Deep Scattering: Rendering Atmospheric Clouds with Radiance-Predicting Neural Networks - Supplementary Material

SIMON KALLWEIT, Disney Research and ETH Zürich
THOMAS MÜLLER, Disney Research and ETH Zürich
BRIAN MCWILLIAMS, Disney Research
MARKUS GROSS, Disney Research and ETH Zürich
JAN NOVÁK, Disney Research

Here we discuss the details of obtaining the data for the Lorenz-Mie phase function and chopping of the peak. We also detail the CNN architecture that we compared to in Figure 10 of the main paper. Lastly, we provide additional results and evaluations of our method, and a comparison to the flux-limited diffusion by Koerner et al. [2014]. While this document provides an overview of the images, we recommend using the interactive viewer to best inspect the differences between methods.

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

Additional Key Words and Phrases: Monte Carlo Rendering, Deep Learning, Machine Learning, Participating Media, Phase Functions, Clouds

ACM Reference format:
Simon Kallweit, Thomas Müller, Brian McWilliams, Markus Gross, and Jan Novák. 2017. Deep Scattering: Rendering Atmospheric Clouds with Radiance-Predicting Neural Networks - Supplementary Material. *ACM Trans. Graph.* 36, 6, Article 231 (November 2017), 14 pages.
https://doi.org/10.1145/3130800.3130880

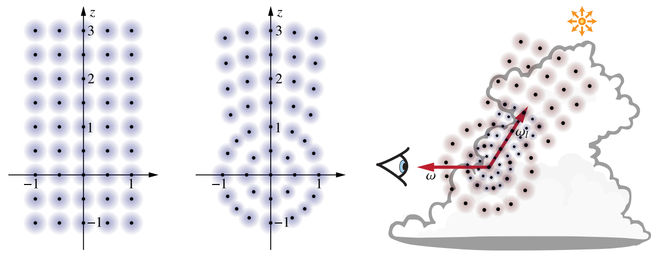
1 LORENZ-MIE PHASE FUNCTION

We used MiePlot [Laven 2017] to extract a tabulated version of Lorenz-Mie phase function for an aerosol with gamma-distributed radii (in μm) with shape $k = 2$ and scale $\theta = 2$. To obtain the final shape, we averaged phase-function profiles for three visible colors, red (650 nm), green (530 nm), and blue (450 nm), across 50 randomly sampled radii from the gamma distribution.

To remove the diffraction peak, we replace the values within the $[0°, 8°]$ by taking the slope at $8°$ and extrapolating the function as a line towards $0°$. We then renormalize the function to yield a valid phase function and reduce the extinction coefficient according to the fraction of scattered light contained in the chopped peak. More precisely, the reduced extinction coefficient μ_t' is given by $\mu_t'(\mathbf{x}) = r \cdot \mu_t(\mathbf{x})$ with

$$r = 1 - \int_{S^2} p_p(\cos\theta)\,\mathrm{d}\widehat{\omega},$$

where μ_t is the original extinction, p_p is the portion of the phase function which is chopped away, and $\cos\theta$ is the z-component of $\widehat{\omega}$. In our case, $r = 0.559$. For modeling the scattering distribution with a HG phase function we leave $\mu_t(\mathbf{x})$ unchanged and set the g-parameter to the g measured from the Lorenz-Mie phase function. In the case of modeling the scattering distribution with an isotropic phase function, $\mu_t'(\mathbf{x}) = \mu_t(\mathbf{x})(1-g)$ is computed via similarity theory. For our data, $g = 0.857$.



(a) Cartesian grid (b) Spherical shells (c) Local frame

Fig. 1. Warping the Cartesian $5 \times 5 \times 9$ stencil into spherical shells.

2 CNN ARCHITECTURE

We experimented also with CNN architectures. The one used in the paper in Figure 10 uses 2 layers of 3D convolution on the stencil features. Both layers use 3x3x3 convolution kernels. The first layer reduces the 5x5x9 stencil features to 3x3x7 values, the second reduces further to 1x1x5 values. The first convolution layer is using 50 filters, the second is using 100 filters. The filters are shared among the different stencil levels (scales). The outputs of the second convolutional layers are combined across stencil levels and with the angle feature and fed into 3 fully connected layers of width 400, 200 and 200 and then passed to the output unit. We use ReLU as activation funtion on the fully connected layers. Overall, this network architecture used around twice the number of parameters than the MLP-based architecture that we propose, yet still performed slightly worse in terms of quality and incurred significantly higher computational overhead.

3 ALTERNATIVE POINT STENCILS

3.1 Spherical Stencil

We tried warping the Cartesian grid into spherical shells: each point $\mathbf{p}$ is warped into point $\mathbf{q}$ on the shell as $\mathbf{q} = \mathbf{p}\,\|\mathbf{p}\|_\infty / \|\mathbf{p}\|_2$; see Figure 1 for an illustration. While this makes the stencil rounder, it did not significantly improve the quality. Nevertheless, due to its more symmetrical shape it might be worth considering in cases when orientation-dependent artifacts arise.

0730-0301/2017/11-ART231 $15.00
https://doi.org/10.1145/3130800.3130880

3.2 Data-driven Stencil

We also investigated shaping the point stencil according to the spatial distribution of fluence around the local frame centered at $\mathbf{x}$ as described in the paper. We tabulated this distribution in a grid by averaging Monte Carlo samples obtained from configurations $\mathcal{S}$ which we generated in the same way as we do for generating training data. We then warped a $7 \times 7 \times 7$ stencil (we tried regular and jittered spacing) such that the density of stencil points was proportional to the tabulated fluence. This process produced slightly anisotropic stencils on small scales and mostly isotropic ones on larger scales. We suspect, that these stencils did not work as well as our simpler ones because of their irregular spacing, making it difficult to optimally filter the cloud density during training and rendering.

REFERENCES

David Koerner, Jamie Portsmouth, Filip Sadlo, Thomas Ertl, and Bernd Eberhardt. 2014. Flux-Limited Diffusion for Multiple Scattering in Participating Media. *Computer Graphics Forum* 33, 6 (Sept. 2014), 178–189. https://doi.org/10.1111/cgf.12342

Philip Laven. 2017. MiePlot. (2017). http://www.philiplaven.com/mieplot.htm.

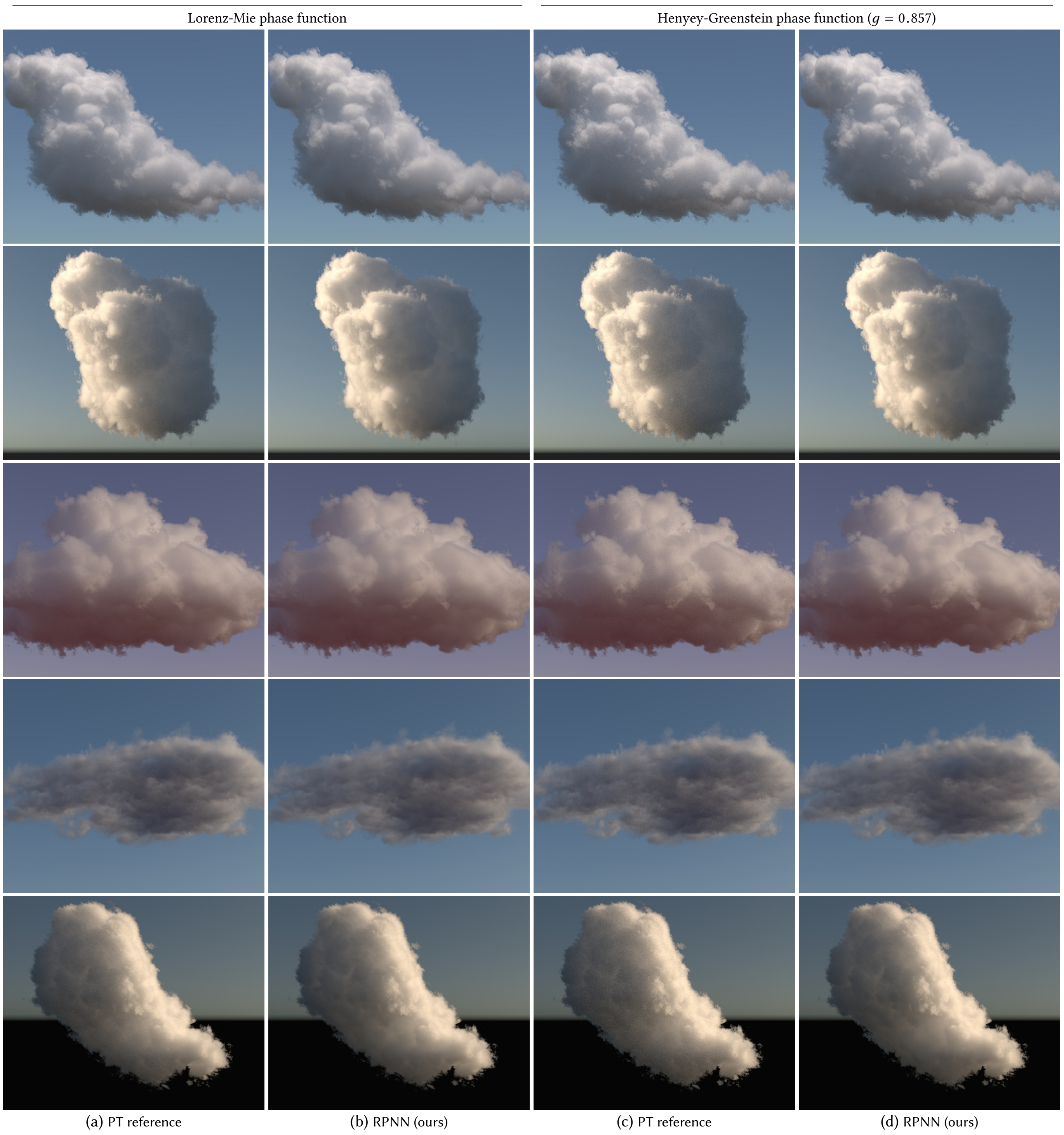

Lorenz-Mie phase function
Henyey-Greenstein phase function ($g = 0.857$)
(a) PT reference
(b) RPNN (ours)
(c) PT reference
(d) RPNN (ours)

Fig. 2. Comparison of a path-traced reference and our radiance-predicting neural network (PRNN) on side-lit clouds illuminated by a sun-sky model.

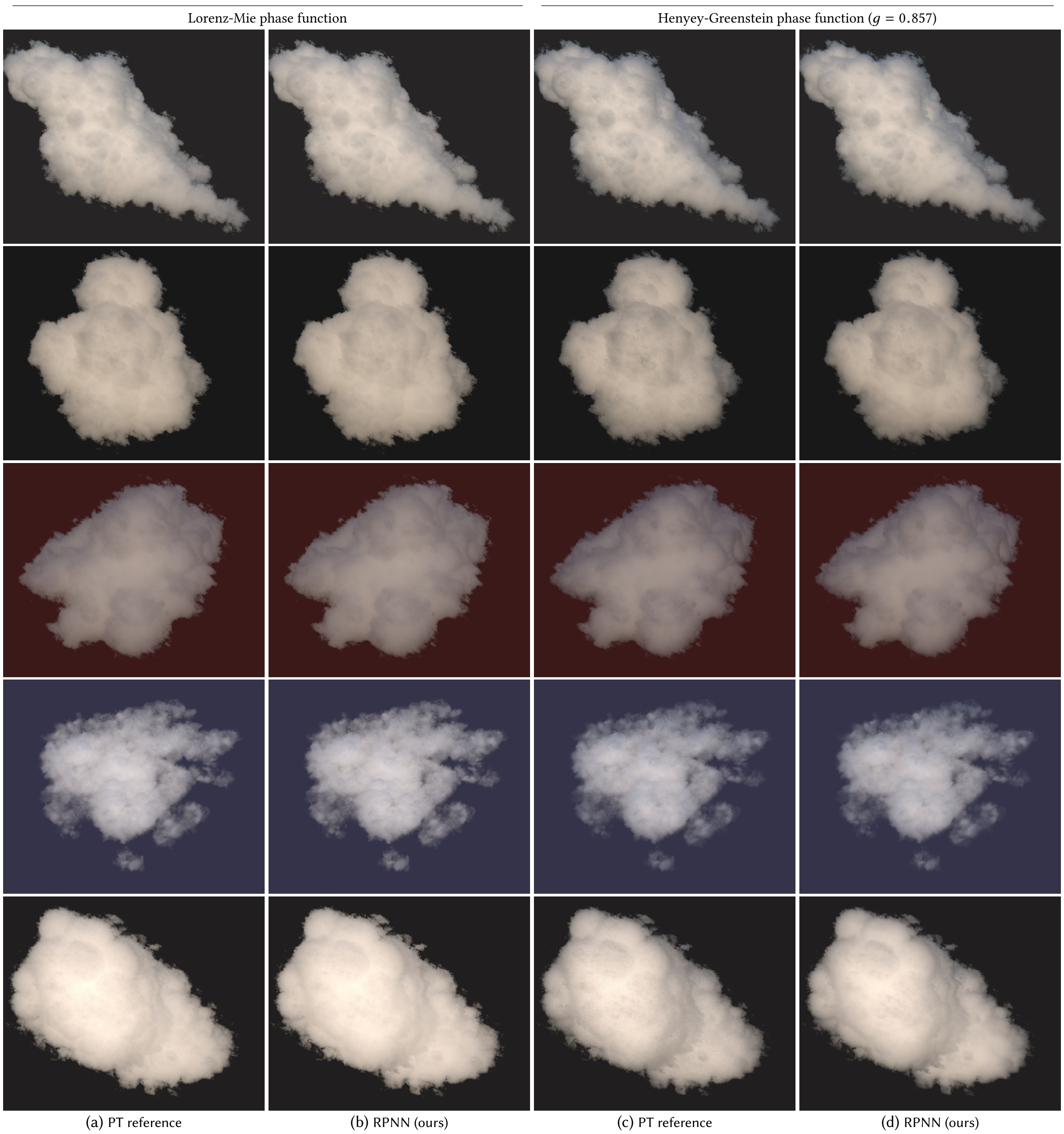

Lorenz-Mie phase function
Henyey-Greenstein phase function ($g = 0.857$)
(a) PT reference
(b) RPNN (ours)
(c) PT reference
(d) RPNN (ours)

Fig. 3. Comparison of a path-traced reference and our radiance-predicting neural network (PRNN) on front-lit clouds illuminated by a sun-sky model.

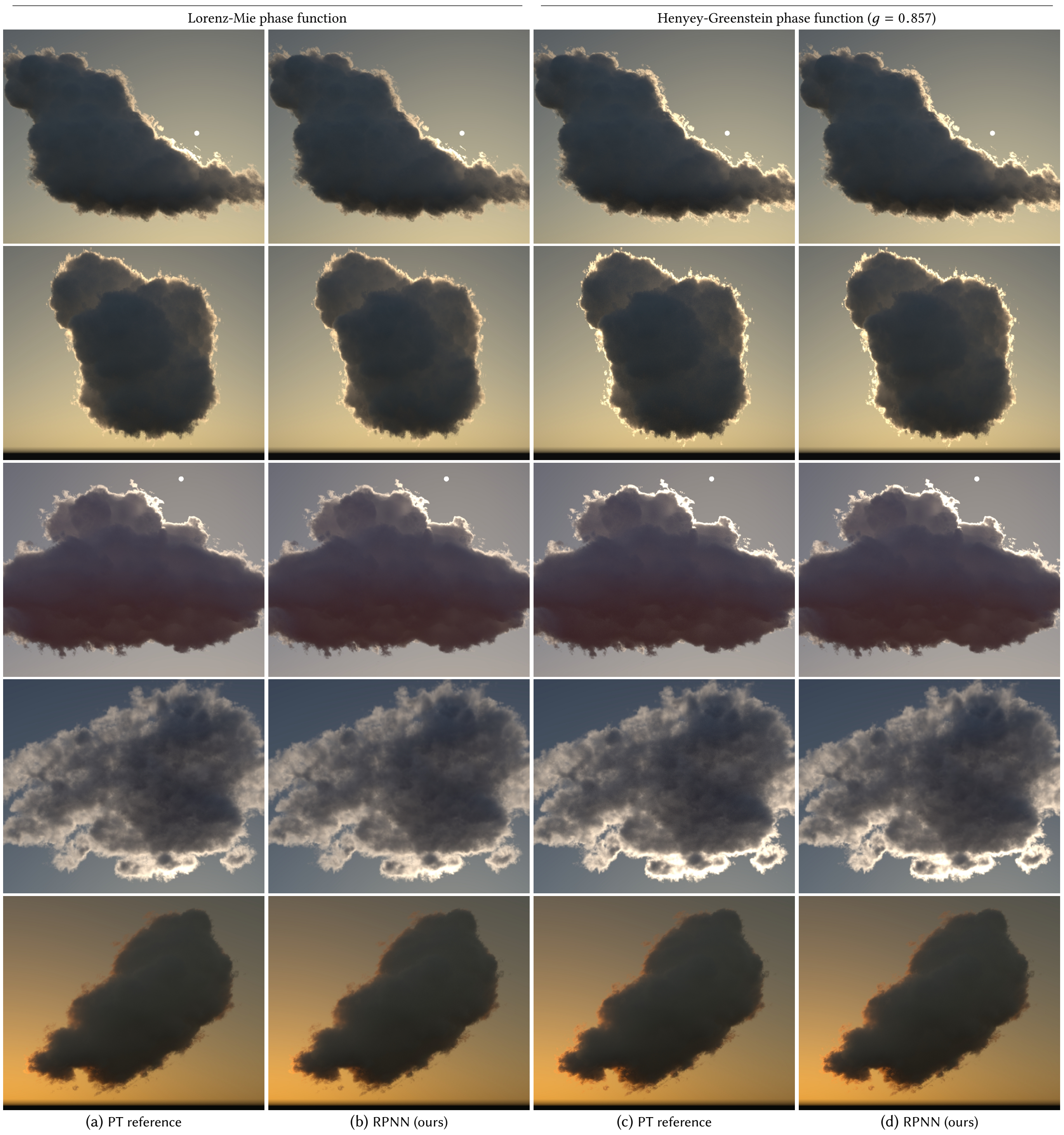

Lorenz-Mie phase function
Henyey-Greenstein phase function (g = 0.857)
(a) PT reference
(b) RPNN (ours)
(c) PT reference
(d) RPNN (ours)

Fig. 4. Comparison of a path-traced reference and our radiance-predicting neural network (PRNN) on back-lit clouds illuminated by a sun-sky model.

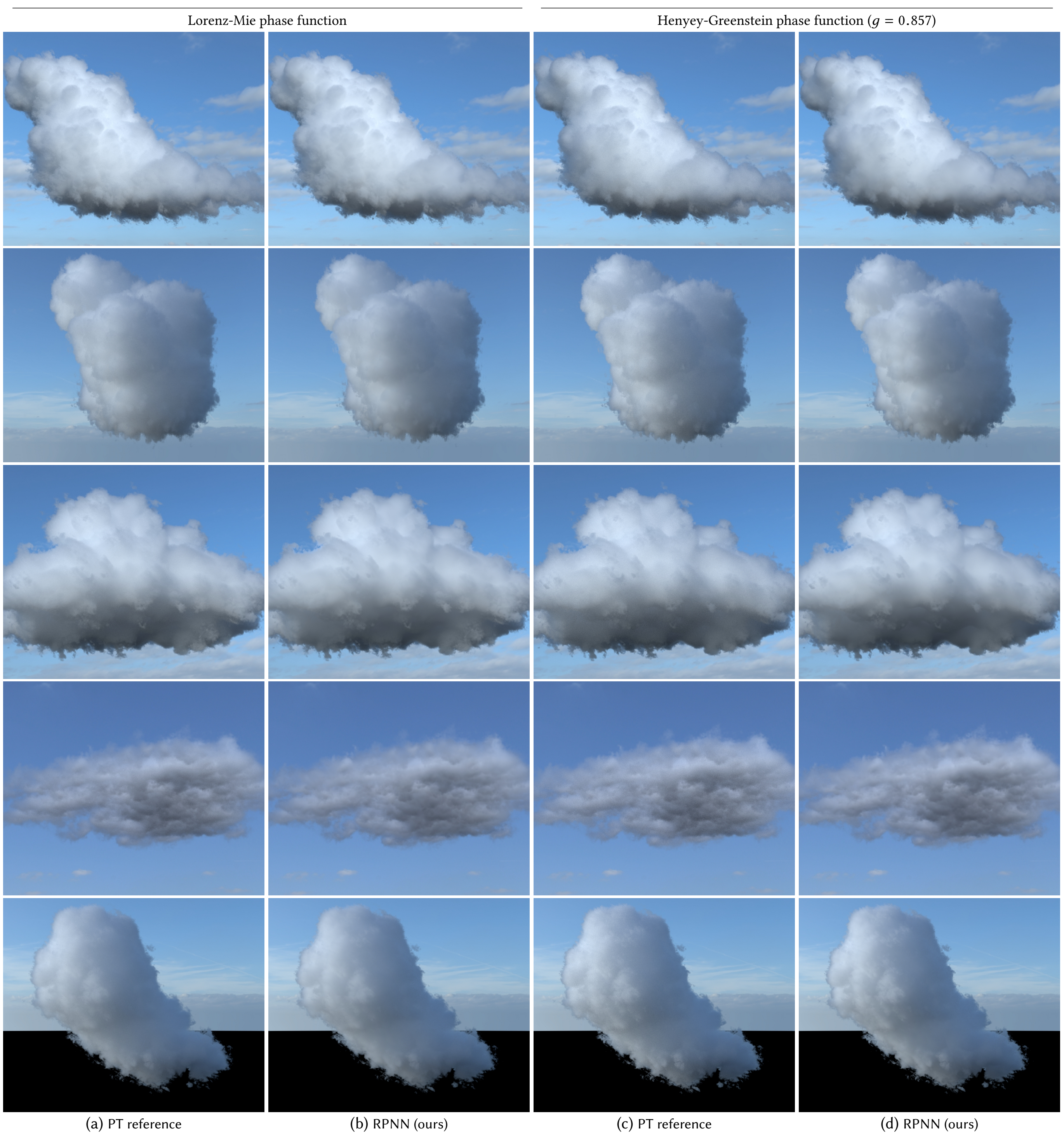

Lorenz-Mie phase function
Henyey-Greenstein phase function ($g = 0.857$)
(a) PT reference
(b) RPNN (ours)
(c) PT reference
(d) RPNN (ours)

Fig. 5. Comparison of a path-traced reference and our radiance-predicting neural network (PRNN) on side-lit clouds illuminated by an environment map.

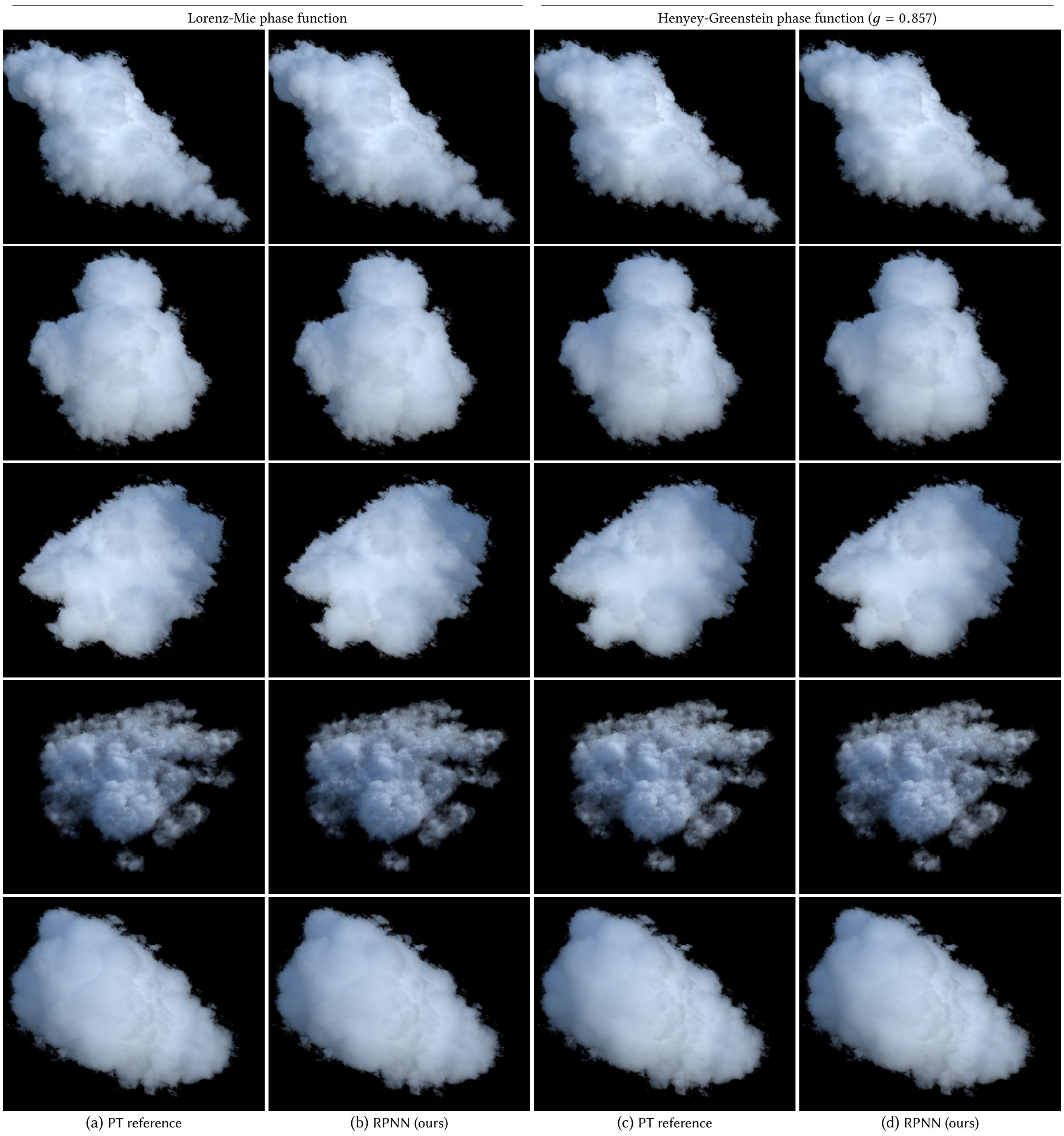
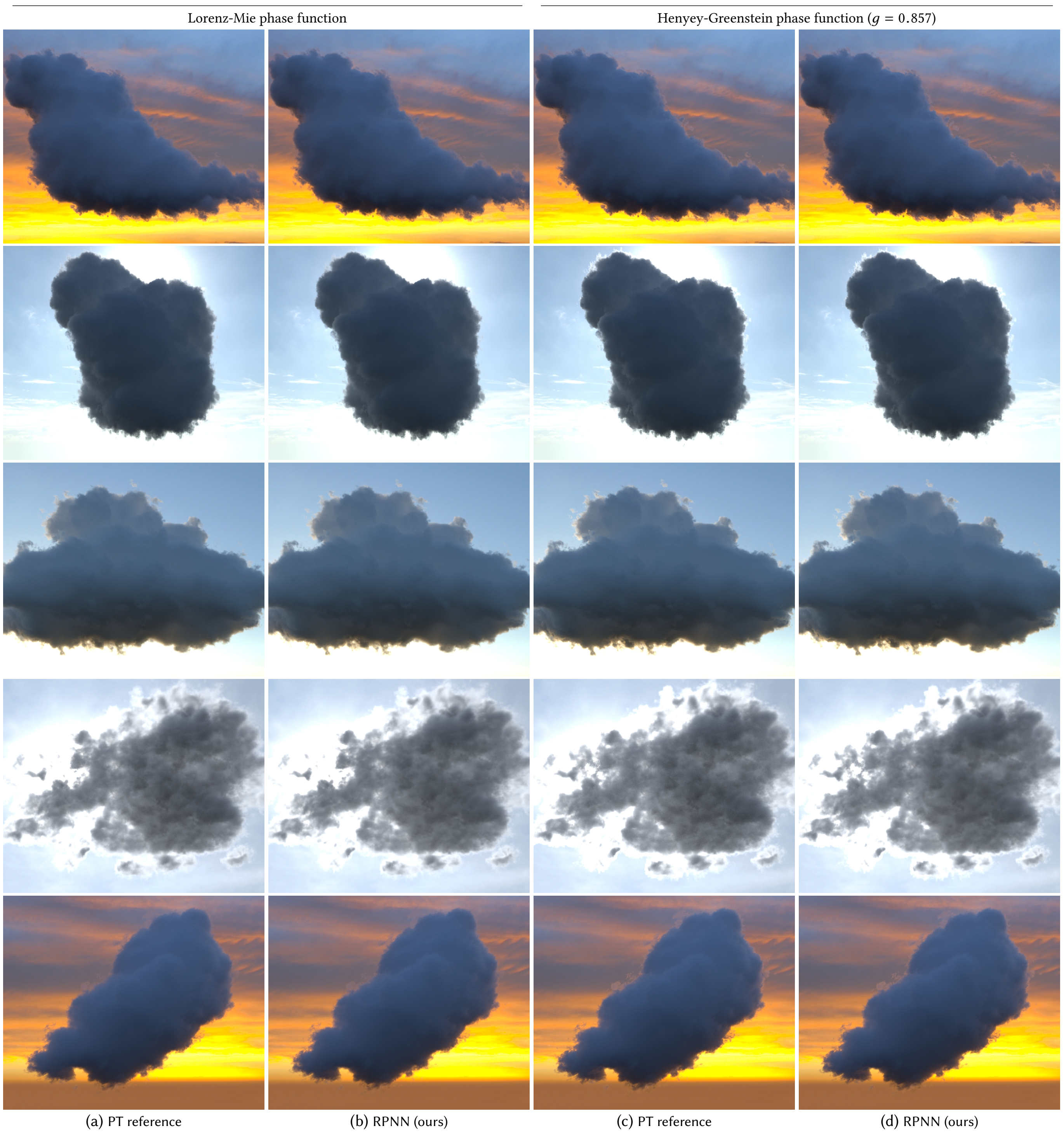
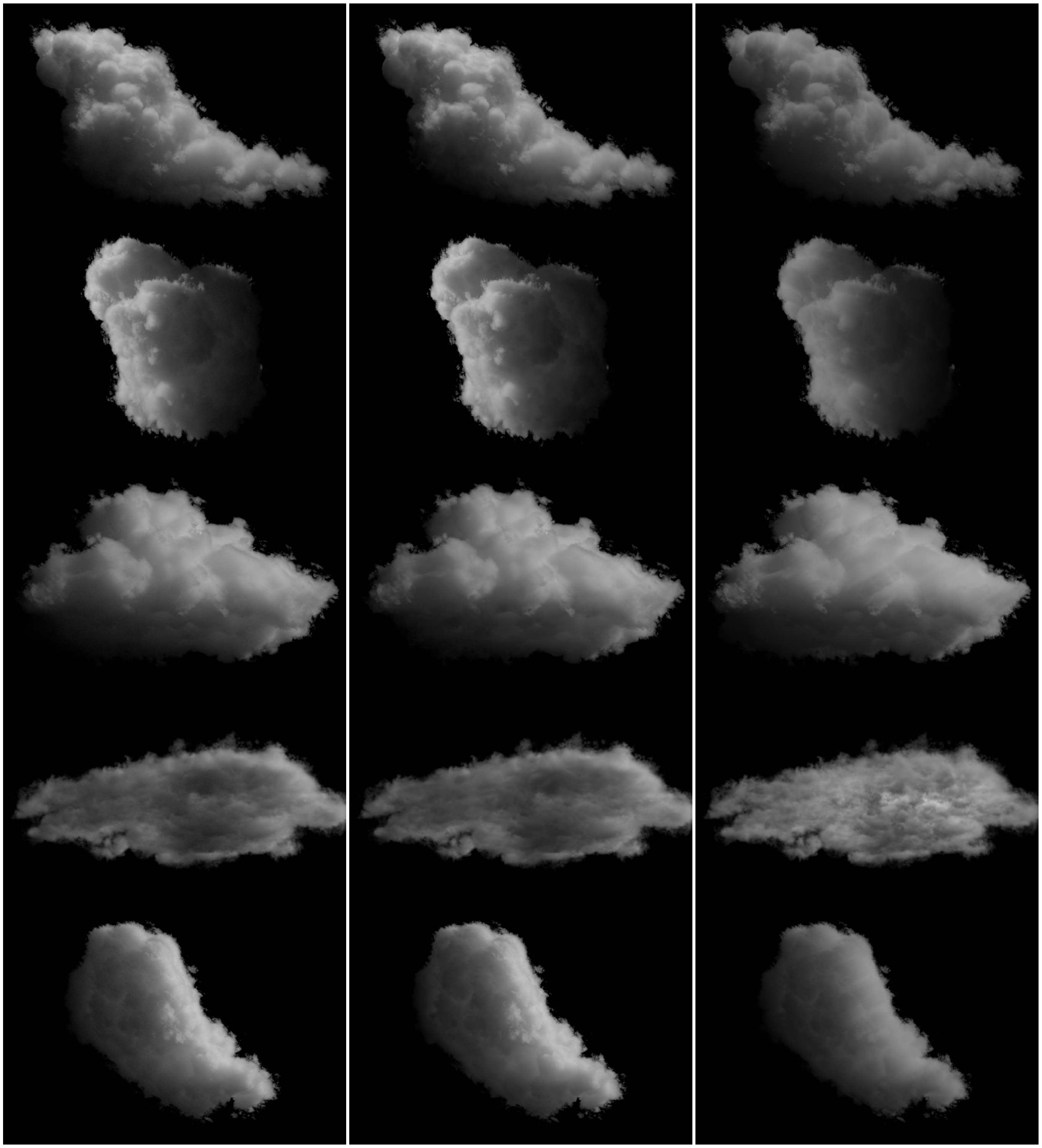

Lorenz-Mie phase function
Henyey-Greenstein phase function ($g = 0.857$)
(a) PT reference
(b) RPNN (ours)
(c) PT reference
(d) RPNN (ours)

Fig. 6. Comparison of a path-traced reference and our radiance-predicting neural network (PRNN) on front-lit clouds illuminated by an environment map.



Lorenz-Mie phase function
Henyey-Greenstein phase function (g = 0.857)
(a) PT reference
(b) RPNN (ours)
(c) PT reference
(d) RPNN (ours)

Fig. 7. Comparison of a path-traced reference and our radiance-predicting neural network (PRNN) on back-lit clouds illuminated by an environment map.



(a) PT reference (b) RPNN (ours) (c) Flux-limited Diffusion

Fig. 8. Comparison of a path-traced reference, our radiance-predicting neural network (PRNN), and flux-limited diffusion (FLD) on side-lit clouds.

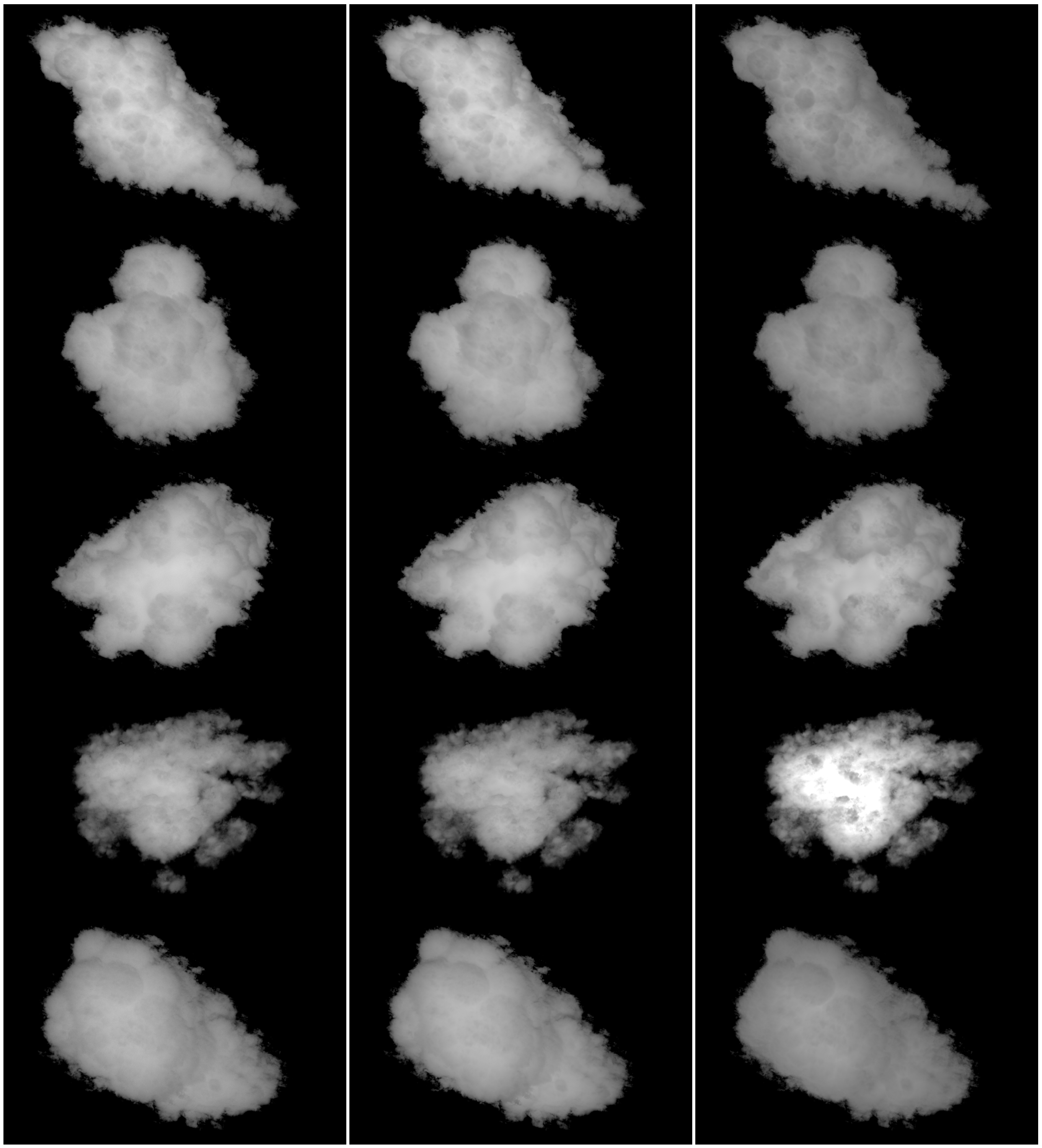


Fig. 9. Comparison of a path-traced reference, our radiance-predicting neural network (PRNN), and flux-limited diffusion (FLD) on front-lit clouds.

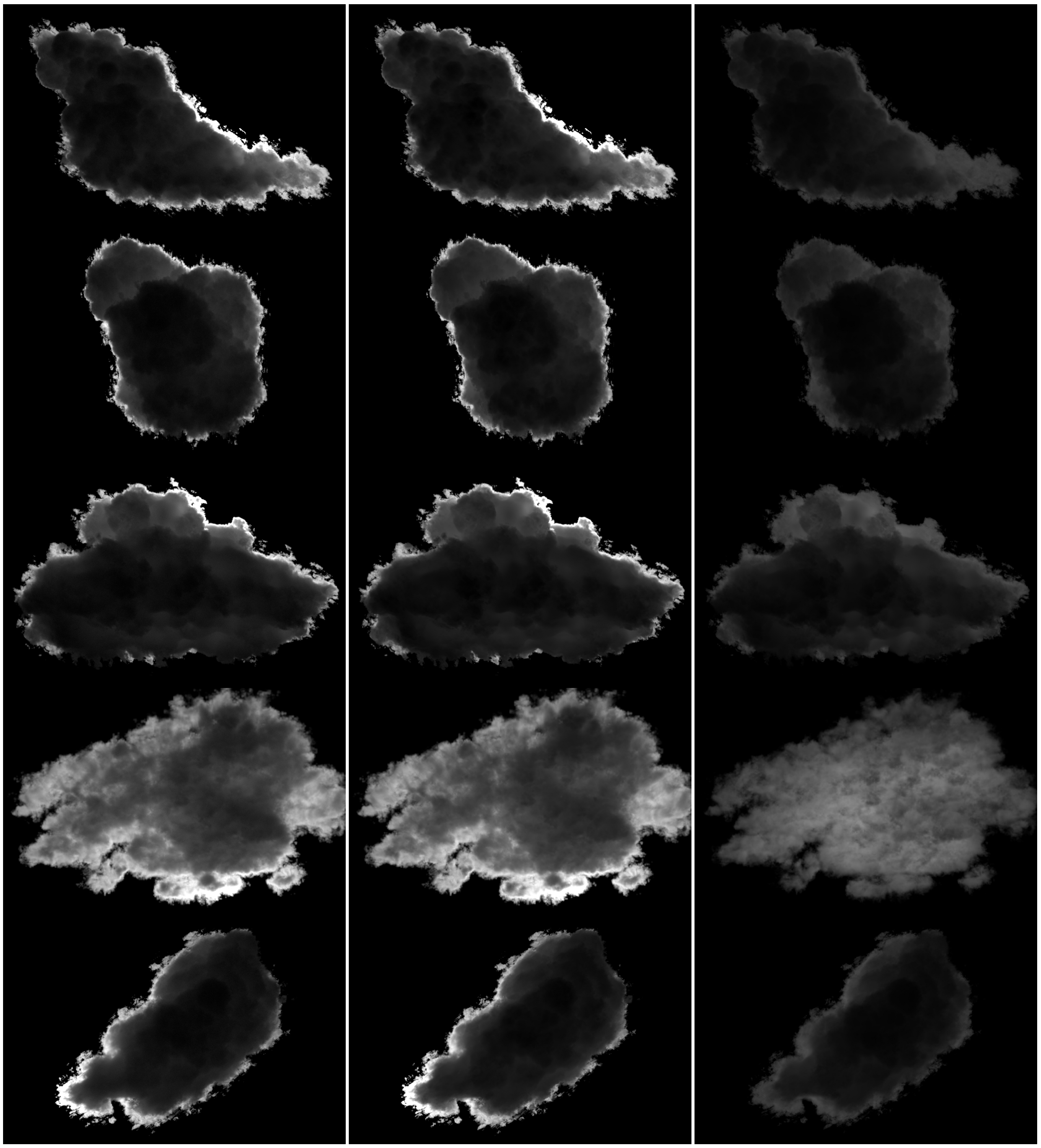

(a) PT reference (b) RPNN (ours) (c) Flux-limited Diffusion

Fig. 10. Comparison of a path-traced reference, our radiance-predicting neural network (PRNN), and flux-limited diffusion (FLD) on back-lit clouds.

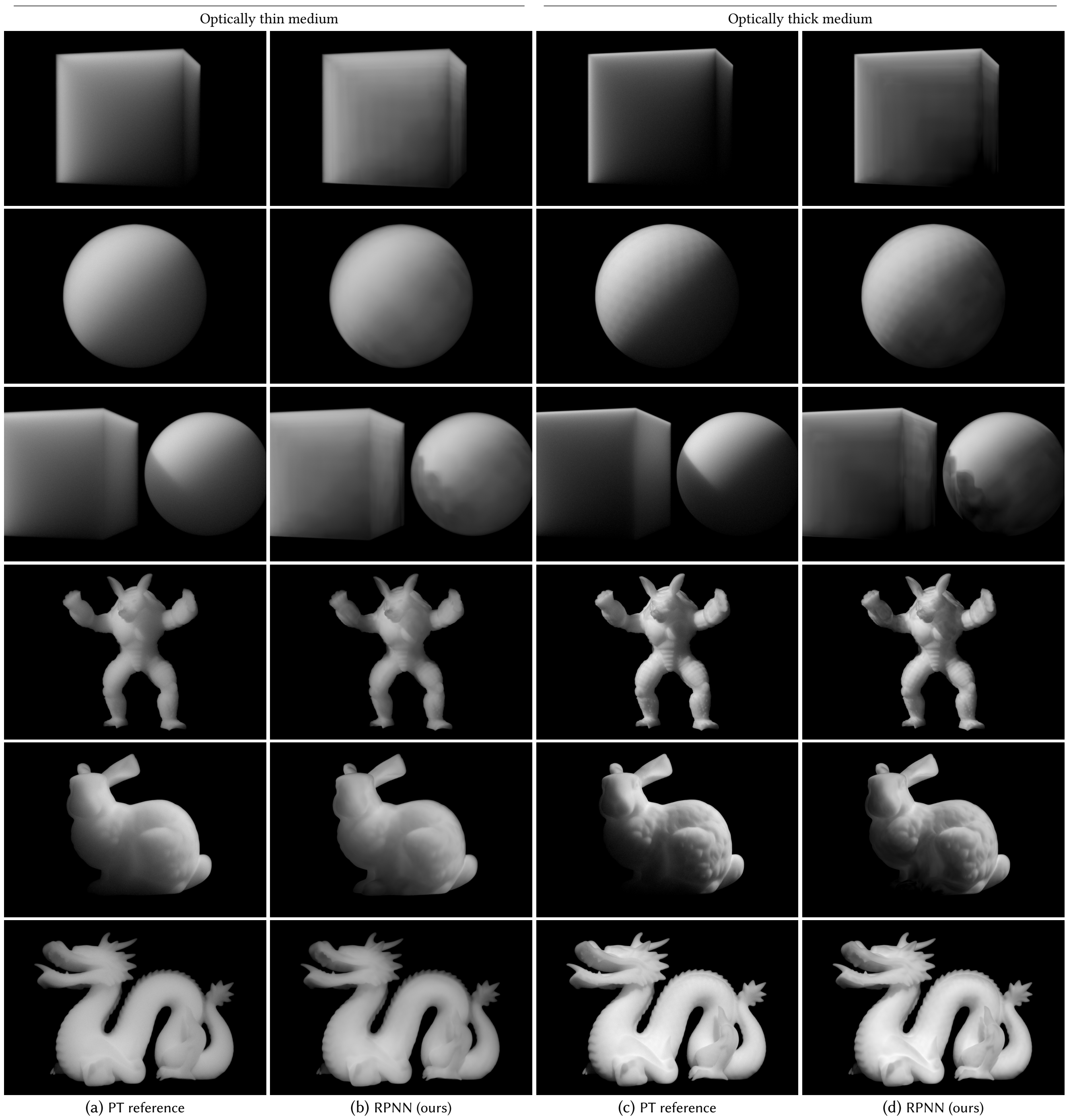


Fig. 11. Comparison of a path-traced reference and our radiance-predicting neural network (PRNN), which was trained on clouds, on side-lit non-cloud shapes filled with an optically thin cloud-like medium (left) and an optically thick cloud-like medium (right).

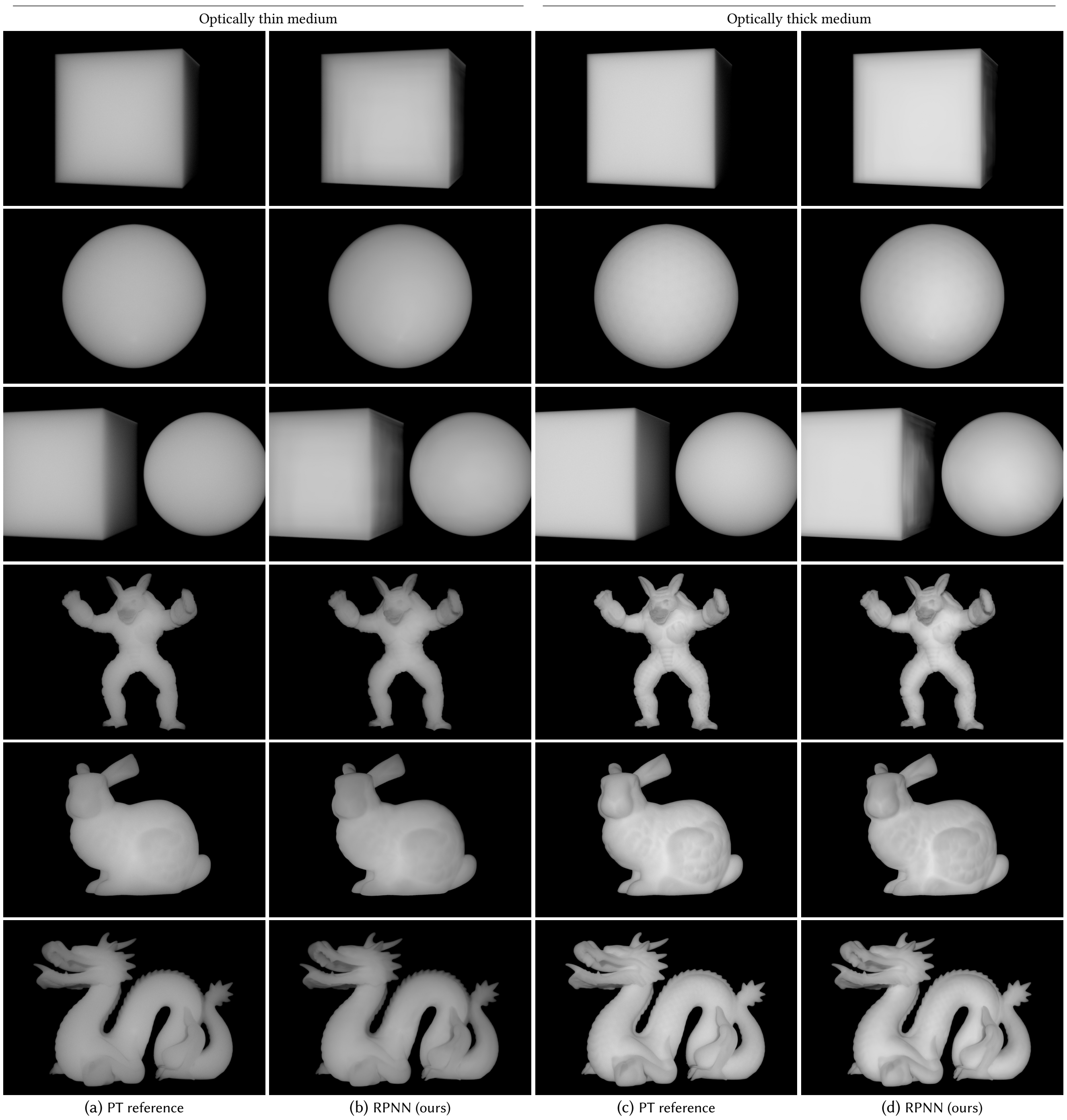


Fig. 12. Comparison of a path-traced reference and our radiance-predicting neural network (PRNN), which was trained on clouds, on front-lit non-cloud shapes filled with an optically thin cloud-like medium (left) and an optically thick cloud-like medium (right).

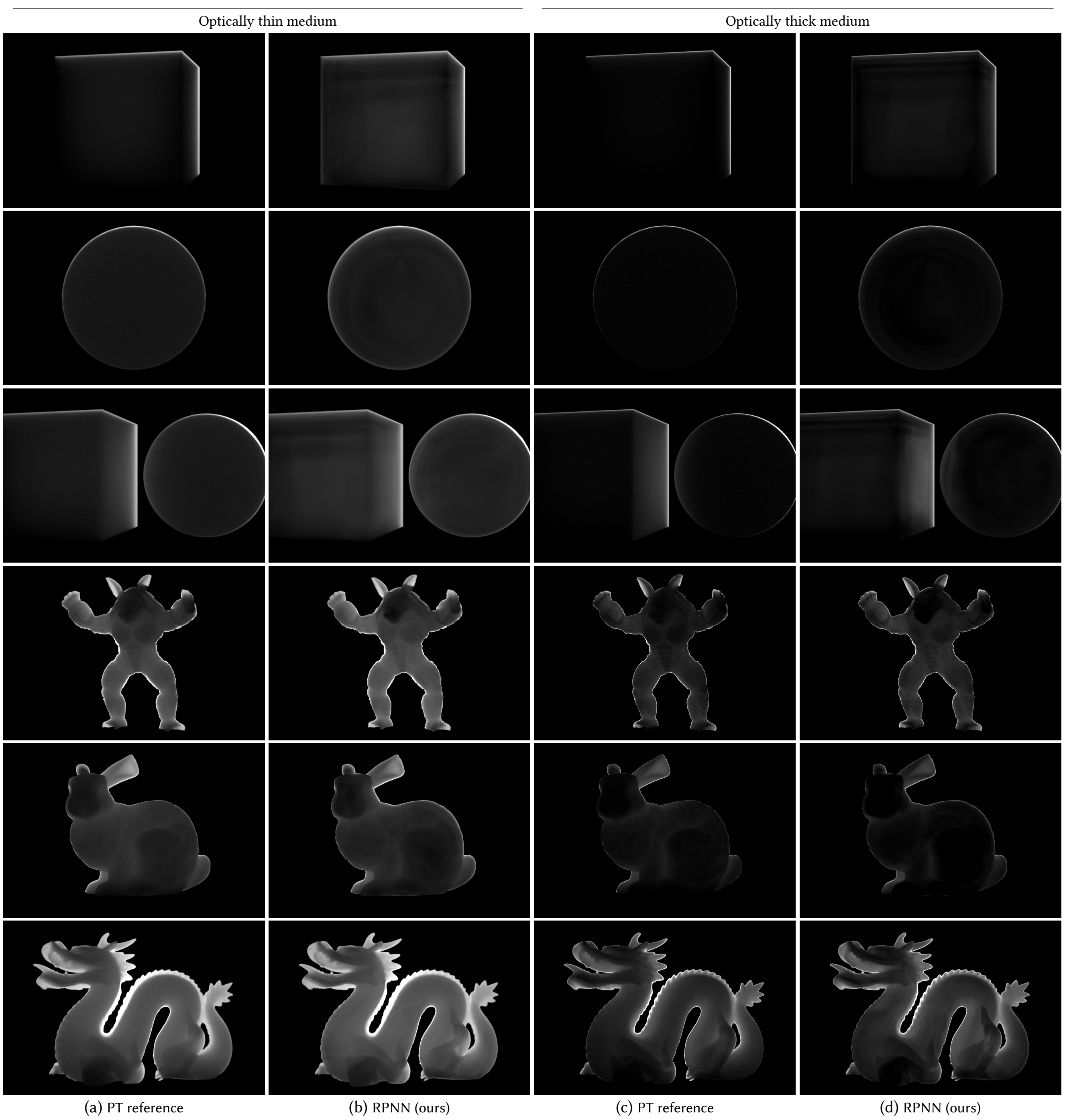


Fig. 13. Comparison of a path-traced reference and our radiance-predicting neural network (PRNN), which was trained on clouds, on back-lit non-cloud shapes filled with an optically thin cloud-like medium (left) and an optically thick cloud-like medium (right).