Light Transport-aware Diffusion Posterior Sampling for Single-View Reconstruction of 3D Volumes

Supplementary Material

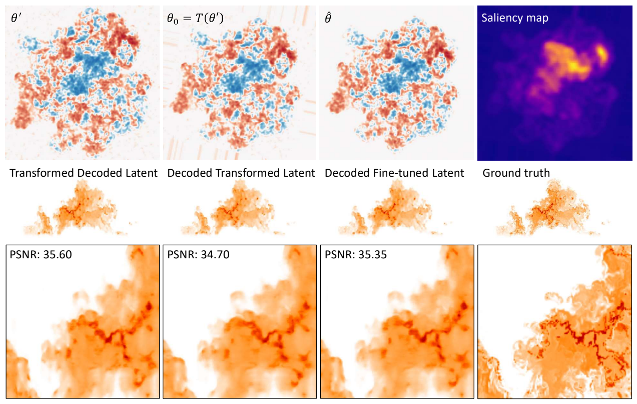

Figure 11. Latent enhancement. Starting with a latent code θ' obtained from the original volume, a transformed version serves as the initial solution θ_0. A few optimization steps are performed to refine the latent representation $\hat{\theta}$, reducing artifacts and enhancing the peak signal-to-noise ratio (PSNR). During optimization, a saliency map derived from θ_0 guides the process by adaptively sampling positions in regions with more prominent features.

6. Enhancing Latent Space

Augmenting the original $1,000$ instances in the Cloudy dataset with additional volumes obtained via transformations requires increasing the encoding time significantly. For example, if encoding $1,000$ clouds requires 2 days on an NVIDIA GeForce RTX 3090, performing a 14-fold multiplication would result in a total computational time of approximately one month.

We leverage the transformation consistency of our monoplanar representation with respect to the xy-plane. The key to reducing the encoding time from 2 minutes to approximately 12 seconds lies in initializing the latent code by applying the desired transformation directly to the original latent representation. Instead of evaluating the representation loss uniformly across all locations, we concentrate sampling in regions where features are most prominent, guided by a distribution derived from a saliency map. This approach uses the features of the initial solution, as the final solutions are expected to remain close to the initialization (see Figure 11).

Another benefit of this refinement is the reduction of patterns that typically emerge from clamping at the domain boundaries when sampling rotated or scaled positions. This helps prevent the generative model from misinterpreting those artifacts as valid structures.

Notation	Description
$\sigma_t(x)$	Extinction field, informally, the density distribution of the particles in the space.
$\varphi(x)$	Scattering albedo: the probability of light to be scattered after a particle interaction.
$\rho(\omega_i, \omega_o)$	Phase function: directional distribution of the scattered light.
$B(\omega)$	Environment radiance coming from ω.
$T(x_a \leftrightarrow x_b)$	Transmittance between two positions.
$L_s(x, \omega)$	Scattered light at x towards ω.
$L_e(x, \omega)$	Emitted light at x towards ω.
$L_i(x, \omega)$	Incoming radiance at x from direction ω.
$L_o(x, \omega)$	Outgoing radiance at surface position x towards direction ω.

Table 3. Terms involved in the volume rendering equation. Notice that all terms are wavelength-dependent.

7. Differentiable Volume Rendering Module

The rendering equation assumes that light travels unchanged between visible surface positions, i.e., the incoming radiance at a point x_a from x_b remains unchanged; $L_i(x_a, \omega) = L_o(x_b, -\omega)$. However, incorporating participating media like clouds requires considering the interactions of light with particles within the volume, due to scattering and/or absorption effects (see Table 3 for the notation used).

7.1. Volume Rendering Equation

The *Volume Rendering Equation (VRE)* computes the incoming radiance $L_i(x_0, \omega)$ by integrating the contributions of scattered and emitted light along a ray, as well as direct contributions from surfaces. It accounts for transmittance (T), scattering properties (σ_t, φ, and ρ), and either volume emission or surface exiting radiance (L_e or L_o).

Given the scattered radiance at x in the direction ω:

$$L_s(x, \omega) = \int_{\omega_i} \rho(-\omega_i, \omega) L_i(x, \omega_i)\, d\omega_i,$$

the incoming radiance at any point in space, including camera sensors, is computed as

$$\begin{aligned} L_i(x_0, \omega) = \int_0^d & T(x_0 \leftrightarrow x_t)\sigma_t(x_t)\left[\varphi(x) L_s(x, -\omega)\right. \\ & \left. + (1 - \varphi(x)) L_e(x, -\omega)\right] dt \\ & + T(x_0 \leftrightarrow x_d) L_o(x_d, -\omega). \end{aligned} \quad (4)$$

The recursive nature of equation 4 is typically addressed using path sampling methods. In the path-based approach, a path $z = x_0, \ldots, x_N$ is sampled, where intermediate vertices correspond to scattering events and the final vertex represents either an absorption event or a surface interaction. The *path throughput* $\Gamma(z)$ captures the cumulative effects of transmittance, densities, scattering albedo, and phase functions along the path. In path-space, the expected radiance is expressed as

$$L_i(x_0, \omega) = \int_z \Gamma(z) E(z) \, dz,$$

where $E(z)$ represents either volume emission (L_e) or outgoing surface radiance (L_o), depending on the final vertex. For simplicity, our analysis considers a single medium surrounded by a "radiative environment shell" that emits radiance inward ($L_o(x, -\omega) = B(\omega)$).

Volumetric path tracing is a standard method for sampling paths proportional to $\Gamma(z)$. However, in its basic form, this approach often experiences high variance due to a mismatch between the path throughput distribution $\Gamma(z)$ and the radiance distribution of the environment. To address this, *next-event estimation* reduces variance by considering direct contributions from the environment at each vertex along the primary path.

7.2. Differentiable Rendering

Let $\mathcal{R}$ be the process of computing the appearance of the volume $\mathcal{D}(\theta)$ subject to physical parameters ϕ, by measuring the arriving radiance L_i to an array of $W \times H$ sensors, i.e.,

$$\mathcal{R}(\mathcal{D}(\theta); \phi) := \{I_k\}_{k=1}^{W \times H}$$

with $I_k = \int_{x_0, \omega} W_e^{(k)}(x_0, \omega) L_i(x_0, \omega) dx_0 d\omega$. Here, x_0, ω represents the incoming ray to the sensor, and $W_e^{(k)}$ is a function that models the sensor's response, typically used to simulate complex lens optics or filter effects. The integral is approximated by averaging multiple samples per pixel, typically 64 in most cases.

Since camera parameters (which could affect W_e or the integral's limits) are not considered, derivatives of $\mathcal{R}$ with respect to its parameters propagate directly through the integral, i.e.:

$$\partial_{\theta\phi} \mathcal{R}(\cdot) = \left\{ \int W_e^{(k)}(x_0, \omega) \partial_{\theta\phi} L_i(x_0, \omega) dx_0 d\omega \right\}_{k=1}^{W \times H}.$$

The propagation of the gradients $\nabla_{\mathcal{R}} \mathcal{L}$ through all volumetric fields requires complex light-path sampling depositing the radiative quantities at every path interactions.

7.3. Differentiable VRE

The propagation of gradients to the argument of an integral operator must adhere to the Leibniz Integral Rule. In this case, the integral limits are independent of the parameters, and there are no discontinuities in the fields. As a result, gradients with respect to L_i can be "propagated" directly to the integral argument. Specifically,

$$\partial_{\theta\phi} L_i(x_0, \omega) = \int_z \partial_{\theta\phi} \left[\Gamma(z) E(z)\right] \, dz.$$

By applying the chain rule, the gradient of the loss function becomes

$$\nabla \mathcal{L} = \int_z \nabla_{L_i} \mathcal{L} \cdot \partial_{\theta\phi} \left[\Gamma(z) E(z)\right] \, dz.$$

This is the idea proposed by Niemier et al. [45], where path sampling is used to "deposit" gradients across all fields involved in the product Γ. In [67], the same z is replayed to compute both Γ and $\partial\Gamma$. A tailored sampler [46] is used to compute $\partial_{\sigma(x_i)}\Gamma$, which becomes problematic when $\sigma(x_i)$ is small. A weighted path sampler [27] includes singular paths with no more than one $\sigma(x_i) = 0$.

Summarizing, using techniques like DRT [46] or SPS [27], gradients with respect to the fields, such as $\partial\mathcal{L}/\partial\sigma(x)$, can be computed. These fields may be represented using various spatial structures, including complex neural models. As long as the representations are differentiable, gradients can propagate to their underlying parameters.

In practice, we use regular grids because they can be efficiently queried and are easily differentiable. If a more complex model is required, such as the volume decoder $\mathcal{D}$, values at the grid vertices are evaluated to obtain the intermediate parameters γ. Then, the gradients $\nabla_\gamma \mathcal{L}$ are backpropagated through the model.

Finally, derivatives of $\mathcal{R}$ with respect to θ and ϕ can be obtained using the differentiable volume renderer, and with this, the gradients of the loss function:

$$\mathcal{L} = \|y - \mathcal{R}(\mathcal{D}(\theta), \phi)\|_2^2,$$

that are required by the Diffusion Posterior Sampling and the OPTIMIZATION method. In Fig. 12 we show some examples of the joint reconstruction of physical parameters ϕ (environment map) and density distributions of the cloud determined by θ with our proposed technique.

8. Parameterized Diffusion Posterior Sampling

Algorithm 2 outlines the adapted DPS method tailored for our parameterized posterior sampling approach. Here, α_t denotes the noise scheduling parameter at time step t. In practice, we sample only 100 time steps with a stride of 10, rather than sampling all steps. This adjustment also impacts the scaling factor ζ_t, which is proportionally amplified.

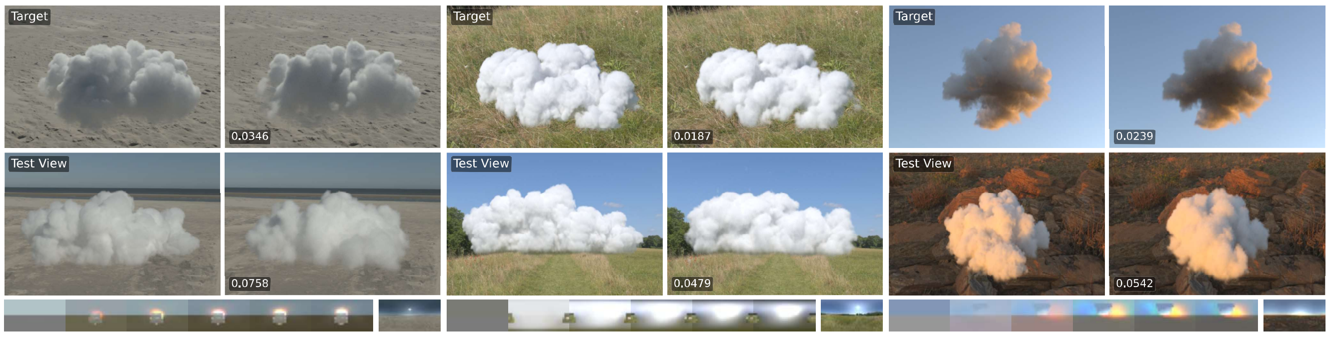

Figure 12. Additional results for reconstructions of both, cloud and lighting conditions, varying the material settings of the cloud and targeting different environment maps.

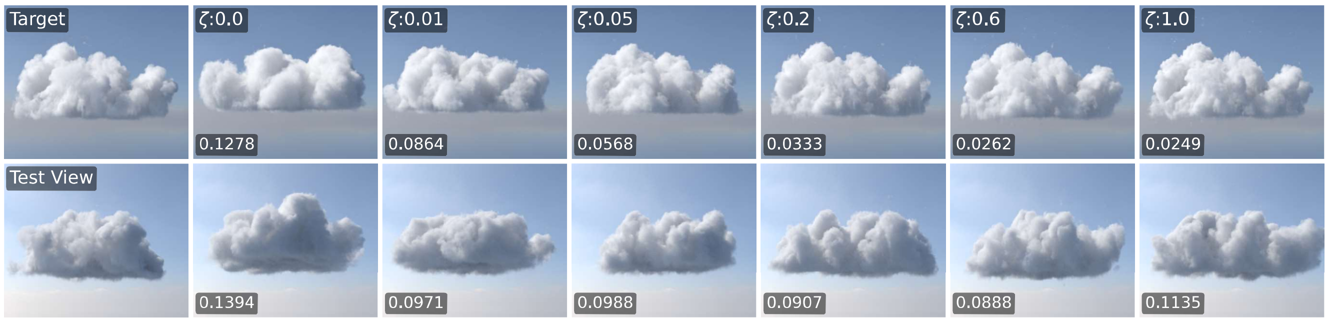

Figure 13. Effect of ζ: Multiple DPS runs were performed with varying values of the ζ multiplier. The top row shows the reconstruction's approximation to the target view, while the bottom row presents the reconstruction from a different perspective. Higher ζ values lead to better alignment with the observation but deviate from the prior, resulting in less cloud-like formations. In contrast, smaller ζ values remain closer to the cloudy prior but exhibit weaker alignment with the observation.

Algorithm 2 Parameterized DPS

```
Require:
  $y, \mathcal{R}, \mathcal{D}, \phi$
  $\theta_k, k$                                   ▷ Start noisy version
Ensure:
  $\theta \sim p(\theta \,|\, y; \phi)$

  for $t = k \dots 1$ do
    $\boldsymbol{\epsilon} \leftarrow \epsilon_\Phi(\theta_t, t)$
    $\hat{\theta}_0 \leftarrow \left(\theta_t - \sqrt{1-\alpha_t}\boldsymbol{\epsilon}\right) / \sqrt{\alpha_t}$
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
                                                  ▷ DDIM step
    $\theta'_{t-1} \leftarrow \sqrt{\alpha_{t-1}}\hat{\theta}_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\boldsymbol{\epsilon} + \sigma_t \mathbf{z}$
                                                  ▷ DPS step
    $\theta_{t-1} \leftarrow \theta'_{t-1} - \zeta_t \nabla_{\theta_t} \|y - \mathcal{R}(\mathcal{D}(\hat{\theta}_0), \phi)\|_2^2$
  return $\hat{\theta}_0$
```

8.1. Influence of ζ in DPS

During diffusion posterior sampling, the gradients' scaling factor that guides the state toward the observation plays a crucial role in balancing the trade-off between prior enforcement and observation fidelity. The authors of [9] proposed the following formulation:

$$\zeta_t = \frac{\zeta}{\|y - \mathcal{A}(\hat{x}_0(x_t))\|},$$

where the hyperparameter ζ is chosen within the range $[0.1, 1.0]$. Figure 13 illustrates how this choice impacts reconstruction accuracy and adherence to the prior.

9. Common diffusion-base tasks

In this section, we present several applications of our proposed generative model and the parameterized diffusion posterior sampling technique, demonstrating their effectiveness across a variety of tasks. These applications highlight the versatility and power of our approach in addressing different challenges within the domain of volumetric scene reconstruction and rendering.

9.1. Generative model

One notable property of our proposed DDPM is its ability to generate new clouds. The generated clouds look similar to the original clouds in Cloudy, and their internal struc-

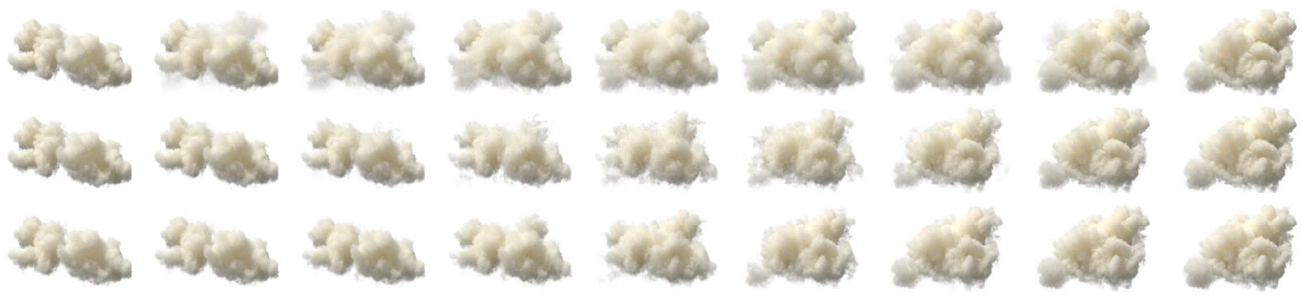

Figure 14. Cloud Interpolation. Top row: linear interpolation between grids, showing a straightforward blending of two cloud structures. Middle row: Linear interpolation between latent representations, offering smoother transitions compared to direct grid interpolation, but still revealing limitations such as ghosting effects. Bottom row: DPS (Diffusion Posterior Sampling) using the linear interpolation in latent space as the target, resulting in more coherent and natural transitions, with the prior enforced to avoid artifacts like ghosting.

ture closely resembles that of a physical simulation. This is demonstrated in Fig. 4 in the main document.

Interpolation: Interestingly, linear interpolation in the cloud's latent space—i.e., between different latent representations—produces plausible transitions between cloud shapes. However, when the cloud distributions differ significantly in terms of lobes or fine elongations, ghosting effects may occur as structures fade out linearly.

To address this issue, we propose an interpolation method based on posterior sampling: The mixture in the latent representation serves as the target, defined as $y := (1-\alpha)\theta_a + \alpha\theta_b$, where θ_a and θ_b are the latent representations of two different clouds, and α controls the blending factor. This method ensures smoother transitions by taking the cloud structure into account during the interpolation process, and enforcing the prior to prevent the appearance of ghost artifacts. By integrating posterior sampling, the model adapts to the natural distribution of clouds, resulting in more physically consistent transitions.

Figure 14 showcases the differences between the linear interpolation strategy and our proposed method, highlighting the improved transitions and the reduction of ghosting effects in complex cloud distributions.

9.2. Super-resolution and In-painting

Super-resolution and in-painting are common use cases in image restoration with diffusion models. These tasks are particularly well-suited for diffusers because the denoiser can easily preserve parts of the existing signal while filling in missing or low-resolution regions with consistent and coherent information. The diffusion process naturally integrates prior knowledge, making it effective at reconstructing fine details and completing structures in a visually plausible manner.

For the case of super-resolution, our measurement function is $\mathcal{A}(\theta) := \mathcal{C}(\mathcal{D}(\theta))$, where $\mathcal{C}$ is a coarse jittered sampling of the decoded grid $\mathcal{D}$. In the case of in-painting, we assume a mask of interest M and consider $\mathcal{A}(\theta) := M \otimes \mathcal{D}(\theta)$.

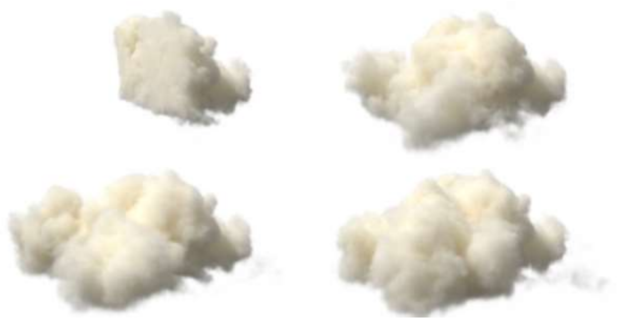

Figure 15. Cloud Inpainting. The diffuser is employed to generate a cloud that is consistent with a visible portion of the cloud. Three different instances are generated and displayed, demonstrating the model's ability to generalize and create diverse cloud formations, each unique yet adhering to the visible parts provided.

Figures 7 and 15 demonstrate the performance of our diffuser on super-resolution and in-painting tasks respectively. While these tasks are typically linear in explicit cases, we continue to use Diffusion Posterior Sampling (DPS) due to the non-linearity of our latent decoder. This non-linearity complicates the optimization, and therefore approaching the solution at x_t to satisfy $y = \mathcal{A}(x_0(x_t))$ requires careful computation of the gradients with respect to x_t.

10. Extended comparisons

Fig. 16 shows visual examples from the 32 test cases.

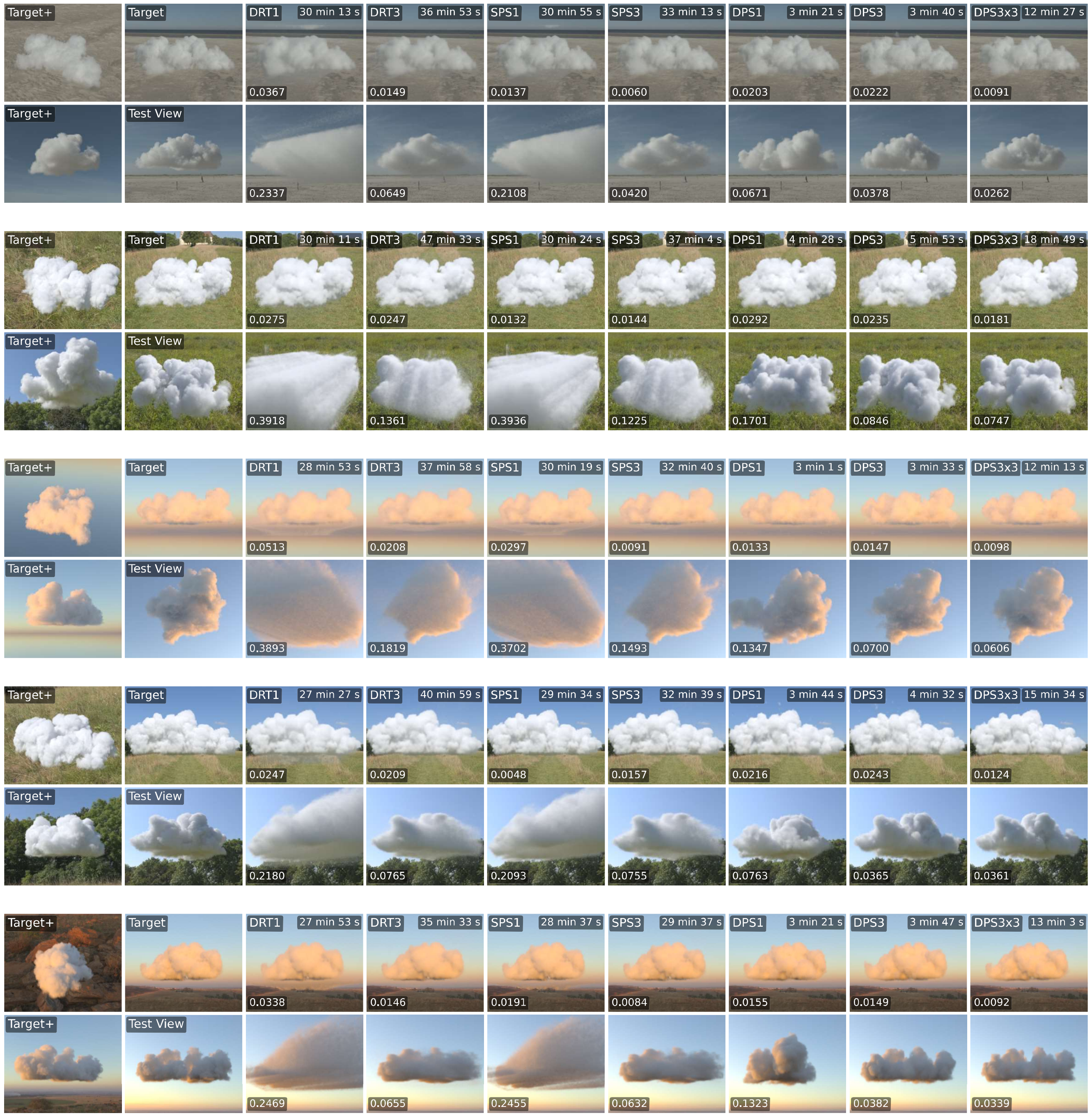

Figure 16. Further comparisons between different reconstruction techniques for single- and sparse-view settings.