FREQUENCY-DECOUPLED COMPRESSION FOR 3D GAUSSIAN SPLATTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing anchor-based 3DGS compression methods primarily focus on spatial context construction, while the interaction between coding order and frequency statistics remains less explored. This ordering can affect probability estimation and, consequently, compression performance. In context-based entropy coding, each anchor estimates its probability based on the causal history. However, due to the coding order, the finite causal window may contain many statistically inconsistent anchors, which degrades rate estimation accuracy. We propose FDC-GS, a frequency-decoupled compression framework that organizes anchors into frequency-consistent levels. This organization provides a coding-level frequency state to the entropy model and reorganizes the composition of the decoded history, yielding more statistically relevant causal context and tighter probability estimates without redesigning the context backbone. FDC-GS first applies saliency-guided decomposition to separate a low-frequency base from informative high-frequency residuals, and then aligns these components with coarse and refinement levels for base-to-refinement coding. Finally, a dynamic R–D counterbalancing schedule increases the rate penalty as high-frequency supervision grows, controlling the bitrate growth induced by detail refinement. Based on widely used hash-grid and triplane contexts, FDC-GS achieves average BD-PSNR gains of 0.249 dB over HAC++ and 0.443 dB over TC-GS across four datasets.

1 INTRODUCTION

3D Gaussian Splatting (3DGS) has emerged as a powerful representation for real-time novel view synthesis (Kerbl et al., 2023), but its explicit storage of millions of Gaussian primitives imposes a significant storage burden (Ali et al., 2025; Bagdasarian et al., 2025). To address this, anchor-based 3DGS compression methods have been developed, which organize Gaussians into structured anchors and employ entropy coding to reduce bitstream size (Lu et al., 2024; Chen et al., 2024; 2025). In these methods, the accuracy of probability estimation is a key factor in the achievable rate-distortion (R–D) performance (Chen et al., 2025; Liu et al., 2024). Consequently, substantial efforts have been devoted to improving context construction, such as hash grids (Chen et al., 2024; 2025), triplanes (Zhan et al., 2025; Wang et al., 2025), cross-attribute autoregressive contexts (Zhan et al., 2025; Liu et al., 2024), and other conditional entropy modeling mechanisms (Ballé et al., 2018; Minnen et al., 2018; Ballé et al., 2016; Minnen & Singh, 2020; Cheng et al., 2020; He et al., 2022).

However, existing anchor-based 3DGS compression methods primarily improve what spatial or hierarchical context is constructed (Chen et al., 2024; 2025; Wang et al., 2024b; Zhan et al., 2025; Liu et al., 2024), while the interaction between coding order and frequency statistics remains less explored. As illustrated in Fig. 1, an ordering that mixes anchors with different frequency characteristics can place statistically mismatched references in a finite causal window. Low- and high-frequency residuals can also exhibit different statistics (Mallat, 1989; Simoncelli & Olshausen, 2001), yet current codecs generally do not expose an explicit frequency state to the likelihood model. This motivates organizing base and refinement coding so that the causal history is more frequency-consistent and the entropy model can condition on the corresponding frequency state.

Motivated by this observation, we propose FDC-GS, which aligns anchor coding levels with frequency characteristics. This alignment provides frequency-consistent causal histories and an explicit coding-level frequency state for likelihood modeling. FDC-GS realizes this design through saliency-guided

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.

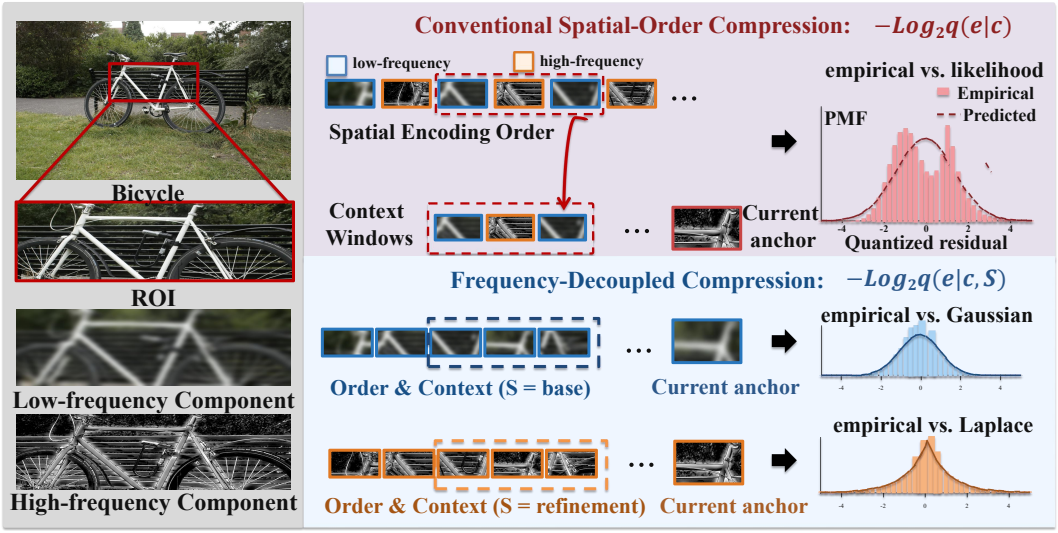


Figure 1: Motivation of frequency-decoupled coding. Conventional spatial or hierarchical ordering can mix frequency-inconsistent anchors in the causal context. Frequency-decoupled ordering aligns coding levels with frequency states, providing frequency-consistent reference histories for more accurate likelihood modeling.

frequency decomposition, progressive frequency-to-level alignment, and dynamic R–D counterbalancing, while preserving the original spatial context backbone and decoder.

FDC-GS realizes this principle via three components. First, after separating the low-frequency base and high-frequency residuals, a saliency-guided decomposition applies a saliency mask to the high-frequency component to retain details while suppressing noise-like fluctuations. Second, to align frequency components with coding levels, we introduce annealing-based supervision that progressively shifts from the low-frequency base to high-frequency refinement. Third, to prevent excessive high-frequency supervision at deeper levels from causing anchor growth and rate increase, we introduce R–D counterbalancing that increases the rate penalty as high-frequency supervision grows. As an entropy-modeling realization of this hierarchy, the coding-level state selects the conditional likelihood family: base residuals use a Gaussian prior and refinement residuals use a Laplace prior, with their parameters predicted from the original spatial context. The framework is built on widely used hash-grid and triplane contexts (Chen et al., 2025; Wang et al., 2025), preserving their context lookup and decoder unchanged. Our contributions are:

- We formulate coding level as an explicit frequency-state variable for anchor-based 3DGS entropy coding, decoupling frequency organization from the underlying spatial context.
- We couple saliency-guided frequency supervision with coarse-to-fine coding: frequency-to-level annealing progressively introduces detail, while a synchronized R–D schedule increases rate pressure to control its compression cost.
- Across four datasets and two context backbones, FDC-GS improves average BD-PSNR by 0.249 dB over HAC++ and 0.443 dB over TC-GS. Ablations validate the three components and the associated level-dependent likelihood selection, while the complexity study characterizes the added training and coding cost.

2 Related Work

3DGS Compression and Context Modeling. Prior compact 3DGS methods reduce storage through pruning, quantization, vector codebooks, and structured parameter representations (Lee et al., 2024; Niedermayr et al., 2024; Fan et al., 2024; Girish et al., 2024; Navaneet et al., 2024; Morgenstern et al., 2024; Tran et al., 2025). Anchor-based codecs build on Scaffold-GS (Lu et al., 2024), which groups nearby Gaussians around anchors and predicts Gaussian attributes from anchor features. HAC (Chen et al., 2024) and HAC++ (Chen et al., 2025) use hash-grid contexts to predict quantization and

entropy-model parameters; ContextGS (Wang et al., 2024b) introduces hierarchical anchor coding; CAT-3DGS (Zhan et al., 2025) uses multi-scale triplanes with spatial and channel-wise autoregressive coding; and HEMGS (Liu et al., 2024) combines a hyperprior, autoregressive context, and variable-rate prediction. These methods mainly improve spatial, hierarchical, or attribute context construction. FDC-GS is complementary: it keeps the spatial context backbone but makes the coding level an explicit frequency state that affects both causal ordering and the conditional likelihood family.

Progressive and Variable-Rate Compression. Learned image compression has long explored progressive and variable-rate coding for bandwidth- and storage-adaptive transmission (Toderici et al., 2017; Choi et al., 2019; Cui et al., 2021; Lee et al., 2022; Xu et al., 2026). In 3DGS, PCGS (Chen et al., 2026) jointly controls anchor quantity and quality to produce a progressive bitstream; GoDe (Di Sario et al., 2025) organizes Gaussians into a gradient-sensitivity hierarchy; LapisGS (Shi et al., 2025) constructs cumulative layers for progressive rendering; HEMGS (Liu et al., 2024) supports variable-rate compression with a single model; and RDO-Gaussian (Wang et al., 2024a) performs end-to-end R–D optimization with dynamic pruning and entropy-constrained vector quantization. These methods primarily target scalability or rate control. FDC-GS instead uses the coding hierarchy to expose frequency state to the entropy model and to make the decoded history more frequency-consistent.

Frequency-Aware Optimization for 3DGS. Frequency-aware signals have also been used to improve 3DGS reconstruction and optimization (Li et al., 2024; Xia et al., 2026; Zhang & Lin, 2024; Zhu et al., 2025). FreGS (Zhang et al., 2024) applies low-to-high frequency regularization during reconstruction and densification, while CAT-3DGS (Zhan et al., 2025) uses view-frequency—how often a primitive contributes across training views—for masking. FDC-GS addresses a different role of frequency: image-frequency statistics are aligned with coding levels and used to condition entropy modeling, with the goal of improving compression rather than reconstruction alone.

3 METHOD

As illustrated in Fig. 2, FDC-GS augments the standard 3D Gaussian training, coding, and rendering pipeline, shown in gray, with three frequency-aware components detailed in Modules A–C. Module A performs saliency-guided frequency decomposition to separate the training signal into low- and high-frequency components. Module B progressively aligns these components with the coarse-to-refinement coding hierarchy and increases the rate constraint as high-frequency supervision grows. Module C combines the resulting reconstruction, frequency, and rate terms into the overall optimization objective. In this way, FDC-GS reorganizes frequency information throughout training and coding while preserving the original spatial context backbone and rendering pipeline.

3.1 PROBLEM FORMULATION

In context-based entropy coding, each anchor is coded from a finite causal history. For a fixed spatial context c, let e denote a quantized residual and $s \in \{l, h\}$ its frequency state. Define $\pi_s(c) = p(s \mid c)$ and $p_s(e \mid c) = p(e \mid c, s)$. The conditional residual distribution is then:

$$p(e \mid c) = \sum_{s \in \{l,h\}} \pi_s(c)\, p_s(e \mid c), \tag{1}$$

where p_l and p_h denote residual distributions of the low- and high-frequency states, respectively.

When the coding order mixes frequency states within the same causal context, a shared likelihood $q(e \mid c)$ must model the resulting mixture distribution. Its cross-entropy is:

$$H(p, q \mid c) = -\mathbb{E}_{p(e|c)} \left[\log q(e \mid c)\right]. \tag{2}$$

Let q^*_{single} denote the optimal shared likelihood within a fixed model class, and let q^*_l and q^*_h denote the corresponding optimal state-conditioned likelihoods. Allowing the likelihood model to condition on the frequency state increases its modeling flexibility and therefore cannot increase the minimum achievable cross-entropy:

$$H(p, q^*_{\text{single}} \mid c) \geq \sum_{s \in \{l,h\}} \pi_s(c)\, H(p_s, q^*_s \mid c). \tag{3}$$

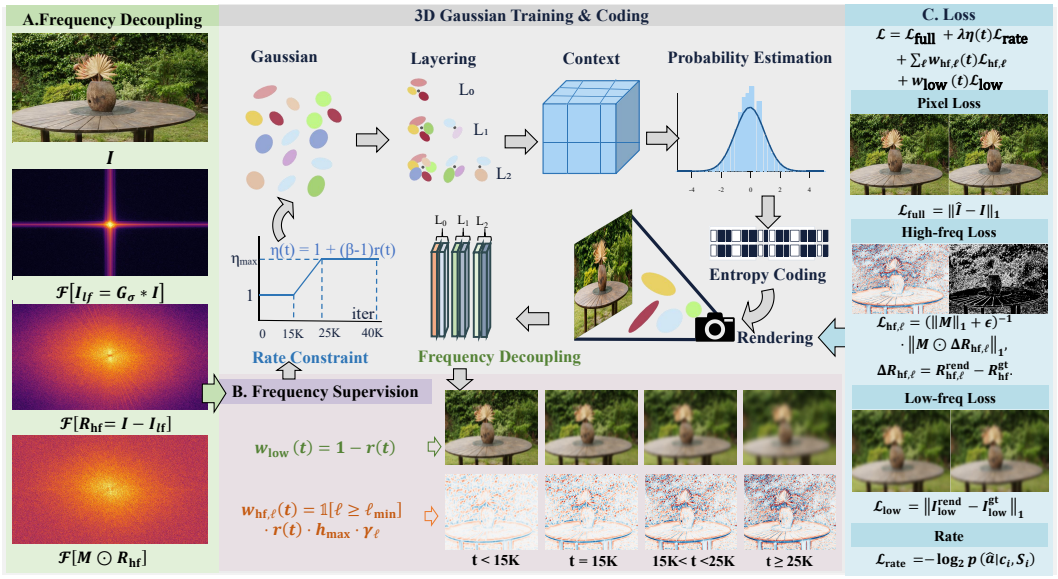


Figure 2: **Overview of FDC-GS.** The gray region shows the overall 3D Gaussian training, coding, and rendering pipeline, while Modules A–C detail the key components introduced by FDC-GS. **A**: Frequency decomposition derives low- and high-frequency supervision signals. **B**: Frequency-to-level alignment progressively couples these signals with the coding hierarchy, together with dynamic R–D counterbalancing. **C**: The resulting reconstruction, frequency, and rate terms are combined into the final training objective.

Averaging Eq. (3) over the spatial context c yields the corresponding expected cross-entropy relation. The gap in Eq. (3) represents the potential penalty of modeling frequency-mixed residuals with a shared likelihood. Under ideal entropy coding, lower cross-entropy corresponds to a lower expected rate, although this relation does not guarantee a lower realized bitrate in a learned codec.

FDC-GS uses frequency-decomposed image supervision to associate the base and refinement levels with low- and high-frequency objectives, respectively. The coding level therefore serves as a deterministic frequency-state proxy s, requiring no additional signaling. We model base residuals with a Gaussian likelihood and refinement residuals with a Laplace likelihood, while their parameters remain predicted from the original spatial context c.

3.2 Saliency-Guided Frequency Decomposition

For both the ground-truth image I and rendered prediction $\hat{I}$, we use the same Gaussian low-pass filter G_σ (Mallat, 1989; Simoncelli & Olshausen, 2001) to obtain low- and high-frequency components:

$$\begin{cases} X_{\text{low}} = G_\sigma * X, \\ R_{\text{hf}} = X - X_{\text{low}}, \end{cases} \qquad X \in \{I, \hat{I}\}. \tag{4}$$

The corresponding components are denoted by $(I^{\text{gt}}_{\text{low}}, R^{\text{gt}}_{\text{hf}})$ and $(I^{\text{pred}}_{\text{low}}, R^{\text{pred}}_{\text{hf}})$, respectively. The high-frequency residual contains both informative edges and small fluctuations in smooth regions. Supervising all residual pixels equally can allocate optimization effort and coding capacity to visually weak structures. We therefore compute Sobel gradient magnitudes from the ground-truth high-frequency residual. Specifically, Sobel responses are computed independently for each RGB channel and averaged across channels to obtain a scalar saliency value per pixel:

$$S = \frac{1}{3} \sum_{k \in \{r,g,b\}} \sqrt{\left(K_x * R^{\text{gt}}_{\text{hf},k}\right)^2 + \left(K_y * R^{\text{gt}}_{\text{hf},k}\right)^2}. \tag{5}$$

We retain only the most salient fraction of pixels:

$$M = \mathbb{I}\left[S \geq \tau_\rho(S)\right], \tag{6}$$

where $\tau_\rho(S)$ denotes the $(1-\rho)$ quantile of the pixel-wise saliency magnitudes in each training image, such that approximately a fraction ρ of pixels is retained.

The resulting high-frequency supervision is

$$\mathcal{L}_{\mathrm{hf}} = \frac{\left\| M \odot \left(R_{\mathrm{hf}}^{\mathrm{pred}} - R_{\mathrm{hf}}^{\mathrm{gt}} \right) \right\|_1}{\|M\|_1 + \epsilon}, \tag{7}$$

while the low-frequency component is supervised by

$$\mathcal{L}_{\mathrm{low}} = \left\| I_{\mathrm{low}}^{\mathrm{pred}} - I_{\mathrm{low}}^{\mathrm{gt}} \right\|_1. \tag{8}$$

The saliency mask M is used only for training supervision and introduces no bitstream overhead.

3.3 Progressive Frequency-to-Level Alignment

The anchor-based representation is organized into multiple coarse-to-fine coding levels $\ell \in \{0, \ldots, L-1\}$ obtained through monotonically increasing level masks. Anchors are encoded and decoded in ascending level order, from the base level to successive refinement levels. Within each level, anchors retain the Morton order of the original backbone to preserve local spatial locality. Thus, FDC-GS changes the global composition of the causal history through frequency-aligned level partitioning while retaining a locally efficient spatial order within each level. We align frequency with this hierarchy instead of applying the same objective to every level. A progress variable $r(t) \in [0,1]$ increases during training:

$$r(t) = \mathrm{clip}\left(\frac{t - t_0}{t_1 - t_0}, 0, 1 \right). \tag{9}$$

We define the supervision weights as:

$$w_{\mathrm{low}}(t) = 1 - r(t), \qquad w_{\mathrm{hf},\ell}(t) = \mathbb{I}[\ell \geq \ell_{\min}]\, r(t)\, h_{\max}\, \gamma_\ell, \tag{10}$$

where

$$\gamma_\ell = \frac{\ell - \ell_{\min} + 1}{L - \ell_{\min}}.$$

The same frequency state controls the entropy model. To instantiate the frequency-dependent statistics exposed by this hierarchy, we use a Gaussian likelihood for base-level residuals and a heavier-tailed Laplace likelihood for refinement residuals:

$$\mathcal{P}_s = \begin{cases} \mathcal{N}(\mu, \sigma^2), & s = l, \\ \mathrm{Laplace}(\mu, b), & s = h, \end{cases} \tag{11}$$

where the distribution parameters are predicted solely from the unchanged spatial context c_i, while the level state only selects the likelihood family. This separates two roles: the spatial backbone determines where context comes from, while the coding-level frequency state determines how the resulting residual is modeled.

3.4 Dynamic R–D Counterbalancing

High-frequency supervision improves local detail but can also increase the incentive to create or retain more anchors during refinement. To counteract this effect, we tie the rate pressure to the same training-progress variable used for progressive frequency supervision:

$$\eta(t) = 1 + (\beta - 1) r(t), \qquad \beta \geq 1. \tag{12}$$

Here, $r(t)$ denotes the normalized training progress, and $\eta(t)$ gradually increases the strength of the rate penalty as high-frequency refinement becomes more prominent. Let $\mathcal{L}_{\mathrm{full}}$ denote the original rendering objective and $\boldsymbol{\theta}_i = g(\mathbf{c}_i)$ the distribution parameters predicted by the unchanged spatial-context model. The estimated rate is:

$$\mathcal{L}_{\mathrm{rate}} = -\sum_i \log_2 p_{s_i}(\hat{\mathbf{a}}_i \mid \boldsymbol{\theta}_i(\mathbf{c}_i)). \tag{13}$$

It denotes the estimated coding rate of the quantized anchor representations. Our overall training objective is defined for the level ℓ_t activated at iteration t. Here, $\mathcal{L}_{\mathrm{hf},\ell_t}$ evaluates Eq. (7) using the rendering of the anchors assigned to this single level, rather than a cumulative reconstruction over levels. The objective is:

$$\mathcal{L} = \mathcal{L}_{\mathrm{full}} + \lambda\eta(t)\mathcal{L}_{\mathrm{rate}} + w_{\mathrm{low}}(t)\mathcal{L}_{\mathrm{low}} + w_{\mathrm{hf},\ell_t}(t)\mathcal{L}_{\mathrm{hf},\ell_t}. \tag{14}$$

The objective jointly balances full-image reconstruction, rate, low-frequency supervision, and the high-frequency loss of the active refinement level. Losses from multiple levels are not accumulated in one iteration. We refer to this mechanism as dynamic R–D counterbalancing: as the auxiliary supervision becomes increasingly detail-oriented, explicit rate pressure is increased simultaneously. $\eta(t)$ discourages excessive anchor growth rather than explicitly constraining anchor count; its benefit is therefore evaluated through measured R–D performance.

3.5 Application to Hash-Grid and Triplane Context

FDC-GS is designed as a backbone-compatible framework for anchor-based 3DGS codecs. We demonstrate two applications: FDC-Hash on HAC++ (Chen et al., 2025) and FDC-Triplane on TC-GS (Wang et al., 2025). In both cases, the anchor parameterization, spatial context coordinates and lookup, quantization configuration, and rendering decoder are inherited from the host codec. FDC-GS adds only (i) the decomposition losses of Eqs. (7) and (8), (ii) coding-level conditioning of the likelihood family as in Eq. (11), and (iii) the shared schedules of Eqs. (10) and (12). For the hash-grid context, c_i is the original multi-resolution hash query; for the triplane context, it is the original projected multi-scale plane query. By keeping the context lookup and rendering decoder unchanged, this design isolates frequency decoupling from backbone modifications and allows us to test its transfer across two distinct context parameterizations.

4 Experiments

4.1 Experimental Setup

Implementation details. We implement FDC-GS in PyTorch based on the official HAC++ (Chen et al., 2025) and TC-GS (Wang et al., 2025) codebases, which use hash-grid and triplane contexts, respectively. We denote the two instantiations as FDC-Hash and FDC-Triplane. FDC-Hash is trained for 40K iterations on Mip-NeRF 360, Tanks & Temples, and Deep Blending and for 30K iterations on BungeeNeRF, where longer optimization may lead to overfitting due to the scale discrepancy between training and test views, as observed in PCGS (Chen et al., 2026). Full details about parameters are provided in Appendix A.

Datasets. We conduct experiments on four datasets: Mip-NeRF 360 (Barron et al., 2022), Tanks & Temples (Knapitsch et al., 2017), Deep Blending (Hedman et al., 2018), and BungeeNeRF (Xiangli et al., 2022). These benchmarks span scenes with substantially different scales and frequency characteristics, providing a diverse test bed for compression performance.

Baselines. We compare against representative 3DGS compression methods, including HAC (Chen et al., 2024), HAC++ (Chen et al., 2025), ContextGS (Wang et al., 2024b), HEMGS (Liu et al., 2024), CAT-3DGS (Zhan et al., 2025), and TC-GS (Wang et al., 2025). We additionally include the progressive compression methods PCGS (Chen et al., 2026) and GoDe (Di Sario et al., 2025). HAC++ and TC-GS are the direct comparisons for FDC-Hash and FDC-Triplane, respectively.

Metrics. We report PSNR, SSIM (Wang et al., 2004), LPIPS (Zhang et al., 2018), bitstream size, Bjontegaard-delta bitrate (BD-BR), and Bjontegaard-delta PSNR (BD-PSNR) (Bjontegaard, 2001). Unless otherwise stated, the BD metrics are computed for FDC-GS relative to the corresponding baseline over their common R–D interval. BD-BR measures the average percentage change in bitrate at equal PSNR, whereas BD-PSNR measures the average PSNR change in dB at equal bitrate. Therefore, negative BD-BR and positive BD-PSNR indicate better compression performance.

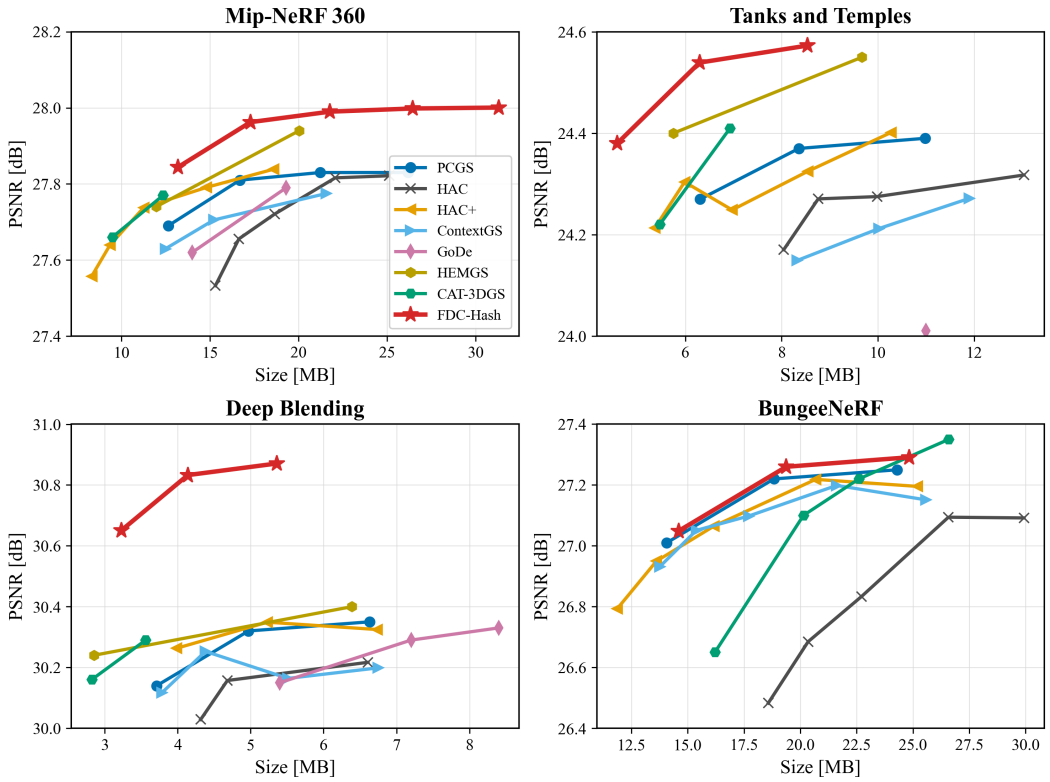


Figure 3: R–D comparison of FDC-Hash and existing 3DGS compression methods on four datasets.

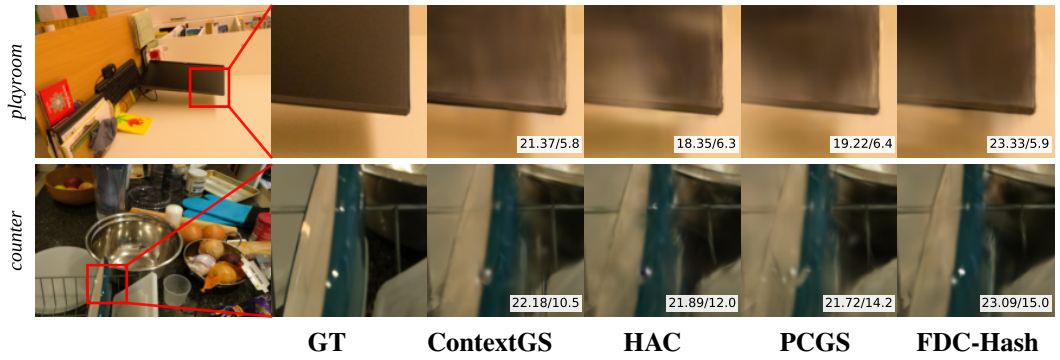


Figure 4: Qualitative comparison on *playroom* and *counter*: PSNR (dB) and size (MB).

4.2 Main Results

4.2.1 FDC-Hash: FDC-GS Applied to Hash

Fig. 3 shows that FDC-Hash consistently shifts the R–D frontier toward higher PSNR across all four datasets. Against its backbone, HAC++, FDC-Hash improves BD-PSNR by 0.122, 0.268, 0.533, and 0.072 dB on Mip-NeRF 360, Tanks & Temples, Deep Blending, and BungeeNeRF, respectively, yielding an average gain of 0.249 dB. Detailed results are provided in Appendix Table 3. These results show that frequency decoupling improves R–D performance while retaining the same context.

The advantage also holds against progressive codecs. Relative to PCGS, which likewise uses a HAC++-style hash context, FDC-Hash obtains 0.246 dB average BD-PSNR over all four datasets. Over the three datasets with valid BD-PSNR overlap, FDC-Hash improves over CAT-3DGS by 0.285 dB on average. GoDe provides only one valid comparison, on which the gain is 0.238 dB. Compared with the remaining strong compressors, FDC-Hash gains 0.456 dB over HAC, 0.341 dB

Table 1: Layer-wise R–D comparison between TC-GS and FDC-TRIPLANE.

Dataset	Method	S_0		S_1		S_2		BD-PSNR (dB) ↑
		PSNR ↑	Size ↓	PSNR ↑	Size ↓	PSNR ↑	Size ↓	
Mip-NeRF 360	TC-GS	27.690	13.85	27.679	**16.27**	27.649	**18.76**	–
	FDC-TRIPLANE	**27.750**	**13.64**	**27.870**	17.90	**27.893**	22.47	0.155
Tanks & Temples	TC-GS	23.795	**5.09**	23.899	**6.41**	23.896	**8.11**	–
	FDC-TRIPLANE	**24.120**	6.34	**24.215**	8.48	**24.229**	11.06	0.280
Deep Blending	TC-GS	29.951	**2.51**	30.011	**2.70**	29.924	**3.37**	–
	FDC-TRIPLANE	**30.018**	3.29	**30.172**	4.48	**30.206**	5.97	0.090
BungeeNeRF	TC-GS	25.391	**12.82**	25.808	**17.46**	25.847	**22.53**	–
	FDC-TRIPLANE	**26.836**	13.29	**27.028**	17.77	**27.078**	23.10	1.247

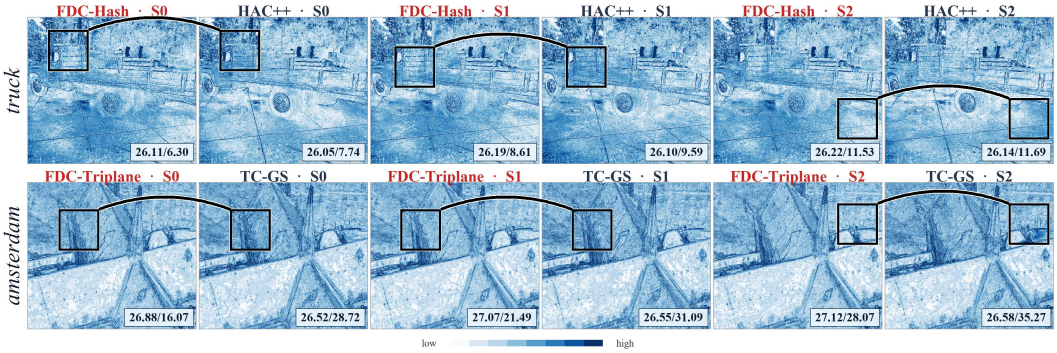


Figure 5: Level-wise comparison of FDC-Hash/HAC++ and FDC-Triplane/TC-GS. Values report PSNR (dB) and size (MB).

over ContextGS, and 0.222 dB over HEMGS across the three datasets. Fig. 4 presents a qualitative comparison among the above methods. The highlighted crops illustrate the visual differences corresponding to the quantitative improvements, with FDC-GS achieving the highest reported PSNR among the shown reconstructions. Additional results are provided in Appendix Fig. 7.

4.2.2 FDC-TRIPLANE: FDC-GS APPLIED TO TRIPLANE

FDC-TRIPLANE inherits TC-GS plane resolution, feature dimensionality, and decoder, introducing only the frequency-decoupled framework. As shown in Table 1, its R–D curve is above that of TC-GS on every dataset, with BD-PSNR gains of 0.155, 0.280, 0.090, and 1.247 dB on Mip-NeRF 360, Tanks & Temples, Deep Blending, and BungeeNeRF, respectively. The gain is substantially larger on BungeeNeRF than on the other datasets. One possible explanation is its broader range of spatial scales. Complete SSIM and LPIPS measurements are given in Appendix Tables 4 and 5. Together with the conclusion in Section 4.2.1, the proposed FDC-GS improves both hash and triplane contexts rather than relying on a specific one.

Scene-Level Evidence and Size Savings. Fig. 5 compares the level-wise performance at S_0–S_2. Gains are largest at the base level and narrow toward refinement. On *truck*, FDC-HASH reduces size from 7.74 to 6.30 MB (18.6%) at the lowest rate while slightly raising PSNR; on *amsterdam*, FDC-TRIPLANE raises PSNR by 0.36–0.54 dB and reduces size by 20.4–44.0%. Together with Tables 3 and 1, these results show that frequency decoupling improves the measured R–D trade-off across both backbones. The controlled conditional-NLL analysis in Appendix G further isolates the effect of coding organization under fixed samples, residuals, features, and model capacity. Frequency-consistent grouping reduces refinement-level NLL by 0.11–0.28 bits/dimension, directly linking the reorganized causal history to more accurate conditional probability estimation. Appendix Fig. 12 further shows that the high-frequency response of the decoded reconstruction increases from the base level to successive refinement levels.

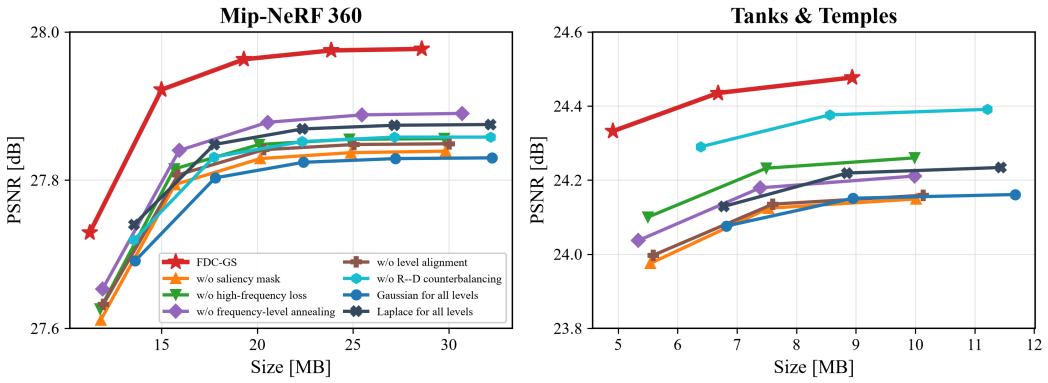


Figure 6: R–D curves of FDC-GS, its component ablations, and its single-family likelihood variants.

Table 2: Average complexity over four datasets. ✓ denotes multi-rate support.

Method	Multi-rate	Anchor count (10^3)	Training (s)	Encoding (s)	Decoding (s)	Rendering (FPS)
TC-GS		321.1	2541	39.5	39.9	128.7
FDC-Triplane	✓	398.0	3931	18.3	23.5	177.7
HAC++		403.7	2043	9.3	15.1	194.7
FDC-Hash	✓	408.6	4629	19.9	20.2	191.3
ContextGS		403.0	5020	77.5	67.7	87.0
PCGS	✓	400.8	2626	16.6	22.1	187.9

4.3 Ablation Studies

We ablate the components associated with frequency decomposition, progressive level assignment, and dynamic R–D counterbalancing. We also replace the level-dependent likelihood selection with a single Gaussian or Laplace family at all levels. Fig. 6 shows their R–D curves on two representative datasets, while Appendix Table 6 reports BD-BR over all four datasets. Each ablated variant is evaluated relative to the full FDC-Hash model, so positive BD-BR indicates degradation. All ablated variants degrade average R–D performance, while the -0.03% result on BungeeNeRF is effectively neutral. Removing the saliency mask and high-frequency loss increases average BD-BR by 16.95% and 15.87%, respectively. Disabling frequency-level annealing results in a 9.29% increase, while removing level alignment increases BD-BR by 19.32%. The largest degradation is caused by removing R–D counterbalancing, which increases average BD-BR by 27.04%. The all-Gaussian and all-Laplace variants increase average BD-BR by 20.20% and 29.66%, respectively, confirming that level-dependent likelihood selection is a useful realization of the frequency-consistent hierarchy.

4.4 Complexity Comparison

Table 2 summarizes average complexity. FDC-Hash keeps anchor count and rendering speed close to HAC++ (408.6K vs. 403.7K; 191.3 vs. 194.7 FPS), while its higher training and encoding cost is associated with processing the full hierarchy within one scene. FDC-Triplane increases anchor count and training time but reduces encoding and decoding time and raises rendering FPS relative to TC-GS. Thus, the added cost is concentrated in training and, for FDC-Hash, encoding rather than rendering. Both FDC-GS variants also encode and decode faster than ContextGS and support multiple rates from a single model; per-dataset measurements are reported in Appendix Table 11.

5 Conclusion

We presented FDC-GS, a frequency-decoupled framework that aligns coding levels with frequency-aware supervision and entropy modeling. FDC-GS consistently improves the R–D performance of both hash-grid and triplane backbones across four datasets, with ablations confirming the contribution of its key components. These results suggest that frequency organization provides a complementary design dimension to spatial context modeling for anchor-based 3DGS compression.

AI USE STATEMENT

In this work, generative AI tools were used for language polishing, structural editing, and LaTeX organization. They were not used to generate experimental measurements. All AI-assisted text, equations, citations, and presentation choices were reviewed and verified by the authors. The authors take responsibility for the final content of the submission.

ETHICS STATEMENT

This work studies compression of 3D scene representations and does not involve human subjects or collection of new personally identifiable data. Experiments use established public novel-view-synthesis benchmarks under their respective licenses. Compressed 3D reconstructions may inherit privacy or copyright concerns from source imagery; deployment should follow the governance requirements of the underlying data.

REPRODUCIBILITY STATEMENT

Section 3 specifies the decomposition, frequency-to-level schedule, conditional priors, and R–D objective. The current artifacts include the R–D calculation scripts, per-scene operating-point PSNR, SSIM, LPIPS, encoded size, and the complexity measurements reported in Table 2. The anonymous release will include the calculation scripts, exact overlap and interpolation convention, configurations, and complete bitstream accounting.

REFERENCES

Muhammad Salman Ali, Chaoning Zhang, Marco Cagnazzo, Giuseppe Valenzise, Enzo Tartaglione, and Sung-Ho Bae. Compression in 3D gaussian splatting: A survey of methods, trends, and future directions. *arXiv preprint arXiv:2502.19457*, 2025.

Milena T. Bagdasarian, Paul Knoll, Yi-Hsin Li, Florian Barthel, Anna Hilsmann, Peter Eisert, and Wieland Morgenstern. 3DGS.zip: A survey on 3D gaussian splatting compression methods. *Computer Graphics Forum*, pp. e70078, 2025. doi: 10.1111/cgf.70078. URL `https://w-m.github.io/3dgs-compression-survey/`.

Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.

Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, 2018.

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

Gisle Bjontegaard. Calculation of average PSNR differences between RD-curves. Technical report, ITU-T SG16/Q6, VCEG-M33, 2001.

Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. HAC: Hash-grid assisted context for 3D gaussian splatting compression. In *European Conference on Computer Vision*, 2024.

Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. Hac++: Towards 100x compression of 3d gaussian splatting. *arXiv preprint arXiv:2501.12255*, 2025.

Yihang Chen, Mengyao Li, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. Pcgs: Progressive compression of 3d gaussian splatting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 3111–3119, 2026.

Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7936–7945. IEEE, 2020.

Yoojin Choi, Mostafa El-Khamy, and Jungwon Lee. Variable rate deep image compression with a conditional autoencoder. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3146–3154. IEEE, 2019.

Ze Cui, Jing Wang, Shangyin Gao, Tiansheng Guo, Yihui Feng, and Bo Bai. Asymmetric gained deep image compression with continuous rate adaptation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10527–10536. IEEE, 2021.

Francesco Di Sario, Riccardo Renzulli, Marco Grangetto, Akihiro Sugimoto, and Enzo Tartaglione. Gode: Gaussians on demand for progressive level of detail and scalable compression. *arXiv preprint arXiv:2501.13558*, 2025.

Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. *Advances in neural information processing systems*, 37:140138–140158, 2024.

Sharath Girish, Kamal Gupta, and Abhinav Shrivastava. Eagles: Efficient accelerated 3d gaussians with lightweight encodings. In *European Conference on Computer Vision*, pp. 54–71. Springer, 2024.

Dailan He, Ziming Yang, Weikun Peng, Rui Ma, Hongwei Qin, and Yan Wang. Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5708–5717. IEEE, 2022.

Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics*, 37(6), 2018.

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.

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.

Jae-Han Lee, Seungmin Jeon, Kwang Pyo Choi, Youngo Park, and Chang-Su Kim. Dpict: Deep progressive image compression using trit-planes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16113–16122, 2022.

Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3d gaussian representation for radiance field. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21719–21728. IEEE, 2024.

Han Li, Shaohui Li, Wenrui Dai, Chenglin Li, Junni Zou, and Hongkai Xiong. Frequency-aware transformer for learned image compression. In *International conference on learning representations*, volume 2024, pp. 30447–30465, 2024.

Lei Liu, Zhenghao Chen, and Dong Xu. Hemgs: A hybrid entropy model for 3d gaussian splatting data compression. *arXiv preprint arXiv:2411.18473*, 2024.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20654–20664, 2024.

Stephane G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

David Minnen and Saurabh Singh. Channel-wise autoregressive entropy models for learned image compression. In *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 3339–3343. IEEE, 2020.

David Minnen, Johannes Ballé, and George D. Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Wieland Morgenstern, Florian Barthel, Anna Hilsmann, and Peter Eisert. Compact 3d scene representation via self-organizing gaussian grids. In *European Conference on Computer Vision*, pp. 18–34. Springer, 2024.

KL Navaneet, Kossar Pourahmadi Meibodi, Soroush Abbasi Koohpayegani, and Hamed Pirsiavash. Compgs: Smaller and faster gaussian splatting with vector quantization. In *European Conference on Computer Vision*, pp. 330–349. Springer, 2024.

Simon Niedermayr, Josef Stumpfegger, and Rüdiger Westermann. Compressed 3d gaussian splatting for accelerated novel view synthesis. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10349–10358. IEEE, 2024.

Yuang Shi, Simone Gasparini, Géraldine Morin, and Wei Tsang Ooi. Lapisgs: Layered progressive 3d gaussian splatting for adaptive streaming. In *International Conference on 3D Vision (3DV)*, pp. 991–1000, 2025. doi: 10.1109/3DV66043.2025.00096.

Eero P. Simoncelli and Bruno A. Olshausen. Natural image statistics and neural representation. *Annual Review of Neuroscience*, 24:1193–1216, 2001.

George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 5306–5314, 2017.

Hoang-Nhat Tran, Francesco Di Sario, Gabriele Spadaro, Giuseppe Valenzise, and Enzo Tartaglione. Rave: Rate-adaptive visual encoding for 3d gaussian splatting. *arXiv preprint arXiv:2512.07052*, 2025.

Henan Wang, Hanxin Zhu, Tianyu He, Runsen Feng, Jiajun Deng, Jiang Bian, and Zhibo Chen. End-to-end rate-distortion optimized 3d gaussian representation. *arXiv preprint arXiv:2406.01597*, 2024a.

Taorui Wang, Zitong Yu, and Yong Xu. TC-GS: Tri-plane based compression for 3D gaussian splatting. In *IEEE International Conference on Multimedia and Expo*, 2025.

Yufei Wang, Zhihao Li, Lanqing Guo, Wenhan Yang, Alex C. Kot, and Bihan Wen. ContextGS: Compact 3D gaussian splatting with anchor level context model. In *Advances in Neural Information Processing Systems*, volume 37, 2024b.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

Yichong Xia, Yimin Zhou, Jinpeng Wang, and Bin Chen. Towards efficient low-rate image compression with frequency-aware diffusion prior refinement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 10942–10950, 2026.

Yuanbo Xiangli, Linning Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering. In *European Conference on Computer Vision*, 2022.

Hao Xu, Xiaolin Wu, and Xi Zhang. Improving 3d gaussian splatting compression by scene-adaptive lattice vector quantization. *IEEE Transactions on Image Processing*, 2026.

Yu-Ting Zhan, Cheng-Yuan Ho, Hebi Yang, Yi-Hsin Chen, Jui Chiu Chiang, Yu-Lun Liu, and Wen-Hsiao Peng. CAT-3DGS: A context-adaptive triplane approach to rate-distortion-optimized 3DGS compression. In *International Conference on Learning Representations*, 2025.

Jiahui Zhang, Fangneng Zhan, Muyu Xu, Shijian Lu, and Eric Xing. FreGS: 3D gaussian splatting with progressive frequency regularization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21424–21433, 2024.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

Yuefeng Zhang and Kai Lin. End-to-end optimized image compression with the frequency-oriented transform: Y. zhang, k. lin. *Machine Vision and Applications*, 35(2):27, 2024.

Lingyu Zhu, Xiangrui Zeng, Bolin Chen, Peilin Chen, Yung-Hui Li, and Shiqi Wang. Leveraging diffusion knowledge for generative image compression with fractal frequency-aware band learning. *arXiv preprint arXiv:2503.11321*, 2025.

A Implementation Details

Both instantiations build on the official HAC++ (Chen et al., 2025) and TC-GS (Wang et al., 2025) releases. The anchor parameterization, spatial context lookup, rendering decoder, quantization configuration, optimizer, learning-rate schedule, and densification/pruning settings are inherited from those releases. The coding-level frequency state described in Sec. 3.3 selects the likelihood family, while the original entropy head predicts its parameters from the spatial context. The FDC-GS hyperparameters are shared across backbones without per-dataset tuning or per-backbone retuning. The frequency decomposition uses a Gaussian low-pass with $\sigma = 2.0$ and retains the top $\rho = 0.3$ saliency responses; frequency annealing starts after $t_0 = 15$K iterations and completes at $t_1 = 25$K; the maximum high-frequency weight is $h_{\max} = 0.2$, the entropy multiplier is bounded by $\beta = 2.0$, and the minimum high-frequency level is $\ell_{\min} = 1$. Furthermore, base-level residuals use a Gaussian family and refinement levels a Laplace family. Within a method–dataset setting, all rate points share one training protocol and differ only in the rate-control configuration. Coding uses a single trained model that produces the complete base-to-refinement hierarchy, and the reported bitstream sizes and coding times cover that complete hierarchy.

B Additional Qualitative Comparisons

Figures 8–11 extend the main-paper comparison to additional scenes from all four benchmarks. In every figure, each row shows the rendered view and selected region, followed by the ground-truth crop and results from ContextGS, HAC, HAC++, PCGS, TC-GS, and FDC-GS (Ours). Values below the reconstructions are scene-level PSNR (dB) and model size (MB); red borders highlight the ground truth and our results.

Across indoor, outdoor, bounded, and large-scale scenes, FDC-GS preserves thin structures, repeated textures, and object boundaries at smaller or comparable model sizes. The consistent behavior across the four benchmarks complements the aggregate R–D results and shows that the bitrate reduction does not come from visibly suppressing high-frequency scene content.

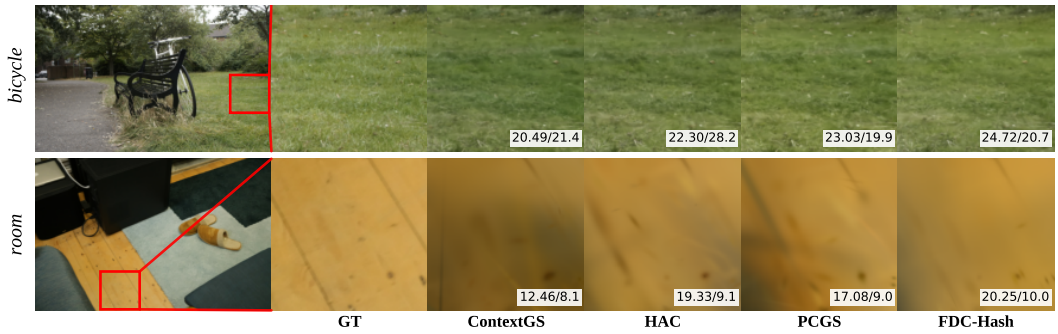


Figure 7: Additional qualitative comparison on the *bicycle* and *room* scenes: PSNR (dB) and size (MB).

C Operating Points of the R–D Curves

Tables 4 and 5 list the complete PSNR, SSIM, LPIPS, and size measurements underlying Fig. 3 and Table 1. These measurements expose the individual operating points behind the interpolated

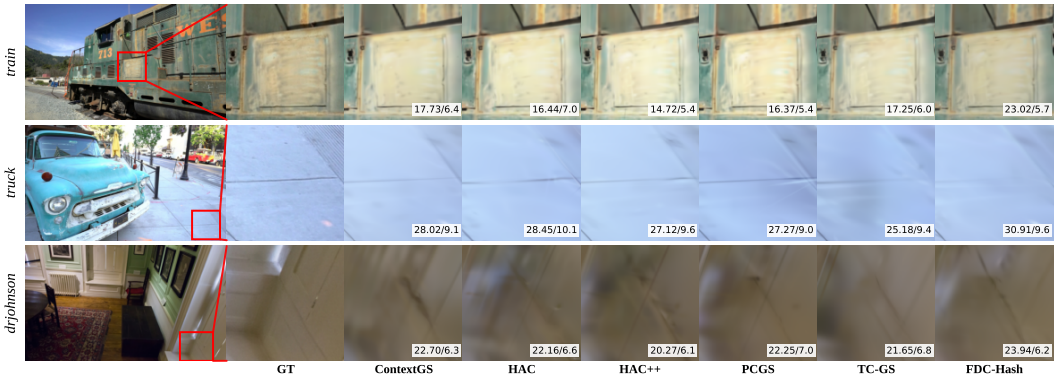


Figure 8: **Additional qualitative comparisons** on the *train*, *truck*, and *drjohnson* scenes.

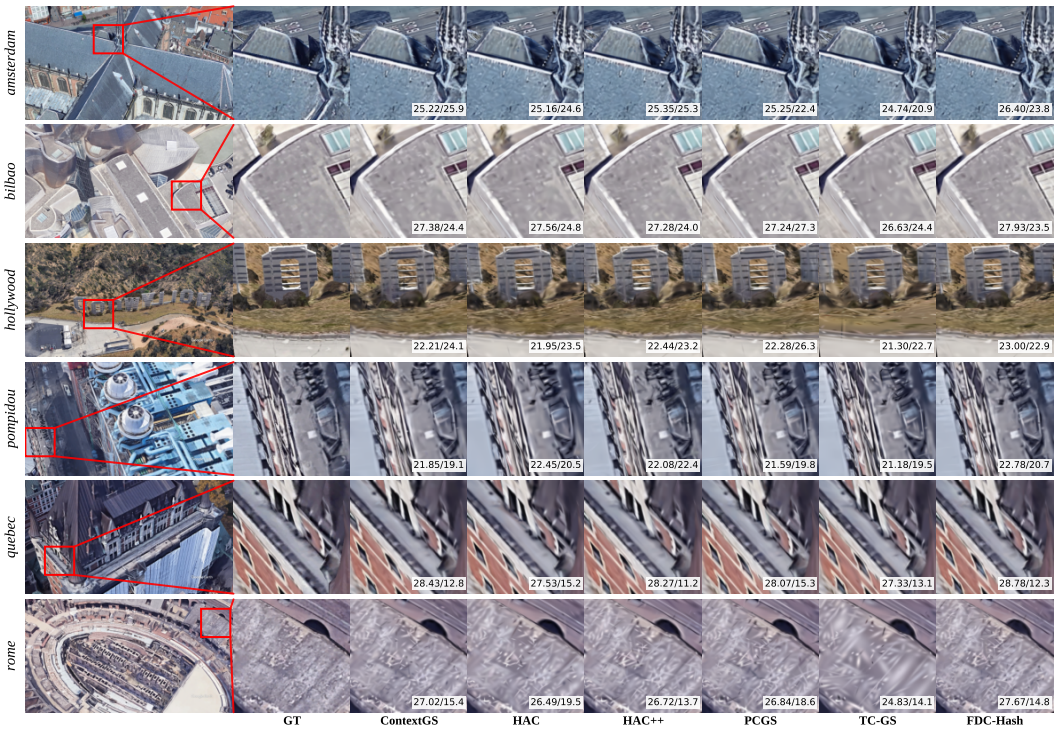


Figure 9: **Additional qualitative comparisons** on the *amsterdam*, *bilbao*, *hollywood*, *pompidou*, *quebec*, and *rome* scenes.

R–D summaries. Across successive rates, FDC-Hash and FDC-Triplane maintain competitive reconstruction quality while shifting the corresponding curves toward lower storage, confirming that the reported gains are not caused by a single favorable operating point.

The complete-coverage BD-PSNR comparison is valid for HAC, HAC++, ContextGS, and PCGS on all four datasets. FDC-Hash improves over HAC++ by 0.249 dB on average and over ContextGS by 0.341 dB, demonstrating that frequency-decoupled organization remains beneficial relative to strong spatial-context baselines. Missing BD-BR entries occur only when the measured PSNR intervals do not overlap; they are not treated as zero-valued comparisons.

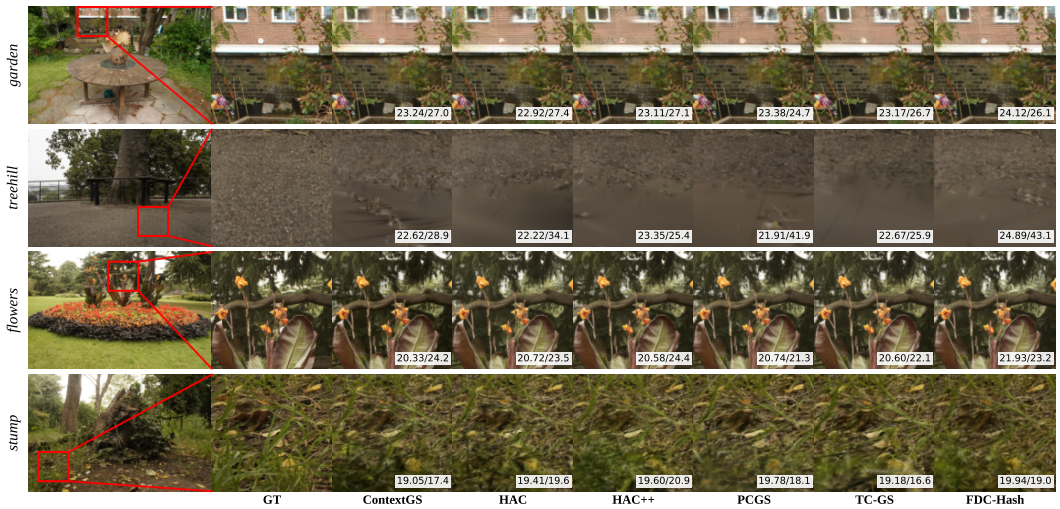


Figure 10: **Additional qualitative comparisons** on the *garden*, *treehill*, *flowers*, and *stump* scenes.

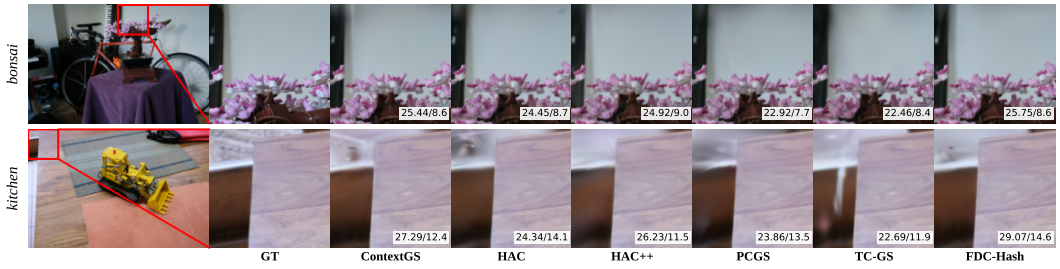


Figure 11: **Additional qualitative comparisons** on the *bicycle*, *bonsai*, *room*, and *kitchen* scenes.

Table 3: BD-BR and BD-PSNR of FDC-HASH relative to each baseline. Negative BD-BR and positive BD-PSNR indicate better performance. "–" denotes no valid common R–D interval; averages use valid datasets only.

Methods	Metrics	Mip-NeRF 360	Tanks & Temples	Deep Blending	BungeeNeRF	Average	Coverage
PCGS	BD-BR (%) ↓	–	-54.69	–	-3.71	-29.20	2/4
	BD-PSNR (dB) ↑	0.150	0.229	0.580	0.026	0.246	4/4
HAC	BD-BR (%) ↓	–	–	–	-50.67	-50.67	1/4
	BD-PSNR (dB) ↑	0.237	0.356	0.711	0.520	0.456	4/4
HAC++	BD-BR (%) ↓	–	-55.06	–	-24.36	-39.71	2/4
	BD-PSNR (dB) ↑	0.122	0.268	0.533	0.072	0.249	4/4
ContextGS	BD-BR (%) ↓	–	–	–	-25.36	-25.36	1/4
	BD-PSNR (dB) ↑	0.215	0.420	0.637	0.093	0.341	4/4
GoDe	BD-BR (%) ↓	–	–	–	–	–	0/4
	BD-PSNR (dB) ↑	0.238	–	–	–	0.238	1/4
HEMGS	BD-BR (%) ↓	-20.55	-29.31	–	–	-24.93	2/4
	BD-PSNR (dB) ↑	0.076	0.096	0.494	–	0.222	3/4
CAT-3DGS	BD-BR (%) ↓	–	-32.31	–	-22.86	-27.59	2/4
	BD-PSNR (dB) ↑	–	0.215	0.436	0.202	0.285	3/4

D OPERATING POINTS AND RUNTIME OF THE ABLATION STUDY

All ablated variants degrade average R–D performance, while the -0.03% result on BungeeNeRF is effectively neutral. Relative to the full FDC-Hash model, removing the saliency mask and high-frequency loss increases average BD-BR by 16.95% and 15.87%, respectively. Disabling frequency-level annealing results in a 9.29% increase, while removing level alignment increases

Table 4: Operating points on Mip-NeRF 360 and Tanks and Temples (Chen et al., 2026; Zhan et al., 2025; Di Sario et al., 2025).

	Mip-NeRF 360					Tanks and Temples			
Method	PSNR↑	SSIM↑	LPIPS↓	Size (MB)↓	Method	PSNR↑	SSIM↑	LPIPS↓	Size (MB)↓
PCGS	27.690	0.808	0.237	12.64	PCGS	24.270	0.850	0.184	6.31
	27.810	0.810	0.235	16.68		24.370	0.854	0.180	8.36
	27.830	0.811	0.234	21.22		24.390	0.854	0.179	10.98
	27.830	0.811	0.234	26.17		—	—	—	—
HAC	27.533	0.807	0.239	15.31	HAC	24.171	0.850	0.185	8.05
	27.655	0.809	0.235	16.63		24.270	0.852	0.181	8.76
	27.721	0.810	0.233	18.65		24.275	0.853	0.179	9.98
	27.816	0.811	0.230	22.07		24.318	0.856	0.174	13.04
	27.822	0.812	0.229	25.12		—	—	—	—
HAC++	27.557	0.803	0.253	8.32	HAC++	24.213	0.849	0.190	5.37
	27.639	0.805	0.249	9.35		24.304	0.851	0.185	5.99
	27.738	0.807	0.242	11.26		24.249	0.852	0.182	6.97
	27.791	0.810	0.235	14.80		24.325	0.855	0.176	8.54
	27.839	0.812	0.230	18.59		24.401	0.857	0.173	10.27
ContextGS	27.629	0.808	0.237	12.48	ContextGS	24.149	0.854	0.180	8.33
	27.705	0.809	0.233	15.22		24.212	0.854	0.176	10.03
	27.775	0.811	0.230	21.57		24.271	0.856	0.175	11.90
GoDe	27.620	0.803	0.297	14.00	GoDe	24.010	0.848	0.224	11.00
	27.790	0.810	0.284	19.30		—	—	—	—
HEMGS	27.740	0.807	0.249	11.96	HEMGS	24.400	0.848	0.192	5.75
	27.940	0.813	0.230	20.03		24.550	0.856	0.176	9.67
CAT-3DGS	27.660	0.807	0.249	9.49	CAT-3DGS	24.220	0.850	0.194	5.47
	27.770	0.809	0.241	12.35		24.410	0.853	0.189	6.93
TC-GS	27.297	0.791	0.180	7.82	TC-GS	23.795	0.838	0.124	5.09
	27.546	0.798	0.171	10.83		23.899	0.844	0.118	6.41
	27.690	0.802	0.165	13.85		23.896	0.847	0.114	8.11
	27.679	0.802	0.164	16.27		23.971	0.846	0.111	9.46
	27.649	0.803	0.162	18.76		—	—	—	—
FDC-Triplane	27.750	0.808	0.236	13.64	FDC-Triplane	24.120	0.845	0.186	6.34
	27.870	0.810	0.234	17.90		24.215	0.849	0.182	8.48
	27.893	0.811	0.234	22.47		24.229	0.850	0.181	11.06
	27.898	0.811	0.233	27.23		—	—	—	—
	27.899	0.811	0.233	32.25		—	—	—	—
FDC-Hash	27.844	0.809	0.238	13.19	FDC-Hash	24.380	0.847	0.192	4.58
	27.962	0.811	0.235	17.28		24.539	0.853	0.186	6.30
	27.990	0.812	0.234	21.77		24.573	0.854	0.185	8.54
	27.998	0.812	0.234	26.44		—	—	—	—
	28.001	0.812	0.234	31.29		—	—	—	—

Rows within each method are the operating points retained in Figure 3, ordered by increasing bitstream size. Removed operating points are not included.

Table 5: Operating points on Deep Blending and BungeeNeRF (Chen et al., 2026; Zhan et al., 2025; Di Sario et al., 2025).

	Deep Blending					BungeeNeRF			
Method	PSNR↑	SSIM↑	LPIPS↓	Size (MB)↓	Method	PSNR↑	SSIM↑	LPIPS↓	Size (MB)↓
PCGS	30.140	0.906	0.264	3.71	PCGS	27.010	0.873	0.206	14.07
	30.320	0.909	0.260	4.97		27.220	0.879	0.200	18.83
	30.350	0.910	0.259	6.63		27.250	0.880	0.199	24.30
HAC	30.029	0.906	0.266	4.31	HAC	26.483	0.847	0.248	18.58
	30.157	0.907	0.262	4.68		26.685	0.855	0.238	20.35
	30.217	0.909	0.255	6.61		26.833	0.862	0.228	22.72
	—	—	—	—		27.094	0.871	0.213	26.58
	—	—	—	—		27.092	0.874	0.207	29.93
HAC++	30.263	0.909	0.260	3.99	HAC++	26.793	0.858	0.234	11.87
	30.349	0.910	0.255	5.23		26.950	0.866	0.223	13.61
	30.325	0.910	0.253	6.73		27.065	0.872	0.211	16.17
	—	—	—	—		27.218	0.879	0.197	20.68
	—	—	—	—		27.195	0.882	0.188	25.20
ContextGS	30.117	0.908	0.263	3.78	ContextGS	26.931	0.866	0.222	13.76
	30.253	0.909	0.260	4.38		27.051	0.870	0.217	15.39
	30.164	0.910	0.256	5.51		27.098	0.873	0.211	17.71
	30.199	0.910	0.255	6.75		27.198	0.876	0.205	21.61
	—	—	—	—		27.151	0.876	0.201	25.58
GoDe	30.150	0.904	0.343	5.40		—	—	—	—
	30.290	0.908	0.334	7.20		—	—	—	—
	30.330	0.909	0.331	8.40		—	—	—	—
HEMGS	30.240	0.909	0.270	2.85		—	—	—	—
	30.400	0.912	0.258	6.39		—	—	—	—
CAT-3DGS	30.160	0.906	0.275	2.82	CAT-3DGS	26.650	0.859	0.226	16.21
	30.290	0.909	0.269	3.56		27.100	0.875	0.204	20.14
	—	—	—	—		27.220	0.880	0.195	22.60
	—	—	—	—		27.350	0.886	0.184	26.59
TC-GS	29.951	0.899	0.131	2.51	TC-GS	25.391	0.791	0.187	12.82
	30.011	0.902	0.127	2.70		25.808	0.813	0.165	17.46
	29.924	0.901	0.128	3.37		25.847	0.816	0.159	22.53
	—	—	—	—		25.869	0.819	0.154	24.99
	—	—	—	—		25.733	0.819	0.153	28.24
FDC-Triplane	30.018	0.902	0.268	3.29	FDC-Triplane	26.836	0.861	0.228	13.29
	30.172	0.904	0.264	4.48		27.028	0.866	0.221	17.77
	30.206	0.905	0.263	5.97		27.078	0.868	0.218	23.10
FDC-Hash	30.651	0.908	0.267	3.23	FDC-Hash	27.049	0.873	0.205	14.60
	30.832	0.911	0.264	4.14		27.260	0.879	0.199	19.37
	30.871	0.911	0.263	5.36		27.291	0.880	0.197	24.83

Rows within each method are ordered by increasing bitstream size. For Deep Blending TC-GS, the third displayed point averages the fourth rate point of *drjohnson* and the first rate point of *playroom* from `triplane002`, following the aggregation used in Table 1. Removed Figure 3 operating points are not included.

Table 6: Ablation results measured by BD-BR (%). Each ablated variant is evaluated relative to the full FDC-Hash model; therefore, a positive value indicates increased bitrate and degraded R–D performance. "–" denotes an invalid or insufficient common R–D interval.

	BD-BR (%) ↓				
Ablation Item	Mip-NeRF 360	Tanks & Temples	Deep Blending	BungeeNeRF	Average
Frequency decomposition					
w/o saliency mask	36.57	–	14.31	-0.03	16.95
w/o high-frequency loss	29.62	–	13.31	4.69	15.87
Frequency-driven layering					
w/o frequency-level annealing	26.64	–	0.61	0.63	9.29
w/o level alignment	32.96	–	22.06	2.93	19.32
R–D counterbalancing					
w/o R–D counterbalancing	35.68	56.18	2.75	13.54	27.04
Conditional likelihood model					
Gaussian for all levels	44.70	–	11.22	4.69	20.20
Laplace for all levels	32.33	–	52.42	4.22	29.66

BD-BR by 19.32%. The largest degradation is caused by removing R–D counterbalancing, which increases average BD-BR by 27.04%. Using a single likelihood family for all coding levels increases average BD-BR by 20.20% with the Gaussian model and 29.66% with the Laplace model, supporting level-dependent likelihood selection.

Tables 7–10 report the dataset-average measurements underlying these comparisons. The rate parameters are those used by the experiment launcher, and runtime is averaged over the available scene logs for each dataset and configuration. "Gaussian for all levels" and "Laplace for all levels" apply one likelihood family throughout the coding hierarchy, whereas FDC-GS selects the likelihood according to the coding level. Together, the operating points verify that the BD-BR changes persist across the progressive hierarchy rather than arising from one isolated rate.

E Per-Dataset Complexity Results

Table 11 provides the dataset-level measurements summarized by Table 2. FDC-Hash keeps the anchor count close to HAC++ on every dataset and maintains practical rendering throughput. FDC-Triplane incurs additional training cost but consistently reduces encoding and decoding time relative to TC-GS, with especially large rendering-speed gains on BungeeNeRF. These results locate the additional cost primarily in optimization rather than deployment-time rendering.

Table 7: Ablation operating points on Mip-NeRF 360.

Method	λ_s	PSNR↑	SSIM↑	LPIPS↓	Δ Size↓	Size↓	Train time	Enc. time	Dec. time
FDC-GS	4	27.729	0.805	0.242	11.25	11.25	4986	21.4	21.3
	2.5	27.922	0.809	0.238	3.75	15.00		3.9	3.3
	1	27.963	0.810	0.236	4.30	19.30		4.1	2.6
	0.5	27.975	0.811	0.236	4.56	23.86		4.1	2.9
	0.2	27.977	0.811	0.235	4.74	28.59		4.7	3.3
w/o saliency mask	4	27.611	0.806	0.239	11.82	11.82	3426	15.9	18.4
	2.5	27.794	0.809	0.235	3.88	15.70		3.5	3.1
	1	27.829	0.810	0.234	4.46	20.16		3.7	2.6
	0.5	27.837	0.811	0.234	4.72	24.88		3.7	2.9
	0.2	27.839	0.811	0.233	4.96	29.84		4.2	3.3
w/o $\mathcal{L}_{\mathrm{HF}}$	4	27.626	0.806	0.239	11.79	11.79	3332	15.8	18.1
	2.5	27.815	0.809	0.235	3.86	15.66		3.4	3.1
	1	27.848	0.810	0.235	4.44	20.10		3.5	2.6
	0.5	27.855	0.811	0.234	4.71	24.81		3.6	2.9
	0.2	27.856	0.811	0.234	4.95	29.76		4.1	3.3
w/o annealing	4	27.653	0.807	0.239	11.92	11.92	4069	11.2	18.7
	2.5	27.840	0.811	0.235	4.00	15.92		2.4	2.7
	1	27.878	0.812	0.234	4.62	20.54		2.6	2.8
	0.5	27.888	0.812	0.233	4.93	25.47		2.8	3.0
	0.2	27.890	0.812	0.233	5.23	30.70		3.1	3.5
w/o level alignment	4	27.632	0.806	0.239	11.95	11.95	3717	33.7	18.9
	2.5	27.807	0.809	0.236	3.91	15.86		6.9	4.8
	1	27.841	0.810	0.235	4.47	20.34		7.0	2.7
	0.5	27.848	0.810	0.235	4.73	25.06		7.6	3.0
	0.2	27.849	0.810	0.234	4.96	30.03		8.2	3.6
w/o entropy boost	4	27.719	0.808	0.237	13.55	13.55	3674	12.6	21.4
	2.5	27.831	0.810	0.234	4.18	17.73		2.5	2.7
	1	27.852	0.811	0.234	4.61	22.34		2.5	2.7
	0.5	27.858	0.811	0.233	4.81	27.15		2.7	3.1
	0.2	27.858	0.811	0.233	5.04	32.19		3.0	3.5

Table 8: Ablation operating points on Tanks & Temples.

Method	λ_s	PSNR↑	SSIM↑	LPIPS↓	Δ Size↓	Size↓	Train time	Enc. time	Dec. time
FDC-GS	8	24.332	0.847	0.191	4.91	4.91	7608	7.7	10.4
	4	24.435	0.852	0.186	1.77	6.68		1.9	2.0
	0.5	24.477	0.853	0.184	2.26	8.94		2.6	2.8
w/o saliency mask	8	23.975	0.846	0.189	5.54	5.54	5512	8.0	11.3
	4	24.124	0.851	0.184	1.99	7.53		2.5	2.6
	0.5	24.149	0.852	0.183	2.48	10.01		3.3	3.5
w/o $\mathcal{L}_{\mathrm{HF}}$	8	24.100	0.845	0.189	5.50	5.50	5196	6.9	10.0
	4	24.232	0.851	0.184	1.99	7.49		2.0	2.2
	0.5	24.260	0.852	0.183	2.50	9.99		2.8	3.0
w/o annealing	8	24.037	0.846	0.189	5.34	5.34	5291	7.2	10.4
	4	24.179	0.851	0.183	2.05	7.39		2.3	2.3
	0.5	24.211	0.852	0.182	2.61	9.99		3.1	3.2
w/o level alignment	8	23.997	0.846	0.188	5.59	5.59	5022	7.0	10.3
	4	24.135	0.851	0.183	2.01	7.60		2.0	2.2
	0.5	24.159	0.852	0.182	2.53	10.13		2.8	3.0
w/o entropy boost	8	24.290	0.851	0.183	6.39	6.39	5513	8.8	12.5
	4	24.376	0.854	0.179	2.18	8.56		2.6	2.7
	0.5	24.391	0.855	0.178	2.65	11.21		3.4	3.6

Table 9: Ablation operating points on Deep Blending.

Method	λ_s	PSNR↑	SSIM↑	LPIPS↓	Δ Size↓	Size↓	Train time	Enc. time	Dec. time
FDC-GS	8	29.998	0.904	0.269	3.28	3.28	2424	2.9	4.1
	4	30.220	0.907	0.264	0.95	4.23		0.7	0.7
	0.5	30.281	0.908	0.263	1.38	5.61		0.8	0.9
w/o saliency mask	8	29.970	0.904	0.269	3.68	3.68	2202	3.6	5.5
	4	30.212	0.908	0.264	1.05	4.73		0.9	1.0
	0.5	30.269	0.908	0.263	1.54	6.27		1.1	1.3
w/o $\mathcal{L}_{\text{HF}}$	8	29.956	0.9036	0.2682	3.63	3.63	1902	3.1	4.8
	4	30.212	0.908	0.263	1.04	4.67		0.8	0.9
	0.5	30.272	0.908	0.262	1.51	6.19		0.9	1.1
w/o annealing	8	30.028	0.905	0.268	3.49	3.49	2013	3.1	4.6
	4	30.253	0.908	0.263	1.03	4.52		1.0	0.9
	0.5	30.312	0.909	0.261	1.54	6.06		1.1	1.2
w/o level alignment	8	29.928	0.904	0.268	3.70	3.70	2035	3.2	4.8
	4	30.181	0.907	0.263	1.08	4.79		0.9	0.9
	0.5	30.242	0.908	0.262	1.55	6.33		1.0	1.1
w/o entropy boost	8	30.133	0.906	0.264	4.12	4.12	2021	4.0	6.1
	4	30.316	0.909	0.260	1.27	5.39		1.0	1.0
	0.5	30.348	0.909	0.259	1.66	7.05		1.1	1.3

Table 10: Ablation operating points on BungeeNeRF.

Method	λ_s	PSNR↑	SSIM↑	LPIPS↓	Δ Size↓	Size↓	Train time	Enc. time	Dec. time
FDC-GS	8	26.882	0.867	0.211	12.45	12.45	3499	17.0	19.6
	4	27.258	0.877	0.202	4.44	16.89		3.6	3.4
	0.5	27.314	0.879	0.199	5.36	22.25		4.2	3.5
w/o saliency mask	8	26.878	0.867	0.212	12.37	12.37	3749	11.9	18.3
	4	27.254	0.877	0.202	4.42	16.78		2.9	3.1
	0.5	27.310	0.879	0.199	5.34	22.13		3.4	3.8
w/o $\mathcal{L}_{\text{HF}}$	8	26.831	0.867	0.213	12.27	12.27	3687	11.6	17.2
	4	27.207	0.877	0.203	4.38	16.66		2.8	3.0
	0.5	27.264	0.878	0.200	5.34	22.00		3.2	3.7
w/o annealing	8	26.900	0.867	0.214	12.41	12.41	4135	11.2	17.4
	4	27.253	0.876	0.205	4.63	17.05		2.9	3.3
	0.5	27.308	0.878	0.202	5.76	22.81		3.5	4.2
w/o level alignment	8	26.918	0.869	0.210	12.64	12.64	3737	11.6	17.6
	4	27.238	0.877	0.202	4.48	17.12		2.8	3.0
	0.5	27.286	0.879	0.199	5.37	22.49		3.2	3.6
w/o entropy boost	8	27.031	0.873	0.205	14.67	14.67	3819	13.6	20.6
	4	27.246	0.878	0.199	4.81	19.48		2.9	3.1
	0.5	27.278	0.879	0.197	5.49	24.97		3.3	3.8

Table 11: Per-dataset complexity. A checkmark denotes multi-rate support.

Dataset	Method	Multi-rate	Anchor count (10^3)	Training (s)	Encoding (s)	Decoding (s)	Rendering (FPS)
Mip-NeRF 360	TC-GS		442.4	3193	40.3	41.2	89.6
	FDC-Triplane	✓	560.0	5192	30.4	38.3	119.7
	HAC++		566.2	2385	11.9	20.1	145.1
	FDC-Hash	✓	577.9	4986	38.3	33.4	126.4
	ContextGS		587.7	6350	98.3	88.1	71.0
	PCGS	✓	561.9	3258	22.5	31.3	135.1
Tanks & Temples	TC-GS		227.9	1810	39.2	38.5	138.9
	FDC-Triplane	✓	281.9	2917	12.1	16.1	148.7
	HAC++		292.7	1575	6.9	11.3	167.8
	FDC-Hash	✓	306.7	7608	12.2	15.2	171.1
	ContextGS		304.8	3961	85.4	71.0	88.5
	PCGS	✓	290.3	1982	12.4	16.5	163.1
Deep Blending	TC-GS		124.9	1599	14.9	14.6	210.8
	FDC-Triplane	✓	192.4	2823	7.0	9.4	208.1
	HAC++		190.1	1462	4.0	6.1	225.6
	FDC-Hash	✓	194.2	2424	4.4	5.7	247.2
	ContextGS		188.0	3419	43.6	35.0	88.4
	PCGS	✓	191.3	1790	7.2	9.2	219.1
BungeeNeRF	TC-GS		489.1	3560	63.5	65.5	75.4
	FDC-Triplane	✓	557.7	4790	23.5	30.2	234.4
	HAC++		566.0	2753	14.5	22.8	240.5
	FDC-Hash	✓	555.4	3499	24.8	26.5	220.4
	ContextGS		531.6	6350	82.6	77.0	100.0
	PCGS	✓	559.7	3473	24.4	31.3	234.5

F LEVEL-WISE FREQUENCY STATISTICS

We quantify the frequency content exposed by the decoded cumulative levels using the same Gaussian low-pass operator as in Sec. 3. For the reconstruction decoded through level s, the mean high-frequency response is:

$$E_{\mathrm{hf}}^{(s)} = \frac{1}{3HW} \left\| \hat{I}^{(s)} - G_\sigma * \hat{I}^{(s)} \right\|_1 . \tag{15}$$

Fig. 12 reports this response relative to S_0 for the two real scene-level comparisons in Fig. 5. For both context backbones, the response increases monotonically from S_0 to S_2. This observable output-level statistic verifies that successive decoded levels introduce progressively more high-frequency content; it does not assume or require a transmitted spectral label for each anchor.

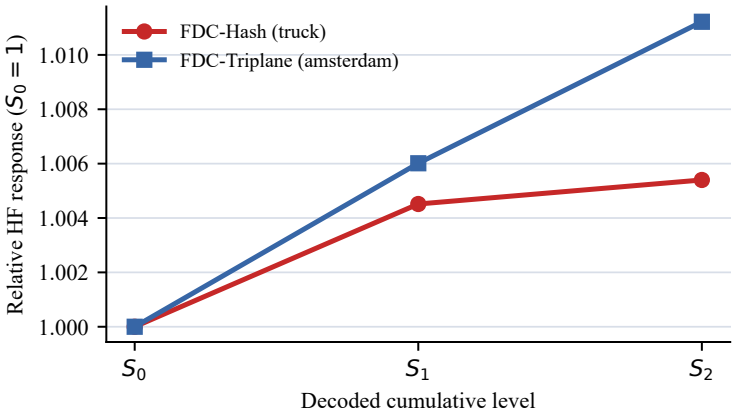


Figure 12: Level-wise high-frequency response of the cumulative reconstructions used in Fig. 5, normalized by the base level S_0.

G RESIDUAL-STATISTICS PROTOCOL

The controlled conditional-NLL analysis in Table 12 isolates whether frequency-consistent coding order improves conditional probability estimation under fixed samples and model capacity. For each scene and backbone, we collect coded residuals and evaluate a fixed-capacity diagnostic head on frozen features. We report sample kurtosis, maximum-likelihood Gaussian and Laplace fits, empirical-to-model KL divergence, cross-entropy, and near-zero mass. For histogram P and model probabilities Q:

$$D_{\mathrm{KL}}(P\|Q) = \sum_b P_b \log \frac{P_b + \epsilon}{Q_b + \epsilon}. \tag{16}$$

Table 12 reports three seeds per scene. The base group is essentially unchanged ($|\Delta| \leq 0.012$ bits/dimension), whereas refinement improves by 0.11–0.28 bits/dimension. Because the samples, residual values, frozen features, and diagnostic-head capacity are fixed, the difference isolates conditional ordering rather than sparsity or capacity. These diagnostic NLLs are not production bitrates; complete-bitstream reductions are reported in Section 4.2. Near-zero-mass checks and complete per-group accounting are included in the released analysis scripts.

Table 12: Controlled conditional NLL (bits/dimension; lower is better). Samples and diagnostic-head capacity are identical.

Scene	Group	Spatial	Ordered	Δ
bicycle	low	1.891 ± 0.020	1.880 ± 0.025	-0.011
	high	1.396 ± 0.035	1.115 ± 0.028	-0.282
counter	low	1.711 ± 0.032	1.710 ± 0.034	-0.002
	high	1.191 ± 0.087	1.078 ± 0.086	-0.112
garden	low	1.869 ± 0.044	1.862 ± 0.048	-0.007
	high	1.436 ± 0.048	1.274 ± 0.023	-0.162