# LIGHTMAP-GS: NEURAL COMPRESSION OF DYNAMIC LIGHTMAPS VIA 2D GAUSSIANS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Dynamic HDR lightmaps store changing illumination on fixed, sparsely occupied UV atlases. Their compression requires spatial capacity to follow atlas detail while sharing information across lighting states. We present Lightmap-GS, a content-specific representation that couples fixed anisotropic 2D Gaussian kernels with a factorized temporal color field. A multiresolution TriPlane decoder predicts color residuals, and tile-local normalized Top-$K$ reconstruction supports full-frame decoding and direct texel evaluation. Capacity-controlled ablations show gains of 2.17 dB from factorized features, 1.30 dB from residual color prediction, and 1.27 dB from adaptive allocation in the tested configurations. On 40 synthetic lightmap sequences, Lightmap-GS achieves 35.41 dB under a shared-scale evaluation protocol. A twelve-state, $512 \times 1024$ sequence occupies approximately 0.41 MiB versus 72 MiB in Float32, with 1.96 ms median full-frame reconstruction on an RTX 4090. Additional lighting states, external assets, and an Unreal Engine integration characterize temporal coverage and deployment costs. A matched-storage diagnostic improves the weakest fitted-state quality over temporal Gaussian alternatives, while higher-precision HEVC achieves substantially better full-atlas fidelity. Optimized neural and hardware-video comparisons further motivate evaluating selective texel access separately from bulk reconstruction and rate–distortion performance.

## 1 INTRODUCTION

Precomputed global illumination moves expensive light transport calculations out of the rendering loop. A lightmap stores the resulting illumination in a surface's UV atlas; multiple lighting states extend this workflow to changing light sources and day–night conditions. Their cost accumulates across states and assets. Even a $512 \times 1024$ RGB lightmap requires 6 MiB in Float32, or 72 MiB for twelve states, before accounting for other scene resources. The relevant question is therefore not only how accurately a representation reconstructs illumination, but also what must be stored and evaluated for the renderer's access pattern.

Two properties distinguish this signal from a sequence of natural images. First, UV charts occupy an irregular subset of a rectangular atlas, and fine illumination boundaries need uneven spatial capacity. Second, when geometry and its UV parameterization are fixed, lighting changes the signal on the surface while the atlas coordinates remain unchanged. These properties suggest sharing spatial support across lighting states and using temporal capacity to model appearance. The representation must accommodate both smooth illumination and sharp spatial boundaries within its storage budget.

Existing representations offer different trade-offs. BC6H provides hardware texture sampling at 8 bits per texel, but a conventional dense atlas retains its rectangular block allocation (Microsoft, 2009). Video codecs exploit temporal redundancy and, with hardware decoding, can reconstruct randomly requested frames quickly (Sullivan et al., 2012). Their native access unit is a decoded frame or block, whereas shader workloads may request selected texels. Coordinate networks and neural texture compression also support direct queries (Müller et al., 2022; Vaidyanathan et al., 2023); their practical cost depends strongly on feature layout, batch size, and inference implementation. A useful evaluation must distinguish these access modes rather than assume that one representation dominates them all.



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.

We investigate a different allocation of capacity: a learned set of 2D Gaussian kernels supplies the spatial basis, and a shared TriPlane network supplies time-dependent color coefficients. The spatial kernels remain fixed at decoding time. Their responses and local candidates consequently do not depend on the lighting state. This separates the cost of choosing spatial support from the cost of updating appearance, and permits queries that touch only local support. The construction builds on Gaussian image representations and factorized neural fields; its contribution is the particular representation and its evaluation for dynamic HDR atlases.

Our contributions are threefold. (1) We formulate a fixed-support, time-varying Gaussian representation with normalized local reconstruction and color residuals, making the separation between spatial allocation and temporal appearance explicit. (2) We evaluate this inductive bias with capacity-controlled comparisons against coordinate decoding, direct color prediction, and uniform valid-region allocation, as well as geometric-deformation ablations. (3) We characterize quality, storage, temporal coverage, and access cost using a synthetic benchmark and deployment studies, including optimized neural and hardware-video alternatives. The evidence supports the design choices within our representation, while also exposing cases where competing representations are preferable.

## 2 Related Work

**Content-specific neural representations.** Implicit neural representations fit continuous functions to individual signals (Sitzmann et al., 2020; Dupont et al., 2021). Adaptive coordinate networks and radial-basis fields allocate capacity according to signal structure (Martel et al., 2021; Chen et al., 2023). Instant-NGP combines multiresolution hash features with small, efficiently implemented networks (Müller et al., 2022). Lightmap-GS belongs to this content-specific family: encoding optimizes a new model for each asset. This offline setting matches light baking, with optimization cost included in the comparison. Learning a reusable encoder for unseen assets is a separate direction.

**Gaussian image representations.** 3D Gaussian splatting couples explicit primitives with efficient rasterization for novel-view synthesis (Kerbl et al., 2023). GaussianImage and Image-GS adapt 2D kernels to image representation, including efficient reconstruction and content-adaptive allocation (Zhang et al., 2024; 2025); multilevel constructions extend this approach to large images (Zhu et al., 2025). We reuse these established ideas in a different signal domain. Unlike independently fitting every lightmap, our model shares a spatial kernel set across states and predicts only color residuals. Gradient-based allocation and Gaussian kernels themselves are not claimed as new.

**Factorized temporal fields and texture compression.** HexPlane and K-Planes factor multidimensional fields into lower-dimensional feature planes (Cao & Johnson, 2023; Fridovich-Keil et al., 2023). 4D Gaussian Splatting combines factorized features with a Gaussian deformation decoder for dynamic scenes (Wu et al., 2024). Our temporal architecture follows this established pattern, specialized to $(u, v, t)$ and color-only updates on fixed UV support. Neural texture compression provides learned, random-access material representations (Vaidyanathan et al., 2023). Although the original task differs from dynamic lightmaps, spatial–temporal hash-grid adaptations are relevant alternatives and are included here. These adaptations are identified explicitly and are not presented as results from the original NTC system. BC6H and HEVC supply complementary deployment references: direct block sampling and predictive frame coding, respectively. Static texture formats can be evaluated frame by frame over an entire sequence; temporal modeling is not a prerequisite for comparison.

## 3 Method

### 3.1 Signal and Spatial Support

Let $I_s : \Omega \to \mathbb{R}^3_+$ be an HDR lightmap at state $t_s$, with a validity mask $M_s$ over an $H \times W$ atlas. We fit the observed states; reconstruction at unobserved times is not assumed accurate. Per-state reference peaks $a_s$ normalize the target to $J_s = I_s/a_s$ and are retained for restoring its scale. This inverse scaling does not make lossy reconstruction lossless. The representation contains $N$ spatial kernels, canonical colors, feature planes, a decoder, and the metadata needed to interpret them.

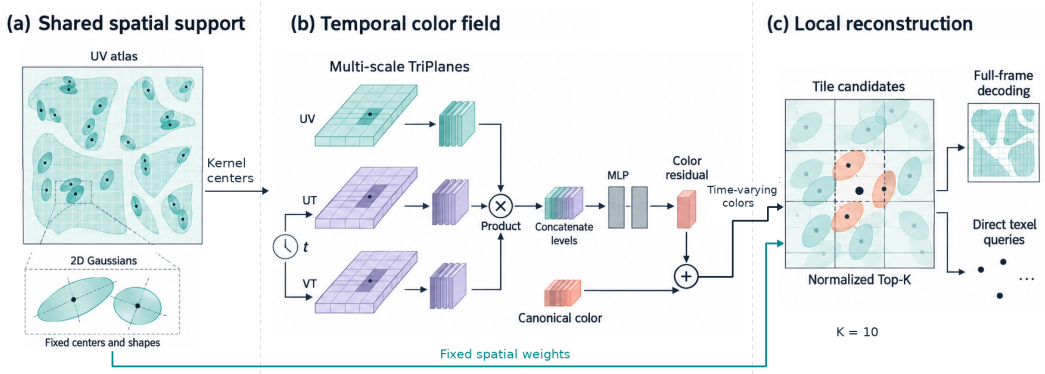



Figure 1: **Representation and access paths.** Spatial kernels are shared across lighting states. Three feature planes at each resolution predict color residuals at kernel centers. Both decoding paths use normalized local kernel weights; full-frame reconstruction and sparse texel queries have different workloads.

Kernel $i$ has center $\boldsymbol{\mu}_i$, rotation $R_i$, positive scales $s_{i1}, s_{i2}$, and canonical color $\mathbf{c}_i^0$:

$$\Sigma_i = R_i \,\mathrm{diag}(s_{i1}^2, s_{i2}^2) R_i^\top, \qquad g_i(\mathbf{x}) = \exp\!\left[-\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^\top \Sigma_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right]. \tag{1}$$

All spatial parameters are learned offline and shared across states. This is an inductive bias appropriate to fixed UV coordinates, not a claim that moving illumination boundaries are geometrically static. Changes in those boundaries must be represented by changes in the kernel coefficients.

### 3.2 Factorized temporal color

For each spatial level $\ell \in \{1, 2, 4\}$, bilinear interpolation samples three 16-channel planes. Their elementwise product forms a level feature, and concatenation joins the levels:

$$\mathbf{h}_i(t) = \mathop{\Big\|}_{\ell\in\{1,2,4\}} \left[P_{uv}^\ell(\mu_{iu}, \mu_{iv}) \odot P_{ut}^\ell(\mu_{iu}, t) \odot P_{vt}^\ell(\mu_{iv}, t)\right], \qquad \mathbf{c}_i(t) = \mathbf{c}_i^0 + f_\theta(\mathbf{h}_i(t)). \tag{2}$$

The base spatial resolution is $16 \times 16$ and the temporal resolution is 10; multiresolution scaling increases only the spatial axes. The 48-dimensional feature enters a width-32 MLP and an RGB residual head. The color-only decoder leaves $\boldsymbol{\mu}_i$ and $\Sigma_i$ unchanged across states. We examine this constraint against additional position, rotation, and scale heads in Section 4.3. Time is normalized according to the recorded state labels; the original implementation does not enforce periodicity across midnight (Appendix A).

### 3.3 Local reconstruction and access cost

We partition the atlas into $16 \times 16$ tiles. Gaussian footprint bounds associate each tile $T_j$ with a candidate list $\mathcal{S}_j$. For a query $\mathbf{x} \in T_j$, let $\mathcal{K}(\mathbf{x})$ contain the $K = 10$ largest valid responses among these candidates. Reconstruction uses

$$w_i(\mathbf{x}) = \frac{g_i(\mathbf{x})}{\epsilon_w + \sum_{k\in\mathcal{K}(\mathbf{x})} g_k(\mathbf{x})}, \qquad \widehat{J}(\mathbf{x}, t) = \sum_{i\in\mathcal{K}(\mathbf{x})} w_i(\mathbf{x}) \mathbf{c}_i(t). \tag{3}$$

An empty candidate set returns zero. This is normalized kernel blending, not depth-ordered opacity compositing or an unnormalized Gaussian sum. Tile binning still requires organizing primitive–tile intersections; eliminating depth compositing does not eliminate all indexing overhead.

The spatial weights are independent of time. Before color clipping and quantization, Equation 2 gives an exact decomposition for a fixed candidate set:

$$\widehat{J}(\mathbf{x}, t) = \underbrace{\sum_i w_i(\mathbf{x}) \mathbf{c}_i^0}_{\text{shared spatial reconstruction}} + \underbrace{\sum_i w_i(\mathbf{x}) f_\theta(\mathbf{h}_i(t))}_{\text{temporal color residual}}. \tag{4}$$

Both sums run over $\mathcal{K}(\mathbf{x})$. Clipping and quantization of the combined color need not preserve this additive identity; the spatial weights remain shared. This makes the allocation of spatial support reusable across states, while temporal reconstruction quality remains dependent on the learned coefficient field and its capacity.

After candidate selection, color accumulation costs $O(K)$ per query. Candidate selection is not free: the current kernel scans the tile list and maintains a length-$K$ array, costing $O(|\mathcal{S}_j|K)$ in the worst case. Full-frame decoding additionally evaluates temporal colors for the primitive set and reconstructs all pixels. Direct queries restrict reconstruction to requested texels, with costs depending on candidate density, batch size, and whether temporal colors and indices are already available. We measure these paths separately.

### 3.4 Allocation, optimization, and storage

**Spatial allocation.** The initial set combines gradient-weighted placement with random exploration. Subsequent additions use reconstruction error to focus capacity on poorly fitted regions. These choices favor atlas detail, while masked photometric supervision prioritizes valid texels. They do not impose a hard constraint keeping all optimized centers inside valid charts: the implementation includes random atlas samples and can allocate kernels outside them. Dense feature planes and indexing also retain overhead. We therefore describe spatial adaptivity as an empirical allocation mechanism, rather than claim zero storage on invalid UV space or prove $N \propto |\Omega_{\text{valid}}|$.

**Optimization.** We jointly optimize spatial parameters, canonical colors, feature planes, and decoder weights with Adam (Kingma & Ba, 2014). The training loss combines masked log-domain reconstruction with tone-mapped structural similarity:

$$\mathcal{L} = \frac{1}{3|M_s|} \sum_{\mathbf{x}\in M_s} \left\| \log(\widehat{J}_s^{+}(\mathbf{x}) + \epsilon) - \log(J_s(\mathbf{x}) + \epsilon) \right\|_1 + 0.1 \left[ 1 - \mathrm{SSIM}(\tau(\widehat{J}_s^{+}), \tau(J_s)) \right], \quad (5)$$

where $\widehat{J}_s^{+} = \max(\widehat{J}_s, 0)$, $\epsilon = 10^{-3}$, and $\tau(z) = z/(1+z)$. The implemented SSIM term operates on the full tone-mapped atlas, whereas the log loss selects valid texels. We do not describe both terms as strictly masked. The primary configuration uses 10,000 kernels; the inspected progressive schedule starts with 5,000 and adds 1,250 every 500 steps, four times.

**Quantization and accounting.** Straight-through quantization applies a uniform, tensor-range quantizer to Gaussian attributes during optimization. This quantizer is distinct from casting parameters to IEEE FP16. The compact representation uses reduced-precision parameter storage, with plane and network parameters included in the reported asset footprint. At 16 bits per scalar, the eight spatial/color scalars per kernel alone take 160,000 bytes; this is not the complete model. We report primary rounded assets separately from runtime allocations; the new matched-cap study instead counts complete FP16 decoder files and evaluates after reloading. No entropy-coding gain or learned bit-rate objective is assumed.

## 4 Experiments

### 4.1 Protocol and comparison scope

The primary benchmark contains 40 Blender-generated scenes with twelve $512 \times 1024$ RGB HDR lightmaps per scene. The selected hourly light-on states run from 19:00 through 06:00; a full-day generation trajectory exists, but is not the same as the evaluated twelve-state subset. The data support evaluation of discrete lighting states, not video-rate transients. Each asset is fitted separately with a reported 10,000-step budget, and encoding takes approximately 1–2 minutes on an RTX 4090. Appendix A records the configuration and implementation details.

We report PSNR, SSIM, and LPIPS (Wang et al., 2004; Zhang et al., 2018). The original protocol uses mask-adjusted perceptual scores; an additional audit instead uses a common reference scale, valid-region PSNR, a tone-mapped SSIM map, and cropped tone-mapped LPIPS. These are separate protocols (Appendix B). Size is expressed in MiB ($2^{20}$ bytes). For $B$ stored bytes and $T$ states, $\mathrm{bpp} = 8B/(HWT)$, including all atlas texels in the denominator. From the rounded 0.41 MiB

Table 1: **Primary 40-scene evaluation.** Quality uses the shared-scale audit protocol. Asterisks denote prior author adaptations; 3DGS* uses 2D kernels without depth/opacity, unlike 3DGS-T below. Asset sizes are rounded MiB; latency is synchronized full-frame reconstruction, median/p95 in ms on RTX 4090.

| Representation | PSNR ↑ | SSIM ↑ | LPIPS ↓ | MiB ↓ | ms ↓ |
|---|---|---|---|---|---|
| 3DGS* | 27.42 | .604 | .428 | 1.08 | 2.38 / 2.61 |
| GaussianImage* | 28.58 | .648 | .402 | 0.60 | 2.72 / 3.02 |
| Image-GS* | 26.94 | .876 | .139 | 0.80 | 2.19 / 2.42 |
| Lightmap-GS | 35.41 | .944 | .091 | 0.41 | 1.96 / 2.08 |

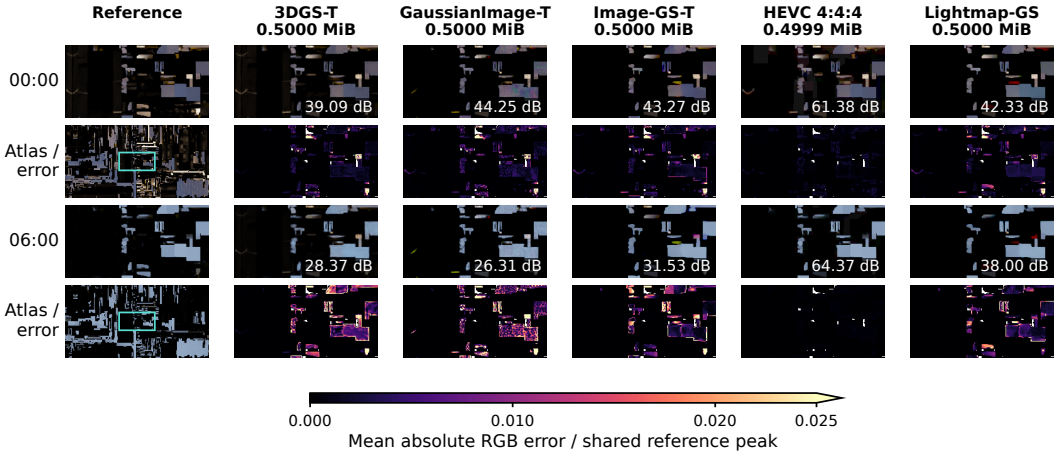



Figure 2: **Temporal reconstruction under a shared 0.5 MiB sequence cap.** Scene 17 at 00:00 and 06:00; all methods share the crop, exposure, and error scale. Cyan boxes locate the crop; inset PSNR evaluates the full valid atlas. Actual file sizes include parameters or bitstream plus HDR metadata. Every Gaussian model jointly fits all twelve states with shared spatial parameters; each displayed output is decoded from its stored asset. HEVC uses software All-Intra 4:4:4 12-bit encoding. Appendix G.1 specifies adaptations and complete storage boundaries.

asset, the corresponding rate is approximately 0.55 bpp and the Float32 reduction is approximately 176×; we do not combine that rounded size with an inconsistent 0.53 bpp claim.

The 40-scene Gaussian comparisons are author adaptations, denoted by asterisks. A five-scene subset (15, 40, 19, 17, 73) supplies BC6H and temporal diagnostics. Optimized learned and video experiments use the three locally available scenes 15, 17, and 19. Their Lightmap-GS quality reference is the original three-scene mean, 36.38 dB, rather than 35.48 dB. The optimized comparisons use an RTX 4090 D, whereas the original Lightmap-GS timings use an RTX 4090; they characterize implemented systems and do not isolate architecture-only speed effects.

### 4.2 Reconstruction Quality and Stronger Alternatives

Table 1 shows higher reconstruction quality and a smaller reported asset than the three Gaussian adaptations. The original 40-scene PSNR is 35.48 dB, with substantial scene variability (27.07–45.72 dB); the shared-scale audit gives 35.41 dB. On the five-scene subset, BC6H reaches 43.62 dB and uses 6 MiB per sequence, while Lightmap-GS uses approximately 0.41 MiB. Thus BC6H's greater storage buys greater fidelity and faster sampling. Static texture compression is a meaningful and strong baseline.

**Matched storage and fitted-state fidelity.** Table 2 and Figure 2 use new Scene 17 fits. Gaussian models share FP16 storage, loss, and 10,000 updates, without QAT. All files are decoded before evaluation; the original timing results do not apply to these assets. At 0.375 MiB, Lightmap-GS exceeds Image-GS-T in mean PSNR by 1.19 dB. At 0.5 MiB, all five complete assets differ by at

Table 2: **Matched-cap sequence quality.** Scene 17, mean/minimum PSNR (dB) over twelve fitted states after asset decoding. GI-T is GaussianImage-T; HEVC is All-Intra 4:4:4 and uses only 0.514 MiB at the 0.75 MiB cap. See Appendix G.1 for exact sizes and per-state results.

| Cap (MiB) | 3DGS-T | GI-T | Image-GS-T | HEVC | Lightmap-GS |
|---|---|---|---|---|---|
| 0.375 | 37.50/27.83 | 41.47/26.31 | 42.16/31.29 | 61.97/57.46 | 43.35/36.55 |
| 0.500 | 38.19/28.37 | 42.88/26.31 | 42.44/31.53 | 65.02/61.38 | 42.13/38.00 |
| 0.750 | 40.14/28.92 | 42.02/26.47 | 41.93/31.53 | 65.35/64.37 | 43.26/37.50 |

Table 3: **Optimized learned alternatives on three scenes.** H and L identify larger and smaller capacity settings, not certified byte-matched codecs. Sizes are exact serialized FP16 states converted to MiB; timing is network evaluation with preallocated coordinates on RTX 4090 D. The Lightmap-GS row is an original-run reference with rounded complete-asset size and separately measured full-frame timing on RTX 4090. Learned-alternative quality is evaluated before storage; these runs have no save–reload quality check.

| Representation | PSNR ↑ | State size (MiB) | Median / p95 (ms) |
|---|---|---|---|
| Lightmap-GS (reference) | 36.38 | ≈0.410 | 1.960 / 2.080 |
| Instant-NGP* H | 38.45 | 0.374 | 0.408 / 0.417 |
| Temporal-NTC* H | 46.12 | 0.386 | 0.887 / 0.913 |
| Instant-NGP* L | 36.42 | 0.280 | 0.395 / 0.405 |
| Temporal-NTC* L | 36.44 | 0.055 | 0.710 / 0.725 |

most 58 bytes. Lightmap-GS's mean/minimum PSNR is 42.13/38.00 dB, versus 42.44/31.53 for Image-GS-T and 42.88/26.31 for GaussianImage-T. Thus it improves the weakest fitted state by 6.46 and 11.68 dB, while slightly reducing mean PSNR; its mean tone-mapped SSIM also exceeds Image-GS-T (0.9886 versus 0.9866). The fixed 06:00 row illustrates this advantage. These fits do not explain the original Scene 17 result of 30.03 dB; configurations differ, so the gap is not an ablation gain. HEVC is much stronger in full-atlas fidelity (65.02 dB mean at this cap). The evidence supports more even fidelity across the fitted states within this Gaussian family, not universal rate–distortion superiority or unseen-time temporal consistency.

Table 3 evaluates two stronger learned alternatives: an Instant-NGP-style $(u, v, t)$ model and Temporal-NTC*, an NTC-inspired spatial–temporal hash-grid adaptation. Both reach high quality with fast full-grid evaluation. Temporal-NTC* takes three-dimensional $(u, v, t)$ input, and its larger setting reaches 46.12 dB on the three-scene subset. The smaller settings were selected near the *aggregate* 36.38 dB reference, rather than matched per scene. These results show that the Gaussian adaptation gains in Table 1 do not extend to general rate–distortion or full-frame speed superiority. Differences in fitting loss, time encoding, storage contents, and timing endpoints are detailed in Appendix C.

Hardware video decoding is also competitive (Appendix D). Persistent HEVC All-Intra decoding produces a GPU-resident RGB frame in 1.108/1.487 ms median/p95 and stores approximately 0.056 MiB per sequence at 36.39 dB on the three-scene diagnostic. These streams use Main10 4:2:0, whereas the software comparison uses 4:4:4. Hardware video is therefore a practical alternative for interactive full-frame retrieval. Direct texel evaluation is a different capability, shared with texture and neural-field methods, and is evaluated separately below.

**Rate, latency, and access granularity.** Table 4 makes the GOP trade-off explicit. At similar three-scene mean PSNR, HEVC GOP reduces the sequence container from 0.0556 to 0.0111 MiB relative to All-Intra, while median/p95 random-frame retrieval rises from 1.108/1.487 to 1.520/2.739 ms. H.264 exhibits the same direction of change. These measurements keep the decoder session open and end at a synchronized GPU-resident RGB frame; process startup and decoder initialization are excluded. A frame-oriented application can benefit from this operating point without requiring direct texel access. The video containers also exclude additional HDR-scale sidecars, so their sizes do not establish exact complete-codec rate matching.

Table 4: **Persistent hardware random-frame retrieval.** Three-scene diagnostic; GPU endpoint. Sizes are exact container lengths converted to MiB. Timing columns average per-scene median/p95.

| Codec | Mode | CRF | PSNR | MiB | Median / p95 (ms) |
|---|---|---|---|---|---|
| H.264 | All-Intra | 46 | 36.36 | 0.0402 | 1.162 / 1.747 |
| H.264 | GOP | 45.5 | 36.36 | 0.0136 | 2.674 / 4.910 |
| HEVC | All-Intra | 49 | 36.39 | 0.0556 | 1.108 / 1.487 |
| HEVC | GOP | 48 | 36.33 | 0.0111 | 1.520 / 2.739 |

Table 5: **Controlled representation ablations.** Original metric protocol and the same data as the primary ablation study; identical Gaussian and 10,000-step budgets. Rounded complete sizes are within 3% of the full model. These tests isolate choices within Lightmap-GS, not optimized implementations of competing codecs.

| Representation | PSNR ↑ | MiB ↓ | Difference (dB) |
|---|---|---|---|
| TriPlane + canonical/residual + adaptive allocation | 35.48 | 0.41 | — |
| Coordinate MLP instead of TriPlane | 33.31 | 0.42 | −2.17 |
| Direct temporal color instead of residual | 34.18 | 0.40 | −1.30 |
| Uniform valid-region instead of adaptive placement | 34.21 | 0.41 | −1.27 |

Per-scene results also vary: Instant-NGP* L improves Scene 17 from the original Lightmap-GS reference of 30.03 to 40.53 dB, but reduces Scene 15 from 45.72 to 38.70 dB (Appendix C). Its three-scene mean proximity therefore does not establish matched quality on individual assets. Optimized learned fits take approximately 0.27–0.52 minutes under different implementations and GPUs; these are system measurements, not architecture-only speed comparisons.

### 4.3 Which design choices help?

Table 5 supports three complementary effects: factorized features improve color decoding under the tested capacity budget; canonical colors make the residual task easier; and spatial adaptation improves reconstruction beyond uniformly filling valid charts. These results do not establish that TriPlane always outperforms a hash grid, or that the benefit transfers to every signal distribution.

Additional ablations in Table 6 favor fixed support. Adding rotation, position, or scale deformation reduces PSNR from 35.48 to 35.12, 34.66, or 29.05 dB, respectively. This is consistent with the UV parameterization being fixed while illumination changes, although it does not prove the optimization mechanism responsible. Removing progressive allocation reduces PSNR to 34.51 dB. The single-scale and no-QAT variants also change storage, so their quality differences are not interpreted as capacity-controlled effects.

### 4.4 Access behavior and temporal coverage

Synchronized profiling gives 1.36/1.47 ms for temporal decoding, 0.58/0.63 ms for projection and rasterization, and 1.96/2.08 ms for the complete standalone pipeline. Individual stage quantiles need not sum to the full-pipeline quantiles. The full-frame median corresponds to about 510 reconstructions/s. Reporting this complete pipeline captures the dominant temporal-decoding cost; application frame rate additionally depends on other rendering passes.

Figure 3 in Appendix E shows why sparse access merits a separate experiment. The recorded Lightmap-GS query path has low latency for small batches, while optimized learned representations amortize overhead and become faster at 65,536 queries. BC6H is fastest at all four measured batch sizes. The smallest-batch neural timings include padding and framework overhead; they do not establish an intrinsic advantage of Gaussian arithmetic. The measurements motivate workload-dependent selection rather than a single FPS ranking. Appendix E includes p95 values and discusses preprocessing and cache boundaries.

**Sparse-query operating points.** Table 7 gives the tail latencies behind Figure 3. At 256 requested texels, Lightmap-GS takes 0.044/0.053 ms median/p95, compared with 0.172/0.185 ms

Table 6: **Original architectural ablations.** PSNR uses the original protocol. Storage categories refer to the original reported rates; exact per-variant serialized lengths are unavailable. Consequently, these rows do not establish byte-exact capacity matching.

| Variant | PSNR (dB) | Reported storage vs. full |
|---|---|---|
| Without gradient/sparsity-oriented initialization | 32.59 | Similar |
| Without progressive allocation | 34.51 | Similar |
| Without spatiotemporal deformation | 30.77 | Larger |
| Without multiresolution hierarchy | 33.17 | Smaller |
| Without QAT | 35.16 | Larger |
| Color + rotation head | 35.12 | Similar |
| Color + position head | 34.66 | Similar |
| Color + scale head | 29.05 | Similar |
| Color-only full model | 35.48 | Reference |

Table 7: **Direct-query median/p95 latency in ms.** Learned models use the H settings. For one requested learned query, 128 queries are executed; throughput counts requested queries. The Lightmap-GS/BC6H and optimized neural measurements are distinct implementation studies, not a common-kernel comparison.

| Method | 1 | 256 | 4,096 | 65,536 |
|---|---|---|---|---|
| Lightmap-GS | 0.028 / 0.036 | 0.044 / 0.053 | 0.094 / 0.111 | 0.580 / 0.670 |
| BC6H | 0.014 / 0.020 | 0.019 / 0.027 | 0.036 / 0.045 | 0.084 / 0.097 |
| Instant-NGP* H | 0.203 / 0.216 | 0.172 / 0.185 | 0.173 / 0.186 | 0.172 / 0.185 |
| Temporal-NTC* H | 0.461 / 0.476 | 0.390 / 0.412 | 0.391 / 0.415 | 0.389 / 0.410 |

for Instant-NGP* H and 0.390/0.412 ms for Temporal-NTC* H. At 65,536 texels, the ordering changes: Lightmap-GS takes 0.580/0.670 ms, while the two neural alternatives take 0.172/0.185 and 0.389/0.410 ms. The corresponding useful median throughput is approximately 113, 380, and 169 million requested texels/s. BC6H reaches 0.084/0.097 ms at this batch size, or approximately 780 million requested texels/s. These are measured operating points; the crossover between sampled batch sizes has not been measured.

The query study does not establish cold-start decoding cost. Neural queries use preallocated coordinates and uniformly sampled $(u, v, t)$ points, while the original Lightmap-GS/BC6H summaries do not itemize every cache and preprocessing boundary. A new state can require temporal-color updates, and a newly loaded asset requires index preparation. The timings therefore support selective evaluation for the recorded execution paths, with application-level gains conditional on reuse and query distribution. They are not evidence that an arbitrary isolated texel can always be decoded with all setup included at the displayed latency.

The paired five-scene diagnostic in Table 8 broadens the evaluated state coverage. Increasing temporal capacity at 48 states recovers approximately 0.85 dB relative to the fixed temporal grid. The separate temporal-setting study in Appendix F reports 0.41, 0.42, and 0.43 MiB for 12, 24, and 48 states at grid resolution 10, with training times of 1.6, 2.8, and 4.8 minutes. Its 48-state, resolution-20 setting uses 0.56 MiB and takes 5.3 minutes to fit; median/p95 reconstruction changes from 2.01/2.15 to 2.17/2.34 ms. This documents an increased temporal-capacity cost, but those aggregate records do not itemize scene membership and must not be merged with the paired five-scene means. Hourly and half-hourly samples do not validate strobes, rapid moving shadows, or temporal interpolation between unseen states; we make no such claim.

### 4.5 External assets and engine integration

Additional Blender demo assets—Classroom, Barcelona Pavilion, and Junk Shop—give a mean lightmap PSNR of 34.01 dB and rendered-view PSNR of 37.34 dB in the reported twelve-state study (Appendix G). These scenes diversify geometry and materials but remain synthetic.

In UE5, the reconstruction pass takes 2.18/2.36 ms median/p95, including temporal decoding, projection, rasterization, and resource transitions. The 0.41 MiB packaged asset occupies 0.78 MiB after

Table 8: **Paired temporal diagnostic on five scenes.** Mean original-protocol PSNR is recomputed from the scene rows in Appendix F. Each model fits all states in its column. This tests denser state sampling and coverage, not held-out-time generalization.

| States | 12 | 24 | 48 | 48 (larger time grid) |
|---|---|---|---|---|
| Sampling | Hourly, light-on | Hourly, full day | 30 min | 30 min |
| Temporal resolution | 10 | 10 | 10 | 20 |
| Mean PSNR (dB) | 35.09 | 34.74 | 34.06 | 34.91 |

Table 9: **External synthetic assets.** Quality triples are PSNR/SSIM/LPIPS; latency is median/p95 in ms. Means give each scene equal weight.

| Scene | Atlas quality | MiB | Latency | Rendered-view quality |
|---|---|---|---|---|
| Classroom | 34.36/.951/.093 | .45 | 2.05/2.19 | 37.82/.965/.061 |
| Barcelona Pavilion | 35.21/.961/.078 | .43 | 2.00/2.13 | 39.14/.974/.044 |
| Junk Shop | 32.47/.925/.139 | .50 | 2.21/2.38 | 35.06/.948/.086 |
| Mean | 34.01/.946/.103 | .46 | 2.09/2.23 | 37.34/.962/.064 |

GPU upload/alignment; shared transient reconstruction buffers use 37.6 MiB and peak incremental GPU allocation is 41.2 MiB. Thus asset compression is not an equivalent reduction in peak runtime memory. The engine study supports integration feasibility, not a measured 120-FPS whole-scene rendering guarantee.

**Deployment implications.** Table 9 shows that external-asset atlas PSNR ranges from 32.47 to 35.21 dB, with 0.43–0.50 MiB assets and 2.00–2.21 ms median reconstruction. Rendered-view PSNR is reported separately because shading and view projection change the evaluated signal. The study fits a new model to each asset and does not test transfer from a shared pretrained representation. The available external summaries also lack a complete camera/lighting manifest, limiting independent reproduction of the view averages.

Appendix Table 14 separates distribution size from rendering resources; compiled shader bytecode adds 0.23 MiB. Five warm-ups precede 100 engine measurements including resource transitions; the 0.22 ms median difference from standalone decoding is not a paired overhead estimate. Full-atlas reconstruction can be reused for shading, while direct-query integration requires a separate application benchmark.

## 5 Conclusion and Limitations

Lightmap-GS represents dynamic lightmaps with shared spatial kernels and factorized temporal colors. The local normalized reconstruction exposes a useful separation between spatial support and lighting-dependent coefficients. Controlled ablations support this design on the studied atlas signals, while temporal and engine studies extend its characterization beyond a single quality–size number.

The limitations are substantive. Encoding is per asset; the evaluation remains synthetic and sparsely sampled in time; validity masks do not enforce chart-preserving support; and direct-query performance depends on preprocessing, query distribution, and implementation overhead. Optimized neural baselines and hardware video are strong alternatives, and the available deployment comparisons do not align every hardware and serialization detail. The matched-cap round trips cover only one scene. A reusable encoder, rapid temporal sequences, broader codec round-trip evaluation, and optimized comparisons remain necessary to establish a wider advantage. The contribution is a tested representation design for dynamic HDR atlases, with a defined applicability range.

### AI USE STATEMENT

Generative AI tools were used only to edit the manuscript for readability and language. We have not used generative AI tools to generate synthetic data, develop theoretical models, formulate mathematical claims, assist with proofs, propose or refine hypotheses, design experiments, implement methods, translate text, clean or reformat data, analyze results, interpret findings, draft the paper, create figures, or write code. The authors reviewed the language edits and take responsibility for the final content of this work.

### REPRODUCIBILITY STATEMENT

The appendices specify the representation, optimization settings, metric definitions, baseline adaptations, timing endpoints, and reporting limitations. The accompanying source bundle contains table records, figure scripts, and the Scene 17 matched-cap decoder assets and training scripts. The matched-cap Scene 17 experiment includes its twelve reference states, masks, and standalone decoding inputs. The original 40-scene study is not a complete training-code or dataset release. Some earlier audit, ablation, temporal, and deployment results are available only as reported summaries, without all underlying checkpoints and logs.

### ETHICS STATEMENT

The experiments use synthetic illumination data and do not involve human subjects or personal data. External scene assets retain their source licenses. Dataset redistribution must respect asset-specific terms; no blanket license is asserted for third-party content. The principal research-integrity consideration is to distinguish demonstrated reconstruction and deployment behavior from untested generalization claims.

## REFERENCES

Blender Foundation. Blender demo files, 2026. URL https://www.blender.org/download/demo-files/.

Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 130–141, 2023.

Zhang Chen, Zhong Li, Liangchen Song, Lele Chen, Jingyi Yu, Junsong Yuan, and Yi Xu. Neurbf: A neural fields representation with adaptive radial basis functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4182–4194, 2023.

Emilien Dupont, Adam Goliński, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet. Coin: Compression with implicit neural representations. *arXiv preprint arXiv:2103.03123*, 2021.

Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12479–12488, 2023.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139:1–139:14, 2023.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Julien NP Martel, David B Lindell, Connor Z Lin, Eric R Chan, Marco Monteiro, and Gordon Wetzstein. Acorn: Adaptive coordinate networks for neural scene representation. *arXiv preprint arXiv:2105.02788*, 2021.

Microsoft. *Texture Block Compression in Direct3D 11*. Microsoft Corporation, 2009. URL https://learn.microsoft.com/en-us/windows/win32/direct3d11/texture-block-compression-in-direct3d-11.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in neural information processing systems*, volume 33, pp. 7462–7473, 2020.

Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012.

Karthik Vaidyanathan, Marco Salvi, Bartlomiej Wronski, Tomas Akenine-Möller, Pontus Ebelin, and Aaron Lefohn. Random-access neural compression of material textures. *arXiv preprint arXiv:2305.17105*, 2023.

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.

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4D gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20310–20320, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Wu_4D_Gaussian_Splatting_for_Real-Time_Dynamic_Scene_Rendering_CVPR_2024_paper.html.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Xinjie Zhang, Xingtong Ge, Tongda Xu, Dailan He, Yan Wang, Hongwei Qin, Guo Lu, Jing Geng, and Jun Zhang. GaussianImage: 1000 FPS image representation and compression by 2D gaussian splatting. In *European Conference on Computer Vision*, pp. 327–345. Springer, 2024.

Yunxiang Zhang, Bingxuan Li, Alexandr Kuznetsov, Akshay Jindal, Stavros Diolatzis, Kenneth Chen, Anton Sochenov, Anton Kaplanyan, and Qi Sun. Image-GS: Content-adaptive image representation via 2D gaussians. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–11, 2025.

Lingting Zhu, Guying Lin, Jinnan Chen, Xinjie Zhang, Zhenchao Jin, Zhao Wang, and Lequan Yu. Large images are gaussians: High-quality large image representation with levels of 2d gaussian splatting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 10977–10985, 2025.

# A Implementation and Representation Details

**Training configuration.** Table 10 specifies the reported main configuration. One lighting state is selected per optimization step. Positions, inverse scales, rotations, colors, the MLP, and feature planes have separate Adam parameter groups. The schedule begins with 5,000 kernels and allocates the remaining 5,000 evenly over four additions. This is the schedule implemented by the fixed per-addition budget; a geometric sequence of additions is not used in the inspected implementation. New colors are initialized from the reconstruction residual, with initial scale 3 pixels and zero rotation. The error sampler operates on the full atlas rather than explicitly restricting every new center to valid UV charts.

Table 10: **Reported main configuration.** Inactive positional-encoding options and unused deformation heads are excluded from the active color-only description. The step budget refers to the reported experiments; other saved development configurations are not substituted for those runs.

| Setting | Value |
|---|---|
| Gaussian budget / initial allocation | 10,000 / 5,000 |
| Additions | 1,250 at steps 500, 1,000, 1,500, 2,000 |
| Random fraction in initialization | 0.3 |
| Tile dimensions / retained contributors | $16 \times 16$ / $K = 10$ |
| Base plane resolutions / channels | $(16, 16, 10)$ / 16 |
| Spatial multipliers / temporal multiplier | $(1, 2, 4)$ / 1 |
| Plane combination | Product within level, concatenation across levels |
| Decoder input / hidden width | 48 / 32 |
| Shared feature network | Linear(48,32), ReLU, Linear(32,32) |
| Color head | ReLU, Linear(32,32), ReLU, Linear(32,3) |
| Training budget / optimizer | 10,000 steps / Adam |
| Position / scale / rotation learning rate | $5\times10^{-4}$ / $2\times10^{-3}$ / $2\times10^{-3}$ |
| Initial color / MLP / grid learning rate | $5\times10^{-3}$ / $10^{-3}$ / $5\times10^{-3}$ |
| Decay check / factor / maximum decays | 1,000 steps / 5 / 2 |
| Log-L1 / tone-mapped $(1 - \mathrm{SSIM})$ weight | 1.0 / 0.1 |
| Log epsilon / L2 weight | $10^{-3}$ / 0 |
| Gaussian attribute quantization levels | $2^{16}$ per tensor range |
| Seed | 123 |

**Optimizer reconstruction in the inspected implementation.** The initial rates in Table 10 are not constant throughout the original code path. Each Gaussian addition recreates Adam and resets its moments; the MLP group then uses the scale rate, $2 \times 10^{-3}$, while the grid group uses the color rate, $5 \times 10^{-3}$. The inspected training loop also clips the total gradient norm to 1.0. The post-allocation schedule can reduce rates and terminate early. Complete optimizer traces for the original benchmark are unavailable, so the nominal 10,000-step budget is not evidence that every archived run executed the same update sequence. The new matched-cap runner instead keeps its declared group rates constant across optimizer resets and performs exactly 10,000 updates.

**Coordinates and temporal coverage.** Gaussian centers are stored in normalized atlas coordinates; projection converts centers to pixels and uses pixel-scale kernel widths. Plane sampling instead transforms stored centers using the inspected reversed AABB $[[1.6, 1.6], [-1.6, -1.6]]$, giving $-\boldsymbol{\mu}/1.6$. The plane notation in Equation 2 includes this coordinate transform. Bilinear sampling uses border padding and aligned corners. The active color network queries the planes at the centers and the scalar time; it does not require the inactive position/time Fourier-feature options. In the original implementation, numeric clock labels are mapped as $(t - t_{\min})/(t_{\max} - t_{\min})$, so the twelve-state experiment maps labels 0000–2300 to $[0, 1]$. The selected set contains a gap between 06:00 and 19:00, and the decoder is not periodic at midnight. The three-scene hash-grid comparison instead indexes the listed states uniformly; the new Gaussian comparison retains clock labels. These choices do not prevent fitting the observed states, but they constrain any interpretation about interpolation or temporal smoothness. Neither interpolation across the unsampled daytime interval nor continuity at midnight is evaluated.

**Mask and loss semantics.** A mask value of at least 127 marks a valid texel in the 8-bit input masks. Valid texels are selected for the photometric loss. In contrast, the fused SSIM loss receives the full tone-mapped atlas, so its windows may include padding. Gradient-weighted initialization can emphasize chart boundaries, but the random fraction samples the atlas and the center updates are not projected back into charts. Gaussian support can cross a chart boundary. This can be undesirable for adjacent UV islands with unrelated illumination and is an additional limitation of the representation.

**Gaussian bounds and normalization.** The renderer associates kernels with tiles using finite footprint bounds. A "three-sigma" footprint is a truncation heuristic, not a universal 99.7% probability-mass statement for an anisotropic 2D Gaussian. The Top-$K$ kernel scans the tile candidate list, maintains the largest responses, and divides each retained response by their sum plus $\epsilon_w = 10^{-4}$ in the tiled kernel. It then accumulates RGB values without alpha compositing. The no-normalization and no-tile variants are different paths and must not be mixed with the default renderer's formulation.

**Quantization.** For tensor $z$, the straight-through operator computes

$$d = \max\left(\frac{z_{\max} - z_{\min}}{2^b - 1}, 10^{-8}\right), \quad Q_b(z) = z_{\min} + d \,\mathrm{clip}\left(\mathrm{round}\,\frac{z - z_{\min}}{d}, 0, 2^b - 1\right). \tag{6}$$

The forward value is $Q_b(z)$ and the backward derivative is approximated by the identity. The inspected renderer clamps scales to $[10^{-4}, 10^4]$ and combined colors to $[-10, 10]$ before quantization. The active quantizer covers position, scale, rotation, and combined temporal color tensors; it does not demonstrate that all plane and network weights undergo the same QAT operation. Casting a saved state to FP16 is a separate serialization choice. Consequently, the no-QAT ablation is reported as the tested variant, not as proof of an optimal fixed-bit entropy code or of a plane-wide QAT effect.

## B Metrics, Aggregation, and Storage

**Original protocol.** For valid texels, PSNR uses RGB mean-squared error after normalization by the ground-truth per-frame peak, with peak value one:

$$\mathrm{PSNR} = 10\log_{10}(1/\mathrm{MSE}), \qquad \mathrm{MSE} = \frac{1}{3|M_s|}\sum_{\mathbf{x}\in M_s} \|\widehat{J}_s(\mathbf{x}) - J_s(\mathbf{x})\|_2^2. \tag{7}$$

The original perceptual reporting adjusts atlas-wide SSIM by $1 + (\mathrm{SSIM} - 1)/r$, where $r$ is valid occupancy. LPIPS uses AlexNet features, independent frame-wise min–max normalization of prediction and reference, and division by $r$. These heuristic occupancy adjustments are not equivalent to computing perceptual metrics only on valid texels, and independent normalization can hide scale errors. They are retained only when explicitly identifying the original protocol.

**Shared-scale audit.** The audit applies the same reference-derived peak scale to prediction and target. PSNR is computed over valid RGB samples. For SSIM, it applies Reinhard tone mapping, zeros the invalid background, computes a spatial SSIM map, and averages its valid centers. For LPIPS, it tone maps using the same scale, zeros invalid pixels, crops to the valid-region bounding box, and evaluates inputs mapped to $[-1, 1]$. This crop can still contain holes, and SSIM windows can cross mask boundaries; neither measure is claimed to be a perfect perceptual metric on disconnected charts. The audit's reported PSNR values differ slightly from the original results because the evaluated artifacts and processing paths are not identical. We do not attribute the entire difference to the mathematical PSNR definition.

Table 11: **Metric audit on the primary comparison.** Triples are PSNR/SSIM/LPIPS. Original and audit protocols must not be mixed within a ranking.

| Method | Original report | Recomputed original | Shared-scale audit |
|---|---|---|---|
| Lightmap-GS | 35.48/.96/.08 | 35.47/.958/.087 | 35.41/.944/.091 |
| 3DGS* | 27.48/.61/.43 | 27.46/.612/.431 | 27.42/.604/.428 |
| GaussianImage* | 28.65/.66/.41 | 28.63/.661/.409 | 28.58/.648/.402 |
| Image-GS* | 26.99/.89/.13 | 26.98/.889/.132 | 26.94/.876/.139 |

The original headline LPIPS of 0.08 is retained as a historical report in Table 11. The displayed 40-scene rows instead average to 0.086625 (Table 21), which rounds to 0.09 at two decimals; ordinary rounding does not reconcile the two summaries. Neither value replaces the shared-scale audit used in Table 1.

**Aggregation.** Unless identified otherwise, metrics average frames within a scene and then give equal weight to each scene. This differs from aggregating MSE across all scenes before conversion to decibels. In the optimized three-scene experiments, the timing summaries average each scene's median and each scene's p95 separately; the resulting p95 column is not the pooled 95th percentile of all requests. We do not claim statistical significance from the reported single-seed comparisons. The inter-scene standard deviation is descriptive variation, not a confidence interval over independent training seeds.

**Storage convention.** A twelve-state Float32 RGB sequence occupies exactly $512 \cdot 1024 \cdot 12 \cdot 3 \cdot 4 = 75{,}497{,}472$ bytes, or 72 MiB. Frame-wise BC6H occupies 6 MiB at 8 bits per texel. The original approximate asset sizes are expressed on this binary scale. The rounded 0.41 MiB Lightmap-GS asset implies approximately 0.547 bpp, $175.6\times$ raw-data reduction, and $14.6\times$ smaller storage than BC6H. These derived values are approximate and do not recover the exact original file length. The source tables' 0.53 bpp and $178\times$ are therefore not reused as independently exact measurements.

For optimized learned baselines, we instead use the exact byte count of the serialized FP16 state dictionary, including its serialization container. Auxiliary frame-scale metadata, deployment indices, and shared runtime resources are not necessarily included in the same way as the rounded Lightmap-GS asset. Similarly, the video sizes count the deployed bitstream/container, not an additional HDR-scale sidecar. These boundaries prevent a strict complete-codec byte-matching claim. Training checkpoints, compressed asset size, GPU residency, and peak working memory are separate quantities.

## C Baseline Implementations and Optimized Diagnostics

**Gaussian adaptations.** The three asterisked Gaussian baselines use the primary data and a reported 10,000-step budget. 3DGS$^*$ is a 2D adaptation with opacity/depth removed and a time-conditioned color MLP with two width-64 hidden layers. GaussianImage$^*$ retains uniform initialization and uses a two-layer width-32 temporal color MLP, with the HDR loss replacing the original image loss. Image-GS$^*$ retains gradient-guided placement and uses a shared $(u, v, t)$ color-residual MLP with three width-64 hidden layers and spatial/time positional encodings with four/two frequencies. These are implementation choices by the authors, not original authors' recommended dynamic-lightmap codecs. Their differing capacity and training behavior limit conclusions about the underlying published methods. The controlled tests in Table 5 address particular components more directly.

**Optimized hash-grid alternatives.** Instant-NGP$^*$ uses a 3D multiresolution hash encoding of $(u, v, t)$ followed by a fused width-64, two-hidden-layer ReLU MLP with sigmoid RGB output. Temporal-NTC$^*$ concatenates a 2D spatial hash encoding and a 3D spatiotemporal encoding before a fused MLP of the same width and depth. It is an NTC-inspired alternative, not a reproduction of the original material codec. Both use two features per hash level, base resolution 16, and a geometric progression toward resolution 1024. The larger Instant-NGP$^*$ setting uses 12 levels and a $2^{13}$ hash cap; the smaller setting uses 9 levels with the same cap. The larger Temporal-NTC$^*$ uses 10 spatial and 10 spatiotemporal levels with a $2^{13}$ cap; the smaller uses 7 and 8 levels with a $2^{10}$ cap. The temporal encoding uses a hash cap one power of two smaller, bounded below by $2^8$.

All twelve archived runs use tiny-cuda-nn 1.7, 10,000 Adam steps, batch size 65,536, and masked MSE rather than Lightmap-GS's log-L1/SSIM objective. Learning rates are $10^{-3}$ for the larger settings and $5 \times 10^{-4}$ for the smaller selected settings. Training times average approximately 0.28/0.27 minutes for the larger/smaller Instant-NGP$^*$ and 0.52/0.51 minutes for Temporal-NTC$^*$. Time is the normalized index in the selected state list. No claim of a common physical-time parameterization across these experiments is made.

**Timing and quality boundaries.** Learned full-grid timing evaluates the network on a preallocated $512 \times 1024$ coordinate tensor at normalized time 0.5. It includes feature lookup and MLP evaluation but excludes coordinate construction, file loading, and application integration. Five warm-ups precede 100 synchronized CUDA-event measurements. The training/evaluation script measures quality from the trained model and then writes an FP16 state; it does not reload the saved state and re-evaluate quality. Thus Table 3 reports representation quality paired with serialized state size, not a validated end-to-end compressed bitstream round trip. These boundaries and the different GPUs mean precise speedup factors over Lightmap-GS would be misleading.

Table 12: **Per-scene PSNR in the optimized diagnostic.** H/L are the capacity settings in Table 3. The original Lightmap-GS reference comes from Table 20; these are not newly rerun Lightmap-GS checkpoints. Aggregate quality proximity does not imply scene-wise parity.

| Method | Scene 15 | Scene 17 | Scene 19 |
|---|---|---|---|
| Lightmap-GS (original reference) | 45.72 | 30.03 | 33.40 |
| Instant-NGP* H | 38.79 | 44.65 | 31.92 |
| Instant-NGP* L | 38.70 | 40.53 | 30.03 |
| Temporal-NTC* H | 45.76 | 49.80 | 42.79 |
| Temporal-NTC* L | 37.51 | 41.55 | 30.25 |

## D Video Coding and Random Frame Retrieval

**GPU deployment.** We distinguish predictive GOP coding from All-Intra coding and persistent decoding from starting a new process for every request. The hardware experiment uses PyNvVideoCodec 2.0.4 with an open NVDEC session on RTX 4090 D. HEVC uses Main10 4:2:0 and H.264 uses 8-bit 4:2:0. The timer ends when an RGB frame is GPU-resident and the device is synchronized. File opening and decoder initialization are excluded. The protocol uses five warm-ups and 100 requests per scene. PSNR is evaluated on the normalized image signal; HDR scale restoration is additional metadata handling. Codec operating points were chosen near the original three-scene mean of 36.38 dB, not matched per scene.

HEVC and H.264 All-Intra both retrieve a complete frame quickly in this experiment. Their median and p95 timings are below the separately reported Lightmap-GS full-frame values. Differences in bit depth, chroma format, hardware, and timing boundaries remain, but the measurements are sufficient to reject a general assertion that video random access cannot meet interactive latency. Hardware decoding is a relevant deployment alternative even though it uses fixed-function units.

**Software 4:4:4 diagnostic.** The persistent CPU experiment uses OpenCV/FFmpeg on AMD EPYC 7K62 and ends at a CPU RGB frame. It is not a same-endpoint comparison to GPU reconstruction. Table 13 includes initialization-free measurements rather than the much larger new-process-per-request times. The latter include process launch, file opening, demuxing, and decoder initialization and must not be interpreted as an intrinsic codec cost.

Table 13: **Persistent software 4:4:4 retrieval.** Same three scene IDs as the hardware diagnostic, but different bitstreams and CPU output.

| Codec | Mode | CRF | PSNR | MiB | Median / p95 (ms) |
|---|---|---|---|---|---|
| HEVC | GOP | 47 | 36.35 | 0.0116 | 23.32 / 34.10 |
| HEVC | All-Intra | 49 | 36.24 | 0.0549 | 13.63 / 17.36 |
| H.264 | GOP | 46 | 36.30 | 0.0122 | 12.84 / 15.33 |
| H.264 | All-Intra | 47 | 36.35 | 0.0350 | 11.24 / 11.68 |

All measured CPU median latencies exceed the 8.33 ms budget for 120 frame updates/s, whereas the hardware All-Intra results meet it. This supports a statement about the tested execution paths, not a universal algorithmic advantage. Even a reconstruction pass below 8.33 ms does not establish a 120-FPS game engine after all other rendering work is included.

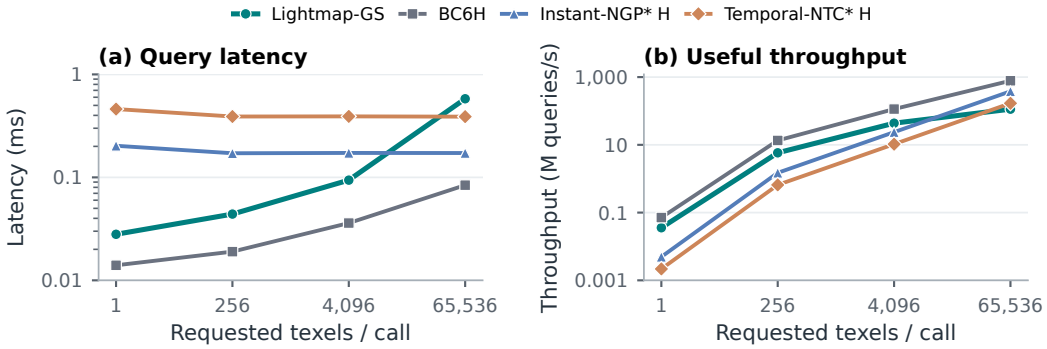



Figure 3: **Batch size changes the access trade-off.** Left: median direct-query latency. Right: useful throughput (requested queries divided by median latency) for the same recorded implementations. Lightmap-GS and BC6H measurements use the original query study; optimized neural results average three scenes on RTX 4090 D. A single requested neural query is padded to 128 executed queries. The graph characterizes these execution paths, not hardware-independent per-texel complexity.

## E Direct Texel Access and Runtime Accounting

The approximately 1670 FPS figure associated with the original implementation comes from a timer that begins immediately before Gaussian projection, after temporal color decoding. It is a stage measurement and is not reused as end-to-end throughput. The synchronized standalone measurements in the main text include both temporal decoding and rasterization.

The requested batches contain 1, 256, 4,096, or 65,536 coordinates. Optimized neural queries are uniformly sampled in the $(u, v, t)$ cube and timed with already allocated coordinates. Small-batch measurements contain framework and launch overhead. The Lightmap-GS and BC6H query summaries establish measured query-call latency, but do not separately identify every cache/preprocessing boundary or the fraction of queries inside valid charts. In particular, the full-frame temporal-decoder latency should not be added to every query or assumed absent from an arbitrary cold query without specifying the caching strategy. We therefore do not treat the query table as proof of a constant-time cold $(u, v, t)$ decoder. An application benchmark should separately specify scene loading, index preparation, temporal-color updates, query distribution, and cache state.

The engine measurement uses GPU timestamps over 100 frames after five warm-ups on RTX 4090. The integrated pass includes temporal decoding, projection, rasterization, and resource transitions. The 0.22 ms difference between median integrated and standalone measurements is descriptive, not the median of paired overhead samples. The shader bytecode size is not the size of an entire plugin distribution. Across 40 equal-sized assets, rounded packaged storage would decrease from 2,880 MiB to 16.4 MiB; this arithmetic does not specify how many decoded resources reside concurrently on the GPU.

Table 14: **UE5 integration accounting.** Size units are MiB. Memory rows are distinct observations, not additive components of a single total.

| Quantity | Reported value |
|---|---|
| Raw twelve-state RGB Float32 asset | 72.00 |
| Compressed per-scene asset | 0.41 |
| Added compiled shader bytecode | 0.23 |
| Persistent per-scene GPU residency after alignment | 0.78 |
| Shared transient reconstruction buffers | 37.6 |
| Peak incremental GPU allocation | 41.2 |
| Integrated reconstruction median / p95 (ms) | 2.18 / 2.36 |
| Standalone reconstruction median / p95 (ms) | 1.96 / 2.08 |

# F TEMPORAL COVERAGE

Table 15: **Paired five-scene temporal diagnostic.** Columns fit 12 hourly light-on states, 24 hourly full-day states, and 48 half-hourly states. The final column doubles temporal grid resolution from 10 to 20. All PSNR values use the original protocol.

| Scene | 12 states | 24 states | 48 fixed | 48 larger grid |
|---|---|---|---|---|
| 15 | 45.72 | 45.31 | 44.58 | 45.49 |
| 40 | 39.13 | 38.79 | 38.14 | 38.96 |
| 19 | 33.40 | 33.08 | 32.42 | 33.19 |
| 17 | 30.03 | 29.72 | 29.10 | 29.84 |
| 73 | 27.18 | 26.80 | 26.06 | 27.07 |
| Mean | 35.09 | 34.74 | 34.06 | 34.91 |

Table 15 supplies an explicit scene-matched comparison. A separate reported temporal-setting study provides the quality, storage, encoding, and decoding aggregates in Table 16. Its scene-level membership is not itemized in the available aggregate record; we therefore do not present its rows as a verified paired 40-scene experiment, and do not replace the five-scene means with these values. In particular, its twelve-state mean of 35.48 dB and the five-scene mean of 35.09 dB have different reporting scopes.

Table 16: **Additional temporal-setting aggregates.** These reported study summaries are separate from the paired five-scene diagnostic above. "Grid" is temporal resolution, training time is minutes, and decoding latency is median/p95 in ms.

| States | Grid | PSNR | SSIM | LPIPS | MiB | Train | Decode |
|---|---|---|---|---|---|---|---|
| 12 | 10 | 35.48 | .957 | .084 | .41 | 1.6 | 1.96 / 2.08 |
| 24 | 10 | 34.77 | .942 | .111 | .42 | 2.8 | 1.98 / 2.11 |
| 48 | 10 | 34.06 | .923 | .122 | .43 | 4.8 | 2.01 / 2.15 |
| 48 | 20 | 34.91 | .935 | .108 | .56 | 5.3 | 2.17 / 2.34 |

Temporal sampling density and temporal frequency content are distinct. These sequences evaluate gradual lighting trajectories with hourly or half-hourly sampling. They do not establish reconstruction of brief flashes, discontinuous shadow motion, or frame-rate dynamics. All states in these comparisons are fitted; no held-out-time interpolation claim follows from their PSNR. The larger temporal grid increases capacity as well as storage, so its gain is not a capacity-matched architectural improvement.

# G EXTERNAL SCENES AND VISUAL EVIDENCE

Classroom, Barcelona Pavilion, and Junk Shop are Blender demo assets (Blender Foundation, 2026). Each reported external experiment has twelve lighting states. Table 9 distinguishes atlas fidelity from rendered-view fidelity: the latter includes the mapping from illumination to a scene image and is not interchangeable with atlas PSNR. These results are additional asset demonstrations rather than evidence of generalization from a shared trained encoder. The available summaries do not provide a full camera/lighting manifest for reproducing the rendered-view averages; their role here is supplementary deployment evidence.

## G.1 MATCHED-STORAGE SEQUENCE EXPERIMENT

The new Scene 17 diagnostic uses complete sequence-file caps of 0.375, 0.5, and 0.75 MiB. Each rate point is a separate fit from scratch over the same twelve states, not twelve independently fitted models. The primary visual comparison uses the predeclared 0.5 MiB cap. Neural assets use 99–100% of each cap by selecting the Gaussian count from actual serialized byte length, without filler bytes. All active learned tensors are stored in FP16 and decoded into FP32 for rendering. A single uncompressed container includes named tensor arrays, shapes, architecture metadata, time labels, and all twelve HDR peaks. Optimizer states, training data, and evaluation masks are excluded. The

decoder regenerates tile acceleration structures; no hidden saved mask is needed to reconstruct a full atlas. File sizes are neither developer-checkpoint sizes nor estimates from parameter counts.

Every Gaussian method uses 10,000 joint Adam updates, seed 123, and the same uniform sequence of time-state draws (792–901 selections per state). We export the final update rather than select the best checkpoint. The loss is identical: valid-texel log-L1 with $\epsilon = 10^{-3}$ plus 0.1 full-atlas SSIM after nonnegative clipping and $z/(1+z)$ tone mapping. SSIM uses the same functional Gaussian-window implementation (size 11, $\sigma = 1.5$) in all environments, averaging valid window centers without an atlas mask; the five-pixel image border is excluded. Learning rates are constant, and no method receives quantization-aware training or entropy coding in this diagnostic. This isolates a common storage precision and training budget, rather than each method's best available codec configuration. Training time and Gaussian count are not matched: capacity allocation and per-step work differ. This is a single-seed, single-scene comparison, separate from the primary 40-scene benchmark. Fixed-budget optimization is not guaranteed to converge equally at different capacities; all raw operating points are retained even when quality is nonmonotonic in stored size.

**Temporal adaptations and capacity allocation.** All four models share their spatial parameters across the twelve states and condition appearance on the noncyclic clock label divided by 2300. Lightmap-GS uses only its active color branch: three levels of $(u, v)$, $(u, t)$, and $(v, t)$ planes at spatial resolutions 16, 32, and 64, temporal resolution 10, and 16 channels per plane. Products of the three sampled planes are concatenated across levels and processed by a width-32 color-residual network. Inactive deformation heads, unused time encodings, and visualization attributes are omitted from the stored decoder. This is a controlled FP16-storage configuration; it does not reproduce the primary table's QAT deployment asset. Its 0.5 MiB version has 16,682 kernels rather than the nominal original 10,000, uses a different optimizer schedule and SSIM implementation, and evaluates stored-decoder outputs. No single-factor explanation is established for the original versus new Scene 17 quality difference.

Image-GS-T retains native scale/rotation 2D projection and normalized Top-10 reconstruction. Its coordinate/time RGB-residual MLP has three width-64 hidden layers, four spatial Fourier bands and two temporal bands. GaussianImage-T (GI-T) retains native Cholesky 2D projection and additive Gaussian reconstruction, with a two-hidden-layer width-32 coordinate/time RGB-residual MLP. It has no learned per-frame spatial parameters or frame embeddings. 3DGS-T uses gsplat 1.4.0 classic perspective projection and depth-ordered opacity compositing from a fixed virtual atlas camera. Its 3D centers, three scales, quaternions, opacities, and canonical RGB logits are shared across time; a two-hidden-layer width-64 MLP predicts RGB-logit residuals from normalized 3D coordinates and time. No camera learning, densification, opacity reset, or view-dependent spherical harmonics are used. These are author temporal atlas adaptations of their native renderers, not claims about official dynamic-lightmap implementations.

Lightmap-GS and Image-GS-T use gradient-based initialization with a 30% uniformly random component, starting with half the final count and allocating the remainder in four residual-guided additions at steps 500, 1000, 1500, and 2000. Initial scales are 3 and 5 pixels, respectively. As in the inspected allocation code, sampling covers the atlas rather than enforcing valid-chart support throughout training. Adam states reset after each addition. GaussianImage-T uses uniformly placed 2D kernels with native random Cholesky initialization; 3DGS-T uses uniform centers on a $16 \times 8$ plane, viewed at distance 8 with focal length 512 pixels and a 3-pixel initial scale. Canonical colors are initialized from state 00:00 for all methods; the shared state list is sorted by numeric clock label. Native rasterization and representation-specific initialization are preserved while training data and loss are shared.

For 2D methods, center, scale/rotation or Cholesky, color, and temporal-network learning rates are 0.0005, 0.002, 0.005, and 0.001. Lightmap-GS's feature planes use 0.005. For 3DGS-T, the corresponding world-coordinate center rate is 0.005; scale/rotation, color/opacity, and temporal-network rates are 0.002, 0.005, and 0.001. All runs use an RTX 4090 D. The native Image-GS renderer uses PyTorch 1.12.1/CUDA 11.6; GaussianImage uses PyTorch 2.4.1/CUDA 11.8; 3DGS uses PyTorch 2.4.1/CUDA 12.1. Framework differences remain an implementation limitation despite the shared loss code. The source bundle includes the complete training and decoding scripts with upstream renderer versions. For numerical stability, only Lightmap-GS retains its inspected pre-rasterization scale/color clamps; this implementation difference is not a controlled ablation.

Table 17: **Complete storage components at the 0.5 MiB cap, in bytes.** Tensor payloads exclude their file headers; the final column includes all container, array-header, and HDR/architecture metadata bytes. Entries sum to each asset's actual file length. Zero denotes an absent component.

| Method | Gaussians | Planes | MLP | Bitstream | Headers/metadata |
|---|---|---|---|---|---|
| 3DGS-T | 511,420 | 0 | 9,350 | 0 | 3,512 |
| GaussianImage-T | 518,656 | 0 | 2,566 | 0 | 3,065 |
| Image-GS-T | 500,416 | 0 | 20,102 | 0 | 3,765 |
| HEVC 4:4:4 | 0 | 0 | 0 | 523,423 | 807 |
| Lightmap-GS | 266,912 | 243,712 | 7,558 | 0 | 6,106 |

**Independent decoding and metrics.** A fresh decoder is constructed using only the exported sequence file. Its rendered outputs agree with the original model after applying the same FP16 parameter rounding, with maximum absolute difference below $10^{-6}$. Quality is computed on these decoded outputs after clipping to $[0, 1]$ in the normalized signal domain; the file's HDR peaks restore physical values. Reported PSNR uses valid texels and the reference peak of each state; sequence means average the twelve state PSNR values. Tone-mapped SSIM covers the full atlas. Final reported PSNR and SSIM are recomputed through a common CPU Float64 evaluator; the original run evaluators are retained for auditing, avoiding framework-dependent metric rounding. The minimum state PSNR describes the weakest fitted lighting state, not unseen-time interpolation or perceptual flicker. Pre-storage metrics and quantization changes are retained in the machine-readable records.

**Video control.** HEVC uses software All-Intra 4:4:4 encoding with the slow preset, PSNR tune, and BT.709 color conversion. Rate selection uses file length only, including the bitstream and HDR metadata container. The initial 10-bit search saturated below the primary cap, so all final video points use 12-bit input to the codec; no padding is added to meet a target. If CRF 0 already fits below a cap, its actual smaller size is reported rather than filling the file; this is a ceiling of the tested fixed configuration, not a claim about every HEVC encoder. Decoding uses FFmpeg RGB48 output, followed by HDR restoration and the same quality evaluation. This control uses higher chroma/bit precision than the earlier hardware Main10 experiment, and no NVDEC timing result is assigned to it. Uniform storage caps do not imply identical computational cost or access capabilities.

Table 18: **Actual complete sequence assets at all tested caps.** Scene 17, twelve fitted states; PSNR in dB, full-atlas tone-mapped SSIM. The minimum is over the twelve fitted states. Each neural asset includes all FP16 parameters and HDR metadata.

| Cap (MiB) | Method | Bytes | Kernels | Mean PSNR | Min PSNR | SSIM |
|---|---|---|---|---|---|---|
| 0.375 | 3DGS-T | 393,214 | 13584 | 37.50 | 27.83 | 0.9778 |
| 0.375 | GaussianImage-T | 393,215 | 24224 | 41.47 | 26.31 | 0.9782 |
| 0.375 | Image-GS-T | 393,211 | 23084 | 42.16 | 31.29 | 0.9846 |
| 0.375 | HEVC 4:4:4 | 388,291 | – | 61.97 | 57.46 | 0.9991 |
| 0.375 | Lightmap-GS | 393,215 | 8490 | 43.35 | 36.55 | 0.9866 |
| 0.500 | 3DGS-T | 524,282 | 18265 | 38.19 | 28.37 | 0.9829 |
| 0.500 | GaussianImage-T | 524,287 | 32416 | 42.88 | 26.31 | 0.9795 |
| 0.500 | Image-GS-T | 524,283 | 31276 | 42.44 | 31.53 | 0.9866 |
| 0.500 | HEVC 4:4:4 | 524,230 | – | 65.02 | 61.38 | 0.9996 |
| 0.500 | Lightmap-GS | 524,288 | 16682 | 42.13 | 38.00 | 0.9886 |
| 0.750 | 3DGS-T | 786,418 | 27627 | 40.14 | 28.92 | 0.9871 |
| 0.750 | GaussianImage-T | 786,431 | 48800 | 42.02 | 26.47 | 0.9722 |
| 0.750 | Image-GS-T | 786,427 | 47660 | 41.93 | 31.53 | 0.9864 |
| 0.750 | HEVC 4:4:4 | 539,110 | – | 65.35 | 64.37 | 0.9996 |
| 0.750 | Lightmap-GS | 786,432 | 33066 | 43.26 | 37.50 | 0.9887 |

**Visual protocol.** Figure 2 uses the same fixed geometric crop (columns 384–639, rows 192–319) and states 00:00/06:00 as the preceding diagnostic, without error-based reselection. One shared reference peak across the three displayed states, exposure 12, Reinhard mapping, and gamma 2.2 are used for every image panel. Error is mean absolute linear-HDR RGB difference divided by that

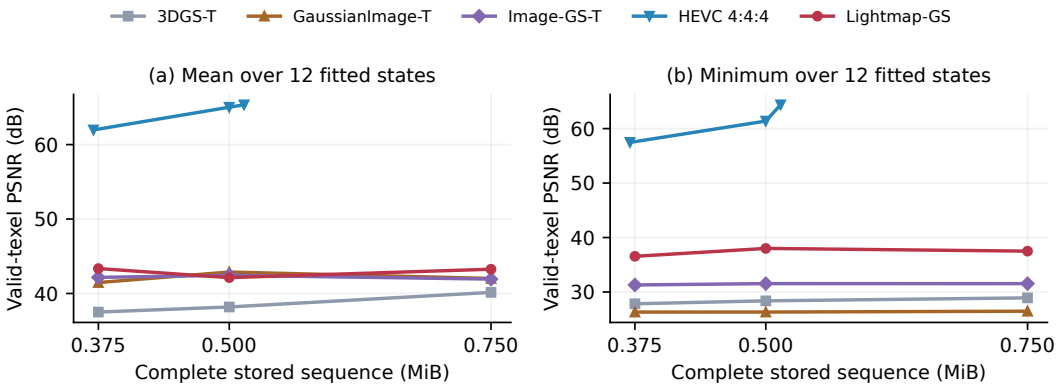



Figure 4: **Quality versus actual complete-file size.** All three predeclared caps and all methods are shown. Mean and minimum fitted-state PSNR capture different aspects of the same twelve-state experiment; lines connect independently trained or encoded assets, not additional measurements.

Table 19: **All fitted states at the 0.5 MiB cap.** Valid-texel PSNR from independently decoded assets. GI-T denotes GaussianImage-T. These results are not substituted for the original 40-scene metrics.

| Time | 3DGS-T | GI-T | Image-GS-T | HEVC | Lightmap-GS |
|---|---|---|---|---|---|
| 00:00 | 39.09 | 44.25 | 43.27 | 61.38 | 42.33 |
| 01:00 | 39.08 | 44.28 | 43.43 | 65.42 | 42.43 |
| 02:00 | 39.28 | 45.13 | 43.75 | 65.65 | 42.56 |
| 03:00 | 39.55 | 46.10 | 43.91 | 65.74 | 42.71 |
| 04:00 | 39.52 | 44.89 | 43.58 | 65.40 | 42.76 |
| 05:00 | 37.27 | 40.96 | 41.99 | 64.98 | 42.02 |
| 06:00 | 28.37 | 26.31 | 31.53 | 64.37 | 38.00 |
| 19:00 | 38.57 | 40.91 | 42.95 | 65.25 | 42.45 |
| 20:00 | 39.37 | 45.17 | 43.73 | 65.62 | 42.70 |
| 21:00 | 39.48 | 45.63 | 43.75 | 65.51 | 42.60 |
| 22:00 | 39.37 | 45.54 | 43.66 | 65.45 | 42.51 |
| 23:00 | 39.29 | 45.37 | 43.68 | 65.49 | 42.51 |
| Mean | 38.19 | 42.88 | 42.44 | 65.02 | 42.13 |
| Minimum | 28.37 | 26.31 | 31.53 | 61.38 | 38.00 |

shared peak; invalid texels are masked. Inset PSNR always evaluates the full valid atlas, not the crop. Figure 5 adds 03:00 with a larger central crop (rows 128–383) using the same independently decoded 0.5 MiB Lightmap-GS asset. All states are fitted observations. No image-generation model creates reconstruction pixels or error maps.

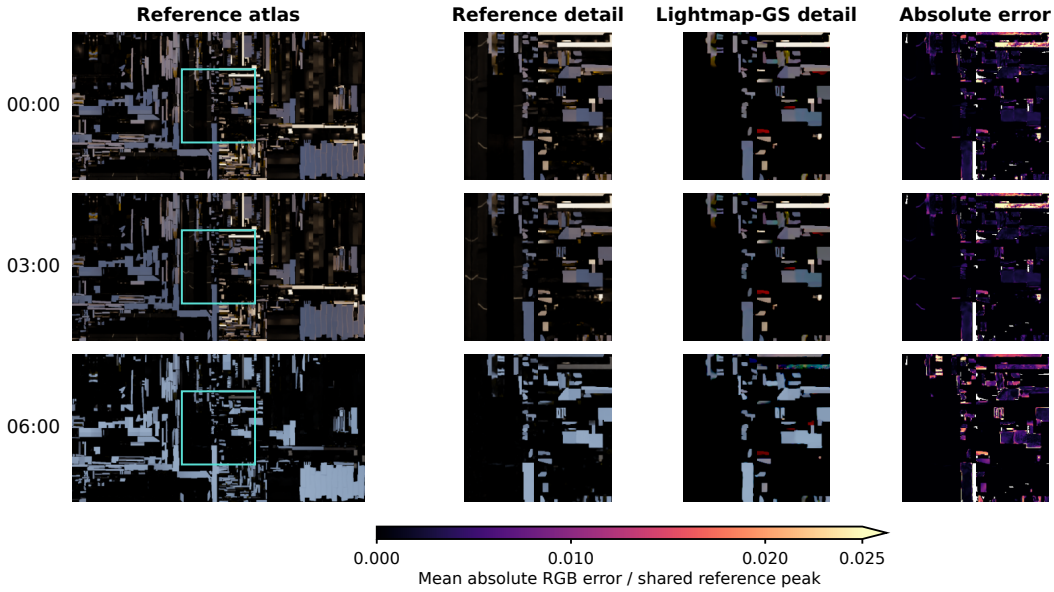



Figure 5: **Additional temporal context at 0.5 MiB.** Reference atlas, reference detail, independently decoded Lightmap-GS detail, and error at three fitted states, with shared exposure and error scale.

## H Further Ablations and Per-Scene Results

The original no-initialization variant and the size-controlled uniform-valid-region experiment are different controls and should not share a label. Removing the hierarchy changes capacity substantially, while the no-QAT configuration changes the reported rate. These results support practical choices in the tested configurations but do not separate every causal mechanism. In particular, the scale-head degradation does not prove a specific local-minimum or instability explanation without additional optimization diagnostics.

The distribution is broad: some scenes have smooth regions that fit compactly, whereas complex atlas boundaries and illumination structure are harder. A single average therefore obscures significant scene dependence. Scene-level differences between original results, later development logs, and saved visual artifacts are not interpreted as additional ablation gains without a matched configuration and checkpoint lineage.

Table 20: **Original 40-scene results.** Each scene averages its twelve fitted states under the original protocol. The aggregate PSNR is 35.48375 dB from these displayed values. Development runs and illustrative checkpoints are not substituted for these scene entries.

| Scene | PSNR | SSIM | LPIPS | Scene | PSNR | SSIM | LPIPS |
|---|---|---|---|---|---|---|---|
| 7 | 32.57 | 0.960 | 0.139 | 40 | 39.13 | 0.990 | 0.021 |
| 9 | 32.15 | 0.968 | 0.057 | 41 | 45.03 | 0.969 | 0.038 |
| 11 | 31.12 | 0.960 | 0.094 | 42 | 41.35 | 0.993 | 0.011 |
| 13 | 31.06 | 0.958 | 0.088 | 44 | 31.32 | 0.956 | 0.083 |
| 15 | 45.72 | 0.973 | 0.032 | 45 | 44.09 | 0.973 | 0.042 |
| 17 | 30.03 | 0.837 | 0.253 | 47 | 30.35 | 0.895 | 0.228 |
| 19 | 33.40 | 0.971 | 0.059 | 48 | 38.50 | 0.989 | 0.028 |
| 21 | 29.48 | 0.954 | 0.285 | 49 | 36.14 | 0.956 | 0.071 |
| 23 | 31.42 | 0.955 | 0.233 | 51 | 37.73 | 0.958 | 0.136 |
| 25 | 31.44 | 0.965 | 0.049 | 53 | 33.12 | 0.929 | 0.138 |
| 27 | 30.95 | 0.962 | 0.091 | 54 | 40.39 | 0.992 | 0.016 |
| 28 | 42.88 | 0.993 | 0.013 | 55 | 33.91 | 0.930 | 0.112 |
| 29 | 32.14 | 0.969 | 0.081 | 59 | 33.34 | 0.943 | 0.101 |
| 30 | 40.24 | 0.988 | 0.030 | 61 | 37.59 | 0.972 | 0.041 |
| 31 | 45.36 | 0.981 | 0.026 | 63 | 32.69 | 0.959 | 0.076 |
| 33 | 33.78 | 0.954 | 0.054 | 65 | 43.38 | 0.980 | 0.033 |
| 34 | 37.83 | 0.991 | 0.031 | 67 | 40.22 | 0.966 | 0.045 |
| 35 | 29.80 | 0.948 | 0.108 | 69 | 31.15 | 0.925 | 0.135 |
| 36 | 27.07 | 0.925 | 0.134 | 71 | 31.27 | 0.938 | 0.072 |
| 37 | 43.03 | 0.980 | 0.027 | 73 | 27.18 | 0.918 | 0.154 |

Table 21: **Statistics recomputed from the displayed 40-scene table.** SD is the sample standard deviation across scenes ($n - 1$ denominator); minor precision differences from unrounded logs are possible.

| Metric | Mean | SD | Median | Min | Max |
|---|---|---|---|---|---|
| PSNR | 35.48 | 5.43 | 33.37 | 27.07 | 45.72 |
| SSIM | 0.958 | 0.030 | 0.961 | 0.837 | 0.993 |
| LPIPS | 0.087 | 0.068 | 0.071 | 0.011 | 0.285 |