Spherical Voronoi
Directional Appearance as a Differentiable Partition of the Sphere

Francesco Di Sario[1,2] Daniel Rebain[3] Dor Verbin[5]
Marco Grangetto[1] Andrea Tagliasacchi[2,4]

[1]University of Torino [2]Simon Fraser University [3]University of British Columbia
[4]University of Toronto [5]Google DeepMind

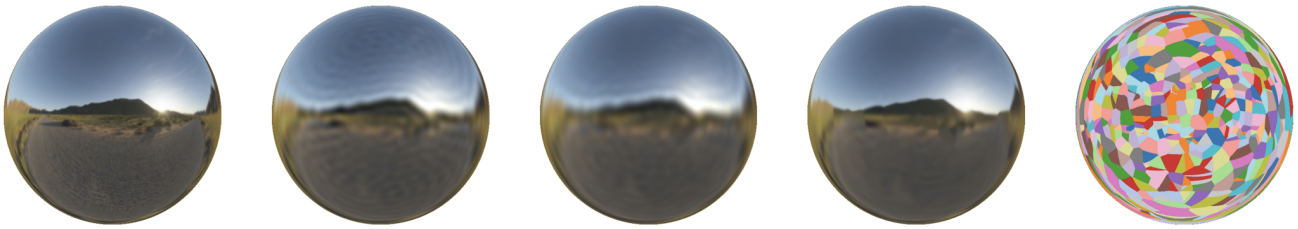


Figure 1. Spherical functions like the shown environment maps have a multitude of applications in Computer Graphics and 3D Computer Vision. Classical representations like Spherical Harmonics optimize well but struggle in representing high-frequency functions. Explicit representations like Spherical Gaussians are capable of representing localized functions, but they are difficult to optimize due to the locality of the Gaussian kernel. We propose Spherical Voronoi as a new explicit representation that is capable of modeling *high frequencies* effectively, provides an *adaptive* decomposition of the spherical domain, and is *easier to optimize*.

Abstract

Radiance field methods (e.g. 3D Gaussian Splatting) have emerged as a powerful paradigm for novel view synthesis, yet their appearance modeling often relies on Spherical Harmonics (SH), which impose fundamental limitations. SH struggle with high-frequency signals, exhibit Gibbs ringing artifacts, and fail to capture specular reflections—a key component of realistic rendering. Although alternatives like spherical Gaussians offer improvements, they add significant optimization complexity. We propose Spherical Voronoi (SV) as a unified framework for appearance representation in 3D Gaussian Splatting. SV partitions the directional domain into learnable regions with smooth boundaries, providing an intuitive and stable parameterization for view-dependent effects. For diffuse appearance, SV achieves competitive results while keeping optimization simpler than existing alternatives. For reflections—where SH fail—we leverage SV as learnable reflection probes, taking reflected directions as input following principles from classical graphics. This formulation attains state-of-the-art results on synthetic and real-world datasets, demonstrating that SV offers a principled, efficient, and general solution for appearance modeling in explicit 3D representations.

1. Introduction

In recent years, novel view synthesis has seen remarkable progress, driven largely by radiance field-based methods. However, while current state-of-the-art approaches effectively capture fine-grained texture and intricate geometric detail, they struggle to reproduce the complex, view-dependent appearance inherent to glossy surfaces. This fact, along with the plateau in photometric accuracy at the top of novel view synthesis benchmarks [12], clearly indicates that further improvement will require fundamental changes in how the *directional* component of radiance fields is modeled. The view-dependent appearance representation of NeRF [19] employs a large multi-layer perceptron (MLP) to model the emitted radiance at each 3D point and viewing direction. While this formulation achieves high-fidelity novel view synthesis, it remains computationally expensive, limiting its applicability to real-time scenarios. For example, Zip-NeRF [2] reports rendering speeds on the order of one frame per second even on high-end GPUs. Many efforts in computer graphics have been made to render glossy reflections accurately and efficiently. Inspired by these approaches, 3D reconstruction methods like Ref-NeRF [25] address these shortcomings by introducing more principled models of light transport into the render-

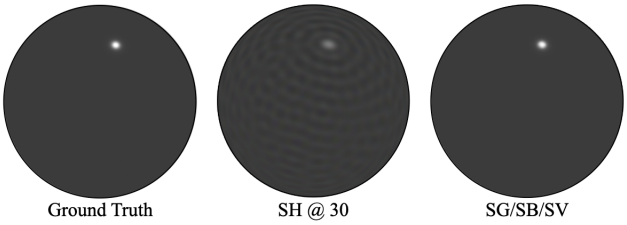


Figure 2. **Gibbs artifacts** – Representing a "glint" on a shiny sphere like shown here is computationally impractical for the popular Spherical Harmonics representation. Many coefficients (30 here) have to be used to reconstruct high-frequency functions accurately, and when this happens, Gibbs artifacts appear.

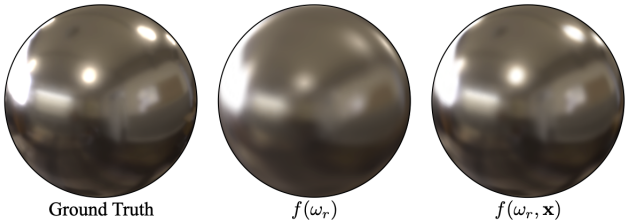


Figure 3. **Spatially varying radiance** – When illumination is spatially varying but a spatially-invariant model is used, conflicting measurements are averaged, resulting in a blurry reconstruction (middle). Conversely, when a spatially varying model is used, crisp functions can be recovered.

ing model. These models are particularly successful at rendering the common case of light reflecting off an opaque surface. Such models take advantage of the fact that incident radiance changes slowly with position. This is because if the illumination originates sufficiently far from the rest of the scene, assuming constant incident radiance at every point in space is an excellent approximation, and modeling reflections by means of a spherical environment map becomes extremely effective [31]. While providing high quality renderings, NeRF-based models can be quite slow to render, as they require many expensive MLP evaluations for capturing and rendering specular highlights. Gaussian splatting [9] and other approaches which focus on rendering speed [3, 4, 18] represent view-dependent appearance using spherical harmonics—the generalization of Fourier series to the spherical domain—which are very cheap to evaluate and therefore enable efficient inference. When only a sparse set of viewing directions is provided as input, the spherical harmonics are also easier to optimize thanks to the smoothness they induce in the loss landscape, and thanks to their global support over the spherical domain. However, despite these advantages, spherical harmonics come with the significant limitation of only allowing band-limited reconstructions. The number of coefficients required to represent a given frequency grows quadratically with the frequency, making it difficult to render accurate representation of sharp view-dependent effects like specular highlights (see Figure 2). Common approaches to rendering glossy reflections use spherical representation of distant illuminants like spherical environment maps [5, 17, 21, 26, 29], or alternatively tracing secondary rays through the scene to also take into account near-field light sources and inter-reflections [20, 27, 29]. Both of these approaches have proven to produce high-quality reflection modeling for glossy surfaces. However, they also create optimization challenges, as reflection computation is highly sensitive to potential errors in the estimated geometry due to the strong dependence of the outgoing radiance on the surface normal. These optimization difficulties often cause such approaches to be outperformed by methods that do not explicitly model reflections when specular highlights are absent. While bridging the gap between lighting decomposition and direct appearance memorization in complex scenes remains an open challenge, we take a step toward unifying these paradigms through a spherical function parameterization that effectively models both the view-dependent component of a traditional 5D radiance field (*i.e.*, extending approaches similar to Zip-NeRF [2]) and the environment map component of a decomposed model (*i.e.*, extending approaches similar to Ref-NeRF [25]). More specifically, taking inspiration from recent work on differentiable Voronoi diagrams [4, 23, 28], we propose *Spherical Voronoi* function approximators, which learn adaptive decompositions of the sphere into cells - simultaneously enabling recovery of high frequency effects like specular highlights, and supporting stable optimization under sparse supervision. Our proposed representation also achieves a better trade-off between parameter budget and reconstruction accuracy on real-world data, compared to alternatives like spherical harmonics and spherical Gaussians.

2. Related Works

NeRFs [19] represent scenes as continuous volumetric functions outputting density and view-dependent color, but originally required long training and struggled with specular surfaces. Kerbl et al. [9] replaced neural fields with millions of anisotropic Gaussians and fast rasterization, achieving real-time, high-quality view synthesis.

Modeling view-dependent effects. NeRF's MLP-based view dependent appearance inspired alternatives that model view-dependent lighting more explicitly. Fridovich-Keil et al. [3] replaces the MLP with a sparse voxel grid storing spherical harmonics (SH) per cell, achieving NeRF-quality results with a significant speed-up. In Gaussian splatting, early methods also used low-order SH, but richer bases have emerged. Niemeyer et al. [22] uses spherical Gaussians and a NeRF-trained prior for robust, ultra-fast optimiza-

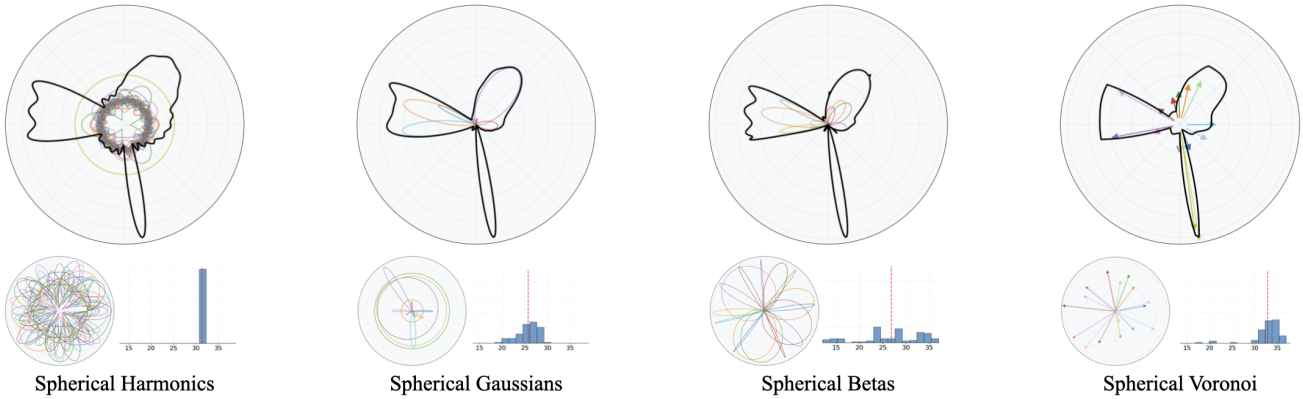


Figure 4. **Fitting functions on spherical domains (2D)** – We fit different spherical representations to the same functions starting from random initializations (samples shown in the bottom-left plots). To measure the robustness of optimization we repeat the fitting 100 times from random initializations, and report the PSNR attained at the end of optimization (bottom-right plots). Spherical Harmonics converge to a unique solution due to the orthogonality of its basis functions, but its band-limited characteristics result in Gibbs fitting artifacts. Spherical Gaussian and Betas can represent locally supported functions, as well as discontinuities, but their optimization is not stable and prone to local minima. Spherical Voronoi *consistently* converges to a *better* reconstruction. We omit drawing the ground truth function, as it is practically identical to the one reconstructed by Spherical Voronoi.

tion, while Liu et al. [14] replaces Gaussians with bounded spherical Beta kernels, capturing sharper highlights without requiring normals. Lu et al. [16] uses anchor Gaussians and an MLP to generate view-dependent attributes per frame, reducing primitive count while accurately modeling reflections and transparency.

Modeling reflections (NeRF). Rendering specular reflections remains a major challenge for radiance fields. Verbin et al. [25] addresses this by factorizing radiance with surface properties and regularizing normals for smoother highlights. Verbin et al. [27] and Wu et al. [29] improve view-consistent reflections by tracing secondary rays through the field. Liu et al. [15] jointly optimizes geometry and BRDF under estimated lighting, while Liang et al. [13] uses a learned neural renderer with an SDF-based model to simulate reflections and enable relighting.

Modeling reflections (3DGS). Gaussian splatting has been extended to handle reflections through methods, like GaussianShader [8], which adds a simplified shading model to each splat by storing diffuse and specular terms, along with a learned residual to capture higher-order effects. It estimates per-splat normals via ellipsoid axes and learned corrections, enforcing consistency with rendered geometry. This improves reflective realism while preserving real-time speed. [30] complements this with deferred shading: splats are rendered to G-buffers, and a second pass computes reflections using estimated normals. By back-propagating reflection errors and refining normals across splats, it achieves sharper specular effects with minimal performance cost. More recently, work has focused on factorizing lighting and reflectance in splatting. Tang and Cham [24] introduces a learned local illumination field via low-rank tensor factorization and assigns each Gaussian a BRDF descriptor, enabling accurate view-dependent effects while keeping training and rendering efficient. Zhang et al. [32] applies directional light factorization in screen space, combining a multi-scale spherical mip-map for roughness-aware reflection blur with a geometry-lighting factorization. This achieves photorealistic, smooth reflections in complex scenes while maintaining interactive performance and accurate geometry.

3. Method

We overview classical representations for spherical functions in Section 3.1, and present our Spherical Voronoi representation in Section 3.2. We then apply Spherical Voronoi to two core applications in radiance fields applications: (i) fitting in a radiance field (Section 3.3), and (ii) modeling of reflections (Section 3.4).

3.1. Background

Functions defined on the sphere $f : \mathbb{S}^2 \to \mathbb{R}^C$ can be represented using a variety of spherical bases. In what follows, and without loss of generality, we only consider scalar functions $C{=}1$. To contextualize our discussion, we first review the representations that are typically found in the differentiable rendering literature.

Spherical Harmonics (SH). A widely adopted choice is the

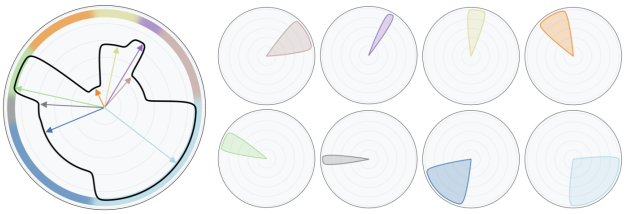


Figure 5. **Spherical Voronoi Bases** – (left) a spherical function is represented by a collection of Voronoi sites. (right) we visualize the support associated with each Voronoi site, and highlight how when combined they span the entire domain.

Figure 6. **Effect of temperature** τ – With a low value of τ the representation is easy to optimize as its degrees of freedom have a large support, while if *sharp* discontinuities are needed, a large value of τ can be chosen; above $\tau = \{1, 5, 25\}$.

Spherical Harmonics expansion:

$$f_{\text{SH}}(\omega; c) = \sum_{l=0}^{L} \sum_{m=-l}^{l} c_{lm} Y_l^m(\omega), \tag{1}$$

where $c_{lm} \in \mathbb{R}$ are optimizable coefficients and Y_l^m are the (real) SH basis functions. Optimizing SH is numerically stable: they form an orthonormal basis, and they are globally supported. However, capturing sharply localized signals (*i.e.*, high-frequencies) requires a large number of basis functions (large L), leading to both large parameter counts, and the introduction of Gibbs-like ringing artifact when representing discontinuities in the function; see Figure 2 for an example.

Spherical Gaussians (SG). To better model localized signals, a linear combination of Spherical Gaussians can be employed:

$$f_{\text{SG}}(\omega; \tau, s, c) = \sum_{k=1}^{K} c_k \exp\left(\tau_k (s_k \cdot \omega - 1)\right), \tag{2}$$

where each lobe is defined by a mean direction s_k, concentration τ_k, and amplitude c_k. SGs describe the function as a combination of smooth and rotationally symmetric lobes. However, as Gaussians (in practice) are compactly supported, the mixture of SG lobes is often sensitive to initialization, and unstable to optimize; see the histograms in Figure 4. This leads to weak gradients when a lobe is misaligned, and to ill-conditioned updates for large concentration parameters τ_k.

Spherical Betas (SB). Spherical Betas extend SG by enabling asymmetric and bounded-support lobes defined as:

$$f_{\text{SB}}(\omega; \alpha, \beta, s) = \sum_{k=1}^{K} (1 + s_k \cdot \omega)^{\alpha_k - 1} (1 - s_k \cdot \omega)^{\beta_k - 1}, \tag{3}$$

where s_k is the principal direction and $\alpha_k, \beta_k > 0$ control the shape. These are more flexible than SGs, as they can model skewed and sharper directional variation. However, this flexibility also makes them harder to optimize in practice. Their contribution remains local, and extreme values of α_k or β_k generate very sharp and flat regions that cause ill-conditioned gradients, and even stronger sensitivity to initialization than spherical Gaussians.

3.2. Spherical Voronoi (SV)

To avoid these limitations, we introduce a new (*soft*) Spherical Voronoi representation. Our representation consists of a set of directional sites $s_1, ..., s_K \in \mathbb{S}^2$ and associated function values $c_1, ..., c_K \in \mathbb{R}$. The function evaluation in direction ω is obtained as a weighted combination of the per-site values:

$$f_{\text{SV}}(\omega; \tau, s, c) = \sum_{k=1}^{K} w_k(\omega; \tau_k) c_k, \tag{4}$$

where the weight w_k is computed using a softmax function:

$$w_k(\omega; \tau) = \frac{\exp(\tau_k s_k \cdot \omega)}{\sum_{k'=1}^{K} \exp(\tau_k s_{k'} \cdot \omega)}. \tag{5}$$

The *temperature* parameters $\tau_k > 0$ control the sharpness of the partition: small values produce smooth color transitions, while large ones approach a hard Voronoi tessellation, as illustrated in Figure 6. When all τ_k share the same value, the formulation corresponds to the standard soft Spherical Voronoi model, whereas assigning distinct temperatures to each site naturally extends it to a weighted variant with locally adaptive angular sharpness. By adjusting τ, the model can effectively represent both smooth and sharp signals, typical in directionally localized structures. This formulation works well in practice because the softmax ensures well-defined gradients for all sites, the temperature provides a way to adjust the sharpness of the representation, and the resulting soft partition induces a clean, non-overlapping decomposition of the sphere that avoids the "competition" between overlapping kernels which representations like SG and SB exhibit. Please refer to Figure 4 for an analysis of approximation and convergence behavior.

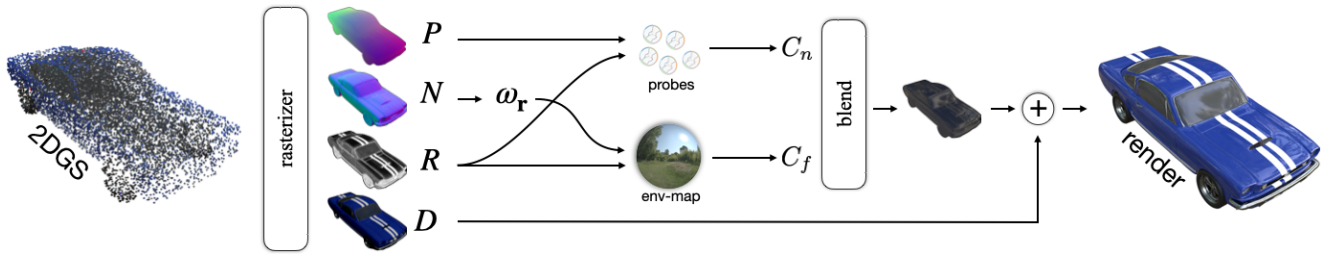


Figure 7. **Deferred rendering pipeline** – The 2DGS scene is rasterized into buffers of (positions, normals, roughness, diffuse color). These are then used in the lighting pass, combining light-probe illumination and an environment cubemap to compute diffuse and specular shading, which are blended to produce the final output.

3.3. View-direction parameterization

For classical radiance fields training, we replace the view-directional representation with our Spherical Voronoi. Given a view-direction ω, the radiance associated with the primitive is evaluated as $f_{\text{SV}}(\omega;\tau,s,c) : \mathbb{S}^2 \to \mathbb{R}^3$, as defined in Equation (4), where the co-domain of the function is the RGB view-dependent color. For 3DGS, that means that *each* Gaussian will be equipped with additional parameters $\big(\{\tau_k\},\{s_k\},\{c_k\},\big)$, and all of these parameters are treated as learnable and jointly optimized.

3.4. Reflection-based Parameterization

Verbin et al. [25] showed how, for glossy surfaces, when the view direction is used to query emitted radiance, a complex function ought to be learned from (relatively) sparse measurements. Instead, they propose to co-learn the normals n of surfaces, and reason in terms of a simpler function – the radiance measured along the *reflected* view direction $\omega_r = 2\,(\omega \cdot n)\,n - \omega$. Spherical Voronoi functions are sufficiently expressive to model highly complex directional behaviors, including multi-modal and discontinuous functions, and particularly well-suited to represent sharp illumination lobes (*e.g.*, a specular highlight / glint on a surface). This makes them appropriate not only for representing smooth view-dependent effects $f(\omega)$, but also for learning fine-grained specular reflections $f(\omega_r)$ in the spirit of Verbin et al. [25].

Spatially varying incoming radiance. Relying exclusively on $f(\omega_r)$ implicitly assumes *far-field* illumination, which, in many real-world scenarios, is an incorrect model. This becomes a problem when a glossy object is in proximity of other objects or light sources, and the appearance becomes a function of not just direction, but also position; see Figure 3. To account for spatial changes in the reflected light field, Verbin et al. [25] conditioned the neural field on *both* reflected direction and position. However, when we parameterize the 3D scene with explicit representations like Gaussian splats, this becomes much more difficult to achieve (vs. conditioning by concatenation in neural fields).

Learnable light probes. To overcome this limitation, we introduce *learnable light probes*, a set of learnable probes placed throughout the scene, each encoding a local reflection field. While we believe we are the first to introduce the concept of learnable light probes in differentiable rendering, we do not claim to have invented them. In fact, light probes are commonly used in real-time rendering engines to cache illumination at discrete spatial positions, so to query illumination efficiently. Each probe stores a compact representation of the incoming radiance, enabling rendering systems to approximate spatially varying lighting by interpolating contributions from nearby probes.

Deferred Rendering with Voronoi Light Probes. Following [30, 32], we adopt a deferred rendering strategy that separates geometry from illumination. We build our framework upon the 2DGS backbone [7], and extend each primitive with two additional learnable material parameters: a roughness value $r \in [0,1]$ and a diffuse color $d \in \mathbb{R}^3$. In the *geometry pass*, all 2D Gaussian primitives are rasterized once to produce per-pixel buffers at image coordinates (u,v), storing the 3D position $P(u,v)$ of the visible surface, its normal $N(u,v)$, roughness $R(u,v)$, and diffuse color $D(u,v)$. In what follows all terms are computed per-pixel, so we omit the explicit dependence on (u,v). The final shaded color for a given pixel is computed as:

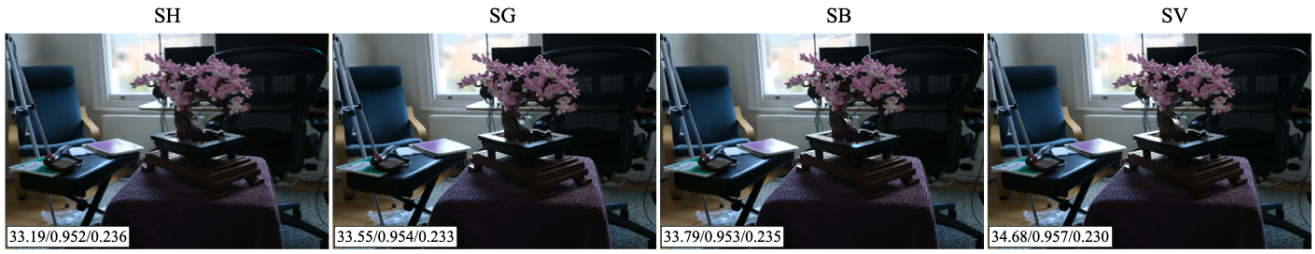
$$C = D + C_{\text{spec}}, \tag{6}$$

where C_{spec} encodes the specular component of the appearance. The specular color is modeled as a spatially-varying blend between near- and far-field illumination:

$$C_{\text{spec}} = \alpha\, C_n + (1-\alpha)\, C_f. \tag{7}$$

The far-field term C_f represents distant illumination and it is implemented as a learnable cubemap evaluated at the reflection direction $\omega_r = 2(\omega \cdot N)N - \omega$.

The near-field term C_n captures spatially varying reflections using a set of learnable *Voronoi light probes*. Each probe i is parameterized by a position $p_i \in \mathbb{R}^3$, a blending weight $\alpha_i \in [0,1]$, and the parameters of a Spherical



Method	Color Model	Mip-NeRF360 PSNR ↑	SSIM ↑	LPIPS ↓	NeRF-Synthetic PSNR ↑	SSIM ↑	LPIPS ↓	DeepBlending PSNR ↑	SSIM ↑	LPIPS ↓	Tanks&Temples PSNR ↑	SSIM ↑	LPIPS ↓
Zip-NeRF [2]	MLP	28.55	0.829	0.218	33.67	0.973	0.036	-	-	-	23.64	0.839	0.100
Beta-Splatting [14]	SH	28.09	0.833	0.231	34.15	0.972	0.033	29.80	0.912	0.303	24.50	0.869	0.175
	SG	28.18	0.833	0.233	34.26	0.972	0.033	29.67	0.912	0.306	24.71	0.870	0.176
	SB	28.12	0.831	0.238	34.10	0.971	0.034	29.56	0.907	0.316	24.54	0.866	0.196
	SV	28.57	0.835	0.230	34.53	0.973	0.032	30.48	0.915	0.296	24.75	0.871	0.171

Table 1. **Modeling radiance** – Our Spherical Voronoi model (SV) consistently improves performance over existing color parameterizations.

Voronoi function (τ_i, s_i, c_i), as defined in Equation (4). All of these parameters are optimized during training. For a surface point P, we query its k-nearest probes:

$$\mathcal{N} = \text{kNN}(P), \tag{8}$$

and compute normalized inverse-distance weights:

$$\tilde{w}_i = \frac{\|P - p_i\|^{-1}}{\sum_{j\in\mathcal{N}} \|P - p_j\|^{-1}}. \tag{9}$$

The near-field color and blending factor are then evaluated as:

$$C_n = \sum_{i\in\mathcal{N}} \tilde{w}_i \, f^i_{\text{SV}}(\omega_r; \tau), \qquad \alpha = \sum_{i\in\mathcal{N}} \tilde{w}_i \, \alpha_i. \tag{10}$$

Here, the roughness R modulates the temperature τ, which controls the angular sharpness of the Spherical Voronoi distribution:

$$\tau = (1 - R)\,\tau_{\max} + R\,\tau_{\min}, \tag{11}$$

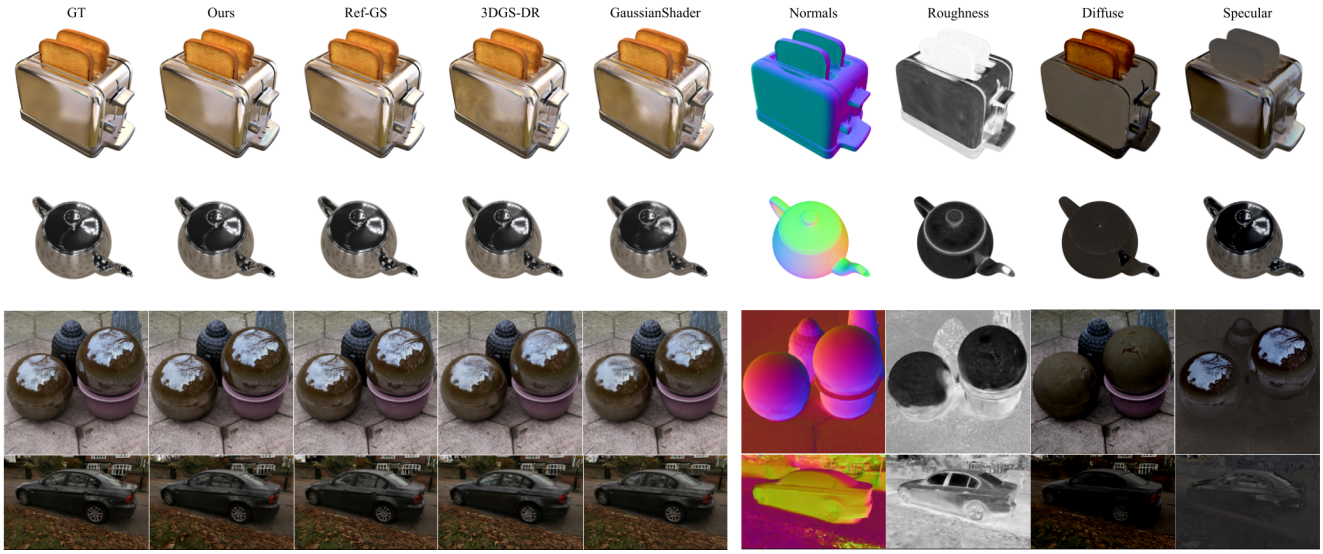
where $\tau_{\min}$ and $\tau_{\max}$ are fixed hyperparameters. Lower roughness regions correspond to higher values of τ, producing sharper reflections, while higher roughness broadens the lobes. This formulation provides a unified, differentiable and explicit model for diffuse and specular appearance. Note that in this reflection-based formulation, τ is not directly learned but derived from the surface roughness R. In contrast, in the radiance modeling setup of Section 3.3, τ is treated as a learnable parameter jointly optimized with the Spherical Voronoi sites and values.

4. Experiments

As mentioned in Section 3, we evaluate our approach in two settings. The first, which we refer to as *view direction parameterization*, is discussed in Section 4.1 where we evaluate our Spherical Voronoi representation for modeling view-dependent appearance in "standard" representations which model outgoing radiance as a function of view direction (*i.e.*, without explicitly modeling reflections). The second, which we refer to as *reflection-based parameterization*, is presented in Section 4.2, where we evaluate our Voronoi Light Probes representation by explicitly modeling reflections. In both settings, the Spherical Voronoi sites are initialized uniformly using Fibonacci sampling, and all remaining hyperparameters (losses, schedulers, *etc.*) follow the default configuration of the respective backbone.

4.1. View Direction Parameterization

To evaluate our view-direction parameterization, we adopt the Beta Splatting backbone [14]. In their implementation, the model is periodically evaluated on the test set (every 500 iterations), and the best-performing checkpoint within 30k iterations is reported. Although this strategy effectively acts as an early-stopping mechanism, it relies on test-set feedback. In our experiments, we instead train for a fixed number of iterations without using the test set for model selection. In this setting, we employ the *weighted* Spherical Voronoi parameterization introduced in Section 3.2. Each site s_k defines both a unit direction $\hat{s}_k = s_k/\|s_k\|$ and an implicit temperature $\tau_k = \|s_k\|$, with the norm of the site vector directly defining the temperature associated with that cell. Additionally, for the *MipNeRF-360* dataset we trained and tested the models on the already downsampled images.



Method	Ref-NeRF [25]			GlossySynthetic			Ref-Real		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Ref-NeRF [25]	32.32	0.956	0.110	27.50	0.927	0.100	23.62	0.646	0.239
NeRO [15]	29.84	0.962	0.072	-	-	-	-	-	-
ENVIDR [13]	32.88	0.969	0.072	29.58	0.952	0.057	23.00	0.606	0.332
3DGS [9]	30.37	0.947	0.083	26.50	0.917	0.092	23.85	0.660	0.230
GShader† [8]	30.91	0.954	0.081	27.54	0.922	0.087	23.32	0.648	0.257
3iGS† [24]	30.91	0.950	0.076	26.86	0.915	0.088	23.67	0.644	0.232
3DGS-DR† [30]	34.13	0.972	0.058	30.36	0.957	0.053	23.83	0.661	0.240
Ref-GS† [32]	35.57	0.975	0.051	31.27	0.962	0.045	23.81	0.659	0.238
Ours	36.09	0.976	0.050	31.30	0.962	0.046	23.91	0.659	0.244

Table 2. **Modeling reflections** - Our method consistently achieves top-tier performance across all datasets. Methods marked with † were retrained for consistency, and most results were successfully reproduced. Ref-GS performs slightly better than originally reported on Ref-NeRF and GlossySynthetic, but worse on Ref-Real, likely because the original implementation did not use the dataset's pre-downsampled input images. In the top-left, we show example outputs: our method produces reflections that are generally on par with—if not superior to—Ref-GS, and clearly better than other baselines (e.g., sharper branch reflections in GardenSpheres and Sedan). On the right, we illustrate the learned decomposition of the scene in diffuse and specular color, roughness and normals.

Our Spherical Voronoi representation employs 8 sites per Gaussian, matching the number of degrees of freedom of degree-3 spherical harmonics. The memory footprint scales linearly with the number of sites: each site stores three floats for its location and three for the radiance, while the temperature τ_k is computed on the fly from the site norm. This results in a total of 48 learnable parameters per Gaussian. We evaluate this configuration on standard radiance fields benchmarks: *Mip-NeRF360* [1], *DeepBlending* [6], *Tanks&Temples* [11], and *NeRF-Synthetic* [19]. Table 1 shows the results of the view-direction parameterization experiments. Our Spherical Voronoi formulation yields consistent improvements over all baselines, achieving higher PSNR than all other color parameterizations (SH, SG, SB). Notably, it also surpasses Zip-NeRF, a strong neural radiance baseline that has dominated benchmarks since its inception.

4.2. Reflection-based Parameterization

For the reflection-based parameterization experiments, we use the 2DGS backbone [7], which provides more accurate normal estimation. For these experiments, we use 128 probes for synthetic scenes and 1024 probes for real scenes, combined with 2048 sites. The cubemap resolution is $256 \times 256 \times 6 \times 3$. We evaluate this configuration on three widely used datasets for reflective scenes: *Ref-NeRF*, *GlossySynthetic*, and *Ref-Real*. Table 2 presents results for the reflection-based parameterization experiments. Our method achieves state-of-the-art performance on *Ref-NeRF* and *GlossySynthetic* datasets, and obtains competi-

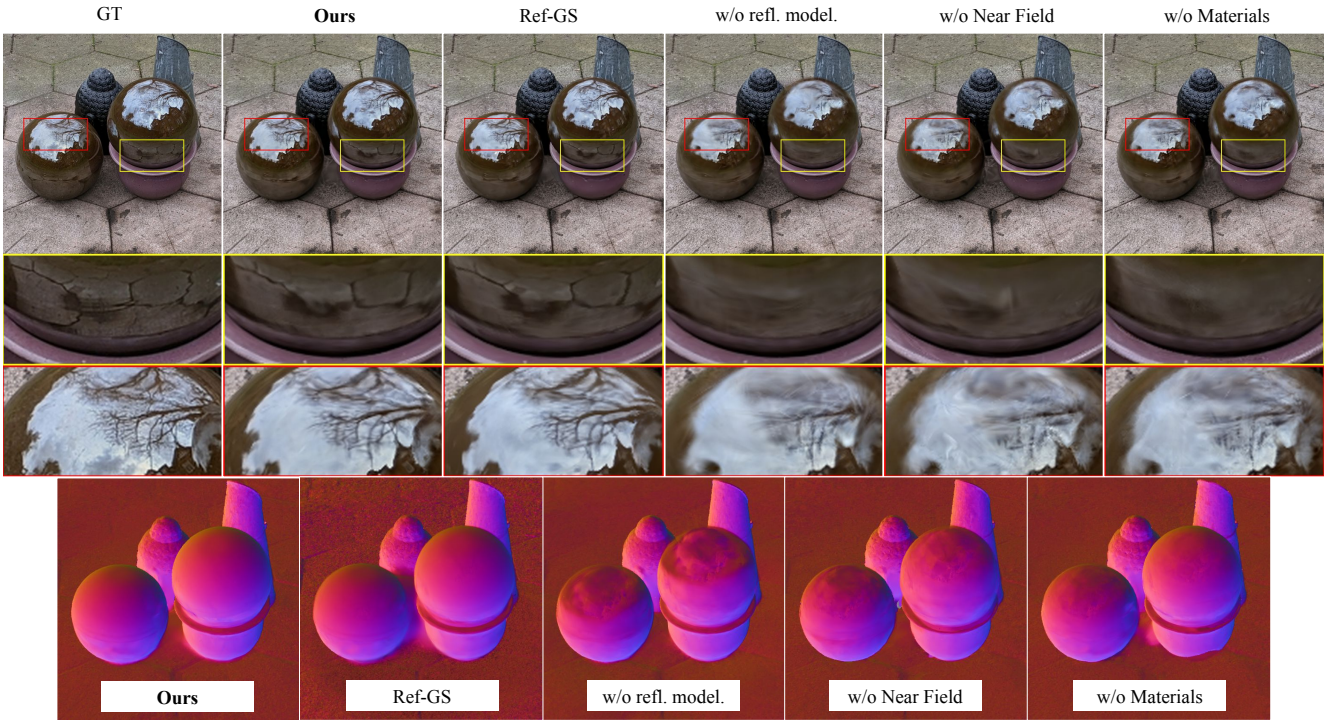


Figure 8. **Qualitative ablation** – We visualize the impact of each component of our model and compare against Ref-GS. Removing reflections, near-field probes, or material parameters (roughness) progressively reduces sharpness, reflection coherence, and normal quality, whereas our full SV-based formulation preserves high-frequency detail and stable shading.

	PSNR ↑	SSIM ↑	LPIPS ↓
SG	34.19	0.967	0.060
SB	34.07	0.966	0.061
Cubemap	34.48	0.970	0.056
SV	36.09	0.976	0.050

Table 3. **Modeling light probes** – Spherical Voronoi delivers substantially higher quality, evaluated on the Ref-NeRF dataset.

tive metrics on *Ref-Real*. Importantly, our method slightly outperforms *Ref-GS*, which uses the same 2DGS backbone but relies on an MLP decoder to model near-field reflections, whereas our approach is fully explicit.

4.3. Ablations

Table 3 reports a quantitative comparison of alternative probe parameterizations used in our framework. We experimented with spherical Gaussians, spherical betas, and cube-map–based representations, keeping the number of learnable parameters *fixed* across settings. Among all tested variants, the Voronoi-based representation consistently achieved the best reconstruction accuracy. This suggests that the inherent spatial partitioning induced by Voronoi cells provides a more expressive and compact basis for modelling local appearance, leading to significantly improved performance under identical capacity constraints. Figure 8 presents a qualitative ablation on the Garden-Spheres scene, which is particularly challenging due to its complex geometry and high-frequency specular and local interactions. We visualize the contribution of each component of our model and compare the final reconstruction to Ref-GS, the current state-of-the-art. Despite relying entirely on an explicit representation, our approach produces reconstructions that are on par with—if not more detailed than—Ref-GS. Notably, our method better resolves fine structures such as tile patterns and other high-frequency textures, without requiring a neural model. It is also important to highlight that standard perceptual and photometric metrics (PSNR, SSIM, LPIPS) remain comparable across all methods, even though models that explicitly model reflections tend to look qualitatively superior.

4.4. Train and Inference Time

For radiance-only modeling, both training and inference remain comparable to the original backbones, since using a small number of SV sites per Gaussian adds only a lightweight computation—essentially a dot product followed by a softmax. In this setting, our runtime closely matches SH-, SG-, and SB-based implementations. The

Table 4. **Efficiency comparison** - Rendering speed and training time reported as factors with respect to 3DGS.

Method	Rendering Speed	Train Time
3DGS	1.00×	1.00×
GaussianShader	0.17×	11.05×
3iGS	0.44×	2.07×
3DGS-DR	0.93×	3.25×
Ref-GS	0.37×	2.63×
Ours	0.45×	2.43×

reflection-based model is naturally slower: deferred shading requires two rendering passes, and evaluating/interpolating the nearest light probes introduces an additional cost. Even with these overheads, our method runs at 0.45× the speed of 3DGS and is faster than Ref-GS, the current state-of-the-art for reflection modeling. Training time follows a similar pattern: probe optimization adds overhead, but our overall cost (2.43×) remains competitive and well below heavier formulations such as GaussianShader.

5. Conclusion

We introduced Spherical Voronoi functions as a new explicit representation for modeling directional appearance in radiance-field–based reconstruction. Thanks to their adaptive decomposition of the sphere and stable optimization behavior, SV consistently outperforms traditional bases such as SH, SG, and SB in radiance-only settings, while maintaining runtime comparable to the underlying forward-rendering backbones. We further extended this representation to spatially varying reflections through Voronoi Light Probes, a fully explicit formulation that captures complex near-field effects without neural decoders. By relying on deferred shading and probe interpolation, our method achieves state-of-the-art quality on reflective benchmarks. Overall, our results demonstrate that carefully designed explicit representations can serve as powerful and efficient alternatives to neural appearance models, enabling high-quality view-dependent effects, encouraging interpretability and practical rendering performance.

References

[1] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022. 7

[2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *ICCV*, 2023. 1, 2, 6

[3] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022. 2

[4] Shrisudhan Govindarajan, Daniel Rebain, Kwang Moo Yi, and Andrea Tagliasacchi. Radiant foam: Real-time differentiable ray tracing. *arXiv:2502.01157*, 2025. 2

[5] Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. Shape, Light, and Material Decomposition from Images using Monte Carlo Rendering and Denoising. *arXiv:2206.03380*, 2022. 2

[6] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. 37(6):257:1–257:15, 2018. 7

[7] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*, New York, NY, USA, 2024. Association for Computing Machinery. 5, 7, 4

[8] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5322–5332, 2024. 3, 7

[9] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 2, 7

[10] Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Jeff Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3d gaussian splatting as markov chain monte carlo. *NeurIPS*, 2024. 4

[11] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017. 7

[12] Jonas Kulhanek and Torsten Sattler. Nerfbaselines: Consistent and reproducible evaluation of novel view synthesis methods. *arXiv preprint arXiv:2406.17345*, 2024. 1

[13] Ruofan Liang, Huiting Chen, Chunlin Li, Fan Chen, Selvakumar Panneer, and Nandita Vijaykumar. Envidr: Implicit differentiable renderer with neural environment lighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 79–89, 2023. 3, 7

[14] Rong Liu, Dylan Sun, Meida Chen, Yue Wang, and Andrew Feng. Deformable beta splatting. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–11, 2025. 3, 6

[15] Yuan Liu, Peng Wang, Cheng Lin, Xiaoxiao Long, Jiepeng Wang, Lingjie Liu, Taku Komura, and Wenping Wang. Nero: Neural geometry and brdf reconstruction of reflective objects from multiview images. *ACM Transactions on Graphics (ToG)*, 42(4):1–22, 2023. 3, 7

[16] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024. 3

[17] Alexander Mai, Dor Verbin, Falko Kuester, and Sara Fridovich-Keil. Neural microfacet fields for inverse rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 408–418, 2023. 2

[18] Alexander Mai, Peter Hedman, George Kopanas, Dor Verbin, David Futschik, Qiangeng Xu, Falko Kuester, Jonathan T. Barron, and Yinda Zhang. Ever: Exact volumetric ellipsoid rendering for real-time view synthesis, 2024. 2

[19] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *ECCV*, 2020. 1, 2, 7

[20] Nicolas Moenne-Loccoz, Ashkan Mirzaei, Or Perel, Riccardo de Lutio, Janick Martinez Esturo, Gavriel State, Sanja Fidler, Nicholas Sharp, and Zan Gojcic. 3d gaussian ray tracing: Fast tracing of particle scenes. *ACM Transactions on Graphics and SIGGRAPH Asia*, 2024. 2

[21] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting Triangular 3D Models, Materials, and Lighting From Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8280–8290, 2022. 2

[22] Michael Niemeyer, Fabian Manhardt, Marie-Julie Rakotosaona, Michael Oechsle, Daniel Duckworth, Rama Gosula, Keisuke Tateno, John Bates, Dominik Kaeser, and Federico Tombari. Radsplat: Radiance field-informed gaussian splatting for robust real-time rendering with 900+ fps. *arXiv.org*, 2024. 2

[23] Daniel Rebain, Wei Jiang, Soroosh Yazdani, Ke Li, Kwang Moo Yi, and Andrea Tagliasacchi. Derf: Decomposed radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14153–14161, 2021. 2

[24] Zhe Jun Tang and Tat-Jen Cham. 3igs: Factorised tensorial illumination for 3d gaussian splatting. In *European Conference on Computer Vision*, pages 143–159. Springer, 2024. 3, 7

[25] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. *CVPR*, 2022. 1, 2, 3, 5, 7

[26] Dor Verbin, Ben Mildenhall, Peter Hedman, Jonathan T Barron, Todd Zickler, and Pratul P Srinivasan. Eclipse: Disambiguating illumination and materials using unintended shadows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 77–86, 2024. 2

[27] Dor Verbin, Pratul P Srinivasan, Peter Hedman, Ben Mildenhall, Benjamin Attal, Richard Szeliski, and Jonathan T Barron. Nerf-casting: Improved view-dependent appearance with consistent reflections. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–10, 2024. 2, 3

[28] Francis Williams, Jerome Parent-Levesque, Derek Nowrouzezahrai, Daniele Panozzo, Kwang Moo Yi, and Andrea Tagliasacchi. Voronoinet: General functional approximators with local support. In *CVPRW*, pages 264–265, 2020. 2

[29] Liwen Wu, Sai Bi, Zexiang Xu, Fujun Luan, Kai Zhang, Iliyan Georgiev, Kalyan Sunkavalli, and Ravi Ramamoorthi. Neural directional encoding for efficient and accurate view-dependent appearance modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21157–21166, 2024. 2, 3

[30] Keyang Ye, Qiming Hou, and Kun Zhou. 3d gaussian splatting with deferred reflection. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–10, 2024. 3, 5, 7

[31] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5453–5462, 2021. 2

[32] Youjia Zhang, Anpei Chen, Yumin Wan, Zikai Song, Junqing Yu, Yawei Luo, and Wei Yang. Ref-gs: Directional factorization for 2d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 3, 5, 7

Spherical Voronoi
Directional Appearance as a Differentiable Partition of the Sphere

Supplementary Material

Supplementary Overview

In Section 6 we expand our deferred rendering formulation, explaining probe querying, near- and far-field illumination, and the role of roughness in controlling the Spherical Voronoi temperature. We then describe in Section 7 our acceleration scheme for Spherical Voronoi evaluation, detailing the cubemap-based partitioning and truncated softmax used to reduce computational cost. Next, in Section 8 we present an analysis of the quality–efficiency tradeoff associated with varying the number of interpolated probes and sites. Section 9 provides additional qualitative visualizations: the adaptive SV decomposition learned when fitting environment maps, the effect of the temperature parameter τ on reflectance sharpness, and the optimization behavior of learnable light probes during training. We also clarify the evaluation protocol adopted to ensure fair comparisons across datasets and prior work in Section 10. Finally, in Section 11 we include additional quantitative results, including SV integration into other baselines and full per-scene metrics across all datasets.

6. Additional Details on Deferred Rendering

In this section, we expand the reflection formulation introduced in the main paper and provide a derivation of the near-field term C_n, the far field term C_f and the spatial blending factor α appearing in Equation (7).

6.1. Geometry Pass

Following the deferred strategy illustrated in Figure 7, we first rasterize all 2DGS primitives once to produce the per-pixel attributes required for shading. For each pixel (u, v), the geometry pass outputs:

- world position $P(u, v) \in \mathbb{R}^3$,
- surface normal $N(u, v) \in \mathbb{S}^2$,
- diffuse color $D(u, v) \in \mathbb{R}^3$,
- roughness $R(u, v) \in [0, 1]$.

These attributes are produced by splatting the corresponding Gaussian parameters, using the same weighted blending used during accumulation in standard 2DGS rendering.

6.2. Lighting Pass

Given the geometry buffer, the lighting pass computes the final shaded color:

$$C(u, v) = D(u, v) + C_{\text{spec}}(u, v), \tag{12}$$

where $C_{\text{spec}}(u, v)$ is the specular term defined in Equation (7). We report it here for brevity:

$$C_{\text{spec}}(u, v) = \alpha(u, v)C_n(u, v) + (1 - \alpha(u, v))C_f(u, v). \tag{13}$$

Both illumination terms depend on the reflected view direction:

$$\omega_r(u, v) = 2(\omega \cdot N(u, v))N(u, v) - \omega. \tag{14}$$

In the following sections we describe how each component is computed.

6.3. Near-field Illumination

Near-field reflections are modeled by a set of learnable light probes distributed in 3D space. Each probe i has:

- Position $p_i \in \mathbb{R}^3$,
- Spherical Voronoi function $f_i(\omega)$,
- Blend parameter $\alpha_i \in [0, 1]$.

Probe selection. We identify the probes that are most relevant for shading a given pixel (u, v). We select the k closest probes using Euclidean nearest-neighbor search:

$$\mathcal{N}(u, v) = k\text{NN}(P(u, v)). \tag{15}$$

This ensures that each pixel only interacts with probes that lie within a meaningful spatial neighborhood, reflecting the intuition that local geometry affects appearance only within a limited radius.

Distance-based weighting. Not all selected probes contribute equally. Probes closer to the shading point should have a stronger influence than those further away. We therefore assign each probe a weight inversely proportional to its distance:

$$w_i(u, v) = \frac{1}{\|P(u, v) - p_i\| + \epsilon}, \tag{16}$$

which is normalized as:

$$\tilde{w}(u, v) = \frac{w_i(u, v)}{\sum_{j \in \mathcal{N}(u,v)} w_j(u, v)}. \tag{17}$$

The resulting weights vary smoothly throughout the image, producing spatially consistent transitions between lighting regions.

Evaluating and aggregating near-field illumination. Each probe stores a SV representation of the reflected radiance at its location, which we query in the reflected direction $\omega_r(u, v)$ to obtain a directional estimate of the local illumination. The response of the selected probes is then combined using the normalized spatial weights $\tilde{w}_i(u, v)$. Formally, the near-field term is:

$$C_n(u, v) = \sum_{i \in \mathcal{N}(u,v)} \tilde{w}_i(u, v) f_i(\omega_r(u, v)). \tag{18}$$

This operation blends the directional information encoded at each probe according to its proximity to the shading point.

6.4. Far-field Illumination

While near-field probes capture illumination effects produced by nearby geometry, many reflective surfaces are dominated by light coming from far more distant parts of the environment, such as skylight, outdoor structures, windows, or large surrounding objects. To represent this global component, we use a learnable environment cubemap. Once the reflected direction $\omega_r(u, v)$ has been computed, obtaining the far-field illumination is straightforward: we simply sample the cubemap at that direction,

$$C_f(u, v) = \text{cubemap}(\omega_r(u, v)). \tag{19}$$

6.5. Blending Factor

The relative influence between near-field and far-field illumination must vary across the scene. Close to complex geometry, reflections are more sensitive to local variations and should rely predominantly on the probes. In open regions, distant illumination captured by the cubemap becomes more important. To achieve this adaptive behavior, each probe carries a learned blend weight α_i, which express how strongly that probe favors near-field effects. At shading time, the per-pixel blend factor $\alpha(u, v)$ is obtained by interpolating these probe parameters using the same spatial weights $\tilde{w}_i(u, v)$ computed above:

$$\alpha(u, v) = \sum_{i \in \mathcal{N}(u,v)} \tilde{w}_i(u, v) \alpha_i. \tag{20}$$

This produces a smooth spatial field that naturally transitions between locally dominated and globally dominated reflection regimes.

6.6. Roughness and Temperature

The sharpness of the Spherical Voronoi representation is controlled by the temperature parameter τ. Each Gaussian primitive stores its own roughness value R, which is splatted into a per-pixel roughness map $\mathbf{R}(\mathbf{x})$ during the geometry pass. Roughness determines how sharp or smooth the directional function should be. To couple the reflectance model with Spherical Voronoi expressivity, we linearly map roughness to temperature:

$$\tau(u, v) = (1 - R(u, v))\tau_{\text{max}} + R(u, v)\tau_{\text{min}}, \tag{21}$$

where we empirically set $\tau_{\text{min}} = 0.2$ and $\tau_{\text{max}} = 1500$. As a result:

- *low roughness* ($R \approx 0$) produces crisp, mirror-like reflections through large temperatures,
- *high roughness* ($R \approx 1$) yields broad, diffuse responses via small temperatures.

7. Speeding up Spherical Voronoi

A naïve evaluation of our Spherical Voronoi representation requires evaluating all K sites for every queried direction ω. While this is not a problem in the view-direction parameterization context, when modeling reflections which require a large number of sites, it becomes impractical. Concretely, for a function

$$f_{\text{SV}}(\omega; \tau, s, c) = \sum_{k=1}^{K} w_k(\omega; \tau)\, c_k, \tag{22}$$

where τ is the temperature controlling the sharpness of the partition, $s = \{s_k\}$ are the unit vectors defining the angular locations of the SV sites, and $c = \{c_k\}$ are their associated radiance values, with

$$w_k(\omega; \tau) = \frac{\exp(\tau\, s_k \cdot \omega)}{\sum_{k'=1}^{K} \exp(\tau\, s_{k'} \cdot \omega)}, \tag{23}$$

the cost of a single evaluation scales linearly with the number of sites K. When using thousands of sites per function, this becomes a major bottleneck during rendering, especially when evaluating directional appearance at every pixel. To mitigate this cost, we introduce a simple acceleration scheme that exploits the fact that *only a small subset of sites is relevant for any given direction*. Our key idea is to partition the unit sphere using a low-resolution cubemap, and to pre-assign to each texel a fixed set of *candidate* sites. At runtime, the softmax is restricted to this candidate set instead of all sites.

7.1. Cubemap-Based Partition of the Sphere

We discretize the sphere using a cubemap of fixed resolution. Each texel t in the cubemap is associated with a direction $\hat{\omega}_t$ corresponding to its center. In a preprocessing step, for each texel t we select a small subset of sites $\mathcal{S}(t) \subset \{1, ..., K\}$ that are the most relevant for directions

around $\hat{\omega}_t$. A natural choice is to pick the sites with highest similarity to $\hat{\omega}_t$, *i.e.*:

$$\mathcal{S}(t) = \text{TopK}_k(s_k \cdot \hat{\omega}_t), \tag{24}$$

where TopK denotes the set of indices of the k closest sites. This yields a lookup table that maps each cubemap texel to a much smaller set of candidate sites. Since the Spherical Voronoi sites are optimized jointly with the rest of the model, the cubemap-to-site assignment $\mathcal{S}(t)$ becomes outdated over the course of training. To account for this, we periodically recompute it. In all our experiments, we rebuild the assignment every 500 optimization steps, which we found to be a good compromise between accuracy of the approximation and preprocessing overhead.

7.2. Softmax Approximation

At rendering time, given a direction ω_r, we determine the cubemap texel $t(\omega_r)$ that contains ω_r. Instead of evaluating the SV weights over all sites, we restrict the computation to the precomputed candidate set. The accelerated SV evaluation becomes:

$$f_{\text{SV}}^{\text{fast}}(\omega_r) = \sum_{k \in \mathcal{S}(t(\omega_r))} \tilde{w}_k(\omega_r; \tau)\, c_k, \tag{25}$$

where the truncated weights are defined as:

$$\tilde{w}_k(\omega_r; \tau) = \frac{\exp(\tau\, s_k \cdot \omega_r)}{\sum_{k' \in \mathcal{S}(t(\omega_r))} \exp(\tau\, s_{k'} \cdot \omega_r)}. \tag{26}$$

In other words, we approximate the full softmax over K sites with a local softmax over a small, direction-dependent subset of sites. Intuitively, for sufficiently fine cubemap resolution and moderately large candidate set size, the omitted sites have negligible contribution to the softmax, as their dot product $s_k \cdot \omega_r$ is much smaller. This reduces the per-direction complexity from $\mathcal{O}(K)$ to $\mathcal{O}(|\mathcal{S}(t)|)$.

8. Quality-Efficiency Tradeoff

We evaluate the effect of kernel capacity by varying the cardinalities $|S(K)|$ and $|N(K)|$. As shown in Figure 9, increasing these values improves reconstruction quality but reduces inference speed, exhibiting a clear quality–efficiency tradeoff. Notably, setting $|N(K)| = 32$ yields a small quality gain but nearly halves the FPS, showing rapidly diminishing returns. For this reason, we adopt $|S(K)| = 8$ and $|N(K)| = 8$ as a balanced configuration that preserves most of the quality benefits while maintaining real-time performance.

9. Visualization of SV Behavior and Probe Dynamics

In this section, we provide qualitative visualizations that further illustrate the behavior of our SV representation and the dynamics of learnable light probes. Figure 10 shows the adaptive decomposition of the spherical domain when learning an environment map. Figure 11 visualizes how increasing τ sharpens SV lobes and enhances specular detail. Figure 12 illustrates the optimization trajectory of the light probes.

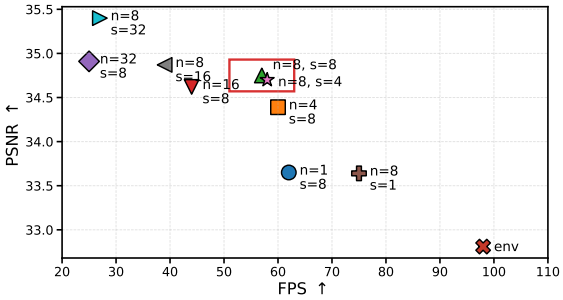


Figure 9. **Quality–Efficiency Tradeoff** - Increasing the kernel capacity ($|\mathcal{N}(K)|$ and $|\mathcal{S}(K)|$; abbreviated as n and s in the plot) leads to higher rendering quality but lower inference speed, highlighting the fundamental tradeoff between accuracy and efficiency. Results are reported for the *coffee* scene of the Ref-NeRF dataset. *Env* denotes the configuration without probes.

10. Clarifications on Evaluation

To ensure a fair and consistent comparison across methods, we reviewed the evaluation protocols commonly used for the Mip-NeRF 360 and Ref-Real datasets. We found that some prior works rely on testing setups that deviate from the dataset guidelines, which can lead to inflated performance metrics. For Mip-NeRF 360, Beta Splatting [14] evaluates models on images that are downsampled on-the-fly from the high-resolution originals (using the `--r` flag in 3DGS codebase). This produces smoother inputs that are easier to fit, increasing PSNR by approximately 0.5 dB on average compared to using the official downsampled images provided in the dataset. Additionally, Beta Splatting employs a checkpoint-selection mechanism: starting from iteration 15k, the model is evaluated on the test set every 500 steps, and the best checkpoint is reported as the final result. While practical, this introduces a form of test-set selection that is not fully aligned with standard evaluation practice. A similar issue appears in the Ref-Real results reported by Ref-GS [32], where non-standard downsampling choices also deviate from the dataset's recommended evaluation setup. For a fully fair and reproducible evaluation, we strictly follow the dataset recommendations. In Mip-NeRF 360, we use `images_2` for indoor scenes and `images_4` for outdoor scenes. For Ref-Real, we use `images_4` for *garden spheres* and *toy car*, and `images_8` for *sedan*. All results in this paper follow these settings.

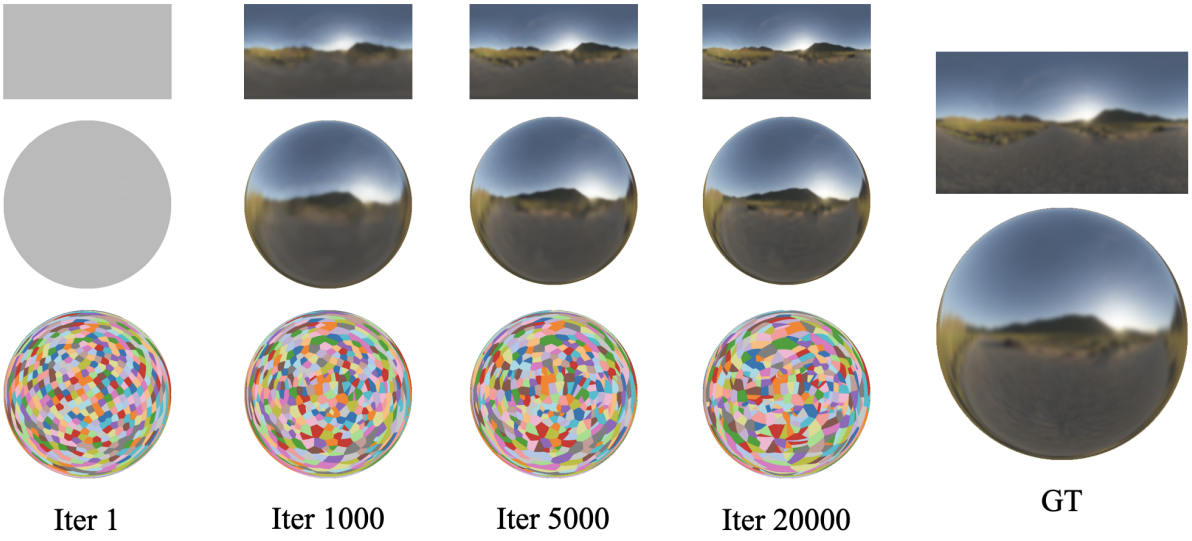


Figure 10. **SV evolution during training (3D)** - Top: predicted environment map in lat-long format. Middle: its rendering on a reflective sphere. Bottom: corresponding SV tessellation (colored by random site IDs). Across training iterations, the SV sites reorganize into a structured decomposition of the sphere. Ground truth is shown on the right.

Figure 11. **Effect of the SV temperature τ (3D)** - Increasing τ progressively sharpens the SV lobes, transitioning from an almost uniform shading ($\tau = 0$) to increasingly crisp, mirror-like reflections. Renderings shown for $\tau \in \{0, 5, 50, 100, 250\}$

11. Additional Results

11.1. Modeling Radiance

Table 5. **Radiance Modeling with SV Voronoi** - Applying SV to existing Gaussian Splatting baselines yields consistent improvements across datasets.

		Vanilla			Ours (SV)		
	Method	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Mip-NeRF 360	*3DGS-mcmc*	27.92	0.833	0.234	28.33	0.832	0.234
	2DGS	26.86	0.798	0.301	27.41	0.800	0.300
	Beta-Splatting	28.12	0.831	0.238	28.57	0.835	0.230
NeRF-Synthetic	*3DGS-mcmc*	33.77	0.972	0.036	34.21	0.972	0.034
	2DGS	33.17	0.968	0.041	33.57	0.969	0.039
	Beta-Splatting	34.10	0.971	0.034	34.53	0.973	0.032
Tanks&Temples	*3DGS-mcmc*	24.24	0.863	0.190	24.45	0.863	0.192
	Beta-Splatting	24.54	0.871	0.171	24.75	0.871	0.171
DeepBlending	*3DGS-mcmc*	29.55	0.901	0.320	29.64	0.905	0.314
	Beta-Splatting	29.56	0.907	0.316	30.48	0.915	0.296

SV can be seamlessly integrated into other Gaussian Splatting pipelines. In Table 5, we report the improvements obtained when augmenting two popular baselines—3DGS-MCMC [10] and 2DGS [7]—with our SV modeling. The consistent gains across both methods demonstrate that SV serves as a general and effective mechanism for enhancing radiance modeling in Gaussian-based representations.

11.2. Per-scene Results

We provide a detailed per-scene evaluation for all datasets used in the main paper. Table 6 reports results on radiance-dominated datasets (Mip-NeRF 360, NeRF-Synthetic, Tanks&Temples, and Deep Blending), while Table 7 presents results for reflection-heavy datasets (Ref-NeRF, Glossy Synthetic, and Ref-Real).

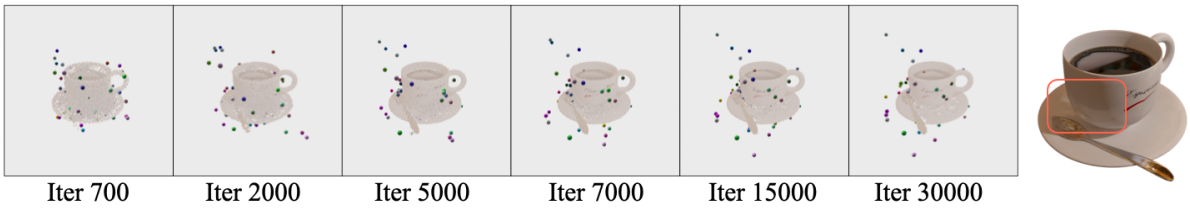


Figure 12. **Evolution of learnable light probes during training** - The probes are initialized at random positions and gradually migrate toward regions exhibiting strong near-field specular interactions. Throughout optimization, they consistently cluster around the reflective side of the object (e.g., the spoon-facing region in the Coffee scene of the Ref-NeRF dataset), which corresponds to an area rich in local specular highlights and interreflections.

Table 6. **Per-scene radiance quality** - Rendering metrics reported per scene across all evaluated datasets.

	Scene	PSNR ↑	SSIM ↑	LPIPS ↓
Mip-NeRF 360	bicycle	25.63	0.791	0.196
	bonsai	34.68	0.957	0.230
	counter	30.97	0.928	0.225
	flowers	22.21	0.639	0.335
	garden	27.74	0.872	0.120
	kitchen	32.57	0.934	0.150
	room	32.97	0.936	0.259
	stump	27.25	0.803	0.212
	treehill	23.08	0.651	0.338
	mean	28.57	0.835	0.230
NeRF-Synthetic	chair	36.85	0.989	0.013
	drums	26.96	0.958	0.039
	ficus	37.01	0.991	0.009
	hotdog	38.31	0.988	0.021
	lego	36.99	0.986	0.016
	materials	30.87	0.965	0.039
	mic	37.60	0.994	0.006
	ship	31.68	0.911	0.117
	mean	34.53	0.973	0.032
Tanks& Temples	train	22.95	0.841	0.214
	truck	26.55	0.900	0.128
	mean	24.75	0.871	0.171
Deep Blending	drjohnson	29.92	0.913	0.301
	playroom	31.04	0.916	0.292
	mean	30.48	0.915	0.296

Table 7. **Per-scene reflection quality** - Rendering metrics reported per scene across all evaluated datasets.

	Scene	PSNR ↑	SSIM ↑	LPIPS ↓
Ref-NeRF	ball	39.40	0.987	0.082
	car	31.26	0.968	0.029
	coffee	34.91	0.973	0.082
	helmet	35.63	0.983	0.031
	teapot	47.59	0.997	0.007
	toaster	27.73	0.950	0.070
	mean	36.09	0.976	0.050
Glossy Synthetic	bell	32.22	0.963	0.044
	cat	33.46	0.975	0.033
	luyu	30.22	0.950	0.042
	potion	33.74	0.961	0.061
	tbell	30.90	0.966	0.050
	teapot	27.25	0.954	0.048
	mean	31.30	0.962	0.046
Ref-Real	gardenspheres	21.97	0.578	0.275
	sedan	25.94	0.757	0.215
	toycar	23.83	0.641	0.243
	mean	23.91	0.659	0.244