SB-HyperGS: Spectral–Spatial Representation Capacity Reallocation for Sparse Hyperspectral Gaussian Splatting

Anonymous authors
Paper under double-blind review

Abstract

Hyperspectral 3D reconstruction aims to recover dense spectral radiance fields from sparse spectral observations. However, directly extending 3D Gaussian Splatting typically requires assigning an independent spectral representation to each Gaussian primitive, resulting in substantial spectral parameter redundancy and limiting spatial representation capacity under a finite parameter budget. We propose a material-efficient hyperspectral Gaussian Splatting framework that replaces independent per-Gaussian spectral storage with a compact material representation. By learning shared material spectral bases and Gaussian-specific local mixing coefficients, our method reduces redundant spectral parameters and reallocates the saved representation capacity to spatial primitives. We further introduce material-aware optimization and adaptive spatial refinement to improve reconstruction from sparse spectral observations. Experiments on multi-scene hyperspectral benchmarks show that our method achieves a better quality–parameter trade-off than direct hyperspectral Gaussian Splatting. Under matched parameter budgets, our method improves reconstruction quality while substantially reducing per-primitive spectral representation overhead, demonstrating the effectiveness of compact material representations for scalable hyperspectral neural rendering.

1 Introduction

Hyperspectral imaging densely samples the spectral response of a scene, providing information beyond RGB for analyzing object appearance and material properties. Extending hyperspectral observations to 3D radiance fields enables view-dependent spectral rendering, material analysis, and 3D scene understanding. Neural Radiance Fields (NeRF) provide a general framework for learning continuous scene representations from multi-view observations (Mildenhall et al., 2020), and recent hyperspectral extensions demonstrate the feasibility of wavelength-dependent radiance fields (Chen et al., 2024).

Acquiring complete hyperspectral data is often constrained by sensor cost, acquisition time, and hardware limitations. In practice, each view may contain only a subset of spectral bands, requiring models to recover full-spectrum radiance from sparse spectral measurements. Unlike conventional few-view reconstruction, where sparsity arises from insufficient cameras (Niemeyer et al., 2022; Yang et al., 2023), the sparsity here lies in the spectral dimension: scene geometry may be sufficiently observed while wavelength information remains incomplete. The central challenge is therefore to establish effective shared representations across incomplete spectral observations.

3D Gaussian Splatting (3DGS) explicitly optimizes spatial Gaussian primitives and uses differentiable rasterization to achieve high-quality novel-view synthesis (Kerbl et al., 2023). A straightforward hyperspectral extension assigns each Gaussian an independent spectral vector covering all target bands. Although expressive locally, this strategy repeatedly stores shared material spectral structures and causes spectral parameters to grow with the number of spatial primitives, creating competition between spectral storage and spatial representation under a finite parameter budget. Therefore, the key issue in sparse hyperspectral Gaussian rendering is not simply increasing spectral degrees of freedom, but allocating limited representation capacity between spectral appearance modeling and spatial structure representation.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

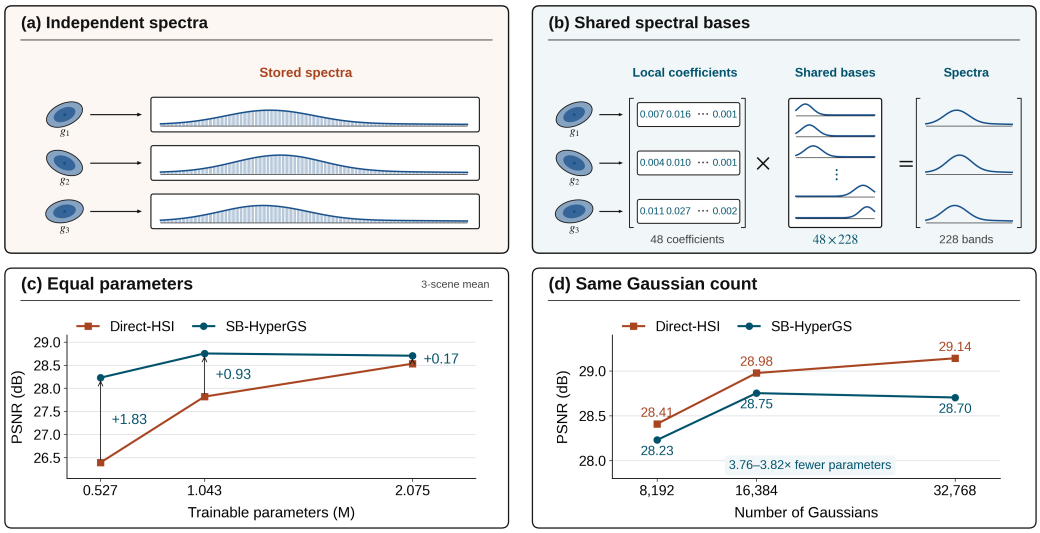

Figure 1: Motivation of SB-HyperGS. Independent spectral storage assigns a complete spectrum to each Gaussian, causing redundant spectral parameters. By using shared spectral bases with local coefficients, SB-HyperGS reduces spectral storage overhead and reallocates representation capacity for spatial modeling.

Motivated by this observation, we propose SB-HyperGS, a material-efficient hyperspectral Gaussian Splatting framework based on spectral–spatial representation reallocation. Instead of conventional per-Gaussian spectral storage, SB-HyperGS represents spectral appearance using scene-level shared material spectral bases and Gaussian-specific local mixing coefficients. The shared bases capture reusable spectral structures, while local coefficients describe variations in material composition. As illustrated in Figure 1, this design reduces redundant per-primitive spectral storage and improves spatial modeling under finite representation capacity. To address sparse-band optimization difficulties, we further introduce material-aware optimization that constrains spectral basis smoothness and local mixture sparsity. Meanwhile, adaptive spatial refinement adjusts Gaussian distribution within a predefined capacity constraint, allocating limited spatial capacity to regions with greater modeling demands.

We evaluate SB-HyperGS from two complementary perspectives. First, under matched parameter budgets, we examine whether shared spectral representations can free capacity for spatial modeling and improve reconstruction quality. Second, with a fixed Gaussian count, we assess whether the method can maintain competitive reconstruction performance with lower per-primitive spectral parameter overhead. These experiments evaluate capacity reallocation and representation efficiency, respectively.

Our contributions are summarized as follows:

- We analyze representation redundancy caused by per-primitive spectral storage in sparse hyperspectral Gaussian Splatting and formulate it as a spectral–spatial capacity allocation problem.
- We propose SB-HyperGS, which replaces independent spectral storage with shared material spectral bases and Gaussian-specific local mixing coefficients, reducing redundant spectral parameters and freeing capacity for spatial representation.
- We develop material-aware optimization and adaptive spatial refinement, combining spectral structure constraints with spatial capacity adjustment to coordinate appearance modeling and spatial structure representation under sparse spectral supervision.
- We demonstrate the quality–parameter trade-off advantages of this representation on real and synthetic hyperspectral scenes through matched-parameter comparisons, fixed-Gaussian-count comparisons, module ablations, and robustness experiments.

2 Related Work

2.1 Hyperspectral Neural Rendering

Hyperspectral neural rendering extends conventional radiance fields from RGB appearance to wavelength-dependent radiance representations. HS-NeRF learns continuous hyperspectral radiance fields from spatial positions, viewing directions, and wavelengths (Chen et al., 2024). SpectralNeRF incorporates the spectral imaging process by combining wavelength-dependent radiance prediction with imaging responses (Li et al., 2024). For explicit representations, HyperGS encodes high-dimensional spectra into a learned latent space and reconstructs hyperspectral novel views through Gaussian rasterization (Thirgood et al., 2024). These methods demonstrate the feasibility of hyperspectral scene reconstruction with both implicit and explicit representations.

Another line of work exploits the intrinsic structure of spectra to reduce redundant wavelength modeling. UnMix-NeRF represents hyperspectral radiance using a global endmember dictionary and spatial abundances, while enabling material-aware analysis (Perez et al., 2025). These studies highlight the value of shared or low-dimensional spectral representations. However, existing methods mainly focus on spectral prediction, latent reconstruction, or material interpretability, while the allocation of limited representation capacity between spectral appearance and spatial primitives under sparse-band supervision remains underexplored.

2.2 Gaussian Splatting and Compact Radiance Representations

3DGS constructs explicit radiance fields using optimizable Gaussian primitives and differentiable rasterization, achieving high-quality novel-view synthesis with efficient rendering (Kerbl et al., 2023). Subsequent works improve Gaussian optimization through primitive generation, densification, pruning, and spatial distribution (Hanson et al., 2025; Wang et al., 2025). For example, 3DGS-MCMC optimizes Gaussian sets through MCMC-based updates (Kheradmand et al., 2024), while compact representations such as LocoGS exploit local correlations among Gaussian attributes to reduce redundancy (Shin et al., 2025). These studies show that Gaussian performance depends not only on primitive quantity but also on how representation capacity is allocated.

Existing compact Gaussian methods mainly target RGB appearance, spatial redundancy, or general attribute compression. In hyperspectral scenes, however, each primitive may carry hundreds of spectral values, making spectral attributes the dominant per-primitive cost and directly limiting the number of spatial primitives under a fixed budget. Therefore, reducing high-dimensional spectral overhead while effectively utilizing the released spatial capacity remains an important open problem.

3 Method

3.1 Problem Formulation

Let $\Lambda = \{\lambda_c\}_{c=1}^{C}$ denote the full set of spectral wavelengths. During training, view v observes only a subset of bands $\Omega_v \subset \{1, \dots, C\}$, yielding sparse spectral observations $Y_v \in \mathbb{R}^{H\times W\times K}$. Complementary band masks across views jointly constrain the full spectral structure. Given sparse multi-view observations, the goal is to recover full C-band spectral images at novel views. The scene is represented by G anisotropic Gaussian primitives. Gaussian i has a center $\mu_i \in \mathbb{R}^3$, three-axis scales $s_i \in \mathbb{R}^3_+$, a rotation matrix R_i, and opacity o_i, with spatial covariance

$$\Sigma_i = R_i \,\mathrm{diag}(s_i^2) R_i^{\mathsf{T}}. \tag{1}$$

Scales and opacity are parameterized by exponential and sigmoid functions, respectively. Geometry, appearance, and opacity are jointly optimized. Figure 2 presents an overview of SB-HyperGS.

3.2 Compact Material Representation

A direct hyperspectral Gaussian representation typically stores an independent C-dimensional spectral vector for each primitive, resulting in $\mathcal{O}(GC)$ spectral parameters that scale linearly with both the Gaussian count G and the band count C. To reduce per-primitive spectral overhead, we repre-

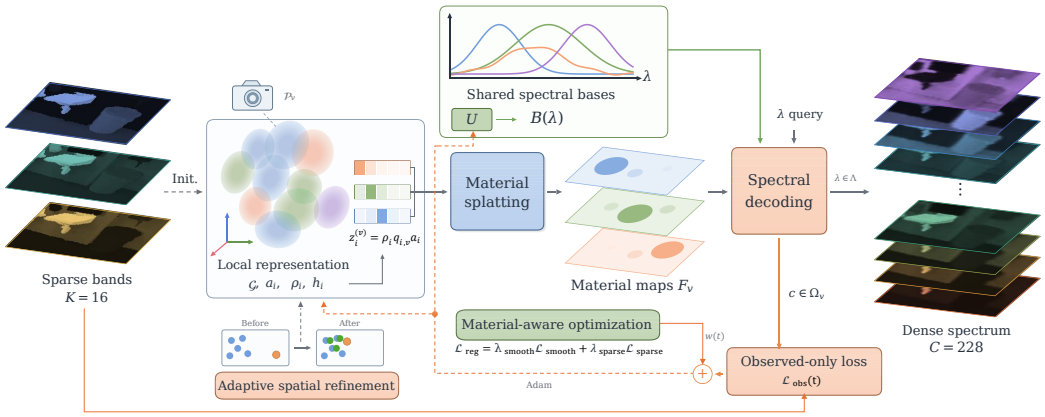

Figure 2: Overview of SB-HyperGS. Sparse spectral observations are represented by Gaussian-specific local coefficients and shared spectral bases, followed by material-space splatting and spectral decoding. Material-aware optimization and adaptive spatial refinement jointly improve spectral reconstruction and spatial representation.

sent scene spectral appearance using scene-level shared material spectral bases and Gaussian-local mixing coefficients.

Shared material spectral bases. We learn $R = 48$ shared spectral bases with a parameter matrix $U \in \mathbb{R}^{R\times C}$. At discrete wavelengths, the r-th basis is represented by $B_r(\lambda_c) = \mathrm{sigmoid}(U_{r,c})$ and its value at an arbitrary query wavelength is obtained by linear interpolation between adjacent wavelengths:

$$B_r(\lambda) = (1-\tau)\,\mathrm{sigmoid}(U_{r,c}) + \tau\,\mathrm{sigmoid}(U_{r,c+1}), \qquad \tau = \frac{\lambda-\lambda_c}{\lambda_{c+1}-\lambda_c}. \tag{2}$$

This representation directly learns shared spectral structure over the full wavelength grid and supports continuous wavelength queries.

Local material mixtures. Each Gaussian learns its own R-dimensional mixing logits ℓ_i, a non-negative intensity $\rho_i = \mathrm{softplus}(\widetilde{\rho}_i)$, and view modulation parameters h_i. To gradually introduce additional spectral components, we progressively activate spectral bases at training step t:

$$g(t) = \mathrm{clip}\left(\frac{t-2000}{4000}, 0, 1\right), \qquad a_i(t) = \mathrm{softmax}\left([\ell_{i,1:32}, \ell_{i,33:48} + \log\max(g(t), 10^{-30})]\right). \tag{3}$$

The first 32 bases participate in training initially, while the remaining bases are gradually activated as training proceeds. Given a camera center c_v, the view modulation factor is determined by the viewing direction and parameters h_i. The Gaussian spectrum at wavelength λ is expressed as

$$q_{i,v} = 1 + 0.1\tanh(h_i^{\mathsf{T}} d_{i,v}), \qquad s_i^{(v)}(\lambda) = \sum_{r=1}^{R} \rho_i q_{i,v} a_{i,r} B_r(\lambda), \tag{4}$$

where $d_{i,v} = (c_v - \mu_i)/\|c_v - \mu_i\|_2$. The background uses independent mixing coefficients and an intensity parameter while sharing the same spectral bases.

Material rendering and decoding. Let $w_{i,v}(p) = T_{i,v}(p)\alpha_{i,v}(p)$ denote the depth-sorted alpha-compositing weights. The material-channel response is then

$$F_{v,r}(p) = \sum_i w_{i,v}(p) z_{i,r}^{(v)} + T_{\mathrm{bg},v}(p) z_{\mathrm{bg},r}, \tag{5}$$

where $z_{i,r}^{(v)} = \rho_i q_{i,v} a_{i,r}$. Radiance at an arbitrary wavelength is decoded using the shared spectral bases:

$$\widehat{I}_v(p,\lambda) = \sum_{r=1}^{R} F_{v,r}(p) B_r(\lambda). \tag{6}$$

Since the material channels are wavelength-independent, rendering is first performed in the low-dimensional shared material space, followed by spectral basis decoding to recover radiance at the target wavelengths. During training, the reconstruction loss is computed only over observed bands; at inference, wavelength queries are extended to the full spectrum.

3.3 Material-Aware Optimization

Sparse spectral supervision provides limited constraints on shared spectral bases and local material mixtures. We therefore introduce material-aware optimization to improve the stability of the spectral representation. The data term combines Charbonnier, SAM, and multi-scale SSIM losses over measured bands:

$$\mathcal{L}_{\text{photo}} = \text{mean}\sqrt{(\widehat{Y}_v - Y_v)^2 + 10^{-6}} + 0.1\mathcal{L}_{\text{SAM}} + 0.2\mathcal{L}_{\text{MS-SSIM}}. \tag{7}$$

Here, multi-scale SSIM encourages spatial structural consistency. The spectral smoothness constraint uses absolute second-order differences on the actual wavelength coordinates to limit local variations in the shared spectral bases. The local mixture sparsity constraint minimizes the normalized entropy of Gaussian mixing coefficients, encouraging compact spectral combinations. The additional regularization term is defined as

$$\mathcal{L}_{\text{reg}} = \lambda_{\text{smooth}}\mathcal{L}_{\text{smooth}} + \lambda_{\text{sparse}}\mathcal{L}_{\text{sparse}}. \tag{8}$$

View-dependent brightness modulation remains part of the appearance representation, and its parameters are jointly learned through the reconstruction objective. Initialization uses only sparse spectral observations from training views: shared spectral bases are initialized by masked NMF, and initial mixing coefficients and intensities are estimated using nonnegative least squares. Gaussian positions are initialized by back-projecting pixels from training views. The shared spectral bases, local mixing coefficients, geometry, opacity, and view modulation parameters are then jointly optimized.

3.4 Adaptive Spatial Refinement

Although the compact material representation reduces the spectral parameter overhead of each Gaussian, limited spatial capacity may still be insufficient to cover regions with complex structures. To improve spatial representation efficiency, we introduce adaptive spatial refinement that adjusts the Gaussian distribution according to current rendering errors, spatial gradients, and primitive contributions, allocating more spatial capacity to regions with greater modeling demands.

Residual and coverage scoring. Spatial adjustments rely only on the measured bands of the current view. We first compute the pixel-wise spectral error:

$$e_v(p) = \mathcal{N}\left(\sqrt{K^{-1}\|\widehat{Y}_v(p) - Y_v(p)\|_2^2}\right) + 0.25\mathcal{N}\left(1 - \cos(\widehat{Y}_v(p), Y_v(p))\right), \tag{9}$$

where the normalization function is defined as $\mathcal{N}(x) = \text{clip}(x/\max(\text{mean}(x), 10^{-8}), 0, 10)$. We sample errors at projected Gaussian centers and neighboring locations and maintain decayed statistics of residuals, screen-space gradients, and normalized contributions, denoted by e_i, d_i, u_i, respectively. We further introduce a balancing term based on the spatial coverage of the currently dominant spectral bases, yielding a spatial-demand score for each primitive:

$$S_i = \mathcal{N}(e_i)\sqrt{\mathcal{N}(d_i)}\sqrt{\mathcal{N}(u_i)}\,\text{clip}\left(\sqrt{\frac{\text{mean}_r\, n_r}{\max(n_{m_i}, 1)}}, 0.5, 2\right). \tag{10}$$

The scores are used to select inefficient primitives for adjustment and identify reference regions with greater spatial modeling demands.

Primitive relocation and refinement. For a Gaussian selected for adjustment, two child primitives are generated along its longest principal axis j:

$$\mu_i^{\pm} = \mu_i \pm \frac{1}{2}s_{i,j}R_i e_j, \qquad s_{i,j}^{\pm} = \sqrt{\frac{3}{4}}s_{i,j}, \qquad o_i^{\pm} = 1 - \sqrt{1 - o_i}. \tag{11}$$

The remaining scales and local appearance parameters are inherited from the source primitive, while the opacity adjustment prevents an abrupt increase in local contributions due to primitive duplication. By reallocating inefficient spatial representations and increasing modeling capacity in high-demand regions, this strategy adaptively optimizes the spatial distribution of Gaussians. All adjustment signals come from currently observed bands and the rendering process, without using full-spectrum test ground truth.

4 EXPERIMENTS

4.1 DATASETS AND SETUP

Real dataset. We use three multi-view sequences from HSI-SC-NeRF: Apple, Maize, and Pear. Each scene contains approximately 90% training views and the remaining views for testing. Images have a resolution of 256×256 with $C = 128$ spectral bands. During training, each view observes only $K = 16$ bands selected by rotating masks, while full spectra are used only for evaluation.

Synthetic dataset. Synthetic experiments use three RRUFF-228 scenes: `0000_00`, `0009_00`, and `0218_01`. Images have a resolution of 128×128 with $C = 228$ bands, and each training view observes $K = 16$ bands by default. This dataset is further used for matched-parameter comparisons, fixed-Gaussian-count comparisons, ablations, and spectral observation robustness studies.

Training and evaluation. All models are trained independently for 30,000 steps with random seed 3407. We use Adam optimization with learning rates of 2×10^{-3} for local appearance and 10^{-3} for shared spectral bases. The material-aware regularization weights are set to $\lambda_{\text{smooth}} = 5 \times 10^{-6}$ and $\lambda_{\text{sparse}} = 10^{-4}$. Evaluation is performed on full test spectra using PSNR, SSIM, RMSE, SAM, SID, parameter counts, and full-spectrum rendering speed. All compared methods follow the same spectral observation protocol and use identical training/test splits to ensure that performance differences mainly reflect representation and optimization strategies rather than differences in supervision.

4.2 BASELINES

Gaussian baselines. We use FastGS (Ren et al., 2026), MCMC-GS (Kheradmand et al., 2024), and SteepGS (Wang et al., 2025) as Gaussian baselines under the same sparse-HSI protocol. Direct-HSI is included as a spectral representation control, which independently stores a C-dimensional spectrum for each anisotropic Gaussian and is compared with SB-HyperGS under matched parameter budgets and fixed Gaussian counts.

Neural field baselines. We use HS-NeRF (Chen et al., 2024), Mip-NeRF 360 (Barron et al., 2022), Nerfacto (Tancik et al., 2023), and TensoRF (Chen et al., 2022). All methods use the same splits, camera parameters, and spectral observation masks, with supervision restricted to measured bands.

Table 1: Main results on synthetic scenes, macro-averaged with equal weight across three scenes. Parameter counts are in millions (M).

Method	Params/M	FPS↑	PSNR↑	SSIM↑	RMSE↓	SAM↓	SID↓
FastGS-HSI	97.918	43.08	28.366	0.9083	0.0387	0.0667	0.1602
MCMC-GS-HSI	1097.700	8.134	29.1066	0.9150	0.03537	0.05588	0.16576
SteepGS-HSI	1242.718	7.469	28.8451	0.9155	0.03637	0.05827	0.09401
TensoRF-HSI	19.460	7.29	29.883	0.9171	0.0321	0.0349	0.0736
Nerfacto-HSI	13.933	20.71	26.765	0.8811	0.0466	0.0362	0.0702
HS-NeRF-HSI	13.913	2.67	15.125	0.5192	0.1764	0.6320	0.8950
Mip-NeRF 360-HSI	9.037	4.642	29.6690	0.9169	0.03293	0.03383	0.14959
SB-HyperGS	2.075	477.87	29.407	0.9188	0.0343	0.0464	0.0641

4.3 SYNTHETIC SCENES

Table 1 reports macro-averaged results over three RRUFF-228 scenes. SB-HyperGS achieves 29.407 dB PSNR and 0.9188 SSIM with 2.075M parameters. Compared with Gaussian baselines,

it achieves better overall reconstruction performance, while compared with Mip-NeRF 360-HSI it uses fewer parameters and achieves higher SSIM with similar PSNR. These results demonstrate that compact material representations can reduce spectral redundancy while maintaining effective hyperspectral reconstruction. Figure 3 further provides qualitative comparisons on held-out synthetic views, where all methods are rendered under the same visualization setting.

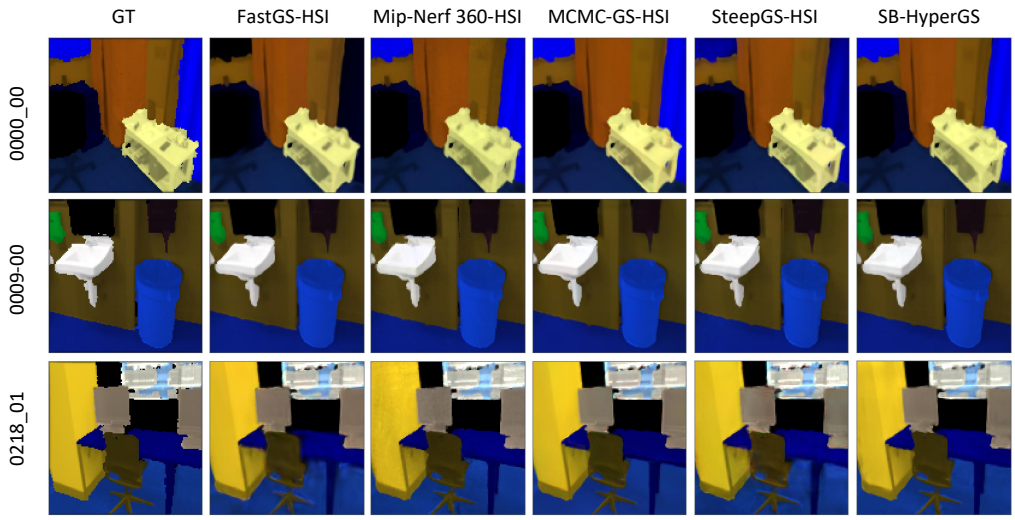

Figure 3: Qualitative comparison on synthetic scenes. All methods are visualized using the same pseudocolor mapping and normalization.

4.4 Real Scenes

Table 2 reports macro-averaged results over the three HSI-SC-NeRF scenes. SB-HyperGS achieves 20.274 dB PSNR and 0.5270 SSIM with 2.071M parameters, providing a better quality–parameter balance than the compared methods. Due to differences in material structures and spectral variations, reconstruction difficulty varies across scenes; therefore, macro-averaged results are reported.

Table 2: Main results on real scenes, macro-averaged with equal weight across three scenes. Parameter counts are in millions (M).

Method	Params/M	FPS↑	PSNR↑	SSIM↑	RMSE↓	SAM↓	SID↓
FastGS-HSI	141.281	55.23	17.587	0.4268	0.1321	0.2624	0.0997
MCMC-GS-HSI	617.700	14.181	18.6059	0.4550	0.11795	0.23835	0.08508
SteepGS-HSI	593.920	21.929	18.1285	0.4529	0.12428	0.24330	0.08728
TensoRF-HSI	19.290	2.79	15.928	0.3535	0.1603	0.2766	0.1203
Nerfacto-HSI	13.846	5.24	15.211	0.4027	0.1770	0.2176	0.0689
HS-NeRF-HSI	13.845	1.22	15.947	0.4376	0.1691	0.1966	0.0668
Mip-NeRF 360-HSI	9.024	1.176	19.0651	0.4661	0.11604	0.24579	0.09939
SB-HyperGS	2.071	434.17	20.274	0.5270	0.1022	0.1634	0.0490

4.5 Matched-Parameter Comparison

Direct-HSI independently learns a nonnegative C-dimensional spectrum for each anisotropic Gaussian. It retains the same geometric representation and view-dependent brightness modulation as SB-HyperGS, but does not use shared material spectral bases or local mixtures. Its parameter count is

$$P_{\text{Direct}} = (C+14)G_{\text{D}} + C, \qquad G_{\text{D}}^{\text{budget}} = \left\lfloor \frac{P_{\text{SB}} - C}{C+14} \right\rfloor. \tag{12}$$

Table 3 compares Direct-HSI and SB-HyperGS under matched parameter budgets. Both methods use fixed topology and identical sparse spectral observations. SB-HyperGS achieves higher macro-averaged PSNR across all budgets on both synthetic and real scenes. This comparison isolates the effect of spectral representation efficiency by removing the influence of different Gaussian optimization strategies, demonstrating that compact spectral representations can release capacity for spatial modeling and improve the quality–parameter trade-off.

Table 3: Comparison with Direct-HSI under matched parameter budgets. Results are macro-averaged over three scenes.

Data	Method	G	Parameters	PSNR↑	SSIM↑	RMSE↓	SAM↓	SID↓
Synthetic scenes								
	SB-HyperGS	8,192	527,089	28.229	0.8847	0.0394	0.0581	0.0858
	Direct-HSI	2,177	527,062	26.394	0.8283	0.0496	0.0749	0.1156
	SB-HyperGS	16,384	1,043,185	28.752	0.9007	0.0370	0.0529	0.0747
	Direct-HSI	4,309	1,043,006	27.819	0.8649	0.0411	0.0527	0.0824
	SB-HyperGS	32,768	2,075,377	28.703	0.9009	0.0372	0.0554	0.0764
	Direct-HSI	8,574	2,075,136	28.530	0.8864	0.0378	0.0445	0.0690
Real scenes								
	SB-HyperGS	8,192	522,289	21.080	0.5667	0.0940	0.1486	0.0400
	Direct-HSI	3,677	522,262	20.195	0.5282	0.0997	0.1994	0.0632
	SB-HyperGS	16,384	1,038,385	20.881	0.5598	0.0951	0.1487	0.0398
	Direct-HSI	7,311	1,038,290	19.698	0.5142	0.1058	0.2123	0.0703
	SB-HyperGS	32,768	2,070,577	20.393	0.5546	0.1007	0.1495	0.0402
	Direct-HSI	14,580	2,070,488	19.107	0.4991	0.1129	0.2303	0.0807

4.6 Fixed Gaussian Count

Table 4 compares the two representations with identical Gaussian counts $G \in \{8192, 16384, 32768\}$. Both methods use fixed topology and the same initial geometry. This setting removes the influence of different primitive numbers and directly evaluates whether the compact representation can preserve reconstruction quality with reduced spectral storage overhead.

Table 4: Comparison with Direct-HSI at fixed Gaussian counts. Results are macro-averaged over three scenes.

Data	Method	G	Parameters	PSNR↑	SSIM↑	RMSE↓	SAM↓	SID↓
Synthetic scenes								
	SB-HyperGS	8,192	527,089	28.229	0.8847	0.0394	0.0581	0.0858
	Direct-HSI	8,192	1,982,692	28.408	0.8834	0.0384	0.0456	0.0719
	SB-HyperGS	16,384	1,043,185	28.752	0.9007	0.0370	0.0529	0.0747
	Direct-HSI	16,384	3,965,156	28.976	0.8993	0.0358	0.0404	0.0612
	SB-HyperGS	32,768	2,075,377	28.703	0.9009	0.0372	0.0554	0.0764
	Direct-HSI	32,768	7,930,084	29.141	0.9043	0.0352	0.0397	0.0582
Real scenes								
	SB-HyperGS	8,192	522,289	21.080	0.5667	0.0940	0.1486	0.0400
	Direct-HSI	8,192	1,163,392	19.532	0.5126	0.1074	0.2164	0.0722
	SB-HyperGS	16,384	1,038,385	20.881	0.5598	0.0951	0.1487	0.0398
	Direct-HSI	16,384	2,326,656	19.080	0.4993	0.1134	0.2306	0.0806
	SB-HyperGS	32,768	2,070,577	20.393	0.5546	0.1007	0.1495	0.0402
	Direct-HSI	32,768	4,653,184	18.735	0.4908	0.1174	0.2374	0.0846

At the same Gaussian count, Direct-HSI achieves higher PSNR on synthetic scenes due to greater local spectral fitting flexibility, while SB-HyperGS uses substantially fewer parameters. On real

scenes, SB-HyperGS provides consistent quality improvements with lower spectral storage overhead. These results demonstrate that shared spectral representations improve parameter efficiency while preserving competitive reconstruction quality.

4.7 Ablations and Robustness

Module ablations. Table 5 evaluates the two main components. Removing material-aware optimization sets $\lambda_{\text{smooth}} = \lambda_{\text{sparse}} = 0$, while removing spatial refinement disables Gaussian growth and relocation with $G = 15981$ fixed during training. Removing material-aware optimization reduces PSNR from 29.407 dB to 29.061 dB, and removing spatial refinement reduces it to 28.747 dB, demonstrating the contribution of both components. The results indicate that spectral regularization and spatial capacity adjustment provide complementary improvements under sparse spectral supervision.

Table 5: Module ablations on synthetic scenes, macro-averaged with equal weight across three scenes.

Setting	PSNR↑	SSIM↑	RMSE↓	SAM↓	SID↓
Full model	**29.407**	**0.9188**	**0.0343**	**0.0464**	**0.0641**
w/o material-aware optimization	29.061	0.9160	0.0356	0.0537	0.0714
w/o spatial refinement	28.747	0.9062	0.0370	0.0523	0.0708

Number of observed bands. Table 6 compares different observation sparsity levels. As K increases from 8 to 32, PSNR improves from 28.913 dB to 29.670 dB, while the model maintains competitive performance under the sparse $K = 8$ setting.

Table 6: Robustness to spectral observations on synthetic scenes. Each row is trained using the corresponding sparse observations and macro-averaged with equal weight across three scenes; $K = 16$ with mask seed 0 is the default setting.

K	Mask seed	PSNR↑	SSIM↑	RMSE↓	SAM↓	SID↓
8	0	28.913	0.9150	0.0366	0.0749	0.0776
16	0	29.407	0.9188	0.0343	0.0464	0.0641
32	0	29.670	0.9205	0.0331	0.0357	0.0612
16	1	29.474	0.9198	0.0340	0.0452	0.0599
16	2	29.501	0.9195	0.0339	0.0434	0.0591

Spectral masks. With $K = 16$ fixed, three mask seeds are used to vary observed wavelength combinations. The similar PSNR values (29.407, 29.474, and 29.501 dB) indicate stable reconstruction across different sparse spectral observation patterns. This suggests that the proposed representation does not rely on a specific wavelength arrangement and can effectively exploit complementary observations from different views.

5 Conclusion

We presented SB-HyperGS, a material-efficient hyperspectral Gaussian Splatting framework that replaces independent spectral storage with shared material spectral bases and Gaussian-specific local mixing coefficients. This reduces redundant per-primitive spectral overhead and uses finite representation capacity more effectively for spatial structure modeling. Building on this representation, material-aware optimization and adaptive spatial refinement further coordinate spectral representation and spatial capacity allocation under sparse spectral supervision. Overall, the results show that SB-HyperGS substantially improves the parameter efficiency of hyperspectral Gaussian representations while maintaining competitive reconstruction performance, providing an effective approach to compact hyperspectral 3D representation from sparse observations.

AI USE STATEMENT

During the preparation of this manuscript, generative AI tools were used as auxiliary tools for language polishing, improving presentation clarity, and checking mathematical expressions and derivation procedures. All generated suggestions were critically examined and independently verified by the authors. The authors remain solely responsible for the research methodology, experimental results, interpretations, and the final manuscript.

REFERENCES

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5470–5479, 2022.

Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. TensoRF: Tensorial radiance fields. In *European Conference on Computer Vision*, pp. 333–350, 2022.

Gerry Chen, Sunil Kumar Narayanan, Thomas Gautier Ottou, Benjamin Missaoui, Harsh Muriki, Cédric Pradalier, and Yongsheng Chen. Hyperspectral neural radiance fields. *arXiv preprint arXiv:2403.14839*, 2024. URL https://arxiv.org/abs/2403.14839.

Alex Hanson, Allen Tu, Vasu Singla, Mayuka Jayawardhana, Matthias Zwicker, and Tom Goldstein. PUP 3D-GS: Principled uncertainty pruning for 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5949–5958, 2025.

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. doi: 10.1145/3592433. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3D Gaussian Splatting as markov chain monte carlo. In *Advances in Neural Information Processing Systems*, volume 37, pp. 80965–80986, 2024. doi: 10.52202/079017-2573.

Ru Li, Jia Liu, Guanghui Liu, Shengping Zhang, Bing Zeng, and Shuaicheng Liu. SpectralNeRF: Physically based spectral rendering with neural radiance field. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 3154–3162, 2024. doi: 10.1609/aaai.v38i4.28099.

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. In *European Conference on Computer Vision*, 2020. URL https://www.matthewtancik.com/nerf.

Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. RegNeRF: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5480–5490, 2022.

Fabian Perez, Sara Rojas, Carlos Hinojosa, Hoover Rueda-Chacón, and Bernard Ghanem. UnMix-NeRF: Spectral unmixing meets neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 26284–26293, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Perez_UnMix-NeRF_Spectral_Unmixing_Meets_Neural_Radiance_Fields_ICCV_2025_paper.html.

Shiwei Ren, Tianci Wen, Yongchun Fang, and Biao Lu. FastGS: Training 3D Gaussian Splatting in 100 seconds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26094–26103, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Ren_FastGS_Training_3D_Gaussian_Splatting_in_100_Seconds_CVPR_2026_paper.html.

Seungjoo Shin, Jaesik Park, and Sunghyun Cho. Locality-aware gaussian compression for fast and high-quality rendering. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=dHYwfV2KeP.

Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. doi: 10.1145/3588432.3591516.

Christopher Thirgood, Oscar Mendez, Erin Chao Ling, Jon Storey, and Simon Hadfield. HyperGS: Hyperspectral 3D Gaussian Splatting. *arXiv preprint arXiv:2412.12849*, 2024. URL https://arxiv.org/abs/2412.12849.

Peihao Wang, Yuehao Wang, Dilin Wang, Sreyas Mohan, Zhiwen Fan, Lemeng Wu, Ruisi Cai, Yu-Ying Yeh, Zhangyang Wang, Qiang Liu, and Rakesh Ranjan. Steepest descent density control for compact 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26663–26672, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wang_Steepest_Descent_Density_Control_for_Compact_3D_Gaussian_Splatting_CVPR_2025_paper.html.

Jiawei Yang, Marco Pavone, and Yue Wang. FreeNeRF: Improving few-shot neural rendering with free frequency regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8254–8263, 2023.