

Supplementary: Multi-feature Radiance Baking Neural Networks for Instant Volumetric Rendering

JIAMING LIANG, School of Computer Science, Peking University, China

HONGLIANG YUAN, Xiaomi Company, China

MENG GAI, School of Computer Science, Peking University, China

GUOPING WANG, School of Computer Science, Peking University, China

SHENG LI*, School of Computer Science, Peking University, China

ACM Reference Format:

Jiaming Liang, Hongliang Yuan, Meng Gai, Guoping Wang, and Sheng Li. 2026. Supplementary: 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, 6 pages. <https://doi.org/10.1145/3799902.3811179>

A Feature Grid Configuration

Though grid resolution and feature dimension depend on volume complexity, an empirical configuration would be to use $\mathcal{B}$ of resolution 128^3 , $\mathcal{L}_{i,l}^m$ of 64^3 with $K = 2$, $\mathcal{V}$ of 48^2 and $\mathcal{G}, \mathcal{A}$ of 160^2 as the finest grain. All grids use a feature dimension of 8, and coarser grids are gradually downscaled by 1.5. $\mathcal{T}$ can be set as 128^3 with corresponding MIPMap generated. This keeps a balance between memory consumption and prediction quality, but can be fine-tuned further for production-ready deployment. Also, although ideally margin consistency could be learned in baking of $\mathcal{V}$, a subtle seam can be observed in a few scenes due to discontinuity between 0 and 2π in spherical coordinates. Given the low resolution of $\mathcal{V}$, an alternative way is to use 3D grid indexed by Cartesian coordinates. The full ablation study table (Tab. 3) shows that this adjustment does not fundamentally affect the final quality.

Table 3. Full table of ablation study (cloud1). In error columns with a/b , a means error on general dataset, and b means with error on challenging dataset. See Sec. 6.2 for how two datasets are generated.

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
3D Grid for $\mathcal{V}$	28.5K	1.67/3.45	4.62/7.56
SH for Base Grid	28.5K	1.78/3.68	4.64/7.52

*Sheng Li is the corresponding author.



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 979-8-4007-2554-8/2026/07
<https://doi.org/10.1145/3799902.3811179>

We also conduct a simple experiment on error with respect to SH order. For $K \leq 4$, prediction performance on test dataset generally increases when K increases as shown in Fig. 12, and such improvement is more pronounced on challenging dataset. However, when $K \geq 5$, we observe performance degradation under same training epochs, which can attribute to the fact that more optimizable parameters make training harder. To keep a balance between accuracy, training speed and runtime sampling cost, we use $K = 2$ as a general configuration. This also illustrates the effectiveness of using low-order compression since a limited band already exhibits good accuracy compared with higher orders.

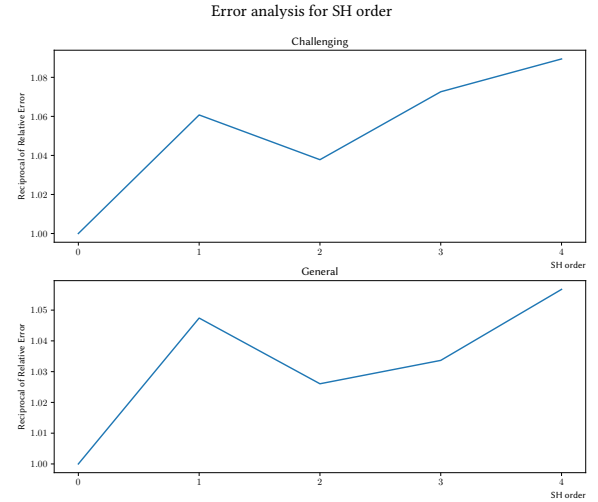


Fig. 12. Plot for reciprocal of relative error under same training epochs when $K \leq 4$ (cloud1), with error of $K = 0$ as normalization factor.

B Network Architecture Figure Illustration

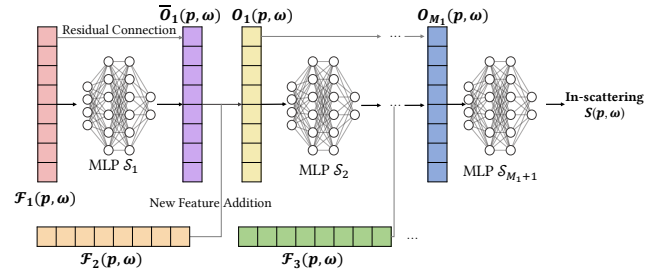


Fig. 13. Network architecture to fuse features in different levels.

C Experiment Details

C.1 Heterogeneous Configurations

In Fig. 9, we display three complex cases with configurations below.

Albedo Grid. Value on each channel is evenly spaced over $[0.8, 1.0]$ and combined. For instance, relative coordinate $(0, 0, 0)$ in volume bounding box has albedo $(0.8, 0.8, 0.8)$ and $(0.1, 0.2, 0.3)$ has albedo $(0.82, 0.84, 0.86)$. During data generation, the albedo grid additionally multiplies an RGB scale for model to learn. That is, the trained model can predict effects of scaling any channel of an albedo grid from baking without runtime cost.

Multiple phase parameters. Eight volumes are grouped in a single scene. To avoid combinatorial amount of ground truth (N^8 if each volume has N choices), we use $g = 0.86, 0.8, 0.72, 0.62, 0.5, 0.4, 0.3, 0.2$ as respective anchors for each volume and train with a uniform random scaling.

Multiple phase functions. Use HG for $x < 0$ and HG-Draine for $x \geq 0$. For HG-Draine, we use the fit function over water droplet diameters provided in [Jendersie and d'Eon 2023]. However, we find that it has a very high g_{HG} in its HG components ($g_{HG} \in [0.97, 0.995]$) and is thus extremely anisotropic. To make PT converge faster, we additionally times 0.9 in g_{HG} , but it is still a tough task to learn well. Data generation will randomly combine the phase parameters and two phase grids are used to encode related information. We also provide training results on volume with pure HG-Draine in Fig. 14.

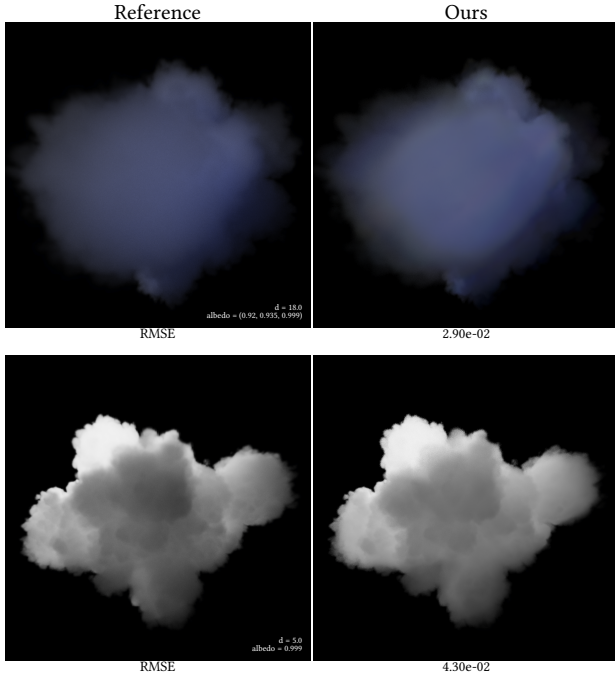


Fig. 14. Experiments on volumes equipped with HG-Draine phase function. It tends to be harder than normal HG since it has a very high g in its HG components.

C.2 Ambient Light Baking Design

The ambient baking is slightly different from the directional light baking for:

- (1) No deterministic light direction, so no ω_s for spherical harmonics interpolation, no $\cos \theta = \omega \cdot \omega_s$ as index and no $\mathcal{T}$ to establish.
- (2) Albedo that has same values in three channels does not lead to same radiance in three color channels.

As we are handling static ambient light, we can instead turn back to Theorem 3.1 and use view direction to march and interpolate. Also, we merge phase module and albedo module to be a 2D grid $\mathcal{PA}$, where the original index $\cos \theta$ is not involved. Finally, we use three $\mathcal{PA}$ to differentiate features on three channels, and use a shared MLP to decode. So the final feature for HG phase function would be:

$$\mathcal{F}_{i,j}(p, \omega) = \left(\mathcal{B}_i(p_i), \sum_{l=0}^K g(p)^l \sum_{m=-l}^l \mathcal{L}_{i,l}^m(p) Y_l^m(\omega), \mathcal{GA}_{i,j}(g(p), \alpha(p)) \right) \quad (11)$$

And the specific configuration is same as directional light but can also be fine-tuned. We do simple tests on training by (1) excluding direct illumination and (2) disjoint MLPs on three channels, but there are no benefits observed. Disjoint MLPs even show slightly worse results under same epochs as larger parameter scale tends to make training harder.

It is worth mentioning that in our scene for Fig. 10, separate baking of ambient light roughly degrades 240 FPS to 160 FPS as a trade-off to overcome inherent limitation of RPN series. It may be achievable to merge ambient light baking with directional light baking in the design to mitigate that performance loss, but then the environment exposure is not independent of directional light luminance, so we do not verify such an idea. MRPNN also needs to recompute transmittance map every frame under environment lighting so such trade-off is reasonable.

It is also worth noting that MRPNN may be able to bake ambient lighting via per-scene optimization theoretically due to approximation capacity of neural networks. However, it would both sacrifice generality without additional speedup and incur much longer training time (four days as Hu et al. [2023] report) for lacking optimization in latent space. Our design significantly boosts training and inference efficiency.

D Proof: Phase Decomposition

D.1 Diagonal Scaling Form of $\mathcal{K}$

On SH basis, coordinates of L can be expressed in two triangular matrices. For $L(p, \omega) = \sum_{l=0}^{\infty} \sum_{m=-l}^l L_l^m(p) Y_l^m(\omega)$, we can split $m \geq 0$ and $m < 0$ and get (omit p for brevity):

$$\mathbf{C}_1 = \begin{bmatrix} L_0^0 & 0 & 0 & \cdots \\ L_1^1 & L_1^0 & 0 & \cdots \\ L_2^2 & L_2^1 & L_2^0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \mathbf{C}_2 = \begin{bmatrix} 0 & 0 & 0 & \cdots \\ L_1^{-1} & 0 & 0 & \cdots \\ L_2^{-2} & L_2^{-1} & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (12)$$

Multiplying $\mathcal{K} = \text{diag}(\eta_0, \eta_1, \dots)$ leads to:

$$\mathcal{K}\mathbf{C}_1 = \begin{bmatrix} \eta_0 L_0^0 & 0 & 0 & \dots \\ \eta_1 L_1^1 & \eta_1 L_1^0 & 0 & \dots \\ \eta_2 L_2^2 & \eta_2 L_2^1 & \eta_2 L_2^0 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \mathcal{K}\mathbf{C}_2 = \begin{bmatrix} 0 & 0 & 0 & \dots \\ \eta_1 L_1^{-1} & 0 & 0 & \dots \\ \eta_2 L_2^{-2} & \eta_2 L_2^{-1} & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (13)$$

To interpret coordinates back to $\mathbb{R}$ given direction ω , define:

$$\mathbf{Y}_1(\omega) = \begin{bmatrix} Y_0^0(\omega) & 0 & 0 & \dots \\ Y_1^1(\omega) & Y_1^0(\omega) & 0 & \dots \\ Y_2^2(\omega) & Y_2^1(\omega) & Y_2^0(\omega) & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \quad (14)$$

$$\mathbf{Y}_2(\omega) = \begin{bmatrix} 0 & 0 & 0 & \dots \\ Y_1^{-1}(\omega) & 0 & 0 & \dots \\ Y_2^{-2}(\omega) & Y_2^{-1}(\omega) & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Then $S(p, \omega) = \text{tr}((\mathcal{K}\mathbf{C}_1(p)) \mathbf{Y}_1(\omega)^T) + \text{tr}((\mathcal{K}\mathbf{C}_2(p)) \mathbf{Y}_2(\omega)^T)$, which is also equivalent to summation of all elements for Hadamard product $(\mathcal{K}\mathbf{C}_1(p)) \odot \mathbf{Y}_1(\omega) + (\mathcal{K}\mathbf{C}_2(p)) \odot \mathbf{Y}_2(\omega)$.

D.2 Detailed Proof of Theorem 3.1

PROOF. Substituting $L(p, \omega)$ in integral, we then get:

$$S(p, \omega) = \int_{S^2} \phi(p, \omega_i, \omega) \sum_{l=0}^{\infty} \sum_{m=-l}^l L_l^m(p) Y_l^m(\omega_i) d\omega_i \quad (15)$$

For simplicity, we omit LHS in the following formulas. We already assume ϕ is a function of $\cos \theta$, therefore we can transform ϕ as series of Legendre polynomials P_l for its orthogonality, leading to:

$$\int_{S^2} \left(\sum_{l=0}^{\infty} f_l(p) P_l(\omega_i \cdot \omega) \right) \cdot \left(\sum_{l=0}^{\infty} \sum_{m=-l}^l L_l^m(p) Y_l^m(\omega_i) \right) d\omega_i \quad (16)$$

According to the addition theorem of SH, this can be further derived into:

$$\int_{S^2} \left(\sum_{l=0}^{\infty} \frac{4\pi f_l(p)}{2l+1} \sum_{m=-l}^l Y_l^m(\omega) \bar{Y}_l^m(\omega_i) \right) \cdot \left(\sum_{l=0}^{\infty} \sum_{m=-l}^l L_l^m(p) Y_l^m(\omega_i) \right) d\omega_i \quad (17)$$

where $\bar{Y}_l^m(\omega)$ is conjugation of $Y_l^m(\omega)$. As both two series converge, let $\eta_l(p) = \frac{4\pi f_l(p)}{2l+1}$, we can fuse the summation and get:

$$\int_{S^2} \left(\sum_{l=0}^{\infty} \sum_{m=-l}^l \sum_{l'=0}^{\infty} \sum_{m'=-l'}^{l'} \eta_l(p) Y_l^m(\omega) \bar{Y}_l^m(\omega_i) L_{l'}^{m'}(p) Y_{l'}^{m'}(\omega_i) \right) d\omega_i \quad (18)$$

Under reasonable assumption (see Sec. D.3), we can rearrange summation and integral as Eq. 19.

$$\sum_{l=0}^{\infty} \sum_{m=-l}^l \sum_{l'=0}^{\infty} \sum_{m'=-l'}^{l'} \eta_l(p) Y_l^m(\omega) L_{l'}^{m'}(p) \int_{S^2} \bar{Y}_l^m(\omega_i) Y_{l'}^{m'}(\omega_i) d\omega_i \quad (19)$$

And by normalization of spherical harmonics, this can be further derived as Eq. 20 with Kronecker delta:

$$\sum_{l=0}^{\infty} \sum_{m=-l}^l \sum_{l'=0}^{\infty} \sum_{m'=-l'}^{l'} \eta_l(p) Y_l^m(\omega) L_{l'}^{m'}(p) \delta_{ll'} \delta_{mm'} \quad (20)$$

So finally, Kronecker delta eliminates summations and we get the target formulation.

$$S(p, \omega) = \sum_{l=0}^{\infty} \eta_l(p) \sum_{m=-l}^l L_l^m(p) Y_l^m(\omega) \quad (21)$$

□

D.3 Rearrange Summation and Integral

We rearrange summation and integral in transformation from Eq. 18 to Eq. 19. Though practically Fubini's Theorem is often applied directly, we justify the claim in theory.

PROOF. Let's first truncate two series so we can rearrange them freely:

$$f^{(N)}(\omega_i \cdot \omega) = \sum_{l=0}^N \sum_{m=-l}^l \eta_l(p) Y_l^m(\omega) \bar{Y}_l^m(\omega_i) \quad (22)$$

$$L^{(M)}(p, \omega_i) = \sum_{l'=0}^M \sum_{m'=-l'}^{l'} L_{l'}^{m'}(p) Y_{l'}^{m'}(\omega_i)$$

Therefore,

$$\begin{aligned} & \int_{S^2} f^{(N)}(\omega_i \cdot \omega) L^{(M)}(p, \omega_i) d\omega_i \\ &= \sum_{l=0}^N \sum_{m=-l}^l \sum_{l'=0}^M \sum_{m'=-l'}^{l'} \eta_l(p) L_{l'}^{m'}(p) Y_{l'}^{m'}(\omega) \int_{S^2} Y_l^m(\omega_i) \bar{Y}_{l'}^{m'}(\omega_i) d\omega_i \\ &= \sum_{l=0}^{\min(N,M)} \eta_l(p) \sum_{m=-l}^l L_l^m(p) Y_l^m(\omega) \end{aligned} \quad (23)$$

And we hope to prove $\int_{S^2} f^{(N)} L^{(M)} d\omega_i \rightarrow \int_{S^2} f L d\omega_i$. Observe that (omit $d\omega_i$ for brevity):

$$\begin{aligned} & \left| \int f L - \int f^{(N)} L^{(M)} \right| \\ &= \left| \int f L - \int f^{(N)} L + \int f^{(N)} L - \int f^{(N)} L^{(M)} \right| \\ &\leq \left| \int (f - f^{(N)}) L \right| + \left| \int f^{(N)} (L - L^{(M)}) \right| \end{aligned} \quad (24)$$

In a rendering system, it is reasonable to assume that decompositions lead to an error that goes to 0 when the polynomial order increases to infinity, and both phase and radiance are bounded. That is,

$$\lim_{N \rightarrow \infty} f^{(N)} = f < \infty, \lim_{N \rightarrow \infty} L^{(M)} = L < \infty \quad (25)$$

So both integral converge to 0 and thus $\lim_{N \rightarrow \infty, M \rightarrow \infty} \int f^{(N)} L^{(M)} = \int f L$, meaning that interchange leads to the same value. □

To be accurate, we note that what Neural Radiosity [Hadadan et al. 2021] verifies is that first estimating and then integrating yields better results than pure estimation. Thus, such a pipeline needs many passes to estimate the integral. We assume it is still

correct for baking features, or in other words, integration first (in the latent space) plus estimation can also lead to better results than pure estimation. This process can eliminate the integration process in our derivation. Such inspiration is also experimentally verified as shown in Fig. 6. The indirect decomposition is generally preferred for reweighting derived from physical equation to smoothen training, but the low-rank compression may make direct decomposition sometimes compensate better.

D.4 Henyey-Greenstein (HG)

For readers' convenience, we provide proof of Lemma 3.2 directly.

PROOF. By definition of the generating function for Legendre polynomials, we know:

$$G(x, t) = \frac{1}{\sqrt{1-2xt+t^2}} = \sum_{l=0}^{\infty} P_l(x)t^l, \text{ where } |x| \leq 1 \quad (26)$$

Getting partial derivative with respect to t on both sides leads to:

$$\frac{\partial G(x, t)}{\partial t} = \frac{x-t}{(1-2xt+t^2)^{3/2}} = \sum_{l=1}^{\infty} lP_l(x)t^{l-1} \quad (27)$$

Combining Eq. 26 and Eq. 27, we can get:

$$\begin{aligned} G(x, t) + 2t \frac{\partial G(x, t)}{\partial t} &= \frac{(1-2xt+t^2) + 2t(x-t)}{(1-2xt+t^2)^{3/2}} = \frac{1-t^2}{(1-2xt+t^2)^{3/2}} \\ &= \sum_{l=0}^{\infty} P_l(x)t^l + 2t \sum_{l=1}^{\infty} lP_l(x)t^{l-1} \\ &= \sum_{l=0}^{\infty} (2l+1)P_l(x)t^l \end{aligned} \quad (28)$$

Let $x = \cos \theta, t = g$, we thus get:

$$\phi_{HG}(\cos \theta) = \frac{1}{4\pi} \sum_{l=0}^{\infty} (2l+1)g^l P_l(\cos \theta) \quad (29)$$

So finally we know $f_l = \frac{2l+1}{4\pi} g^l$ and therefore $\eta_l = \frac{4\pi}{2l+1} f_l = g^l$. $\square$

D.5 Draine

We also use HG-Draine to illustrate our ability to handle different phase functions in a single scene. As HG-Draine is a linear interpolation of HG and Draine's phase function, we just need to calculate the decomposition of Draine, and the specific Legendre polynomial coefficients are interpolated as is.

THEOREM D.1. Given $\phi(\alpha, g, \cos \theta) = \frac{1}{4\pi} \cdot \frac{1-g^2}{(1+g^2-2g\cos \theta)^{3/2}} \cdot \frac{1+\alpha\cos^2 \theta}{K}$

where $K = 1 + \alpha(1+2g^2)/3$, let

- (1) $A_n = \frac{(n-1)n}{2n-1}$ for $n \geq 2$, otherwise 0,
- (2) $B_n = \frac{(n+1)^2}{2n+3} + \frac{n^2}{2n-1}$,
- (3) $C_n = A_{n+2} = \frac{(n+2)(n+1)}{2n+3}$.

then $\eta_n = \frac{1}{(2n+1)K} \left[\alpha A_n g^{n-2} + (\alpha B_n + 2n+1) g^n + \alpha C_n g^{n+2} \right]$.

Particularly, we can give specific results for first few terms:

- $\eta_0 = 1$;
- $\eta_1 = \frac{1}{3K} \left(\left(3 + \frac{9}{5}\alpha \right) g + \frac{6}{5}\alpha g^3 \right)$,

- $\eta_2 = \frac{1}{5K} \left(\frac{2}{3}\alpha + \left(5 + \frac{55}{21}\alpha \right) g^2 + \frac{12}{7}\alpha g^4 \right)$

Or numerically, let $K_i = 1/K$:

- $\eta_0 = 1$;
- $\eta_1 = K_i \left((1 + 0.6\alpha) g + 0.4\alpha g^3 \right)$,
- $\eta_2 = K_i \left(0.1333333\alpha + (1 + 0.5238095\alpha) g^2 + 0.3428571\alpha g^4 \right)$

When $\alpha = 0$, Draine is just HG and thus perfectly matches HG decomposition.

PROOF. For brevity, we first omit the constant factor $\frac{1}{4\pi K}$, leading to:

$$\phi(\alpha, g, \cos \theta) = 4\pi \cdot \phi_{HG}(g, \cos \theta) \cdot (1 + \alpha \cos^2 \theta) \quad (30)$$

We might as well denote $\cos \theta = \mu$ and omit LHS. It's easily computed that $x^2 = \frac{2}{3}P_2(x) + \frac{1}{3}P_0(x)$, and we already know the decomposition of HG, so it can be transformed to product of Legendre polynomials:

$$\left[\sum_{m=0}^{\infty} (2m+1)g^m P_m(\mu) \right] \left[\left(1 + \frac{\alpha}{3} \right) P_0(\mu) + \frac{2\alpha}{3} P_2(\mu) \right] = \sum_{n=0}^{\infty} b_n P_n(\mu) \quad (31)$$

The general decomposition of $P_k(\mu)P_m(\mu)$ is already summarized in [Bailey 1933]. As $P_0(\mu)$ is just constant 1, now we pay attention to decomposition of product $P_m(\mu)P_2(\mu)$ and can get:

$$\begin{aligned} P_2(\mu)P_m(\mu) &= \alpha(m, m-2)P_{m-2}(\mu) + \alpha(m, m)P_m(\mu) + \\ &\quad \alpha(m, m+2)P_{m+2}(\mu) \end{aligned} \quad (32)$$

where

$$\alpha(m, m-2) = \frac{3m(m-1)}{2(2m+1)(2m-1)}, \alpha(m, m+2) = \frac{3(m+1)(m+2)}{2(2m+1)(2m+3)},$$

$$\alpha(m, m) = \frac{3}{2} \left(\frac{(m+1)^2}{(2m+1)(2m+3)} + \frac{m^2}{(2m+1)(2m-1)} \right) - \frac{1}{2}.$$

Therefore, only $m = n-2$, $m = n$ and $m = n+2$ contribute to coefficients of $P_n(\mu)$. Let $a_n = (2n+1)g^n$, we can get:

$$b_n = \left(1 + \frac{\alpha}{3} \right) a_n + \frac{2\alpha}{3} \left[a_n \alpha(n, n) + a_{n+2} \alpha(n, n+2) + a_{n-2} \alpha(n-2, n) \right]. \quad (33)$$

By cancelling common terms, $\frac{2\alpha}{3} a_{n-2} \alpha(n-2, n)$ results in A_n and $\frac{2\alpha}{3} a_{n+2} \alpha(n, n+2)$ results in C_n ; the remaining part can be reduced into B_n . When $n < 2$, A_n does not participate in contribution. Finally, η_l is got by multiplying previously omitted constant and factor of f_l , i.e. $\frac{1}{4\pi K} \cdot \frac{4\pi}{2n+1} = \frac{1}{(2n+1)K}$. $\square$

E Additional Discussions and Experiments

E.1 Supplementary Validation

We conduct some experiments on close-up view to illustrate the robustness. As Fig. 15 shows, our model generates results in the same error level consistently under different view settings. The core reason is that our pipeline still obeys standard physically based volumetric rendering process with mere rendering acceleration of scattering radiance. Only when the training dataset fails to collect data of (p, ω) in some regions should visual effects of the pipeline degrade.

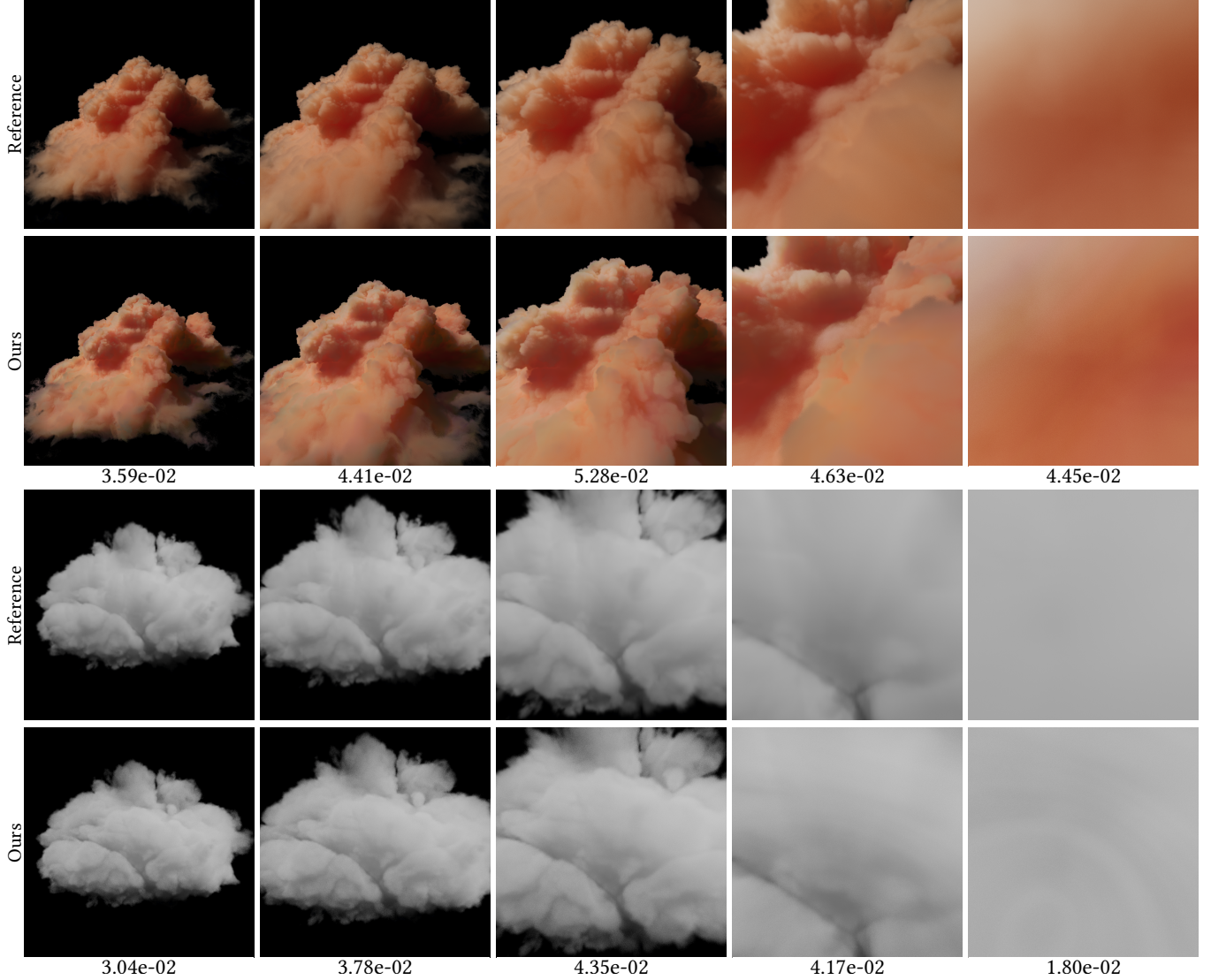


Fig. 15. Gradually close-up views for *cloud0* and *free-cloud-03*, which show consistent RMSE error level in all settings.

We also conduct some experiments on $g < 0$ for thoroughness as shown in Fig. 17. For fairness, we use the same parameter range as MRPNN ($g \in [0, 0.86]$) in the main paper, and Fig. 17 provides test case for backward scattering.

E.2 Predict Bounces Later

Following RPNN family, our model predicts radiance after one scattering event. Nevertheless, it is also possible to predict radiance for bounce $N > 1$, which could improve learning since radiance tends to attenuate when $N \rightarrow \infty$ and is thus easier to bake. However, since VRTE is evaluated using Monte-Carlo method, delaying one bounce would introduce an additional integral estimation, which may result in higher variance. It can be investigated whether better prediction can compensate such variance under same time budget so choosing a later bounce makes result better.

E.3 Possible Artifact

A subtle halo-like artifact can occasionally appear in a few scenes when the camera sees towards the light. We find that it happens when the anisotropy of the volume leads to extremely distorted gradient around grid margin as shown in Fig. 18. Intuitively, this can be solved by allocating more space on $\cos \theta$ near 0 and 1 in $\mathcal{G}$ and $\mathcal{A}$, so we try to use θ/π as index, plus using 1024 as finest grain. We also conduct experiments to apply plain de-ringing techniques [Sloan 2017] on the SH grid. Though Fig. 16 shows that no essential improvement is observed in both trials and thus future investigation is needed, we point out that such artifact can be easily eliminated by multiplying $\cos \theta$ with a constant near 1 when practically used as coordinate. Such trick doesn't raise error but works very efficiently and effectively.

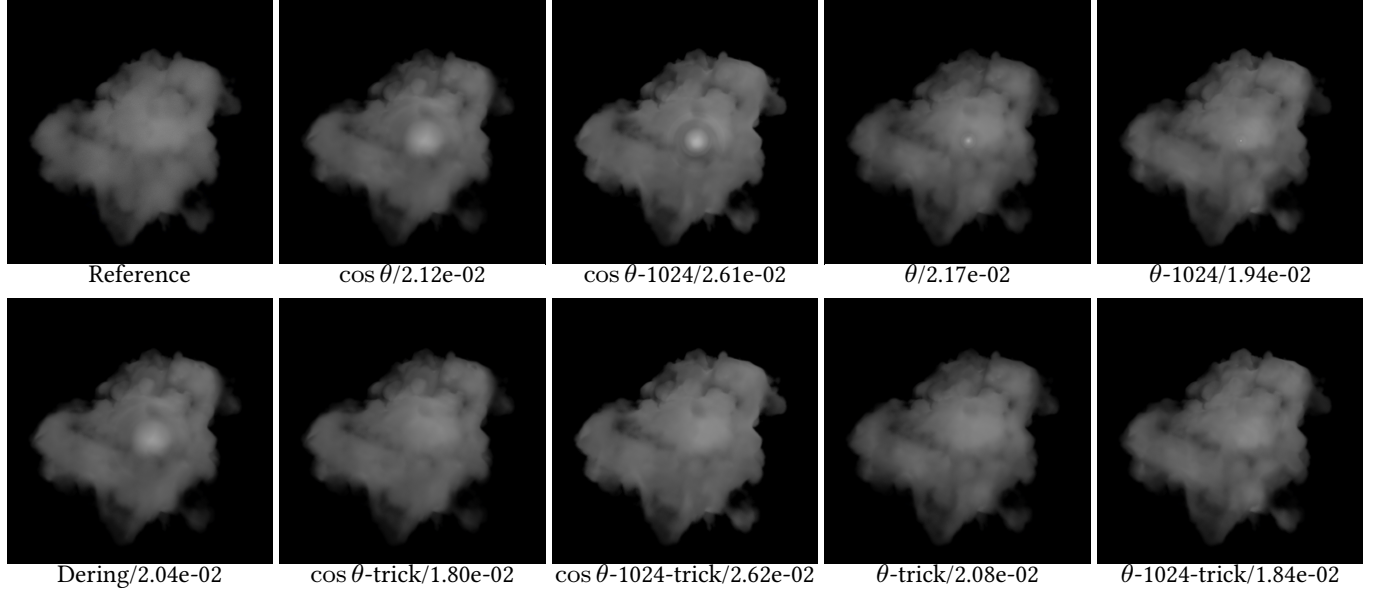


Fig. 16. Possible halo artifacts. $\cos \theta$ means using $(\cos \theta + 1)/2$ proposed in Sec. 5, and θ means using θ/π ; "-1024" means using $\mathcal{G}$ and $\mathcal{A}$ of resolution 1024 while others use configuration in Sec. A; "-trick" means multiplying with an additional constant near 1 (here 0.98). The artifact insists though allocating more space can reduce the size of halo (for θ -1024, zoom in to find it). We also apply the de-ringing window for SH grid but no improvement is observed. Instead of introducing such overhead, using the trick can eliminate it efficiently and effectively. The scene parameters are albedo = 1.0 and $g = 0.86$.

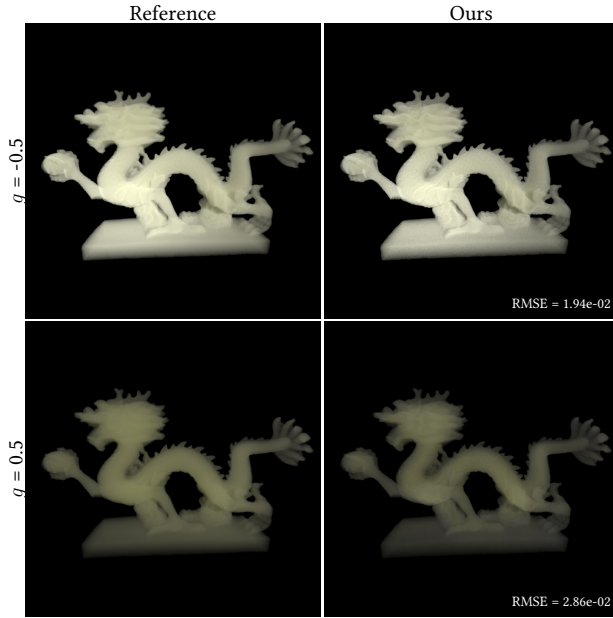


Fig. 17. Backward and forward scattering experiments on *model1*, which show that our method can handle different parameter ranges correctly.

References

- W. N. Bailey. 1933. On the product of two Legendre polynomials. *Mathematical Proceedings of the Cambridge Philosophical Society* 29, 2 (1933), 173–177. doi:10.1017/S0305004100010963
- 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

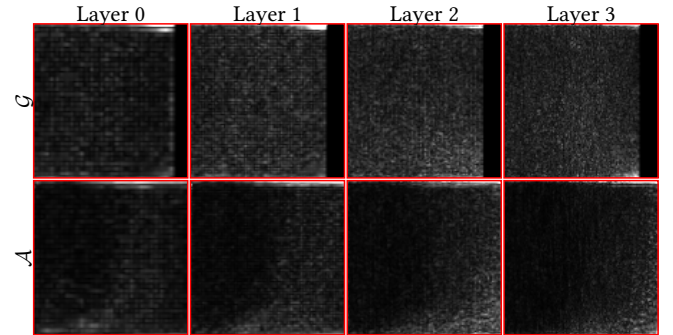


Fig. 18. Let latent vector of the grid be $\mathbf{v}$, $y = (\cos \theta + 1)/2$, the figure shows visualization of gradient norm ($\|\partial \mathbf{v} / \partial y\|_2$) divided by a factor of 40. The vertical axis represents y . It can be found that the top right corner is apparently brighter, which indicates that there exists sharp value transition and finally causes artifact in prediction in the case of high albedo, high anisotropy and towards-light view. Note that a black band appears along the right edge region of $\mathcal{G}$ because we use the same parameter range as MRPNN ($g \in [0, 0.86]$) and the rest of grid is not optimized during supervision.

- 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
- 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
- Peter-Pike Sloan. 2017. Deringing spherical harmonics. In *SIGGRAPH Asia 2017 Technical Briefs* (Bangkok, Thailand) (SA '17). Association for Computing Machinery, New York, NY, USA, Article 11, 4 pages. doi:10.1145/3145749.3149438