since we accumulate more samples but each sample is more biased. Fig. 11 shows comparison of FLIP error map [Andersson et al. 2020] in more scenes under different light conditions and equal spp. Our method is competitive with MRPNN in visual quality while achieving a substantial speedup. See our supplementary video for performance in different scenarios.

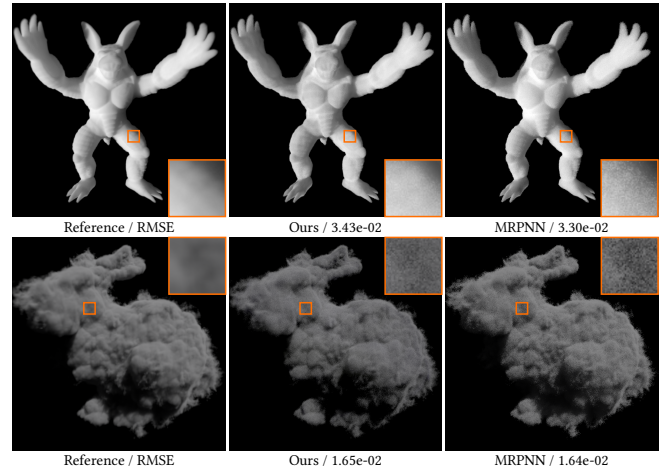


Fig. 5. RMSE and zoom-in comparison under equal-time rendering (armadillo, bunny_cloud). Error is calculated using the original radiance while images are tone-mapped by gamma correction.

Direct decomposition versus indirect decomposition. We propose two approaches to decompose the in-scattering field in Sec. 3.1 and empirically point out that indirect decomposition is better. To

Table 1. Performance measurement breakdown on *cloud0* scene with 1024^2 resolution. "single" means components of albedo RGB are the same, while "full" means three channels have different values. In the former case, MRPNN can skip albedo stage for the other two components and our method can also reduce MLP passes. "Post-process" includes a simple denoising method and drawcall to render buffer on screen. The result shows the effectiveness of our design of compact sampling and network architecture. We also show performance loss when using SH for base grid (i.e. sampling SH grid towards light additionally).

	Delta tracking	Sampling / Accessed elements	MLP inference	Post-process	Final FPS
MRPNN (single)	1.09	4.12 / 766	13.9	< 0.500	51.1
MRPNN (full)			19.0		40.5
Ours (single)		2.55 / 432	0.653		207 (4.05×)
Ours (full)			0.753		203 (5.01×)
SH for base grid (full)		3.12 / 688			182

examine our claim experimentally, we compare two methods in various scenes with respective test sets. Each dataset contains 1k images with random configurations. Fig. 6 demonstrates that in the majority of test scenes, indirect decomposition is either on par with or apparently better compared with direct decomposition.

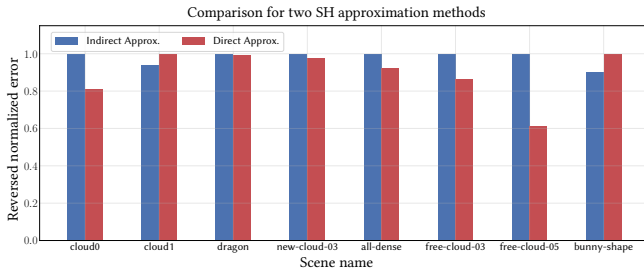


Fig. 6. Comparison for indirect decomposition (blue) and direct decomposition (red), **higher is better**. To normalize different error scales, Y axis is calculated by $1 - (e - l)/l$, where l is the lower error and e is the error. This function maps the lower error to 1.0 and the higher error to ≤ 1.0 .

Complex configurations. Our method is also capable of processing scenes with more complex configurations that cannot be handled by MRPNN. In Fig. 7, we show rendering of scenes with spatially varying albedo and phase function parameters. We also make a tricky volume that has different phase functions (HG and HG-Draine [Jendresie and d'Eon 2023]) in its different parts. These results confirm that our latent-space representation can correctly bake related information. The detailed information and experiments on HG-Draine are given in the Supplementary Sec. C.1 and D.5.

Additionally, our method achieves more stable global illumination under ambient lighting. For simple configurations, this can be directly achieved by stochastically drawing light direction and perform temporal accumulation as implemented in MRPNN. However, as shown in Fig. 9, such strategy incurs severe flickering for high-frequency ambient light and needs seconds to converge. Most importantly, any scene modification, notably camera motion or light direction changes for time-of-day effects, triggers recurrence of this problem. This artifact cannot be resolved by denoising and thus limits the practical use of well-designed skybox in reality for real-time applications. Alternatively, our method can bake the ambient lighting (effectively treating as self-emission) to stabilize this process. Fig. 9 demonstrates that our method maintains high temporal

consistency across frames. Details of environment baking results are provided in our supplementary (Sec. C.2) and video.

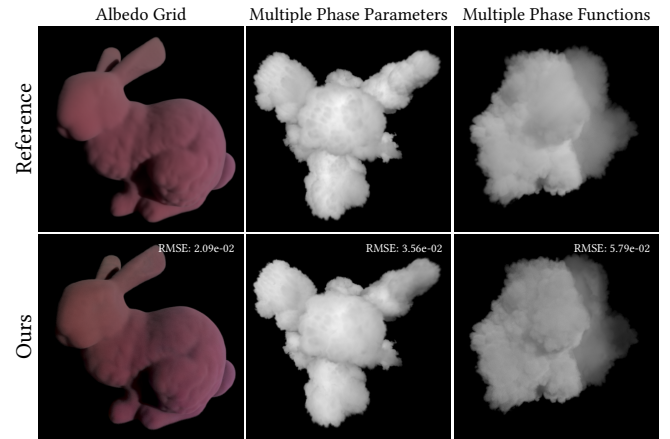


Fig. 7. Rendering result of heterogeneous configurations, with albedo grid (*bunny*), multiple phase parameters (*group*, HG, $g \in [0.16, 0.74]$) and two phase functions (*free-cloud-10*, HG + HG-Draine) respectively.

6.2 Ablation Study

Ablation of Feature Components. We conduct an ablation study to illustrate the contribution from each part of features. We make a general dataset with random configurations and a challenging dataset with high albedo for comprehensive evaluation. Tab. 2 shows the importance of every component on rendering quality. Removing any single component consistently degrades capability of the model.

Use SH for base grid. We introduce a base grid in Sec. 4.1 to sample towards light for avoiding cost of sampling SH grid, and point out that such simplification does not necessarily lead to worse model. We test several scenes and find that trained models show basically equivalent ability. The training plot (Fig. 8) and test error (Tab. 2) support our claim that using SH for base grid yields no consistent improvement in accuracy. However, the simplified design achieves around 10% time reduction in the rendering pipeline (Tab.1).

6.3 Training with Noisy Ground Truth

Since it is relatively computationally expensive to generate low-error radiance estimation, we also compare models trained on ground

Table 2. Ablation Study (*cloud1*). In error columns reported as a/b , a denotes error on general dataset while b denotes error on challenging dataset. Removing any component increases error, while using SH for the base grid yields similar error. The albedo module is not ablated as it is required to model albedo variation. MLP Scale refers to total number of parameters.

Ablation	MLP Scale	MSE (e-3)	ReLMSE (e-2)
w/o View Module ($\mathcal{V}$)	23.4K	2.53/5.46	7.63/14.2
w/o Phase Module ($\mathcal{P}$)	23.4K	2.74/5.78	6.86/11.9
w/o Base Grid ($\mathcal{B}$)	23.4K	2.68/5.70	8.24/14.5
w/o SH Grid ($\mathcal{L}$)	23.4K	2.36/5.00	6.54/11.3
w/o Tr Map ($\mathcal{T}$)	25.9K	2.39/4.98	7.30/12.7
Complete Features	28.5K	1.59/3.28	4.29/6.82
SH for Base Grid	28.5K	1.78/3.68	4.64/7.52

truth produced by MRPNN and by path tracing with less spp in Fig. 10. As expected, model performance improves as spp increases, confirming that less noisy ground truth leads to better-trained baking. However, under some challenging light conditions such as back light for dense volume, models struggle to learn an accurate mapping and MRPNN will also produce over-illumination effects. In these cases, our model alleviates such overestimation, and noisy training data can occasionally yield better results during optimization. Data from MRPNN can also train a comparable model that does not inherit over-illumination in MRPNN. We speculate that radiance generated by MRPNN exhibits a simpler latent pattern compared with path tracing, making it easier for our model to learn better. This resembles model distillation to some extent.

7 Conclusion, Limitations and Future Work

In this paper, we propose a novel lightning-fast neural baking method for high-quality volumetric rendering. We derive a novel diagonal scaling form of in-scattering integral in spherical harmonics basis through Legendre polynomials that eliminates explicit integration, and propose practical methods to employ feature grids accordingly. With our design for dual-pattern feature sampling and compact network architecture, our system is four to five times faster than the state-of-the-art and achieves hundreds of FPS. Beyond ordinary configurations supported by prior work, our method is also capable of handling more complex scenes like ones with spatially-varying albedo and phase or high-frequency ambient lighting. We validate our claims through extensive experiments.

Despite the effectiveness of our method, there still exist some limitations. First, our approach bakes volumes with fixed density field, which covers many rendering tasks but cannot handle dynamic media (notably clouds generated by Worley-Perlin noise). A temporal baking method could be used to mitigate this problem. Second, in line with previous work in the RPNN series, we assume a directional light as the light condition to represent time-of-day effects. This restricts its applicability in more complex dynamic incident light environment. Finally, the training cost in our framework is still relatively high. Although the inference speed is remarkable, long training time makes it inconvenient for users to manufacture baking assets. As a neural method, some delicate approach for tuning with few-shot ground truth may be designed.

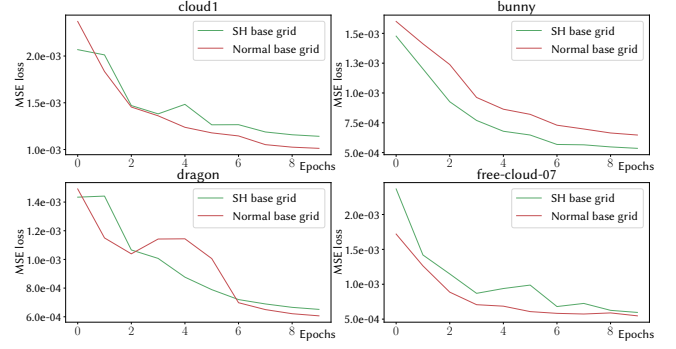


Fig. 8. MSE loss comparison on evaluation dataset between using SH for base grid $\mathcal{B}$ or not (i.e. normal). Normal one leads to lower loss in *cloud1*, higher loss in *bunny* and similar loss in *dragon* and *free-cloud-07*. Both convergence speed and final quality are roughly equivalent too. This shows that using normal grid for $\mathcal{B}$ to accelerate sampling process is acceptable.

Acknowledgments

This project is supported by the National Key R&D Program of China (No. 2023YFF0905103). It is also partially supported by the Shenzhen Science and Technology Program (No. KJZD20230923114114028) and the Shandong Key Research and Development Program (No. 2024CXGC010801). We also thank the following creators of the test scenes: Disney Clouds (*cloud0*), OpenVDB Samples (*dragon*, *bunny*, *armadillo*, *bunny_cloud*), CS Motion Design (clouds beginning with *free-*), sample data from MRPNN (*model1*, *cloud1*), and JangaFX (*new-cloud-03*).

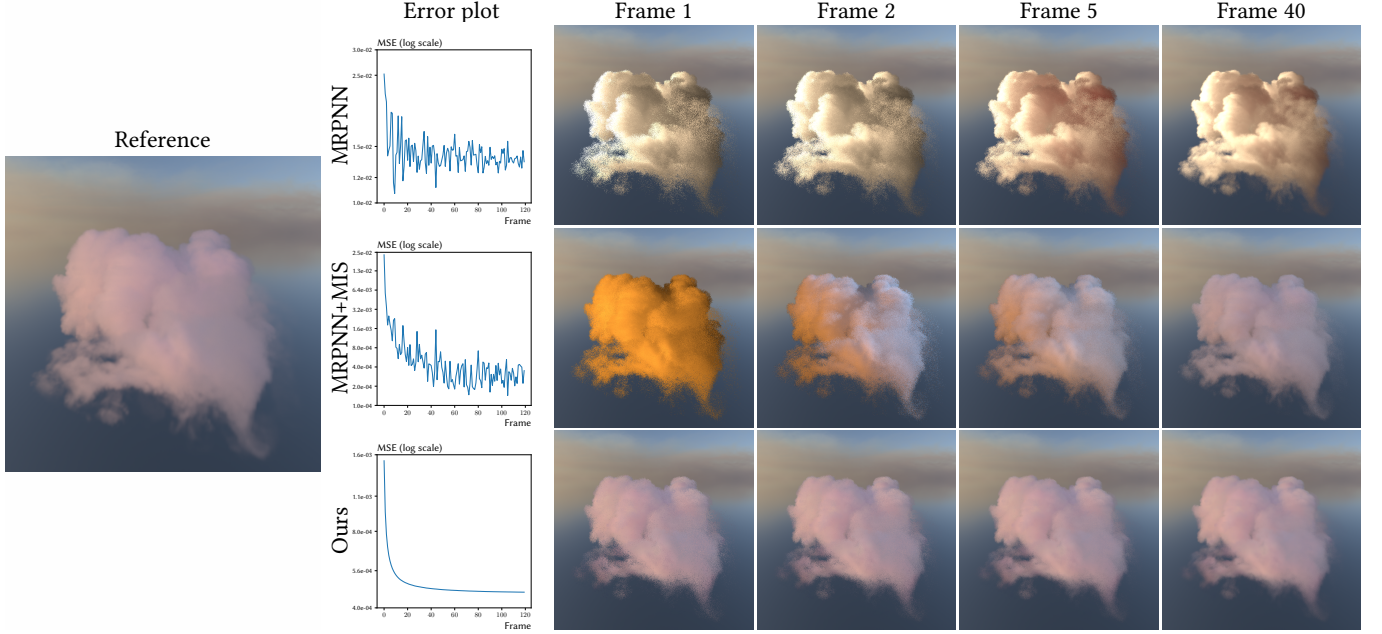


Fig. 9. Comparison between MRPNN and our method for rendering global illumination by environment lighting plus a sunset red directional light (*new-cloud-03*, with albedo = 0.999 and $g = 0.5$). For fair comparison, we additionally implement multiple importance sampling (MIS) for MRPNN, which performs better than original uniform sampling implementation but still has severe flickering. By contrast, our method produces a temporally stable result with high fidelity. See our supplementary video to show frames continuously for a more intuitive comparison.

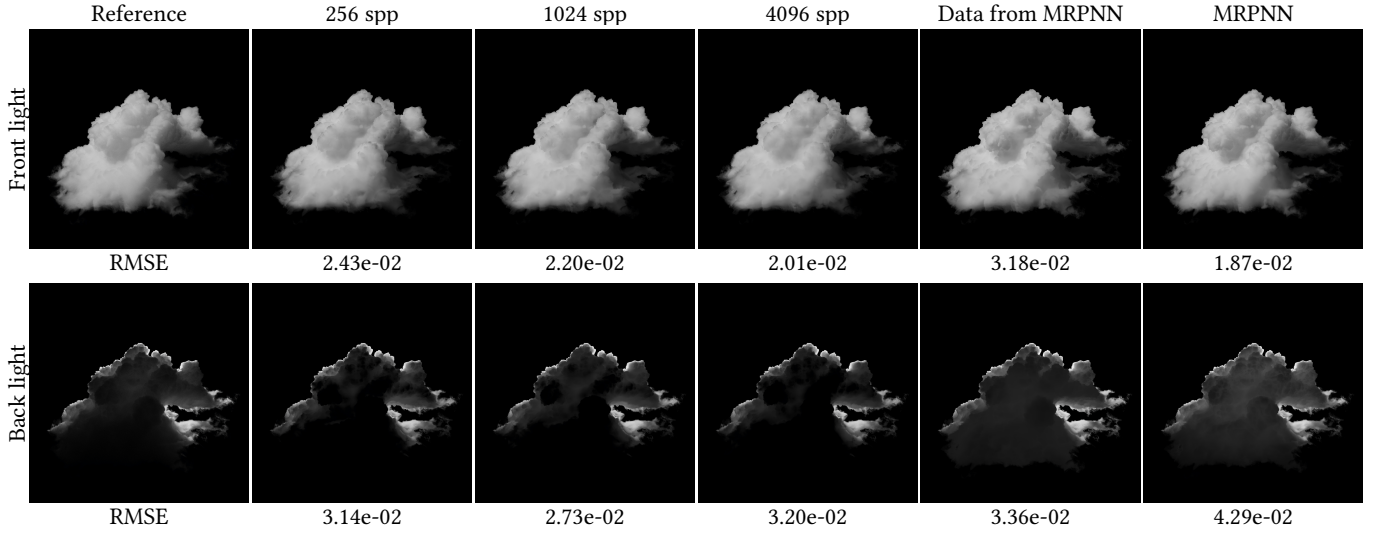


Fig. 10. Comparison for models trained by ground truth with different configurations on *cloud0*, with albedo = 0.999 and $g = 0.86$. Higher spp tends to lead to better result; but for tough light condition, noisy or biased data (MRPNN) can also train comparable models. It can be observed that all of our models alleviate over-illumination effects.

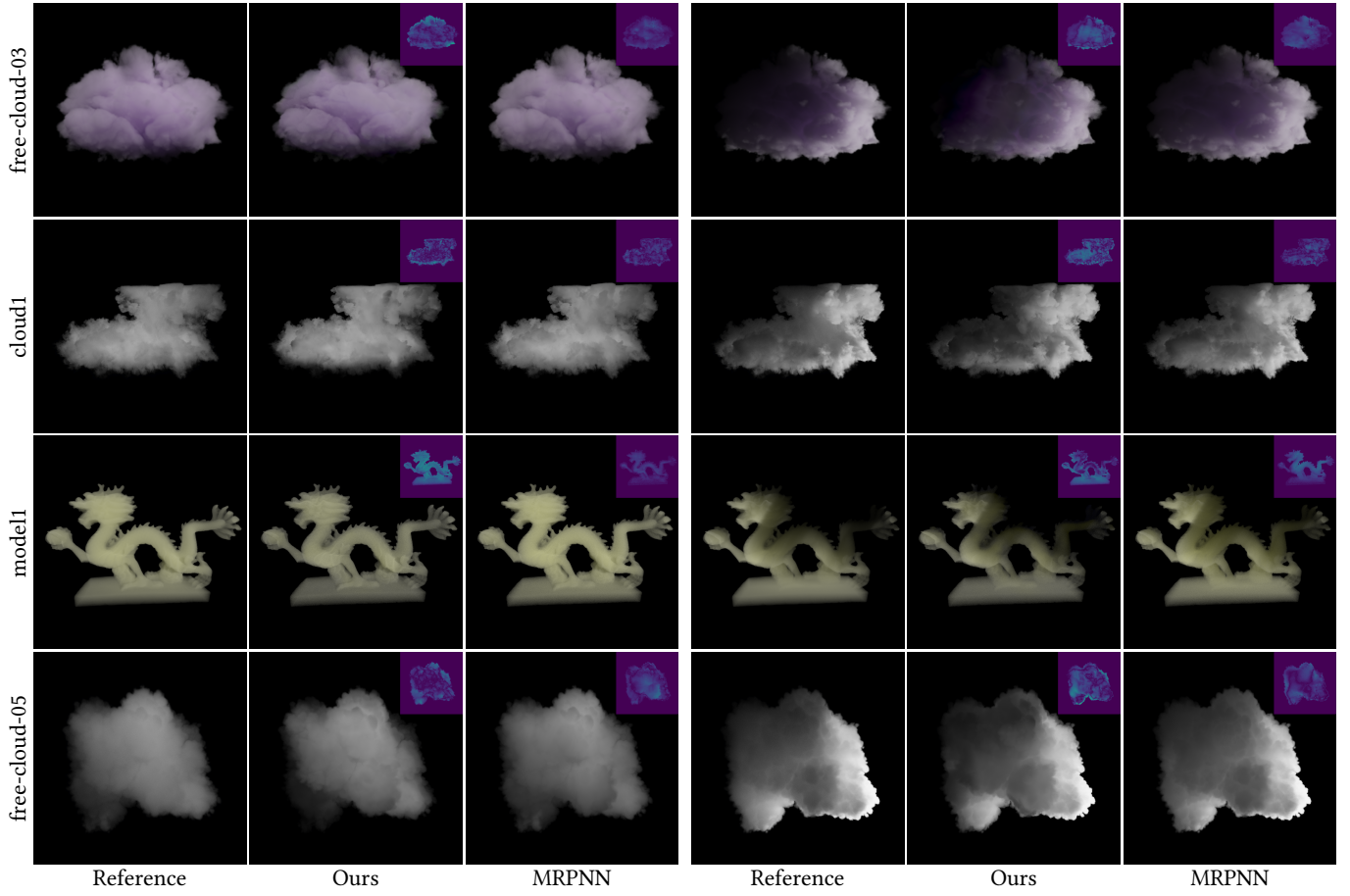


Fig. 11. MRPNN versus our method, converged results in a variety of scenes with front light (left) and side light (right). Under equal spp, our method achieves comparable visual quality to MRPNN while being significantly faster, and maintains high fidelity compared with references generated by path tracing. See our supplementary video for detailed FPS in different scenarios. For the four scenes (top to bottom), the albedo values are: $[0.984, 0.951, 0.999]$, 0.999 , $[0.999, 0.999, 0.793]$, 0.999 ; and the corresponding g values are: $0.5, 0.86, 0, 0.857$.

References

- Di An, Liangfu Kang, and Kun Xu. 2025. Real-Time Neural Homogeneous Translucent Material Rendering Using Diffusion Blocks. *IEEE Transactions on Visualization and Computer Graphics* 31, 10 (2025), 7519–7534. doi:10.1109/TVCG.2025.3548442
- Pontus Andersson, Jim Nilsson, Tomas Akenine-Möller, Magnus Oskarsson, Kalle Åström, and Mark D. Fairchild. 2020. FLIP: A Difference Evaluator for Alternating Images. *Proc. ACM Comput. Graph. Interact. Tech.* 3, 2, Article 15 (Aug. 2020), 23 pages. doi:10.1145/3406183
- Ang Cao and Justin Johnson. 2023. HexPlane: A Fast Representation for Dynamic Scenes. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023), 130–141. <https://api.semanticscholar.org/CorpusID:256105095>
- Chakravarty R. Alla Chaitanya, Anton S. Kaplanyan, Christoph Schied, Marco Salvi, Aaron Lefohn, Derek Nowrouzezahrai, and Timo Aila. 2017. Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder. *ACM Trans. Graph.* 36, 4, Article 98 (July 2017), 12 pages. doi:10.1145/3072959.3073601
- Jorge Condor, Sebastien Speierer, Lukas Bode, Aljaz Bozic, Simon Green, Piotr Didyk, and Adrian Jarabo. 2025. Don't Splat your Gaussians: Volumetric Ray-Traced Primitives for Modeling and Rendering Scattering and Emissive Media. *ACM Trans. Graph.* 44, 1, Article 10 (Feb. 2025), 17 pages. doi:10.1145/3711853
- Honghao Dong, Guoping Wang, and Sheng Li. 2023. Neural Parametric Mixtures for Path Guiding. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) (SIGGRAPH '23). Association for Computing Machinery, New York, NY, USA, Article 29, 10 pages. doi:10.1145/3588432.3591533
- Bruce Fowler. 1983. Expansion of Mie-theory phase functions in series of Legendre polynomials. *Journal of the Optical Society of America* 73 (01 1983), 19–22. doi:10.1364/JOSA.73.000019
- Iliyan Georgiev, Zackary Misso, Toshiya Hachisuka, Derek Nowrouzezahrai, Jaroslav Krivánek, and Wojciech Jarosz. 2019. Integral formulations of volumetric transmittance. *ACM Trans. Graph.* 38, 6, Article 154 (Nov. 2019), 17 pages. doi:10.1145/3355089.3356559
- Saeed Hadadan, Shuhong Chen, and Matthias Zwicker. 2021. Neural radiosity. *ACM Trans. Graph.* 40, 6, Article 236 (Dec. 2021), 11 pages. doi:10.1145/3478513.3480569
- Pat Hanrahan and Wolfgang Krueger. 1993. Reflection from layered surfaces due to subsurface scattering. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques* (Anaheim, CA) (SIGGRAPH '93). Association for Computing Machinery, New York, NY, USA, 165–174. doi:10.1145/166117.166139
- Sebastian Herholz, Yangyang Zhao, Oskar Elek, Derek Nowrouzezahrai, Hendrik P. A. Lensch, and Jaroslav Krivánek. 2019. Volume Path Guiding Based on Zero-Variance Random Walk Theory. *ACM Trans. Graph.* 38, 3, Article 25 (June 2019), 19 pages. doi:10.1145/3230635
- Jinkai Hu, Chengzhong Yu, Hongli Liu, Lingqi Yan, Yiqian Wu, and Xiaogang Jin. 2023. Deep Real-time Volumetric Rendering Using Multi-feature Fusion. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) (SIGGRAPH '23). Association for Computing Machinery, New York, NY, USA, Article 61, 10 pages. doi:10.1145/3588432.3591493
- Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, Merlin Nimier-David, Delio Vicini, Tizian Zeltner, Baptiste Nicolet, Miguel Crespo, Vincent Leroy, and Ziyi Zhang. 2022. *Mitsuba 3 renderer*. <https://mitsuba-renderer.org>.
- Johannes Jendersie and Eugene d'Eon. 2023. An Approximate Mie Scattering Function for Fog and Cloud Rendering. In *ACM SIGGRAPH 2023 Talks* (Los Angeles, CA, USA) (SIGGRAPH '23). Association for Computing Machinery, New York, NY, USA, Article 47, 2 pages. doi:10.1145/3587421.3595409
- James T. Kajiya and Brian P. Von Herzen. 1984. Ray tracing volume densities. *SIGGRAPH Comput. Graph.* 18, 3 (Jan. 1984), 165–174. doi:10.1145/964965.808594
- Simon Kallweit, Thomas Müller, Brian McWilliams, Markus Gross, and Jan Novák. 2017. Deep scattering: rendering atmospheric clouds with radiance-predicting neural networks. *ACM Trans. Graph.* 36, 6, Article 231 (Nov. 2017), 11 pages. doi:10.1145/3130800.3130880
- Markus Kettunen, Eugene D'Eon, Jacopo Pantaleoni, and Jan Novák. 2021. An unbiased ray-marching transmittance estimator. *ACM Trans. Graph.* 40, 4, Article 137 (July 2021), 20 pages. doi:10.1145/3450626.3459937
- Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. [arXiv:1412.6980 \[cs.LG\]](https://arxiv.org/abs/1412.6980) <https://arxiv.org/abs/1412.6980>
- Eric P. Lafortune and Yves D. Willems. 1996. Rendering participating media with bidirectional path tracing. In *Proceedings of the Eurographics Workshop on Rendering Techniques '96* (Porto, Portugal). Springer-Verlag, Berlin, Heidelberg, 91–100.
- Daqi Lin, Chris Wyman, and Cem Yuksel. 2021. Fast volume rendering with spatiotemporal reservoir resampling. *ACM Trans. Graph.* 40, 6, Article 279 (Dec. 2021), 18 pages. doi:10.1145/3478513.3480499
- Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2021. NeRF: representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (Dec. 2021), 99–106. doi:10.1145/3503250
- Bailey Miller, Iliyan Georgiev, and Wojciech Jarosz. 2019. A null-scattering path integral formulation of light transport. *ACM Trans. Graph.* 38, 4, Article 44 (July 2019), 13 pages. doi:10.1145/3306346.3323025
- Zackary Misso, Yining Karl Li, Brent Burley, Daniel Teece, and Wojciech Jarosz. 2023. Progressive null-tracking for volumetric rendering. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) (SIGGRAPH '23). Association for Computing Machinery, New York, NY, USA, Article 31, 10 pages. doi:10.1145/3588432.3591557
- Thomas Müller. 2021. *tiny-cuda-nn*. <https://github.com/NVlabs/tiny-cuda-nn>
- Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.* 41, 4, Article 102 (July 2022), 15 pages. doi:10.1145/3528223.3530127
- Thomas Müller, Fabrice Rousselle, Jan Novák, and Alexander Keller. 2021. Real-time neural radiance caching for path tracing. *ACM Trans. Graph.* 40, 4, Article 36 (July 2021), 16 pages. doi:10.1145/3450626.3459812
- Jan Novák, Iliyan Georgiev, Johannes Hanika, and Wojciech Jarosz. 2018. Monte Carlo methods for volumetric light transport simulation. In *Computer graphics forum*, Vol. 37. Wiley Online Library, 551–576.
- Minghao Pan, Rui Wang Xinguo Liu, Qunsheng Peng, and Hujun Bao. 2007. Precomputed Radiance Transfer Field for Rendering Interreflections in Dynamic Scenes. *Computer Graphics Forum* 26, 3 (2007), 485–493. [arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2007.01071.x](https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2007.01071.x) doi:10.1111/j.1467-8659.2007.01071.x
- Mikhail Panin and Sergey Nikolenko. 2019. Faster RPNN: Rendering Clouds with Latent Space Light Probes. In *SIGGRAPH Asia 2019 Technical Briefs* (Brisbane, QLD, Australia) (SA '19). Association for Computing Machinery, New York, NY, USA, 21–24. doi:10.1145/3355088.3365150
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates Inc., Red Hook, NY, USA.
- Mark Pauly, Thomas Kolli, and Alexander Keller. 2000. Metropolis Light Transport for Participating Media. In *Proceedings of the Eurographics Workshop on Rendering Techniques 2000*. Springer-Verlag, Berlin, Heidelberg, 11–22.
- Gilles Rainer, Adrien Bousseau, Tobias Ritschel, and George Drettakis. 2022. Neural Precomputed Radiance Transfer. *Computer Graphics Forum* 41 (05 2022). doi:10.1111/cgf.14480
- G. Sansone. 1959. *Orthogonal Functions*. R. E. Krieger Publishing Company.
- Nicholas Sharp, Souhaib Attaki, Keenan Crane, and Maks Ovsjanikov. 2022. Diffusion-Net: Discretization Agnostic Learning on Surfaces. *ACM Trans. Graph.* 41, 3, Article 27 (March 2022), 16 pages. doi:10.1145/3507905
- Peter-Pike Sloan, Jan Kautz, and John Snyder. 2002. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Trans. Graph.* 21, 3 (July 2002), 527–536. doi:10.1145/566654.566612
- Peter-Pike Sloan, Ben Luna, and John Snyder. 2005. Local, deformable precomputed radiance transfer. *ACM Trans. Graph.* 24, 3 (July 2005), 1216–1224. doi:10.1145/1073204.1073335
- Rui Su, Honghao Dong, Haojie Jin, Yisong Chen, Guoping Wang, and Sheng Li. 2025. Vertex Features for Neural Global Illumination. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers* (SA Conference Papers '25). Association for Computing Machinery, New York, NY, USA, Article 10, 11 pages. doi:10.1145/3757377.3763826
- Rui Su, Honghao Dong, Jierui Ren, Haojie Jin, Yisong Chen, Guoping Wang, and Sheng Li. 2024. Dynamic Neural Radiosity with Multi-grid Decomposition. In *SIGGRAPH Asia 2024 Conference Papers* (Tokyo, Japan) (SA '24). Association for Computing Machinery, New York, NY, USA, Article 23, 12 pages. doi:10.1145/3680528.3687685
- Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. 2021. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11358–11367.
- E Woodcock, T Murphy, P Hemmings, and S Longworth. 1965. Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry. In *Proc. Conf. Applications of Computing Methods to Reactor Problems*, Vol. 557. Argonne National Laboratory.
- Kehan Xu, Sebastian Herholz, Marco Manzi, Marios Papas, and Markus Gross. 2024. Volume Scattering Probability Guiding. *ACM Trans. Graph.* 43, 6, Article 184 (Nov. 2024), 17 pages. doi:10.1145/3687982
- Chong Zeng, Yue Dong, Pieter Peers, Hongzhi Wu, and Xin Tong. 2025. RenderFormer: Transformer-based Neural Rendering of Triangle Meshes with Global Illumination. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Papers* (SIGGRAPH Conference Papers '25). Association for Computing Machinery, New York, NY, USA, Article 48, 11 pages. doi:10.1145/3721238.3730595
- Chuanqun Zheng, Yuchi Huo, Hongxiang Huang, Hongtao Sheng, Junrong Huang, Rui Tang, Hao Zhu, Rui Wang, and Hujun Bao. 2024. Neural Global Illumination via Superposed Deformable Feature Fields. In *SIGGRAPH Asia 2024 Conference Papers* (Tokyo, Japan) (SA '24). Association for Computing Machinery, New York, NY, USA, Article 24, 11 pages. doi:10.1145/3680528.3687680