CORD: COMPETITIVE REGIONAL DIAGRAMS FOR FAST AND COMPACT LARGE IMAGE REPRESENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Explicit primitive-based methods have achieved high fidelity but remain costly to scale to large images. Recent competitive representations improve reconstruction through per-pixel competition. We observe that the same geometry-dependent competition recurs throughout the pipeline: it is repeatedly updated over millions of pixels during fitting, altered by geometry quantization, and costly to rediscover during rendering. We introduce COmpetitive Regional Diagrams (CORD), which localizes this recurring competition within independently fitted regional diagrams that serve as common units of fitting, coding, and rendering. Bounded competitive fitting limits the size of each optimization problem and completes pruning early, leaving retained primitives time to adapt. Decoded-state coding adapts appearance to decoded geometry and evaluates alternatives by their decoded distortion and actual encoded size. Image-wide rate allocation reuses these alternatives across budgets and directs bytes by reconstruction gain, without per-budget retuning or retraining. Persistent contributor mapping compiles decoder-selected contributor lists into compact GPU state, sharing IDs across neighboring pixels to avoid repeated discovery at fixed geometry and output grid. Experiments show that CORD achieves better rate–distortion trade-offs than the evaluated baselines. On FGF2, CORD High gains 4.60 dB over SGI with 19% smaller payloads, an 18.9× optimization speedup, and nearly 200 FPS for repeated rendering on an A800 GPU.

1 INTRODUCTION

High-resolution imaging supports applications such as digital pathology (Lee et al., 2025), remote sensing (Xia et al., 2018), and heritage digitization (Rijksmuseum, 2022), where fine structures must remain visible within a large field of view. At archive scale, even modest per-image storage and processing costs accumulate quickly: NASA lists a 36.89 MB JPEG for its Landsat 9 view of Los Angeles (NASA Earth Observatory, 2022), while the USGS Landsat archive had reached 12 million scenes by January 2025 (U.S. Geological Survey, 2025). The storage demands of large collections motivate compact representations that preserve fine detail and support efficient processing.

Explicit image representations encode visual content using spatial primitives whose geometry and appearance are fitted to the image (Li et al., 2020; Zhang et al., 2024). Sharing primitive parameters across pixels can reduce storage, with fewer primitives needed in smooth regions and more allocated to fine structures. Recent competitive diagrams select and blend these primitives at each pixel, capturing both sharper boundaries and smoother color transitions (Iinbor et al., 2026). At large-image scale, however, preserving fine detail can still require large primitive sets (Zhu et al., 2025; Pan et al., 2026). Their practical value therefore depends on the storage needed to attain a given reconstruction quality, alongside the cost of fitting and rendering.

Reducing this storage cost requires deciding how many primitives to retain and how precisely to encode their geometry and colors. In competitive representations, pruning and geometry quantization can change which primitives contribute to each pixel and how their colors blend, so previously fitted colors may need adjustment. The best use of limited bytes also varies across regions: some benefit more from additional primitives, others from more precise geometry or greater color capacity. Evaluating these choices over millions of pixels adds computational cost, while the resulting compact representation must remain efficient to render. These requirements connect three evaluation dimensions—**reconstruction fidelity**, **storage efficiency**, and **computational efficiency**—and

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.

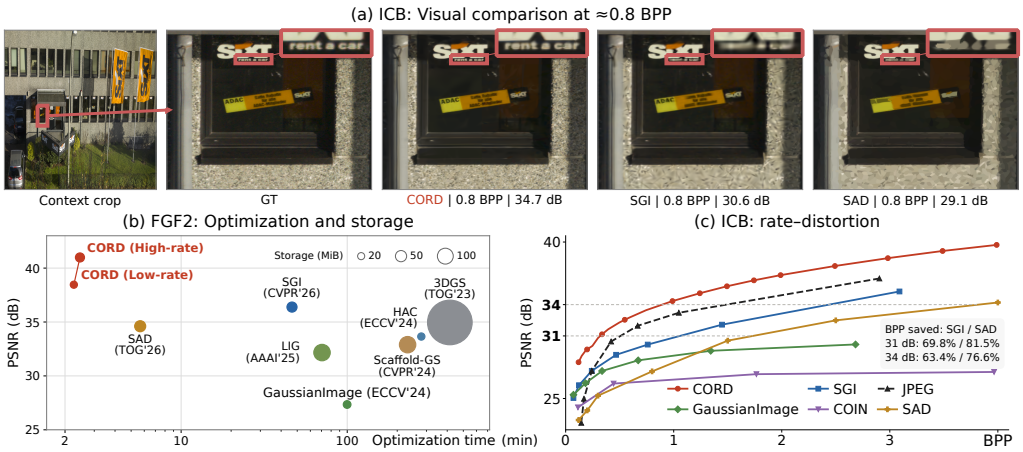


Figure 1: **Fidelity, storage, and computation.** **(a)** At comparable rates near 0.8 BPP on ICB, CORD retains finer lettering and facade texture than SGI and SAD: annotations report full-image BPP and PSNR. **(b)** PSNR versus single-A800 optimization time on FGF2: marker area denotes storage size. **(c)** ICB Rate–distortion curves. The inset reports BPP savings over SGI / SAD at matched PSNR.

raise a central question: *How can we reduce the storage required for high-fidelity large images while keeping fitting and rendering efficient?*

We address this trade-off with COmpetitive Regional Diagrams (CORD), a framework that uses decoded reconstruction quality to guide how limited bytes are spent. CORD organizes each image into independently fitted regional diagrams, allowing storage choices to adapt to local content under an image-wide budget. For each regional coding candidate, it decodes geometry, adapts appearance, and measures reconstruction error together with the complete encoded size. These measurements determine whether bytes are better spent on more primitives, more precise geometry, or greater color capacity in each region. Regional optimization keeps fitting problems bounded, while compact GPU mappings reuse the decoded primitive selections for fast rendering.

We make three technical contributions: **i) Bounded competitive fitting:** We fit regional diagrams with bounded pixel and primitive counts, jointly initialize their colors, and complete pruning early so retained primitives have time to adapt (Section 3.2). **ii) Decoded-state coding and rate allocation:** We select primitive counts, geometric precision, and color capacity under an image-wide budget using encoded sizes and decoded errors after appearance adaptation to decoded geometry. The candidate bank supports multiple budgets without refitting (Section 3.3). **iii) Persistent contributor mapping:** We share cached primitive IDs across neighboring pixels, reducing GPU memory while preserving each pixel's decoder-selected order and avoiding repeated discovery when geometry and pixel positions remain fixed (Section 3.4).

Figure 1 summarizes CORD's fidelity–storage–computation trade-off together with its preservation of fine detail at low bitrate. On FGF2, CORD's High setting outperforms the SGI baseline by 4.60 dB with 19% smaller payloads and an 18.9× optimization speedup.

2 Related Work

Image representations and compression. Coordinate networks encode images as a learned function (Sitzmann et al., 2020; Müller et al., 2022), and have also been used for compression: COIN stores an image in quantized network weights (Dupont et al., 2021). For large signals, ACORN uses a multiscale block decomposition (Martel et al., 2021), while MINER fits local implicit networks over a Laplacian pyramid (Saragadam et al., 2022). Explicit representations instead use spatial primitives, including gradient meshes (Sun et al., 2007) and vector paths (Li et al., 2020). Gaussian-based image representations provide another instance: GaussianImage combines a compact representation with quantization (Zhang et al., 2024), and Image-GS adapts primitive allocation to image con-

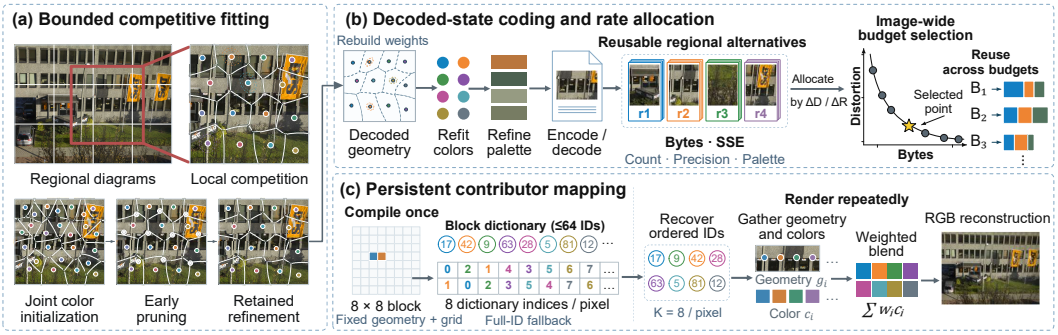


Figure 2: **Overview of CORD. (a)** *Bounded competitive fitting* decomposes a large image into independent regional diagrams with joint initialization, early pruning, and retained refinement. **(b)** *Decoded-state coding and rate allocation* adapts appearance to decoded geometry, forms reusable alternatives, and selects under an image-wide byte budget. **(c)** *Persistent contributor mapping* compiles decoder-selected contributors into compact shared state for repeated rendering.

tent (Zhang et al., 2025). For large images, LIG uses hierarchical residual levels (Zhu et al., 2025), and SGI combines seed-structured Gaussians with coarse-to-fine fitting (Pan et al., 2026).

Competitive diagrams. Differentiable diagrams form adaptive partitions through competition between sites, with applications to cellular simulation (Numerow et al., 2024) and directional appearance (Di Sario et al., 2026). For image representation, SAD uses soft anisotropic scores to select and blend a sparse set of sites per pixel (Iinbor et al., 2026). CORD bounds competition within independently fitted regional diagrams and compiles decoded contributor lists for reuse.

Compression of 3D explicit representations. For 3D Gaussian representations, HAC exploits hash-grid context for entropy coding (Chen et al., 2024b). Compressed3DGS combines sensitivity-aware vector quantization with quantization-aware training (Niedermayr et al., 2024), and LightGaussian combines pruning, recovery optimization, and attribute compression (Fan et al., 2024). These methods develop effective techniques for compressing primitive attributes. CORD instead focuses on the additional coupling introduced by competitive reconstruction, where geometry changes can alter both the selected contributors and their blending weights.

3 Method

As shown in Figure 2, CORD represents a large image with independently fitted regional diagrams. Fitting jointly refines geometry and appearance within bounded local problems. Coding adapts appearance to decoded geometry and constructs reusable regional alternatives for image-wide rate allocation. The selected diagrams then supply compact contributor mappings for repeated rendering. We first define the competitive reconstruction shared by these stages.

3.1 Competitive Representation

Within each region, primitives reconstruct the image through soft anisotropic competition (Iinbor et al., 2026). For a region with P pixels, let $I \in [0,1]^{P\times 3}$ contain the target RGB values and let M denote the set of active primitive indices. We write their geometry as $G = \{g_i\}_{i\in M}$ and stack their RGB colors c_i as the rows of $C \in [0,1]^{|M|\times 3}$. Each primitive's geometry comprises position, radius, sharpness, orientation, and anisotropy. Appendix B.1 specifies the anisotropic score and its regional parameterization. Each primitive assigns a spatial score $\ell_i(x; g_i)$ to pixel x. At each pixel, the top-K scores determine the contributors, normalized blending weights, and reconstruction:

$$K_G(x) = \mathrm{TopK}_{i\in M}\, \ell_i(x; g_i),\ w_i(x) = \frac{\exp \ell_i(x; g_i)}{\sum_{j\in K_G(x)} \exp \ell_j(x; g_j)},\ R(x) = \sum_{i\in K_G(x)} w_i(x) c_i. \quad (1)$$

Following Iinbor et al. (2026), we use $K = 8$. The decoder uses finite spatial search rather than exhaustive Top-K selection, producing an ordered list $K_{\mathrm{dec}}(x)$ that replaces $K_G(x)$ in Equation 1. All coding and rendering evaluations use this decoder-selected list.

For fixed geometry and contributor lists, blending is linear in color: $R = AC$, where $A \in \mathbb{R}^{P\times|M|}$ contains the blending weights and $R \in [0,1]^{P\times 3}$ is the reconstructed image. Each row of A has at most K nonzero entries, which sum to one. We measure reconstruction error by the sum of squared errors (SSE), $D = \|I - R\|_F^2$. Pruning or geometry quantization can change A, leaving previously fitted colors mismatched to the new blending weights. This conditional linear structure supports joint color initialization during fitting and appearance adaptation during coding.

3.2 Bounded competitive fitting

Bounded regional optimization. We partition the image into non-overlapping rectangular regions with comparable dimensions and pixel counts. A target side length L controls regional scale, while a separate cap limits the initial primitive count. Each region is fitted independently, with primitives competing only for pixels in that region. Larger images therefore add regional fitting problems while the scale of each optimization remains bounded. The reconstructed regions are stitched without overlap. With regional origin o_r and side scale $S_r = \max(w_r, h_r)$, the score uses normalized positions $q_i = (p_i - o_r)/S_r$ and radii $\rho_i = r_i/S_r$.

Joint color initialization. Image-GS and SAD initialize colors from target pixels at sampled primitive positions (Zhang et al., 2025; Iinbor et al., 2026). These pointwise assignments do not account for blending among competing primitives. We therefore rebuild contributors for the initial geometry and jointly initialize regional colors under the fixed blending operator A_0 by approximately solving:

$$\min_{0\le C\le 1} \ \|A_0 C - I\|_F^2 + \lambda\|C - C_{\text{init}}\|_F^2, \tag{2}$$

where C_{init} is the preliminary color assignment and λ regularizes departures from it. We keep colors in the RGB range and accept an update only if the fixed-A_0 reconstruction error does not increase. This conditional solve (Golub & Pereyra, 1973) is performed once, and subsequent Adam iterations (Kingma & Ba, 2015) update both geometry and colors.

Pruning and scheduled refinement. Pruning changes the competitors of retained primitives. We gradually reduce an overcomplete regional diagram to its target population, completing pruning early to leave time for geometry and colors to adapt to the retained population. Contributor lists are refreshed periodically during pruning and at every update during final refinement.

Efficient fitting through pixel sampling. Repeated pixel evaluation remains costly during Adam optimization. We partition regional pixels into disjoint, spatially interleaved subsets $\Omega_0, \ldots, \Omega_{m-1}$ ($m = 3$) and cycle through them during both optimization phases. For subset SSE $D_\Omega = \sum_{x\in\Omega}\|I(x) - R(x)\|_2^2$, non-pruning updates use $g_t \propto m\nabla_{G,C} D_{\Omega_{t \bmod m}}$, where m compensates for subset sampling. Every pixel is covered within one cycle. On pruning iterations, both gradient computation ($g_t \propto \nabla_{G,C} D$) and removal-statistic accumulation use all regional pixels: a primitive that appears unnecessary on one subset may still represent detail elsewhere in the region.

Quantization-aware refinement. When used, quantization-aware refinement (QAT) (Jacob et al., 2018) follows the main fit and precedes final appearance adaptation to decoded geometry. Unlike main-fit refinement, QAT keeps contributor identities fixed.

3.3 Decoded-state coding and rate allocation

Primitive count, geometry precision, and color capacity provide different ways to improve a regional reconstruction, but their benefits vary across regions and interact. We therefore separate the construction of complete, appearance-adapted regional alternatives from image-wide budget selection, comparing each alternative by its final decoded distortion and actual encoded size.

Construct regional coding candidates. For target region I_r, we construct a set $\mathcal{S}_r$ of complete encodings spanning primitive count, geometry bit widths, and palette size. Alternatives with the same primitive count share the corresponding fitted primitives, while different populations use separate regional fits (Section 3.2).

Adapt appearance to decoded geometry. Geometry quantization can change contributors and blending weights, so inherited colors may no longer match the decoded diagram. For candidate s, we decode its quantized geometry and rebuild contributors to obtain the blending operator A_s. Holding A_s fixed, we first refit all primitive colors jointly to reconstruct I_r under the decoded blending

Table 1: **Quantitative comparison on FGF2 and STimage.** Dataset-average reconstruction fidelity, optimization time in minutes (on an A800 GPU), and representation size in MiB. Optimization time covers fitting and QAT, excluding subsequent coding. Our CORD is evaluated in two fixed-capacity settings: Low and High rate. Shaded bands group the lower- and higher-rate operating points, and bold values mark the best result for each metric within each group.

Method	FGF2					STimage				
Metric	PSNR↑	SSIM↑	LPIPS↓	Opt. Time↓	Size↓	PSNR↑	SSIM↑	LPIPS↓	Opt. Time↓	Size↓
SIREN (NeurIPS'20)	21.120	0.4961	0.5088	172.13	**15.79**	26.228	0.7900	0.3609	258.77	15.79
I-NGP (TOG'22)	29.798	0.8576	0.1914	**2.11**	21.47	34.953	0.9215	0.1834	**2.21**	23.09
HAC (ECCV'24)	33.652	0.9571	0.0739	279.28	20.66	35.900	0.9577	0.1223	631.98	**12.12**
GaussianImage (ECCV'24)	27.348	0.8393	0.2079	99.84	23.37	25.819	0.6755	0.4194	131.76	23.37
Ours (Low)	**38.461**	**0.9787**	**0.0294**	2.26	20.10	**43.268**	**0.9884**	**0.0342**	3.29	24.80
3DGS (TOG'23)	34.959	0.9675	0.0559	414.76	787.73	38.501	0.9736	0.0707	525.52	787.73
Scaffold-GS (CVPR'24)	32.896	0.9535	0.0793	230.93	106.15	Out of memory				
LIG (AAAI'25)	32.165	0.9210	0.1174	70.51	106.81	34.046	0.8730	0.2669	105.58	106.81
SGI (CVPR'26)	36.385	0.9690	0.0522	46.50	40.15	38.485	0.9756	0.0673	82.89	**28.98**
SAD (TOG'26)	34.609	0.9398	0.1058	5.67	48.09	42.414	0.9790	0.0585	10.53	92.13
Ours (High)	**40.988**	**0.9880**	**0.0138**	**2.46**	**32.48**	**45.010**	**0.9912**	**0.0273**	**3.52**	36.66

weights. These colors initialize a palette of J_s entries. Palette sharing introduces an additional appearance constraint, so we further refine the palette values against pixel reconstruction error. Let Q_s be the one-hot assignment matrix, with one row per retained primitive and one column per palette entry, and let $V_s \in [0,1]^{J_s \times 3}$ contain the palette RGB values. Holding assignments fixed during each refinement step, we approximately solve:

$$\min_{0 \le V_s \le 1} \|I_r - A_s Q_s V_s\|_F^2 . \tag{3}$$

We then encode and decode the complete candidate, so its measured distortion includes color sharing, finite-step adaptation, and palette quantization. Each geometry choice is therefore evaluated with the appearance capacity it actually receives. Appendix A.3 analyzes recoverable color mismatch.

Allocate the image budget by reconstruction benefit. For each completed alternative, we record its decoded SSE $D_r(s) = \|I_r - \widehat{I}_r(s)\|_F^2$ and actual regional stream size $R_r(s)$, including metadata. Because regions reconstruct disjoint pixels, their SSEs add exactly. For an image-wide budget of B bytes and shared header cost R_{header}, we seek

$$\min_{s_r \in \mathcal{S}_r} \sum_r D_r(s_r) \quad \text{subject to} \quad R_{\text{header}} + \sum_r R_r(s_r) \le B. \tag{4}$$

We approximate this discrete problem using a slope-based rate–distortion heuristic (Shoham & Gersho, 1988; Taubman, 2000). For each region, we form the lower rate–distortion hull and order its alternatives by increasing byte cost. Moving to the next hull point defines an upgrade with gain $g_{r,j} = \Delta D_{r,j} / \Delta R_{r,j}$, the decoded SSE reduction per additional byte. We rank legal upgrades across regions by decreasing gain while preserving their within-region order. Starting from the lowest-rate alternative in every region, we apply upgrades in this order and stop before the next upgrade would exceed the image budget. The selected regional streams are then assembled and decoded. Budgets below the minimum combined size are infeasible. Appendix A.4 analyzes this approximation.

3.4 Persistent contributor mapping

To render an image, the decoder selects up to K primitives for each pixel and blends their colors. With decoded geometry and pixel positions fixed, these selections remain valid even if colors change. We therefore compute and store the ordered selections once for reuse across renders. To store these selections compactly, we exploit the overlap between neighboring pixels' selected primitives. Pixels are grouped into 8×8 blocks. For each block b containing at most 64 distinct selected primitives, a shared dictionary H_b stores each full primitive ID once. Each pixel $x \in b$ stores $K = 8$ one-byte indices $j_{x,k}$ into this dictionary, preserving its decoder-selected order. Blocks requiring more than 64 distinct IDs instead store full IDs separately for each pixel.

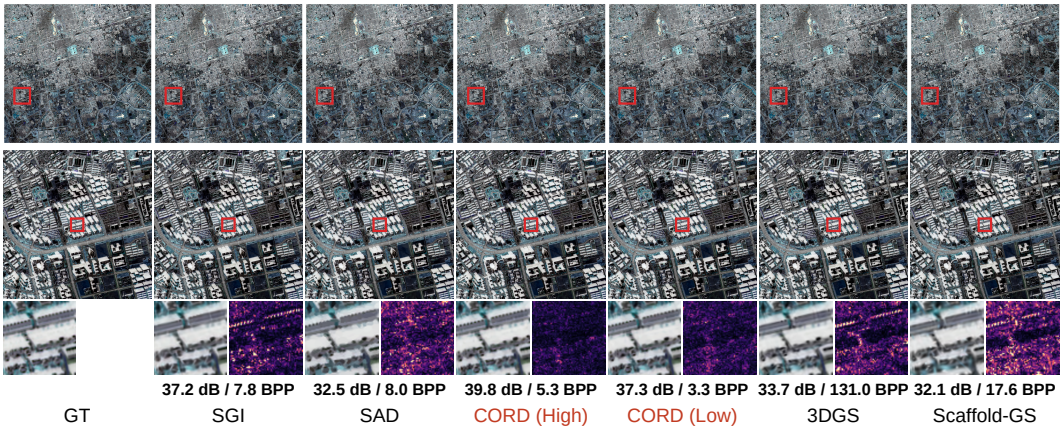


Figure 3: **Fine-detail reconstruction on FGF2.** Rows progress from the full image to an enlarged region and a close-up paired with an error map. Red boxes link locations across zoom levels and methods. Error maps use a shared RMSE scale. Annotations report full-image PSNR and BPP. CORD preserves fine rooftop structure with substantially smaller payloads than baselines.

This GPU-resident mapping is constructed from decoded geometry and is separate from the encoded image stream. A fused CUDA kernel retrieves each pixel's selected primitives, recomputes their scores and blending weights, and blends their colors. For blocks using the shared dictionary,

$$i_{x,k} = H_b[j_{x,k}], \qquad R_{\text{map}}(x) = \sum_{k=1}^{K} w_{i_{x,k}}(x) c_{i_{x,k}} = R_{\text{dec}}(x), \tag{5}$$

where R_{dec} is the reconstruction obtained from the decoder's original ordered primitive lists. Both storage layouts preserve these lists and the resulting reconstruction (Appendix A.5). Colors can change without rebuilding the mapping, and changes to primitive geometry require rebuilding it.

4 Experiments

4.1 Experimental Setup

Datasets, metrics, and baselines. We evaluate nine high-resolution images across natural, satellite, and biomedical imagery: two images from ICB (Rawzor, 2015), four satellite images from Full-resolution Gaofen-2 (FGF2) (Wang et al., 2024), and three histopathology images from STimage (Chen et al., 2024a), ranging from 27.7 to 92.1 megapixels. We evaluate reconstruction fidelity using peak signal-to-noise ratio (PSNR), SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018), and report optimization time and representation size. We compare against coordinate-based neural methods, SIREN (Sitzmann et al., 2020) and I-NGP (Müller et al., 2022), Gaussian-based baselines including 3DGS (Kerbl et al., 2023), Scaffold-GS (Lu et al., 2024), and HAC (Chen et al., 2024b), and image-oriented representations GaussianImage (Zhang et al., 2024), LIG (Zhu et al., 2025), SGI (Pan et al., 2026), and SAD (Iinbor et al., 2026). JPEG provides a conventional image-coding reference in the RD comparisons. Because official SAD degrades on large images (Table 2), we use its corrected best-performing configuration as the baseline (Appendix B.5).

Implementation details. CORD uses balanced regions with a nominal side length of 1024 pixels. The RD curves use reusable regional candidate banks built from seven primitive capacities, each fitted for 1,600 updates, with 500 QAT updates for the two smallest capacities and 1,000 otherwise. Palette variants reuse the corresponding fitted state. For the main comparison, Low and High instead each use one prescribed fitting and coding configuration, with target primitive counts of 7.5% and 11.25% of the regional pixel count, and regional palettes of 4,096 and 32,768 colors. Both configurations use 1,600 fitting updates followed by 1,000 quantization-aware refinement updates, with three interleaved pixel subsets. These settings are kept unchanged across datasets. Each image is optimized on a single A800 GPU. Reported optimization time includes fitting and QAT, with

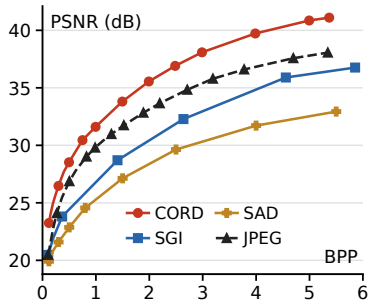


Figure 4: **Rate–distortion performance on FGF2.** CORD lies above the plotted comparison curves throughout their measured-rate ranges.

Configuration	PSNR↑	ΔdB	MiB↓	Δ%	s↓	Δ%	ms↓	Δ%
Official SAD	23.24		—		396.23		—	
SAD + large-image fixes	35.30	+12.06	112.31		323.41	−18.4	1833 R	
+ Init. / scale alignment	**37.07**	+1.77	111.05	−1.1	282.06	−12.8	1339 R	−27.0
+ Bounded fitting	**40.10**	+3.03	111.63	+0.5	282.28	+0.1	1236 R	−7.7
+ Joint init. / sampling	40.21	+0.11	111.66	+0.03	256.34	−9.2	1488 R	+20.3
+ Pruning / refinement	40.02	−0.19	111.68	+0.02	**80.29**	−68.7	1497 R	+0.6
+ QAT / coding	37.05	−2.97	**16.93**	−84.8	111.88	+39.3	1494 R	−0.2
+ Appearance adapt.	**38.68**	+1.63	17.19	+1.5	115.84	+3.5	1211 R	−19.0
+ Persistent spatial idx.	37.98	−0.70	17.19	0.0	116.31	+0.4	**457.2** W	−62.2
+ Persistent cand. cache	37.87	−0.11	17.19	0.0	116.95	+0.6	**32.83** W	−92.8
+ Compact contrib. map	**38.67**	+0.80	17.19	0.0	118.52	+1.3	**4.99** W	−84.8

Table 2: **Cumulative ablation.** Full-resolution FGF2 image on one A800 GPU. Bold values highlight the primary gain at each stage. Time reports the cumulative cost of the operations included up to each row. R: requires contributor search before each render; W: renders directly without search. More details in Appendix D.1.

subsequent coding measured separately. CORD's encoded size counts its complete stored stream. Contributor maps are constructed after decoding and accounted for separately as resident rendering state. More details on implementation are illustrated in Appendix B.

4.2 Comparison with Baselines

Reconstruction quality, storage, and optimization. On FGF2, CORD High reaches 40.99 dB, outperforming the SGI by 4.60 dB while reducing stream size from 40.15 to 32.48 MiB and optimization time from 46.50 to 2.46 GPU-minutes per image. SSIM increases and LPIPS decreases as well. On STimage, CORD Low already exceeds SGI in PSNR while using less storage. On ICB, CORD High improves PSNR, storage, and optimization time over SGI, while SGI retains higher SSIM and lower LPIPS. As shown in Figure 3, both CORD settings preserve FGF2 rooftop boundaries with smaller payloads than the displayed baselines.

Rate–distortion performance. As shown in Figure 1(c) and Figure 4, CORD supports a range of budgets by selecting among reusable regional reconstructions. On both datasets, it exceeds every plotted comparison curve over their overlapping measured rate ranges. On the ICB dataset, it spans 28.48 dB at 0.119 BPP to 39.72 dB at 3.989 BPP.

Repeated rendering efficiency. At comparable PSNR, CORD Low renders ICB building and the third FGF2 image in 3.80 and 4.98 ms (∼263 and 201 FPS), respectively, versus 74.24 and 79.46 ms for SGI, a 16–20× speedup with smaller encoded representations. These median full-image latencies use GPU-resident decoded states and prebuilt mappings on one A800, including RGB writes and GPU synchronization (Table 24). With the runtime initialized, one-time mapping construction from decoded primitives takes 1.2–2.7 s across 12 image pairs from ICB and FGF2, including GPU transfer and contributor discovery. The mapping is then reused at fixed geometry and output grid.

4.3 Ablation Studies

Component attribution. Table 2 traces successive changes from SAD to CORD. Initialization and scale alignment improve PSNR by 1.77 dB, followed by a further 3.03 dB from bounded regional fitting. Subsequent fitting-schedule refinements accelerate optimization by 3.19× with a 0.19 dB quality trade-off. Quantized coding reduces storage by 84.8%, and decoded-state appearance adaptation recovers 1.63 dB. Finally, persistent contributor mapping replaces repeated discovery with reusable rendering state. More details and separate controls for initialization, pruning timing, and contributor refresh are reported in Appendix D.

4.3.1 Bounded competitive fitting

Regional scale and boundaries. On three 2048^2 crops, 1024-pixel regions use 1.14 GiB of peak fitting memory versus 1.77 GiB for 2048-pixel regions, and 12.6% fewer encoded bytes than 512-pixel regions under the same coding profile. Grid shifts reduce decoded PSNR by 0.02–0.44 dB, quanti-

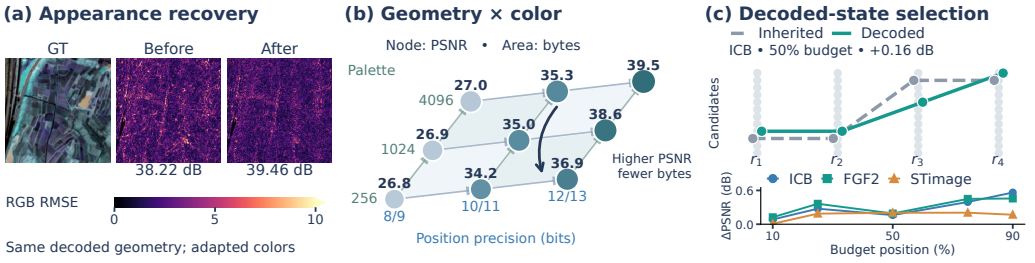


Figure 5: **Decode, adapt, then compare. (a)** Refitting appearance after geometry decoding reduces reconstruction error. Maps share an RGB RMSE scale and labels report full-region PSNR. **(b)** Geometry precision and color capacity interact. The highlighted pair achieves higher PSNR with fewer bytes through a different allocation of geometry and color resources. **(c)** Decoded-state scoring changes regional selections at a fixed budget (top) and improves PSNR over inherited-color scoring across five normalized budget positions on ICB, FGF2, and STimage (bottom).

fying the partition sensitivity of this memory–quality trade-off (Table 10). To assess partitioning at fixed topology, we compare global and 1024-pixel regional fits on 2048^2 crops with shared initialization and no pruning. Regional fitting improves whole-crop PSNR by 0.16–0.22 dB with 5–6% longer solver-call time under sequential regional execution (Table 14). PSNR differences on identical eight-pixel boundary bands range from -0.25 to $+0.01$ dB (three-seed means. Appendix D.7).

Pixel sampling. The proposed three-subset sampling reduces fitting-plus-QAT time by 15–20% and improves decoded PSNR by 0.14–0.21 dB over full-pixel main-fit updates on all three crops, with subsequent QAT and coding unchanged (see Table 11 for more results).

4.3.2 Decoded-state coding and rate allocation

Appearance adaptation after decoding. Geometry quantization changes the blending operator, so colors should adapt to the decoded geometry rather than the original fit. We compare refitting colors before and after geometry decoding while keeping the final geometry, solver iteration budget, and palette-coding procedure fixed. As shown in Figure 5(a), refitting after decoding improves PSNR by 0.27–1.28 dB across six preselected regional states (the full results are shown in Table 15). This isolates the quality recovered by adapting appearance to the decoded state.

Coupled geometry and color choices. The value of color capacity depends on geometry precision. As shown in Figure 5(b), increasing the palette from 256 to 4096 colors gains only 0.23–0.43 dB at coarse position precision, but 1.33–2.59 dB at the finest tested precision. At fixed population, we further compare joint selection with varying only geometry precision or only palette size over five budgets in each four-region ICB and FGF2 problem. Each single-variable control uses its best fixed setting of the other variable. Joint selection gains up to 0.78 dB over geometry-only and 2.50 dB over palette-only selection, showing that each coding resource's value depends on the operating point.

Decoded-state candidate scoring. To isolate candidate scoring, both policies select from the same adapted output streams at the same byte costs; only their scoring errors differ. We exhaustively search 12^4 combinations at five budgets for each of three four-region problems. As shown in Figure 5(c), using final decoded error instead of inherited-color error improves PSNR by 0.08–0.56 dB on ICB, 0.12–0.46 dB on FGF2, and 0.01–0.20 dB on STimage. The regional selections show how appearance-aware scoring redirects the same byte budget toward better decoded reconstructions.

Content-adaptive regional allocation. To isolate regional allocation, we hold the candidate bank and image-wide byte ceiling fixed and compare against the best feasible uniform configuration. Across 12 low-rate budgets per complete ICB image (0.10–0.25 BPP), mean PSNR gains are 0.16 dB on building and 0.18 dB on tree, with maxima of 0.40 and 0.72 dB, respectively. Across 12 medium-to-high-rate budgets per four-region problem (1.5–3.4 BPP), mean gains are 0.44 dB on ICB and 0.56 dB on FGF2, reaching 1.41 and 1.59 dB. Our allocator achieves near-optimal allocations on three full-image candidate banks with 48–56 regions, with an average PSNR gap of 0.009 dB and a maximum of 0.038 dB relative to integer-programming optima across nine tested budgets.

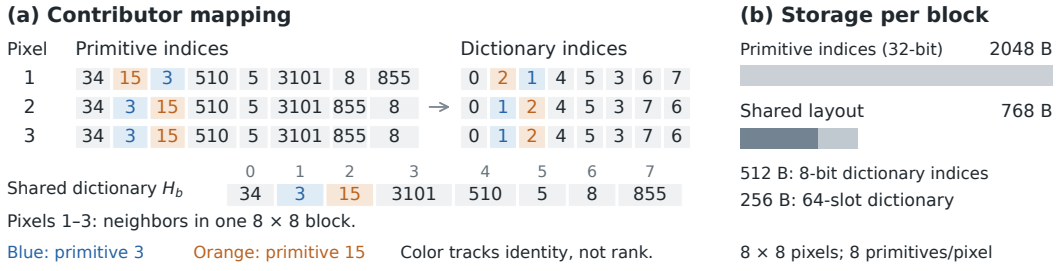


Figure 6: **Compact rendering state with unchanged contributors.** **(a)** Neighboring pixels share dictionary H_b. Dictionary indices recover primitive indices in their original order and colors track primitive identity. **(b)** Per-block selection storage excludes metadata. Across six full-image states, sharing reduces resident memory (including RGB output) by 41–43% and render latency by 4.8–8.4%, with bitwise-identical RGB.

Encoding cost and budget reuse. We distinguish single-capacity encoding, reusable-bank construction, and subsequent budget queries. On the full GF2_0 image with 4,096-color palettes, separately measured fitting, QAT, and post-fitting stages sum to 146.18 s on one A800 with eight CPU workers (see Table 26 for more results). Independent full-image validation and PNG export are excluded. For multiple operating points, constructing the seven-capacity bank on ICB tree takes 406.1 s on one A800, including fitting, QAT, and coding. Once the banks are constructed, budget selection and physical-stream assembly take 39–563 ms per complete ICB image. Additional budgets reuse encoded regional candidates without further fitting, QAT, or appearance adaptation.

4.3.3 Persistent contributor mapping

Storage and rendering. We isolate the mapping layout by comparing full per-pixel IDs with shared dictionaries under identical decoded primitives, ordered contributors, and fused rescoring/blending kernels. Selection storage decreases by 62.4%, from 32 to approximately 12 bytes/pixel including dictionaries. As shown in Figure 6, across six full-image state pairs, resident memory including the RGB output buffer decreases by 41–43% and render latency by 4.8–8.4%, while all paired RGB outputs remain bitwise identical.

5 Discussion and Future Directions

CORD currently uses regular rectangular regions and prescribed fitting and candidate-generation schedules, which are shared across datasets without dataset-specific tuning. Its performance across natural, satellite, and pathology images suggests that the framework is not tied to a particular image domain. Greater adaptivity could learn region boundaries, pruning schedules, or candidate-bank composition, concentrating encoder effort where additional capacity yields the largest decoded reconstruction gains (Yang et al., 2026). Beyond 2D images, the same principle of evaluating complete decoded states and allocating rate by realized reconstruction benefit may extend to explicit 3D representations, where large scenes remain costly to optimize and store (Liang et al., 2026), and fine detail is easily lost under local or low-rate representations (Bai et al., 2026).

6 Conclusion

Compression changes the competition underlying reconstruction, not just stored parameters. CORD addresses this coupling through three coordinated stages. Bounded competitive fitting optimizes retained primitives within manageable regional problems. Decoded-state coding adapts appearance to quantized geometry and allocates bytes across reusable regional alternatives according to decoded distortion and actual encoded size. Persistent contributor mapping compiles decoder-selected contributors into compact state for efficient repeated rendering. Together, these stages improve the fidelity–storage–computation trade-off in large-image representation.

LARGE LANGUAGE MODEL USAGE STATEMENT

We acknowledge the use of large language models (LLMs) as auxiliary tools in the preparation and implementation of this work. Specifically, LLMs were used for language polishing, literature search and retrieval, assistance with selected implementation and figure-generation code, and assistance with mathematical derivations and proofs in the appendix. All AI-assisted outputs, including code, references, mathematical arguments, and manuscript edits, were critically reviewed and verified by the authors before inclusion.

The research problem, methodological design, experimental design, interpretation of results, and scientific conclusions were determined by the authors. We critically reviewed and edited all text generated or modified by the LLM to ensure it accurately reflects our original thoughts and findings.

ETHICS STATEMENT

This work studies efficient representation, compression, and rendering of large images. It does not involve human-subject recruitment, user studies, or the collection of new personal data. Experiments use existing research datasets spanning natural, satellite, and histopathology imagery; for the biomedical data, we use only the released RGB image content and do not use subject identities, demographic attributes, or clinical metadata. CORD is a general-purpose representation method and does not perform recognition, profiling, diagnosis, or other downstream decision-making. Its potential benefits include reducing storage and computational costs for large-image processing, while potential risks primarily depend on the downstream applications where the images are used.

REPRODUCIBILITY STATEMENT

The supplementary materials provide source code for CORD's Low and High configurations, covering fitting, quantization-aware training, encoding, and decoding. The accompanying README includes installation and execution instructions, as well as dataset and verification procedures. Experimental settings, additional evaluations, and theoretical derivations are provided in the appendix.

REFERENCES

Tian Bai, Zheng Qiu, Haojie Chen, and Ziyang Dai. Plug-and-play optimization for 3d gaussian splatting compression: distribution regularization, probabilistic pruning and detail compensation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2372–2380, 2026.

Jiawen Chen, Muqing Zhou, Wenrong Wu, Jinwei Zhang, Yun Li, and Didong Li. Stimage-1k4m: A histopathology image-gene expression dataset for spatial transcriptomics. *Advances in Neural Information Processing Systems*, 37:35796–35823, 2024a.

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*, pp. 422–438. Springer, 2024b.

Francesco Di Sario, Daniel Rebain, Dor Verbin, Marco Grangetto, and Andrea Tagliasacchi. Spherical voronoi: Directional appearance as a differentiable partition of the sphere. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22529–22538, 2026.

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

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.

Gene H Golub and Victor Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on numerical analysis*, 10(2):413–432, 1973.

Laki Iinbor, Zhiyang Dou, and Wojciech Matusik. Soft anisotropic diagrams for differentiable image representation. *ACM Transactions on Graphics (TOG)*, 45(4):1–19, 2026.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 2704–2713. IEEE, 2018.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4), July 2023. ISSN 0730-0301. doi: 10.1145/3592433. URL https://doi.org/10.1145/3592433.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Jonghyun Lee, Lina Takemaru, DM Bappy, Ye Sul Jeong, Won-Ki Jeong, Derek Oldridge, Dokyoon Kim, Sangjeong Ahn, and Sung Hak Lee. Adaptive compression framework for giga-pixel whole slide images. *Nature communications*, 17(1):207, 2025.

Tzu-Mao Li, Michal Lukáč, Michaël Gharbi, and Jonathan Ragan-Kelley. Differentiable vector graphics rasterization for editing and learning. *ACM Transactions on Graphics (TOG)*, 39(6): 1–15, 2020.

Zhicheng Liang, Junhua Liu, Dayou Zhang, Boyan Li, Zijian Cao, Yaxi Liu, Feng Qian, and Fangxin Wang. Mobilesplat: Real-time streaming and rendering of city-scale 3d scenes for mobile platforms. *IEEE Transactions on Mobile Computing*, 2026.

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

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.

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.

NASA Earth Observatory. A super view of LA. NASA Science, February 12, 2022, February 2022. URL https://science.nasa.gov/earth/earth-observatory/a-super-view-of-la-149458/.

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.

Logan Numerow, Yue Li, Stelian Coros, and Bernhard Thomaszewski. Differentiable voronoi diagrams for simulation of cell-based mechanical systems. *ACM Transactions on Graphics (TOG)*, 43(4):1–11, 2024.

Zixuan Pan, Kaiyuan Tang, Jun Xia, Yifan Qin, Lin Gu, Chaoli Wang, Jianxu Chen, and Yiyu Shi. Sgi: Structured 2d gaussians for efficient and compact large image representation. *arXiv preprint arXiv:2603.07789*, 2026.

Rawzor. Image compression benchmark. https://imagecompression.info/test_images/, 2015. Accessed 2026-08-10.

Rijksmuseum. Rijksmuseum publishes 717 gigapixel photograph of The Night Watch. Rijksmuseum press release, January 3, 2022, January 2022. URL https://www.rijksmuseum.nl/en/press/press-releases/rijksmuseum-publishes-717-gigapixel-photograph-of-the-night-watch.

Vishwanath Saragadam, Jasper Tan, Guha Balakrishnan, Richard G Baraniuk, and Ashok Veeraraghavan. Miner: Multiscale implicit neural representation. In *European Conference on Computer Vision*, pp. 318–333. Springer, 2022.

Yair Shoham and Allen Gersho. Efficient bit allocation for an arbitrary set of quantizers (speech coding). *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(9):1445–1453, 1988.

Vincent Sitzmann, Julien N. P. Martel, Alexander Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7462–7473, 2020.

Jian Sun, Lin Liang, Fang Wen, and Heung-Yeung Shum. Image vectorization using optimized gradient meshes. *ACM Transactions on Graphics*, 26(3), 2007.

David Taubman. High performance scalable image compression with ebcot. *IEEE Transactions on image processing*, 9(7):1158–1170, 2000.

U.S. Geological Survey. Landsat 9 sees diverse landscapes of the desert southwest. USGS Landsat Missions, January 9, 2025, January 2025. URL https://www.usgs.gov/media/images/landsat-9-sees-diverse-landscapes-desert-southwest.

Yingying Wang, Xuanhua He, Yuhang Dong, Yunlong Lin, Yue Huang, and Xinghao Ding. Cross-modality interaction network for pan-sharpening. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–16, 2024.

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.

Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3974–3983. IEEE, 2018.

Xianben Yang, Tao Wang, Yuxuan Li, Yi Jin, and Haibin Ling. Gsˆ 2: Graph-based spatial distribution optimization for compact 3d gaussian splatting. *arXiv preprint arXiv:2604.01884*, 2026.

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

We analyze how primitive changes affect competitive reconstruction, how appearance adapts to a decoded blending operator, and how regional rate–distortion costs compose. We then characterize supported allocations within a fixed candidate bank and prove preservation of decoder-selected contributors under compact mapping. Numerical checks on decoded states complete the analysis. We use Y for the target RGB matrix and Id for the identity. The distortion is measured as SSE.

A.1 COMPETITIVE SENSITIVITY

Proposition A.1 (fixed-support differential). Suppose Top-K membership is fixed in a neighborhood of the geometry and each score ℓ_i depends only on g_i. For i in the support,

$$\frac{\partial R}{\partial c_i} = w_i \mathrm{Id}_3, \qquad \frac{\partial R}{\partial g_i} = w_i(c_i - R)\nabla_{g_i}\ell_i^\top.$$

An excluded primitive has zero derivative within that fixed-support neighborhood.

Proof. The softmax derivative is $\partial w_j/\partial \ell_i = w_j(\mathbf{1}_{i=j} - w_i)$. Thus $\partial R/\partial \ell_i = \sum_j c_j w_j(\mathbf{1}_{i=j} - w_i) = w_i(c_i - R)$. The chain rule and linearity give the geometry derivative and color derivative.

Local reconstruction response. Within a fixed contributor set, geometry sensitivity depends on the blending weight, the color contrast $c_i - R$, and the score gradient. Joint perturbations therefore act through the current competitive reconstruction. The differential does not describe contributor transitions.

Lemma A.2 (joint response under sparse competition). Fix contributor lists with at most K entries per pixel and smooth decoded coordinates. For simultaneous primitive perturbations, define

$$z_i(x) = w_i(x)\big[\delta c_i + (c_i - R(x))\delta\ell_i(x)\big], \qquad z_i(x) = 0 \text{ outside primitive } i\text{'s support.}$$

Here z_i also denotes the vector formed by stacking its RGB pixel effects. The joint differential is $\delta R(x) = \sum_i z_i(x)$, and

$$\Big\|\sum_i z_i\Big\|_2^2 = \sum_i \|z_i\|_2^2 + 2\sum_{i<j}\langle z_i, z_j\rangle, \tag{6}$$

$$0 \le \Big\|\sum_i z_i\Big\|_2^2 \le K\sum_i \|z_i\|_2^2. \tag{7}$$

For $K \ge 2$, sparsity alone supplies no positive lower bound relative to the sum of individual energies.

Proof. Summing Proposition A.1 gives the differential; expanding its squared norm gives Equation 6. At each pixel at most K terms are nonzero, so Cauchy–Schwarz bounds their squared sum by K times the sum of squared norms. Summing over pixels proves Equation 7. For two contributors with unchanged colors,

$$z_1 + z_2 = w_1 w_2(c_1 - c_2)(\delta\ell_1 - \delta\ell_2).$$

Equal nonzero shifts cancel when $c_1 \ne c_2$; this example is expressed in logit coordinates.

Overlapping pixel supports produce the cross terms in Equation 6. CORD retains these interactions within regions, while disjoint regional reconstruction domains give the additive composition in Proposition A.7.

A.2 STABILITY OF CONTRIBUTORS UNDER QUANTIZATION

Proposition A.3 (margin certificate). Fix the active topology and a pixel. Let $m = \ell_{(K)} - \ell_{(K+1)}$ and suppose $|\widehat{\ell}_i - \ell_i| \le \epsilon$ for every active primitive. If $m > 2\epsilon$, the exhaustive decoded Top-K set equals the original set.

Proof. For every original selected i and excluded j, $\widehat{\ell}_i - \widehat{\ell}_j \ge \ell_i - \ell_j - 2\epsilon \ge m - 2\epsilon > 0$. No excluded score can cross a selected score.

Score perturbation bounds. The margin condition certifies exhaustive Top-K membership for a fixed active set under a perturbation bound covering every active primitive. If $|\Delta\ell_i| \le L_i\|\Delta g_i\|$ with L_i valid along the quantization segment, one may take $\epsilon = \max_i L_i\|\Delta g_i\|$. This membership certificate is separate from the ordered selections returned by the finite spatial query. This certificate is sufficient, not necessary: failure does not imply a support change.

Lemma A.4 (finite response and contributor changes). Keep the primitive population and output grid fixed. Let $r(\theta)$ be the vectorized reconstruction using the deployed contributor selection. Define r_0 by holding the lists selected at θ fixed while recomputing scores, weights, and colors at its input parameters; hence $r_0(\theta) = r(\theta)$. Put $J_0 = Dr_0(\theta)$ and

$$\eta_{\mathrm{sel}}(\delta) = \|r(\theta + \delta) - r_0(\theta + \delta)\|_2.$$

If Dr_0 is L-Lipschitz in operator norm along the perturbation segment,

$$\|r(\theta+\delta) - r(\theta) - J_0\delta\|_2 \le \tfrac{L}{2}\|\delta\|_2^2 + \eta_{\mathrm{sel}}(\delta). \tag{8}$$

Proof. The integral Taylor remainder of r_0 is bounded by $\int_0^1 Lt\|\delta\|_2^2\,dt$. Add $r(\theta+\delta) - r_0(\theta+\delta)$ and use the triangle inequality.

The bound separates the smooth response of a fixed-list renderer from the reconstruction difference caused by contributor reselection. The latter term compares the two selections at the perturbed state. Different primitive populations are evaluated as separate coding alternatives.

A.3 Recoverable appearance mismatch

For fixed decoded geometry and contributors, appearance remains a linear reconstruction variable. The following decomposition separates the operator's minimum least-squares error from the mismatch of inherited colors and motivates decoded-state appearance adaptation.

Proposition A.5 (decoded-state error decomposition). Let A be the blending matrix determined by decoded geometry and contributors. For target Y and colors $\widehat{C}$, define $C^\star = A^\dagger Y$ and $P = AA^\dagger$, where $A^\dagger$ is the Moore–Penrose pseudoinverse and P is the orthogonal projector onto $\mathrm{col}(A)$. Then

$$\underbrace{\|Y - A\widehat{C}\|_F^2}_{\text{error with inherited colors}} = \underbrace{\|(\mathrm{Id}-P)Y\|_F^2}_{\text{structure-limited error}} + \underbrace{\|A(\widehat{C}-C^\star)\|_F^2}_{\text{appearance mismatch}}. \tag{9}$$

The first term is the minimum error attainable by unrestricted colors for this operator. The second is the appearance mismatch eliminated by unrestricted least-squares refitting. Furthermore, for inherited colors C_0,

$$\|Y - AC_0\|_F^2 - \|Y - AC^\star\|_F^2 = \|P(Y - AC_0)\|_F^2.$$

Proof. P is the orthogonal projector onto $\mathrm{col}(A)$, including when A is rank deficient. Split $Y - A\widehat{C} = (\mathrm{Id}-P)Y + (PY - A\widehat{C})$. The two terms lie in orthogonal subspaces and $PY = AC^\star$, proving the first identity. Also $PAC_0 = AC_0$, which gives the second. Least-squares colors need not be unique, but the minimum-norm solution and the optimal rendered output PY are unique.

Adaptation at fixed decoded geometry. After decoding changes A, unrestricted least-squares refitting to the new operator recovers $\|P(Y - AC_0)\|_F^2$ of the inherited-color error.

Constrained appearance adaptation. For a convex feasible color set $\mathcal{C}$ and an exact constrained minimizer C_b, the first-order optimality condition gives

$$\langle AC_b - Y, A(C - C_b)\rangle_F \ge 0, \qquad C \in \mathcal{C}.$$

Consequently

$$\|Y - AC\|_F^2 - \|Y - AC_b\|_F^2 = \|A(C-C_b)\|_F^2 + 2\langle AC_b - Y, A(C-C_b)\rangle_F \ge \|A(C-C_b)\|_F^2.$$

The orthogonal identity above uses unrestricted colors. With fixed palette assignments Q, continuous palette optimization instead uses AQ. The encoder evaluates decoded SSE after its bounded solves, assignment updates, and palette quantization.

Safeguarded approximate projection. Let $d = \mathrm{clip}(\widehat{C}) - C_0$, with feasible C_0. Every $C_0 + \alpha d$ for $\alpha \in [0,1]$ is feasible. Accepting only trials whose measured fixed-support SSE does not exceed the baseline, with $\alpha = 0$ as a fallback, cannot increase that SSE in exact arithmetic. Numerical acceptance uses an explicit tolerance.

Corollary A.6 (appearance compensation of geometry response). Within a fixed-support smooth branch, let a small geometry change induce $U = (\mathrm{d}A)C$. The joint first-order reconstruction change is $U + A\Delta C$, omitting the second-order product $(\mathrm{d}A)\Delta C$. With $P_A = AA^\dagger$ and unrestricted color increments,

$$\min_{\Delta C} \|U + A\Delta C\|_F^2 = \|(\mathrm{Id} - P_A)U\|_F^2. \tag{10}$$

For target residual $E = Y - AC$, the corresponding local fitting problem is

$$\min_{\Delta C} \|E - U - A\Delta C\|_F^2 = \|(\mathrm{Id} - P_A)(E - U)\|_F^2. \tag{11}$$

If the baseline is palette-compatible, $C = QV$, and assignments Q remain fixed, palette increments instead satisfy

$$\min_{\Delta V} \|U + AQ\Delta V\|_F^2 = \|(\mathrm{Id} - P_{AQ})U\|_F^2 \geq \|(\mathrm{Id} - P_A)U\|_F^2, \tag{12}$$

where $P_{AQ} = (AQ)(AQ)^\dagger$.

Proof. In Equation 10, choose $\Delta C = -A^\dagger U$ to cancel $P_A U$; the orthogonal component is unaffected by any color increment. Applying the same argument to $E-U$ proves Equation 11. Replacing A by AQ gives the palette equality. The inequality follows from $\mathrm{col}(AQ) \subseteq \mathrm{col}(A)$.

Equation 10 measures departure from the baseline reconstruction. Equation 11 measures target error. They coincide for zero baseline residual. Under the stated fixed-assignment, first-order model, palette sharing restricts the appearance compensation space from $\mathrm{col}(A)$ to $\mathrm{col}(AQ)$.

A.4 Regional rate–distortion composition and allocation

Complete regional states can be compared image-wide because their decoded errors and counted rates compose under the same budget.

Proposition A.7 (composition of complete regional states). Suppose each pixel is reconstructed by exactly one region, each regional payload is independently encoded, and all remaining container overhead is included in R_{header} or the appropriate regional rate. Then any tuple $(s_1, \ldots, s_m)$ has

$$D_{\mathrm{image}} = \sum_r D_r(s_r), \qquad R_{\mathrm{image}} = R_{\mathrm{header}} + \sum_r R_r(s_r).$$

Choosing among these tuples under a byte budget gives Equation 4.

Proof. Because each pixel belongs to exactly one region, regional squared errors add. Concatenating the counted payloads and metadata gives the byte sum. Any state-dependent overhead is charged to the corresponding regional rate.

Dominance and supported alternatives. A state can safely be removed if another state in the same region has no greater rate or distortion and is strictly better in at least one. Such a replacement cannot worsen the optimum within that bank. Convex-hull filtering is different. With one region, alternatives $(R, D) = (1, 10), (2, 6), (3, 0)$ have an unsupported middle point. At budget 2, the middle point is nevertheless the best discrete feasible choice. Thus convex-hull filtering and marginal greedy upgrades do not generally preserve the integer-budget optimum. The next result characterizes the supported allocations generated by the regional lower hulls.

Proposition A.8 (supported regional allocations). Fix finite regional alternative sets and the additive errors and complete byte accounting of Proposition A.7. For $\lambda \geq 0$, minimizing

$$D(s) + \lambda R(s) = \lambda R_{\mathrm{header}} + \sum_r \left[D_r(s_r) + \lambda R_r(s_r)\right]$$

separates into regional minimizations. On each regional lower RD hull, consecutive states have increasing rates and decreasing errors, with nonincreasing slopes

$$g_{r,j} = \frac{D_r(s_{r,j}) - D_r(s_{r,j+1})}{R_r(s_{r,j+1}) - R_r(s_{r,j})}.$$

A prefix of the globally decreasing slope order, respecting regional predecessors, minimizes the Lagrangian for a common supporting λ. At tied slopes, any predecessor-legal partial prefix remains a minimizer at that slope.

Proof. Additivity separates the objective. A regional upgrade changes it by $(\lambda - g_{r,j})\Delta R_{r,j}$, so all slopes above λ are preferred, those below are rejected, and equal slopes tie. Hull slopes are nonincreasing, making these choices predecessor-legal. For any merged prefix, choose λ between the next and last selected slopes (equal to their common value for a tie). The empty prefix uses a slope at least as large as every upgrade. The full prefix can use zero. Linear objectives attain their minima on the lower hull, so removing unsupported states does not alter this Lagrangian minimum over the full bank.

Corollary A.9 (budget-slack bound). Let a supported selection have distortion $\widehat{D}$, complete size $\widehat{R} \leq B$, and supporting slope $\lambda \geq 0$. If $D^\star(B)$ is the best distortion under budget B in the same full alternative bank,

$$0 \leq \widehat{D} - D^\star(B) \leq \lambda(B - \widehat{R}). \tag{13}$$

In particular, the selection is budget-optimal within the bank at its own attained rate $B = \widehat{R}$.

Proof. Feasibility gives the lower bound. For a budget-optimal $s^\star$, $\widehat{D} + \lambda\widehat{R} \leq D(s^\star) + \lambda R(s^\star) \leq D^\star(B) + \lambda B$. Rearranging yields the upper bound.

This bound applies to supported-prefix selections in a fixed candidate bank with additive distortion and complete byte accounting. Numerical comparisons of prefix, remainder-filled, and exact selections are reported in Appendix E.

A.5 Lossless contributor mapping

The block dictionary and lossless per-pixel indices preserve the decoder's ordered contributor lists.

Proposition A.10 (lossless selection compilation). Let $K_{\mathrm{dec}}(x) = (i_1, \ldots, i_K)$ be the ordered contributors produced by the deployed decoder. For each block b, let a dictionary H_b store their union and let lossless local indices satisfy $H_b[j_{x,k}] = i_k$. This representation preserves every selected contributor and its order exactly. A full-ID fallback preserves the same property for blocks that exceed dictionary capacity.

Proof. Substituting $H_b[j_{x,k}]$ for i_k recovers the original ordered list at every pixel. Fallback blocks retain that list directly. With the same geometry, colors, score evaluation, and blending arithmetic, the output is unchanged.

Corollary A.11 (exhaustive-candidate sufficiency). Let $K_{\mathrm{exh}}(x)$ be the exhaustive regional Top-K under a fixed tie-breaking rule. If a candidate set contains $\bigcup_{x \in b} K_{\mathrm{exh}}(x)$, reranking within that set reproduces exhaustive rendering, with identical scores and blending.

Proof. Every exhaustive winner remains available, and no other candidate ranks above it under the same scores and tie-breaking. The selected list and its reconstruction are therefore unchanged. This sufficient condition requires coverage of the exhaustive winners. A finite cache capacity or absence of overflow alone does not establish it.

A.6 Numerical checks on decoded states

Contributor stability. We use the first raster-order region containing more than eight primitives from `big_building`, `GF2_0`, and `ST_0` at both fixed-capacity settings. Each state contributes 128 predetermined center-block and dispersed pixels. Positions are additionally rounded to 12-, 10-, and 8-bit grids over the regional extent, with shapes and colors fixed. As shown in Table 3, exhaustive CPU score evaluation measures contributor-set changes and the margin certificate.

Table 3: **Contributor stability under position quantization.** Six states, 128 pixels each. Changed denotes the number of pixels with Top-K membership changes. Certified denotes the number satisfying the margin certificate; Violations denotes changes among certified pixels.

Bits	Changed	Certified	Violations
12	76 / 768	3 / 768	0
10	281 / 768	0 / 768	0
8	630 / 768	0 / 768	0

Table 4: **Contributor coverage of block-level candidate pools.** CPU float64 evaluation on 1,152 pixels. Query and Exhaustive count Top-8 set matches to the respective references; RGB error is the maximum channel difference from exhaustive rendering in $[0, 1]$.

Selection	Entries	Query	Exhaustive	RGB error
Cell center	32	927/1152	599/1152	0.0840
Cell center	64	1002/1152	665/1152	0.1039
Cell center	128	1054/1152	712/1152	0.0533
Pixel union	32	1149/1152	807/1152	0.0030
Pixel union	64	1152/1152	810/1152	0.0030
Pixel union	128	1152/1152	810/1152	0.0030

Coordinate scaling. For each state, we examine three predetermined 8×8 blocks: a central block, a boundary block, and the home block of the most anisotropic primitive. Scaling pixels, positions,

radii, and regional extent together by 1, 2, and 4 leaves all evaluated logits and Top-K sets unchanged in float64 on these 18 blocks.

Candidate-pool coverage. On the same blocks, exhaustive regional scores provide a reference for center-ranked candidate pools and rank-major unions of pixel candidates. As shown in Table 4, the float64 comparison measures agreement with both the spatial query and exhaustive regional selection, together with RGB differences from the exhaustive reconstruction.

B Implementation Details

Algorithm 1 summarizes the common fitting, coding, and rendering stages. The following sections specify the regional parameterization, optimization schedules, stream format, and rendering layout, followed by the SAD baseline implementation.

B.1 Regional representation

Regional partition and coordinates. For an image of width W and height H, the target side length L determines $n_x = \lceil W/L \rceil$ columns and $n_y = \lceil H/L \rceil$ rows. Their boundaries are

$$x_j = \lfloor jW/n_x \rfloor, \quad j = 0, \dots, n_x, \qquad y_k = \lfloor kH/n_y \rfloor, \quad k = 0, \dots, n_y.$$

Cumulative rounding distributes residual pixels across the grid: widths, and separately heights, differ by at most one pixel. For region r, let o_r be its image-space origin and $S_r = \max(w_r, h_r)$ its larger side length. For an image-space pixel x, primitive position p_i, and radius r_i, define

$$\bar{x} = \frac{x - o_r}{S_r}, \qquad q_i = \frac{p_i - o_r}{S_r}, \qquad \rho_i = \frac{r_i}{S_r}.$$

The score normalizes $p_i - o_r$ and r_i, stored in regional pixel units.

Anisotropic score and parameterization. The geometry $g_i = (p_i, r_i, \tau_i, u_i, a_i)$ comprises position, radius, positive sharpness τ_i, a unit orientation vector $u_i \in \mathbb{R}^2$, and positive anisotropy a_i. Its determinant-one metric is

$$H_i = a_i u_i u_i^\top + a_i^{-1}(\mathrm{Id}_2 - u_i u_i^\top).$$

The implemented score is

$$\ell_i(x; g_i) = -\tau_i \left[\sqrt{\max((\bar{x} - q_i)^\top H_i (\bar{x} - q_i), \epsilon)} - \rho_i \right], \qquad \epsilon = 10^{-8}. \tag{14}$$

The floor protects the square root near a primitive center. Sharpness and anisotropy are optimized as $\log \tau_i$ and $\log a_i$; orientation is renormalized after updates and encoded as an angle.

B.2 Fitting and quantization-aware refinement

Regional capacity. Balanced, non-overlapping regions have a nominal side length of 1024 pixels. Low and High initialize at most 128,000 and 153,600 primitives per region, respectively, with capacity proxies $b_{\mathrm{cap}} = 9.6$ and 14.4. For $w_r h_r$ pixels, the requested population is

$$N_r^\star = \max\left(1, \mathrm{round}\, \frac{b_{\mathrm{cap}} w_r h_r}{128}\right).$$

The capacity proxy uses the original 128-bit-per-primitive convention to set a target count; encoded BPP is measured from the resulting stream. The pruning fraction is chosen by simulating the prescribed removal schedule toward $N_r^\star$. Integer removals and inactive-site cleanup produce the recorded final count.

Main-fit schedule. Each fit runs 1,600 updates with a learning-rate multiplier of four and three rotating pixel subsets. Pruning runs every 40 updates from 100 to 1000. At zero-based update t, ordinary updates use raster indices congruent to t modulo three and scale their gradient contributions by three; pruning updates use all pixels with unit weight for gradients and removal statistics. Contributor refresh occurs every 16 updates before update 1000 and at every update thereafter. Sharpness diffusion occurs every four updates. Three persistent regional workers share one GPU.

Algorithm 1 CORD: regional fitting, decoded-state coding, and repeated rendering.

Require: Image I, regions $\{\Omega_r\}$, fixed settings or byte budget B
for each region r and requested fitting capacity **do**
  Initialize regional geometry, colors, and a rich primitive population
  Rebuild contributors; apply safeguarded joint color initialization
  for each main fitting update **do**
    Refresh contributor lists at the prescribed frequency
    Select all regional pixels if pruning is scheduled; otherwise select the rotating stratum
    Compute gradients and, on pruning iterations, accumulate removal statistics
    Update geometry and colors with Adam
    Prune using the accumulated removal statistics, when scheduled
  end for
  Apply quantization-aware refinement when specified by the operating mode
  for each requested geometry-precision and color-capacity choice **do**
    Decode geometry; rebuild contributors and blending weights
    Refit primitive colors; construct and refine the palette; encode and decode
    Record regional decoded SSE and complete payload bytes
  end for
end for
Use the prescribed regional states, or select alternatives under byte budget B
Serialize and independently decode the complete image stream
Compile the decoder's contributors into shared dictionaries and local indices
Render each pixel by evaluating and blending its indexed contributors

Initialization. Initial contributor construction uses eight refresh passes, a radius of 34, and site-coverage handling.

The appearance solver uses at most eight preconditioned conjugate-gradient (PCG) iterations and stops early upon convergence. For active columns of A_0, let $\bar{d} = |M|^{-1} \sum_{i \in M} (A_0^\top A_0)_{ii}$. The regularizer in Equation 2 is $\lambda = \max\{10^{-6} \max(\bar{d}, 10^{-12}), 10^{-12}\}$: 10^{-6} scales the ridge coefficient relative to $\bar{d}$. A clamped proposal is tested at $\alpha \in \{1, 1/2, \ldots, 1/64, 0\}$ against the true fixed-support SSE. The main fitter then resumes scheduled contributor updates and Adam optimization.

Quantization-aware refinement. Both fixed-capacity settings use 1,000 QAT updates at a learning rate 0.01, with three-way pixel subsampling and sharpness diffusion every four updates. Contributor lists are rebuilt at the start of QAT and remain fixed throughout; target-rate schedules follow below.

The nine quantized fields and their bit widths in both fixed-capacity settings are

Field	p_x	p_y	$\log \tau$	r	R	G	B	θ	$\log a$
Bits	12	13	6	6	5	5	5	8	6

Here p_x, p_y are region-local pixel coordinates and $\theta = \operatorname{atan2}(u_y, u_x)$. Quantization bounds are frozen from the active input fit: positions use $[0, w_r - 1]$ and $[0, h_r - 1]$, orientation uses $[-\pi, \pi]$, and other fields use their observed minima and maxima. For a range narrower than 10^{-6}, QAT replaces its upper bound by the next representable float above its lower bound. For bounds $[l, h]$ and b bits, QAT clips v to the range and uses $k = \lfloor (2^b - 1)(v - l)/(h - l) + 1/2 \rfloor$, reconstructing $l + k(h - l)/(2^b - 1)$. The maximum angle code maps to zero because $-\pi$ and π describe the same direction. The RGB bit widths above are used during QAT. The final palette stores eight bits per channel, with 4,096 entries in Low and 32,768 in High. Each fixed-capacity setting fits and QAT-refines each region once.

B.3 Coding and candidate construction

Decoded-state appearance adaptation. For each requested encoding, the encoder decodes geometry and rebuilds contributors and blending weights. With this operator fixed, it refits primitive colors and uses them to initialize the palette, which is then refined against pixel reconstruction error.

Palette coding. The encoder uses weighted color clustering and support-conditioned palette projection, followed by 8-bit palette rounding, reassignment, and a final projection. Palette PCG has a 16-step cap, distinct from initialization. Large-palette acceleration reuses a deterministic histogram seed sequence and tiled exhaustive nearest-center computation, preserving exact nearest-center assignments. Palette workers share the same physical GPU. The stream stores primitives and encoded colors.

Physical palette stream. Each regional payload stores six geometry fields $(p_x, p_y, \log \tau, r, \theta, \log a)$, an 8-bit RGB palette, and one palette index per primitive. The encoder retains the source quantization bounds, rounds normalized scalar codes to nearest with ties to even, and canonicalizes the maximum angle code to zero. Unused palette entries are removed; occupied entries are ordered by their Morton codes in RGB space, and the indices are remapped.

Ordering and entropy coding. The fixed-capacity encoder orders sites lexicographically by coded $(\log \tau, r, \text{palette index}, \log a, \theta)$, with decoded-position Morton rank breaking ties. For each geometry field and the palette-index sequence, it chooses the shortest of absolute bit-packing and zigzag-delta varints, each compressed with either zlib (level 9) or LZMA (preset 6). Palette indices use $\max(1, \lceil \log_2 P_c \rceil)$ bits for P_c occupied colors; the palette bytes use zlib.

Container and byte accounting. The regional header records dimensions, population, palette count, quantization bounds and bit widths, ordering and codec identifiers, payload lengths, and a checksum. The image container records regional placement and payload descriptors. All headers and payloads contribute to reported size. Contributor maps are constructed after decoding and are not part of the stored stream.

Reusable target-rate bank. The bank-construction experiment uses capacity proxies 0.6, 1.2, 2.4, 4.8, 7.2, 9.6, and 14.4 to provide operating points over 0.3–4 BPP. Each capacity is fitted for 1,600 updates; palette variants reuse the corresponding fitted and QAT-refined state.

Capacity proxy	QAT updates	Palette capacities
0.6, 1.2	500	64, 128, 256, 4096
2.4	1000	64, 128, 256, 4096
4.8, 7.2, 9.6	1000	4096
14.4	1000	4096, 8192, 16384, 32768

Strictly dominated alternatives are removed before constructing regional lower hulls (Appendix A.4). When expanding a bank, a previously feasible stream is retained if the expanded selection does not improve its measured objective. Each final operating point is stored as a standalone stream. The ultra-low-rate ICB bank uses the 16-pass candidate renderer; fixed-capacity states and the medium/high-rate bank use spatial contributor reconstruction. Each profile is used consistently for evaluation and compilation of its selected contributors.

B.4 Contributor mapping and rendering

Each 8×8 block stores up to 64 four-byte contributor IDs in a shared dictionary and eight one-byte local indices per pixel. An overflow-block map selects full per-pixel IDs where required. The regular layout uses approximately 12 bytes per pixel, including the dictionary, compared with 32 bytes for eight full IDs; the overflow map and padding are accounted separately. Primitive arrays and the RGB output buffer are also tracked separately. The fused kernel retrieves the selected primitives, recomputes scores and weights, and writes the full RGB output. File parsing, mapping construction, and repeated rendering are distinct operations.

B.5 SAD large-image implementation

Our SAD baseline retains full-image optimization with the original representation, contributor-update rule, pruning schedule, and Adam optimizer. We make two numerical adjustments. First, the R2 low-discrepancy sampler evaluates its seed-dependent products and fractional-part operations in double precision before converting the sampled coordinates to FP32. This avoids the loss of fractional precision at large seed indices. Second, candidate positions are stored in FP32 rather than

FP16 when constructing and ranking contributors, avoiding coarse rounding of pixel-space coordinates on large images; the remaining packed candidate attributes retain their original precision.

We use $\log \tau = 9$ and $N_{\text{init}} = 128{,}000\lceil W/1024 \rceil \lceil H/1024 \rceil$, matching the aggregate initial population of the regional configuration while optimizing all primitives jointly over the full image. This setting gave the highest quality among the initializations in our preliminary large-image comparison, retaining full-image competition and pointwise color initialization.

We retain an initial radius of 4 pixels and optimize for 4000 updates with a unit learning-rate multiplier. Base learning rates for position, log-sharpness, radius, RGB appearance, direction, and log-anisotropy are 0.025, 0.01, 0.01, 0.01, 0.0025, and 0.025, respectively. Adam uses $(\beta_1, \beta_2) = (0.9, 0.999)$ and $\epsilon = 10^{-8}$. Contributors are refreshed every eight updates with one propagation pass on the full-resolution candidate grid. Densification is scheduled every 20 updates within 20–3000, and concurrent pruning every 40 updates within 100–3000. Their base fractions are 0.01 and 0.033, respectively.

C Evaluation Protocol and Additional Comparisons

C.1 Datasets, metrics, and timing

ICB contains *big tree* (6088×4550) and *big building* (7216×5412). The four author-provided FGF2 images are 7300×6908. STimage dimensions are 6954×7435, 9830×8602, and 9272×9931. Evaluation uses the fixed RGB inputs of the dataset manifest.

Dataset-level metrics are arithmetic means of per-image PSNR, SSIM, and LPIPS values computed from the reconstructed RGB images using the same evaluation implementation. A dataset mean is reported only when all images have measurements.

Optimization time is the elapsed fitting-plus-QAT time for an image on one A800, excluding metric evaluation and subsequent coding. Regional workers can overlap on that GPU, so image-level elapsed time is recorded directly rather than summed across workers.

CORD's representation size includes all decoder-required payloads and headers. Endpoint comparisons use each configuration's measured quality and size. The rate–distortion comparisons use the measured operating points. The ICB comparison includes locally measured streams and digitized published curves. Interpolation is restricted to overlapping measured ranges. For encoded representations, sizes count complete streams, and independent decoding produces the RGB reconstructions used for evaluation.

Metrics use each run's decoded reconstruction; support initialization can cause numerical variation.

Rate accounting. For the STimage RD comparison, CORD and SGI use $8S/(WH)$ BPP, where S counts all bytes in the self-contained encoded representation, including decoder-required metadata. SAD uses $128N_{\text{final}}/(WH)$ BPP, where N_{final} is the surviving primitive count after pruning; this parameter-space convention does not include container metadata.

C.2 ICB comparison and visual evaluation

Table 5 reports both fixed-capacity CORD settings and all baseline operating points, averaged over *big tree* and *big building*.

Figure 1(a) compares the same ICB sign detail near 0.8 BPP, with full-image PSNR and BPP annotations. Panel (b) plots FGF2 dataset-average PSNR against single-A800 optimization time on a logarithmic axis; marker area is proportional to stored MiB. Each marker denotes one evaluated operating configuration. Panel (c) presents the ICB rate–distortion comparison.

Figure 3 uses `GF2_2` reconstructions and reports full-image PSNR recomputed from the displayed files. All methods share crop coordinates and the error-map scale; the 768×768 context is centered on the same 64×64 detail. Figure 7 provides additional native 256×256 crops at the quarter point and image center using the same reconstruction files.

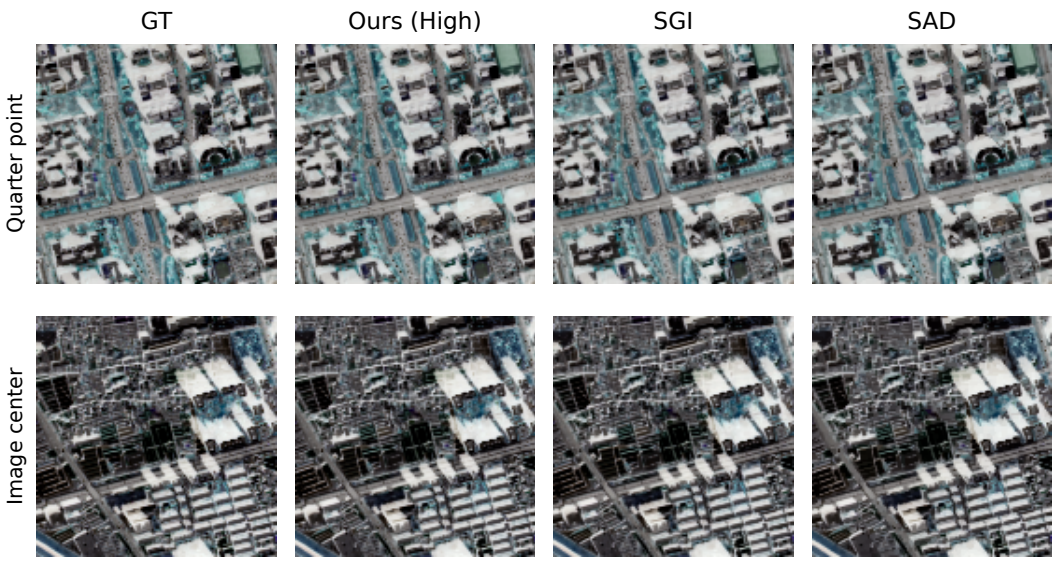


Figure 7: **Reconstruction at fixed FGF2 locations.** Native 256×256 crops from `GF2_2`, centered at the quarter point (top) and image center (bottom), with positions determined by image dimensions.

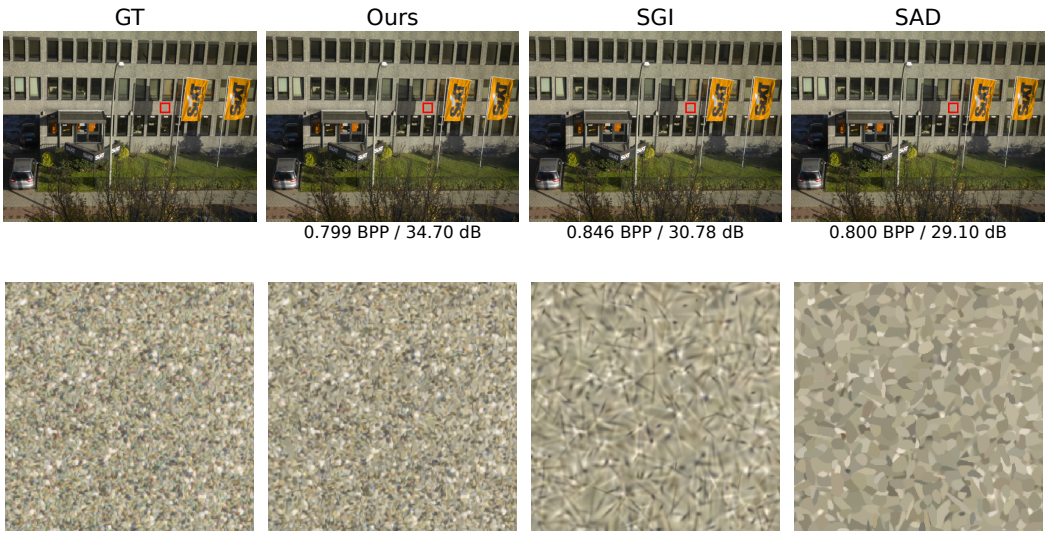


Figure 8: **Facade texture near 0.8 BPP on ICB.** Top: context crops from ICB, with annotations reporting full-image PSNR and BPP. Bottom: the same marked 256×256 facade region in GT and each reconstruction.

C.3 Low-rate visual comparisons

The low-rate comparisons show facade texture and glass lettering from `big_building` using identical crop coordinates across methods. Figure 8 pairs context crops with a native 256×256 facade detail; Figure 9 shows a separate lettering detail. Annotations report full-image PSNR and BPP. The facade comparison uses the SGI operating point at 0.8456 BPP, and the lettering comparison uses its 0.7974-BPP point.

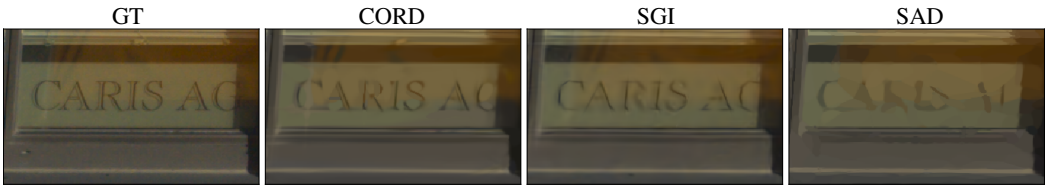
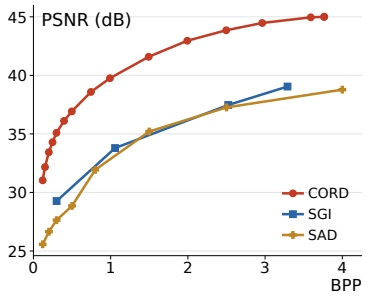


Figure 9: **Glass lettering near 0.8 BPP on ICB.** Identical crops from `big_building` for GT, CORD, SGI, and SAD. Annotations report full-image RGB PSNR and BPP.

Table 5: **Large-image representation on ICB.** Means over both images. Time is fitting plus QAT in A800 min/image; size is in MiB. CORD uses fixed-capacity Low/High settings. Shading groups methods; bold marks group-best results.

Method	PSNR↑	SSIM↑	LPIPS↓	Time↓	MiB↓
SIREN	26.891	0.6831	0.4201	113.72	15.79
I-NGP	34.026	0.8747	0.1746	2.10	20.84
HAC	36.548	0.9427	0.0862	241.85	14.47
GaussianImage	30.946	0.8300	0.2450	83.74	23.37
CORD (Low)	**38.778**	**0.9487**	**0.0743**	**1.52**	**13.42**
3DGS	37.575	0.9529	0.0607	304.06	787.73
Scaffold-GS	38.113	0.9592	0.0493	150.61	105.53
LIG	36.441	0.9274	0.0739	48.41	106.81
SGI	39.972	**0.9733**	**0.0350**	42.82	32.59
SAD	36.623	0.9123	0.1468	3.89	31.83
CORD (High)	**40.995**	0.9676	0.0428	**1.76**	**21.71**

Figure 10: **Rate–distortion performance on STimage.** Results averaged over all three images.



C.4 Additional rate–distortion comparison

Figure 10 averages measured BPP and PSNR over all three images at each operating configuration. All methods use the same decoded reference images from the JPEG-encoded inputs. The plotted lines connect these configuration means. In this comparison, CORD and SGI rates count complete encoded streams. The SAD runs use $\log \tau = 9$ with rate-dependent initialization and pruning.

D Fitting and Optimization Controls

This section shows the cumulative SAD-to-CORD comparison and paired controls for initialization, pruning, contributor refresh, regional partitioning, and pixel sampling. Coding controls use independently decoded reconstructions; fitting-only controls are identified separately.

D.1 Sequential component attribution

Configurations. Table 2 evaluates successive configurations on the full 7300×6908 `GF2_0` image using one A800. The unmodified SAD reference starts with 7,168,000 primitives and retains 131,126. Subsequent configurations apply the large-image adjustments, initialization and scale alignment, regional fitting, joint initialization with sampling and scheduled refresh, and the pruning/refinement schedule. The joint-initialization/sampling row uses 4,000 updates, pruning through update 3000, a learning-rate multiplier of one, and sharpness diffusion at every update. The next row uses 1,600 updates, completes pruning at 1000, applies a learning-rate multiplier of four, and diffuses sharpness every four updates. Both use the 4,000-update learning-rate horizon, three-subset sampling, and periodic-to-dense refresh. Their final populations are 3,151,820 and 3,151,804. Coding and rendering variants reuse one fitted capacity and its QAT state. Size is measured using a common lossless container before quantization and the coded stream after quantization.

The initialization/scale row retains global competition, combines balanced seeding with Adam $\epsilon = 10^{-8}/56$, and averages two executions from the same saved initialization. This row and the regional-

Table 6: **Fitting-component ablations.** Decoded PSNR, encoded BPP, and fitting-plus-QAT time on one A800, averaged over three preselected images: ICB `big_building`, FGF2 `GF2_0`, and STimage `ST_0`. Within each capacity setting, all variants share the final QAT and coding configuration.

Setting	Fitting variant	PSNR	BPP	Fit + QAT (s)
Low	Complete	40.477	2.910	123.2
Low	Refresh every step	40.464	2.908	151.8
Low	Periodic refresh	39.817	2.903	112.7
Low	Support-only initialization	40.450	2.925	125.8
Low	Unprojected initialization	39.625	3.042	117.0
Low	Direct target population	39.929	2.988	119.3
Low	Late target population	40.235	2.891	126.3
High	Complete	42.473	4.613	139.4
High	Refresh every step	42.535	4.630	166.9
High	Periodic refresh	41.031	4.609	124.6
High	Support-only initialization	42.297	4.621	138.0
High	Unprojected initialization	41.023	4.943	129.4

fitting row rebuild the spatial index at each synchronized contributor refresh (one pass every eight updates). Regional fitting uses $\epsilon = 10^{-8}$ with regional loss normalization. Both use 4,000 updates; the regional row reports one complete 56-region fit. For these rows, rendering latency is the median of three cold requests per fitted state.

Counted cost. The cumulative time includes fitting, QAT, color and palette solves, and one setup for the renderer used in that row. Entropy coding, replay validation, clustering, and serialization are outside this measurement. The three rendering layouts use their respective setup times.

Operation	Time (s)
Primitive-color solve	1.315
Palette solve	2.643
Spatial-index setup	0.478
Candidate-cache setup	1.118
Compact-mapping setup	2.679

The three setup rows are alternatives, not cumulative additions.

Rendering measurements. R is the median of three full-image passes, each summing the SAD executable's contributor-build and draw timers. W measures synchronized full-image wall time over 30 frames after five warm-ups, with contributor state prepared once.

D.2 Initialization and fitting controls

The paired fitting controls use `big_building`, `GF2_0`, and `ST_0`, with the final QAT and coding profile fixed within each capacity setting. Support-only initialization rebuilds contributors but omits the color solve; Unprojected omits both contributor rebuilding and the color solve. QAT initialization is unchanged. Direct-start fitting retains the standard inactive-site removal policy.

The complete fitter reaches mean PSNRs of 40.48 dB in Low and 42.47 dB in High. In High, complete initialization improves PSNR by 0.18 dB over Support-only and 1.45 dB over Unprojected. Periodic-to-dense refresh reduces fitting-plus-QAT time by 19% and 16% in Low and High relative to refreshing every step, with PSNR changes of $+0.01$ and -0.06 dB. Tables 6 and 7 report aggregate and per-image results for all configurations.

D.3 Pruning timing and contributor refresh

The crossed control compares pruning windows 100–1000 and 540–1440, both with 40-update spacing, under periodic or dense final contributor refresh. Dense refresh begins at update 1000 in both

Table 7: **Per-image fitting-ablation results.** The three images contributing to the aggregate fitting comparison. Each entry reports independently decoded PSNR (dB), encoded BPP, and fitting-plus-QAT time (s) on one A800, in that order. Variants and coding settings match the aggregate table.

Setting	Variant	ICB: big building	FGF2: first image	STimage: first image
Low	Complete	40.575 / 3.359 / 100.0	39.273 / 3.347 / 129.6	41.582 / 2.023 / 140.1
Low	Refresh every step	40.558 / 3.351 / 129.8	39.436 / 3.345 / 166.1	41.398 / 2.028 / 159.6
Low	Periodic refresh	40.535 / 3.361 / 90.1	37.539 / 3.340 / 118.0	41.377 / 2.006 / 130.1
Low	Support-only initialization	40.574 / 3.366 / 102.8	39.189 / 3.349 / 136.7	41.586 / 2.061 / 137.7
Low	Unprojected initialization	40.044 / 3.392 / 100.4	38.219 / 3.393 / 134.2	40.613 / 2.342 / 116.3
Low	Direct target population	39.929 / 3.398 / 101.1	38.320 / 3.388 / 131.6	41.536 / 2.179 / 125.2
Low	Late target population	40.458 / 3.343 / 105.3	38.981 / 3.329 / 133.6	41.265 / 2.002 / 140.1
High	Complete	42.845 / 5.428 / 118.9	41.336 / 5.410 / 148.8	43.238 / 3.003 / 150.6
High	Refresh every step	42.838 / 5.422 / 138.1	41.528 / 5.410 / 184.0	43.238 / 3.056 / 178.6
High	Periodic refresh	42.763 / 5.428 / 100.3	37.336 / 5.387 / 134.4	42.993 / 3.014 / 139.1
High	Support-only initialization	42.813 / 5.434 / 111.8	41.053 / 5.406 / 150.1	43.025 / 3.023 / 151.9
High	Unprojected initialization	41.612 / 5.517 / 110.6	39.604 / 5.544 / 141.6	41.852 / 3.768 / 135.9

pruning schedules; Periodic refreshes every 16 updates throughout. All runs share the initial population, requested target counts, pixel schedule, QAT/coding configuration, and 1,600-update budget.

The ICB/FGF2 comparison matches final regional populations (Table 8). Under dense final refresh, early completion improves decoded PSNR by 0.21 dB in Low and 0.41 dB in High. Table 9 separately reports all STimage schedule combinations together with their measured final populations.

Table 8: **Pruning timing and contributor refresh at matched population.** ICB/FGF2 means with shared QAT/coding settings. E/L denote Early/Late pruning; P/D denote Periodic/Dense refresh. Time includes fitting and QAT.

Setting	Prune	Refresh	PSNR	BPP	Time (s)
Low	E	P	39.037	3.351	104.1
Low	E	D	39.924	3.353	114.8
Low	L	P	38.567	3.335	106.8
Low	L	D	39.719	3.336	119.4
High	E	P	40.049	5.407	117.4
High	E	D	42.090	5.419	133.8
High	L	P	40.057	5.391	120.1
High	L	D	41.682	5.397	133.7

Table 9: **Pruning and refresh results on STimage.** The same schedules on `ST_0`. N counts final primitives after inactive-site removal; quality and rate are measured after decoding and encoding, respectively.

Setting	Prune	Refresh	Final N	PSNR	BPP
Low	E	P	3280090	41.377	2.006
Low	E	D	3347364	41.582	2.023
Low	L	P	3237119	41.356	1.984
Low	L	D	3257729	41.265	2.002
High	E	P	4991863	42.993	3.014
High	E	D	4852006	43.238	3.003
High	L	P	4855406	42.966	2.982
High	L	D	4833338	42.996	3.010

D.4 Encoded regional partitioning

Inputs and configurations. The encoded partition and sampling studies use the native 2048×2048 window starting at $(1024, 1024)$ in `big_building`, `GF2_0`, and `ST_0`, with coordinates and seed zero fixed. Each has seven configurations: region sides 512, 1024, and 2048 using three pixel subsets; a shifted 1024 layout; and one, two, and four subsets on the unshifted 1024 layout. Across the three images, these seven configurations yield 21 fitted and independently encoded results.

Partition and initialization. We use the balanced partition of Appendix B.1. The shifted layout moves the internal boundaries from 1024 to 1280 on both axes while retaining the outer 2048×2048 extent and four regions. A coordinate-indexed deterministic seed places 8,000 primitives in each native 256^2 cell, producing the same 512,000 initial positions, point-sampled colors, radii, and anisotropies in every layout. Primitives are assigned to regions by their centers. Positions are translated to the regional origin, and log-sharpness is adjusted by $\log(\max(w_r, h_r)/1024)$ to preserve the initial score scale. The target population is 262,144, apportioned by regional area; the table reports the final counts produced by the unchanged percentile-based pruning rule.

Table 10: **Regional extent and placement under a fixed coding profile.** Decoded quality, encoded rate, optimization cost, and maximum sampled device memory on three fixed 2048^2 crops. Side is the nominal region length in pixels; Shift moves both internal 1024-pixel boundaries to 1280. Fit includes preparation, initialization, pruning, optimization, and output; QAT is timed separately. Max. sampled memory is the largest device-memory sample, polled every 20 ms during fitting and including CUDA contexts. N is the final population for a target of 262,144.

Image	Side	PSNR	BPP	Fit (s)	QAT (s)	Max. sampled memory (GiB)	N
ICB	512	38.460	3.268	13.43	5.31	0.98	262144
ICB	1024	38.008	2.857	12.79	4.76	1.14	262148
ICB	2048	36.951	2.606	13.66	5.34	1.77	262142
FGF2	512	40.034	3.243	13.03	5.27	0.98	262144
FGF2	1024	39.366	2.836	13.13	5.36	1.14	262148
FGF2	2048	38.033	2.608	13.22	5.09	1.77	262142
STimage	512	42.527	2.648	11.62	4.99	0.98	262144
STimage	1024	42.635	2.315	11.07	4.84	1.14	262148
STimage	2048	41.397	2.158	11.50	5.00	1.77	262142
Partition placement/shape control (four regions)							
ICB	Shift	37.896	2.834	13.40	5.08	1.27	262144
FGF2	Shift	39.344	2.830	13.14	4.95	1.27	262144
STimage	Shift	42.194	2.189	11.28	4.89	1.27	262144

Fitting, coding, and measurement. Every run executes 1600 main updates on the common 4000-update learning-rate schedule, with a learning-rate multiplier of four. Pruning is enabled from update 100 to 1000 at frequency 40; contributor refresh is periodic at frequency 16 and dense from update 1000. The same initial support/appearance projection and sharpness diffusion are retained. Subsequent QAT uses 1000 updates, learning rate 0.01, and three pixel subsets for every arm. Attribute bits are $(12, 13, 6, 6, 5, 5, 5, 8, 6)$, with the same bound-estimation rule, followed by a 4096-entry regional palette, appearance projection, and entropy coding. Each region has its own palette and header, and position quantization uses the regional extent with fixed bit widths. Final quality and size are measured after the same appearance-adaptation and coding procedure.

Each configuration uses one serial regional worker on one otherwise idle A800; concurrent configurations occupy different GPUs. Elapsed fitting includes crop preparation, initial-site construction, projection, pruning, optimization, final-site output, and server startup/shutdown. The QAT stage is measured separately and added for total training time. Entropy coding and independent replay are not included in that time. Device memory is sampled through NVML every 20 ms during fitting, including CUDA contexts; we report the maximum sampled device memory. Final metrics and byte counts use independent decoding of the complete containers; agreement with encoding-time evaluation is within 0.0003 dB.

D.5 Pixel sampling

We vary only the main-fit subset count $m \in \{1, 2, 3, 4\}$ under the encoded protocol above. Update t uses flattened regional indices satisfying $u \bmod m = t \bmod m$, with sampled gradients scaled by m. Pruning updates use all pixels with unit weight. All arms share regional inputs, initialization, iteration budgets, learning rates, contributor schedules, pruning evaluations, final populations, and subsequent QAT and coding. Three-subset sampling reduces fitting-plus-QAT time by 15–20% and improves decoded PSNR by 0.14–0.21 dB relative to full-pixel main-fit updates on these three crops. Table 11 reports all subset counts.

D.6 Boundary reconstruction

We evaluate identical masks around the default boundaries at 1024 and shifted boundaries at 1280 on every decoded reconstruction. Half-widths are 2, 4, 8, and 16 pixels. Horizontal and vertical strips are unioned, counting intersections once and excluding the outer image boundary. Table 12 reports eight-pixel-half-width comparisons, with $\Delta\mathrm{PSNR} = \mathrm{PSNR}_{\mathrm{layout}} - \mathrm{PSNR}_{\mathrm{single}}$ on the same mask,

Table 11: **Pixel sampling during the main fit.** One subset evaluates all pixels at each update; larger counts cycle through spatially interleaved subsets, with full-pixel pruning updates. Each image shares initialization, final regional populations, and subsequent 1,000-step QAT and coding across variants. Fit + QAT excludes coding and independent decoding.

Image	Subsets	PSNR	BPP	Fit (s)	QAT (s)	Fit + QAT (s)
ICB	1	37.856	2.871	16.39	4.97	21.36
ICB	2	37.938	2.857	13.76	5.05	18.81
ICB	3	38.008	2.857	12.79	4.76	17.55
ICB	4	38.071	2.841	12.44	4.95	17.38
FGF2	1	39.156	2.850	16.54	5.29	21.84
FGF2	2	39.294	2.840	13.35	5.16	18.51
FGF2	3	39.366	2.836	13.13	5.36	18.49
FGF2	4	39.461	2.827	12.54	5.21	17.75
STimage	1	42.496	2.372	15.05	4.89	19.93
STimage	2	42.576	2.358	12.06	4.99	17.05
STimage	3	42.635	2.315	11.07	4.84	15.92
STimage	4	42.517	2.308	10.68	4.84	15.51

Table 12: **Boundary reconstruction under encoded layouts.** (a) Layout-minus-single-region PSNR (dB) on identical eight-pixel-half-width masks around the default (1024) or shifted (1280) lines. (b) Area and decoded-SSE shares of the default mask in the 1024-pixel layout.

Image	Mask	512	1024	Shifted
ICB	Default	+1.697	+1.073	+0.986
ICB	Shifted	+1.291	+1.222	+0.953
FGF2	Default	+2.250	+1.148	+1.210
FGF2	Shifted	+2.146	+1.217	+1.115

Image	Area (%)	SSE (%)
ICB	1.56	2.61
FGF2	1.56	1.12
STimage	1.56	0.73

where Single denotes the single-region encoded reconstruction. It also reports default-mask area and SSE shares. All masks use the same reference RGB pixels. Figure 11 uses the fixed window from $(896, 896)$ to $(1408, 1408)$, spanning both boundary positions. Reconstructions retain their original RGB values, and all error maps share a 0–25 RGB RMSE scale in 8-bit units. Boundary markers appear on GT only.

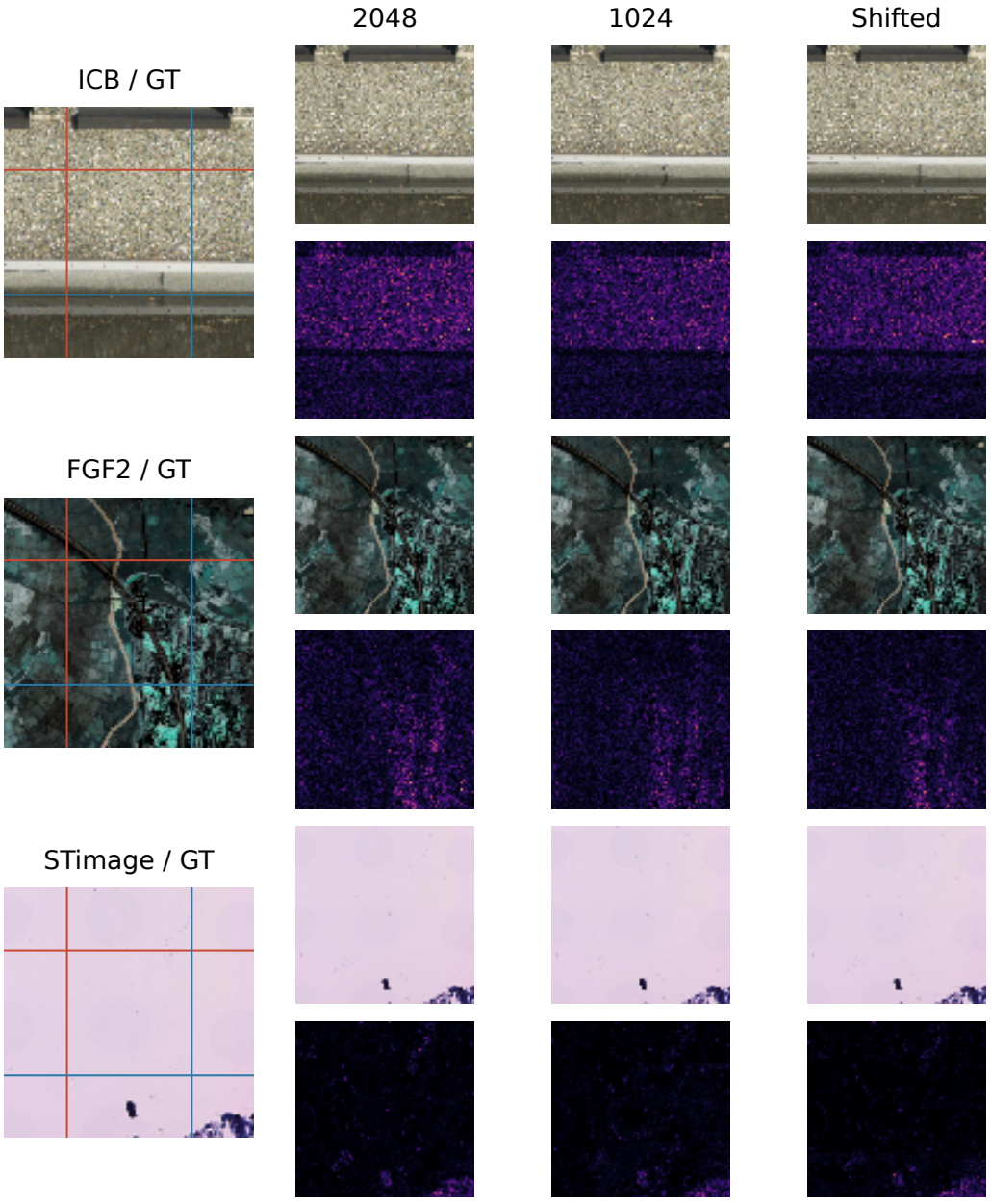


Figure 11: **Reconstruction across regional boundaries.** The same fixed 512×512 window is shown for each diagnostic image under a single 2048-pixel region, the default 1024-pixel layout, and the shifted layout. Red and blue lines on GT mark the default and shifted boundaries. Each reconstruction is paired with its RGB RMSE map, using a shared 0–25 scale in 8-bit units.

D.7 Fixed-population and scale controls

Regional scale. We use native 4096×4096 crops starting at $(1024, 1024)$ in `big_building`, `GF2_0`, and `ST_0`, with region sides 512, 1024, and 2048. Each configuration starts with 2,048,000 primitives and ends with 524,288 after 1,600 updates. This is a fixed primitive budget corresponding to the analytical 4-BPP convention; metrics are measured before QAT and encoding (Table 13).

Fixed-population partitioning. The bounded/global control uses native 2048×2048 crops, three seeds, and 131,072 primitives throughout fitting. Layouts share initial positions and colors, optimizer settings, and iteration budgets. Regional layouts assign primitives by center, translate posi-

tions, and adjust sharpness to preserve the initial score scale. Candidate queries, diffusion boundaries, and the normalized distance floor follow each layout's regional rules. Regions execute sequentially on one A800. Solver-call time includes projection, optimization, and solver I/O, with Python input preparation and stitching excluded.

Boundary comparisons use identical band pixels in the regional and global reconstructions, with half-widths of 2, 4, 8, and 16 pixels. Table 14 includes the eight-pixel-half-width results for the 1024-pixel layout, with $\Delta\mathrm{PSNR} = \mathrm{PSNR}_{\mathrm{regional}} - \mathrm{PSNR}_{\mathrm{global}}$ on the same band pixels. SSE-share-to-area-share ratios describe error concentration; all comparisons use the original RGB crop as reference.

Table 13: **Regional scale before coding.** PSNR and fit time on 4096^2 crops after 1,600 updates. All layouts start/end with 2,048,000/524,288 primitives; QAT and encoding are excluded.

Image	Region	PSNR	Fit (s)
ICB	512	35.964	28.81
ICB	1024	36.302	28.07
ICB	2048	36.265	30.14
FGF2	512	35.838	28.58
FGF2	1024	36.077	28.09
FGF2	2048	36.012	29.89
STimage	512	35.908	28.02
STimage	1024	36.044	28.11
STimage	2048	36.195	29.27

Table 14: **Fixed-population partitioning.** Three-seed means on 2048^2 crops with 131,072 primitives and no pruning. Δ is regional-minus-global PSNR; Time is the regional/global solver-time ratio. Band reports the 1024-layout difference on identical eight-pixel-half-width bands.

Image	Region	Δ	Time	Band
ICB	512	+0.185	1.081	—
ICB	1024	+0.163	1.050	-0.164
ICB	2048	0	1	—
FGF2	512	+0.245	1.089	—
FGF2	1024	+0.199	1.057	-0.255
FGF2	2048	0	1	—
STimage	512	+0.288	1.141	—
STimage	1024	+0.222	1.051	+0.008
STimage	2048	0	1	—

D.8 Initialization and execution repeatability on ICB

We evaluate repeatability on the full ICB `big_building` image at both fixed-capacity settings. Seeds 0, 1, and 2 control deterministic phase shifts of the initialization's gradient-weighted R2 sequence; seed 0 retains the original phase. We additionally repeat seed 0 twice, giving five complete fitting, QAT, and coding runs per setting. Each setting uses the same A800 GPU and unchanged optimization and coding parameters. The different-seed and same-seed groups each contain three runs and share the initial seed-0 run. Values below are means $\pm$ sample standard deviations; time includes fitting and QAT. The SSIM column reports $1 - \mathrm{SSIM}$ in units of 10^{-3}.

Setting	Seeds	PSNR (dB)	$(1-\mathrm{SSIM})\times 10^3$	LPIPS	BPP	Time (s)
Low	Different	40.5795 ± 0.0007	29.439 ± 0.011	0.05483 ± 0.00013	3.3603 ± 0.0010	107.9 ± 1.9
Low	Same	40.5767 ± 0.0021	29.446 ± 0.014	0.05485 ± 0.00011	3.3602 ± 0.0007	108.9 ± 1.0
High	Different	42.8494 ± 0.0062	18.990 ± 0.022	0.03175 ± 0.00006	5.4277 ± 0.0015	124.8 ± 3.1
High	Same	42.8419 ± 0.0027	19.013 ± 0.011	0.03182 ± 0.00003	5.4259 ± 0.0017	124.9 ± 2.8

Both initialization changes and repeated execution yield closely matched reconstructions on this image: PSNR standard deviations remain below 0.007 dB and encoded-rate standard deviations below 0.002 BPP in both Low and High.

E Decoded-State Coding and Rate Allocation

We evaluate appearance adaptation, the interaction between geometric precision and color capacity, the error criterion used to score alternatives, and image-wide allocation under fixed byte ceilings. Candidate construction and all paired controls are specified below; encoding and budget-query costs are reported in Appendix F.

Table 15: **Appearance adaptation before and after geometry decoding.** Six preselected regional states. Under Unquantized, Init. and Refit report PSNR on the original geometry before and after color adaptation. Before decoding and After decoding use identical final decoded geometry, 16 PCG iterations, and subsequent palette coding, differing in when colors are refitted. Their PSNR is measured after decoding; BPP counts the regional payload.

		Unquantized		Before decoding		After decoding	
Image	Setting	Init.	Refit	PSNR	BPP	PSNR	BPP
ICB	Low	44.54	44.80	42.76	3.384	43.17	3.388
ICB	High	45.68	46.18	44.65	5.475	45.11	5.477
FGF2	Low	41.32	41.80	38.22	3.304	39.50	3.305
FGF2	High	40.54	41.20	38.36	5.458	39.53	5.471
STimage	Low	49.22	50.08	48.66	1.213	48.97	1.160
STimage	High	49.18	50.10	49.14	1.394	49.40	1.284

Table 16: **Quantization-aware refinement and final appearance adaptation.** Decoded PSNR and encoded BPP on six fixed regions with shared quantization bounds and palette capacities. Both paths use k-means palette coding; Final refit adds primitive-color refitting and palette projection.

		No QAT				1,000 QAT updates			
		No final refit		Final refit		No final refit		Final refit	
Image	Setting	PSNR	BPP	PSNR	BPP	PSNR	BPP	PSNR	BPP
ICB	Low	41.15	3.321	43.17	3.388	39.84	3.320	43.24	3.389
ICB	High	41.82	5.040	45.11	5.477	40.51	5.040	45.21	5.474
FGF2	Low	37.50	3.261	39.50	3.305	38.00	3.266	39.76	3.306
FGF2	High	37.06	4.913	39.53	5.471	38.17	4.913	40.29	5.470
STimage	Low	43.31	1.074	48.97	1.160	41.81	1.083	48.90	1.153
STimage	High	43.90	1.146	49.40	1.284	42.29	1.156	49.29	1.277

E.1 Appearance adaptation

Controlled inputs. Both experiments use the first eligible raster-order region. The adaptation-order study starts from the fixed-capacity comparison fits; the QAT/refit study uses separate ablation fits with shared quantization ranges and schedules across the 0- and 1,000-QAT arms. Each experiment compares paired states from its own inputs.

Adaptation order. Starting from unquantized main-fit states, the Before path refits colors on the original operator and transfers them through the codec's exact primitive permutation onto decoded geometry, without additional scalar color rounding. The After path refits directly on that decoded geometry. Geometry rounding moves positions by up to 0.11–0.12 pixels. Both paths use 16 PCG iterations and identical subsequent palette coding. The Unquantized columns report PSNR before and after color adaptation on the original geometry; Before decoding and After decoding report the final decoded reconstructions. Refitting after decoding gains 0.27–1.28 dB across the six states.

QAT and final adaptation. The second control crosses 0 and 1,000 QAT updates with final appearance adaptation on six regions, using common quantization bounds and palette capacities. Final adaptation improves decoded PSNR in both QAT settings; Table 16 reports paired PSNR and BPP.

E.2 Geometry precision and color capacity

We reuse the first preselected Low fit from each diagnostic image and cross position precisions $(8, 9)$, $(10, 11)$, and $(12, 13)$ bits with 256, 1024, and 4096 palette entries. Primitive count, fitted state, and all remaining geometry bit settings are fixed. Each of the 27 alternatives receives its own decoded-geometry appearance adaptation and palette projection before physical serialization and evaluation. The resulting quality and rate measurements characterize the interaction between position precision and color capacity.

Increasing the palette from 256 to 4096 entries adds 0.23–0.43 dB at the coarsest position precision and 1.33–2.59 dB at the finest tested precision. Table 17 reports all 27 pairs.

Table 17: **Geometry precision and palette capacity.** Decoded PSNR (dB) / encoded BPP after appearance adaptation and palette coding. Within each image, all alternatives share the fitted region, primitive count, and remaining geometry precisions. Rows specify the bit widths for (p_x, p_y); columns specify palette capacity.

Image	Position bits	256 colors	1024 colors	4096 colors
ICB	8/9	34.44/2.360	34.66/2.559	34.77/2.766
ICB	10/11	39.83/2.676	40.79/2.873	41.29/3.089
ICB	12/13	41.02/2.974	42.32/3.171	43.17/3.388
FGF2	8/9	26.77/2.270	26.92/2.478	26.99/2.691
FGF2	10/11	34.17/2.579	34.99/2.789	35.35/3.006
FGF2	12/13	36.91/2.878	38.59/3.087	39.50/3.306
STimage	8/9	43.32/0.780	43.60/0.923	43.74/0.984
STimage	10/11	46.28/0.875	47.07/1.016	47.33/1.081
STimage	12/13	47.62/0.950	48.50/1.093	48.95/1.162

Table 18: **Low-rate coverage of nested candidate banks.** Actual BPP / decoded PSNR (dB), averaged over both ICB images at each requested budget. The banks successively add compact palettes and smaller fitted populations while reusing common states. Infeasible denotes a budget below the minimum complete-stream size available in that bank.

Budget	Original scalar choices	+ Compact colors	+ Small populations
0.15	Infeasible	Infeasible	0.148 / 29.085
0.20	Infeasible	Infeasible	0.198 / 29.702
0.25	Infeasible	0.250 / 29.566	0.248 / 30.049

Joint and single-variable selection. For each four-region ICB and FGF2 problem, we fix the highest fitted population and compare joint selection with geometry-only and palette-only selection at five byte ceilings. At each budget, geometry-only selection uses the best fixed palette, and palette-only selection uses the best fixed position precision. All policies use the same candidates and byte ceiling.

E.3 Low-rate candidate coverage

We evaluate nested candidate banks that add compact palettes and then smaller fitted populations to the scalar-coded alternatives, reusing all common states. Table 18 reports the encoded BPP and decoded PSNR of both complete ICB images at requested budgets of 0.15, 0.20, and 0.25 BPP. The complete bank provides feasible streams across this range.

E.4 Decoded-state candidate scoring

Both scoring policies select from the same encoded alternatives and use the same per-alternative byte costs and image-header accounting. They differ only in the reconstruction error used for selection. For each alternative, inherited-color error is measured after entropy decoding but before final per-primitive and palette adaptation, using the same decoded geometry and contributors as the completed alternative. Final-error scoring uses that completed alternative's decoded error. Both policies deliver reconstructions decoded from their selected completed alternatives.

Full-image population control. The three images contain 48, 56, and 56 regions. Each region has three population alternatives, with capacity proxies 4.8, 7.2, and 9.6 and a 4,096-entry palette, giving 480 encoded alternatives. The source colors have undergone fitting and QAT. Source and completed streams share the decoded geometry after accounting for primitive ordering; both color states are evaluated with the same rebuilt shared-query contributors. Budgets are fixed at 25, 50, and 75% of the span between the cheapest and most expensive complete selections. Final containers are decoded and their regional SSEs combined against the common reference. Table 19 reports actual rates, unused bytes, and quality for both prefix and remainder-filled allocation.

Table 19: **Candidate scoring and budget utilization.** Both policies select completed encoded alternatives with identical per-candidate byte costs. Inherited/Final score error before/after final appearance adaptation. Prefix uses the supported sequence; Filled adds feasible remainder upgrades. Each policy reports actual BPP, unused bytes (Slack), and final decoded PSNR under the same image-wide BPP ceiling.

Image	Budget	Prefix						Filled					
		Inherited			Final			Inherited			Final		
		BPP	Slack	PSNR	BPP	Slack	PSNR	BPP	Slack	PSNR	BPP	Slack	PSNR
ICB	2.127	2.117	47284	38.984	2.120	33965	38.999	2.117	47284	38.984	2.120	33965	38.999
	2.538	2.528	46234	39.740	2.534	19153	39.754	2.528	46234	39.740	2.534	19153	39.754
	2.949	2.942	33356	40.277	2.942	35070	40.278	2.942	33356	40.277	2.942	35070	40.278
FGF2	2.127	2.126	9948	36.753	2.125	11602	36.753	2.126	9948	36.753	2.125	11602	36.753
	2.534	2.532	11336	37.816	2.533	8919	37.818	2.532	11336	37.816	2.533	8919	37.818
	2.941	2.940	5211	38.754	2.927	86076	38.727	2.940	5211	38.754	2.940	5211	38.754
STimage	1.343	1.337	40351	39.919	1.331	76707	39.955	1.343	385	39.919	1.342	6162	39.963
	1.577	1.570	45602	40.930	1.575	15344	41.003	1.577	1732	40.934	1.577	3358	41.005
	1.811	1.782	184428	41.307	1.810	8680	41.396	1.810	7207	41.312	1.810	8680	41.396

Table 20: **Decoded-state scoring with joint coding alternatives.** Final PSNR gains (dB) over inherited-color scoring at identical byte ceilings. Each problem contains four fixed regions with 12 population/position-precision/palette alternatives per region. Both policies exhaustively evaluate all 12^4 selections from the same encoded alternatives using the same per-alternative byte costs. Column percentages locate budgets between the cheapest and most expensive complete selections.

	10%	25%	50%	75%	90%
ICB	+0.084	+0.276	+0.162	+0.394	+0.563
FGF2	+0.124	+0.363	+0.190	+0.450	+0.457
STimage	+0.008	+0.188	+0.204	+0.204	+0.169

Exhaustive mixed-candidate control. We use the first four raster-order regions of each diagnostic image. Each region combines three capacity proxies (4.8, 7.2, 9.6), two position precisions ($(10, 11)$ and $(12, 13)$ bits), and two palette capacities (1,024 and 4,096), with all other geometry settings fixed. The 144 regional alternatives are independently adapted, encoded, and decoded. Both scoring policies enumerate all 12^4 selections at budgets fixed at 10, 25, 50, 75, and 90% of the feasible byte span, including container overhead. Decoded-state scoring improves final PSNR by 0.08–0.56 dB on ICB, 0.12–0.46 dB on FGF2, and 0.01–0.20 dB on STimage. Table 20 reports all fifteen comparisons. Fixed-palette controls can lose quality at some budgets, whereas fixing the finer position precision does not reduce the optimum in any tested case.

E.5 Discrete allocation accuracy

Eight-region exhaustive reference. For the first eight raster-order regions of each image, we enumerate all $3^8 = 6{,}561$ selections using their original integer-byte costs. Budgets lie midway between consecutive supported rates. The largest supported-prefix SSE excess is 0.23–1.03% across these problems, and all measured gaps satisfy Equation 13. Table 21 reports prefix and remainder-filled selections against the same exact reference.

Full-image allocation optimality. Across nine budgets on three full-image candidate banks with 48–56 regions, our allocator has an average PSNR gap of 0.009 dB and a worst-case gap of 0.038 dB relative to the integer-programming optima. All nine reference solves terminate optimally with zero reported MIP gap; selected states satisfy the integer-byte budgets, including container overhead.

Table 21: **Allocation accuracy against exhaustive search.** Each image contributes its first eight regions and three capacity alternatives per region, giving $3^8 = 6{,}561$ selections. Tested budgets lie midway between consecutive supported rates. Gap is the maximum relative SSE excess over the exact feasible optimum across those budgets. Prefix and Filled denote the supported-prefix and remainder-filled policies; Exact time measures enumeration.

Image	Budgets	Prefix gap (%)	Filled gap (%)	Exact time (s)
ICB	16	0.233	0.233	0.007
FGF2	15	1.032	1.032	0.008
STimage	8	0.814	0.000	0.007

E.6 Content-adaptive regional allocation

We compare content-adaptive regional allocation with the best feasible uniform configuration from the same candidate bank. Each uniform configuration defines a common image-wide byte ceiling, giving 12 comparisons per image or four-region problem. Both policies count physical payloads and container overhead and are evaluated using decoded SSE; the resulting streams may leave different amounts of the budget unused.

The low-rate study uses both complete ICB images over 0.10–0.25 BPP. The medium-to-high-rate study uses the first four raster-order regions of ICB `big_building` and FGF2 `GF2_0` over 1.5–3.4 BPP. Mean PSNR gains are 0.16 and 0.18 dB on the complete ICB images and 0.44 and 0.56 dB on the four-region ICB and FGF2 problems, respectively. At one four-region FGF2 ceiling, the marginal-upgrade selection is 0.26 dB below the best feasible uniform configuration.

F Rendering and Computational Cost

F.1 Lossless mapping and storage

Paired layout comparison. The full-ID and compact layouts use identical decoded primitives, ordered contributor lists, and the same fused rendering kernel. Full IDs use the kernel's lossless fallback layout. We alternate execution order for 60 paired warm frames after five warm-ups and synchronize each completed RGB output. Memory includes the layout, primitive arrays, and one float32 RGB buffer, excluding duplicate tensors used only by the diagnostic. All six full-image pairs produce bitwise-identical RGB (Table 22).

Incremental construction. Both layouts start from the same GPU-resident primitives and selected IDs. Three alternating trials measure layout construction. The break-even render count is the additional layout-construction time divided by the measured per-frame saving, rounded up using unrounded timings. Common entropy decoding, GPU transfer, and contributor discovery precede this comparison (Table 23).

Table 22: **Resident contributor-layout memory.** MiB including primitives, selection, and one float32 RGB buffer. All full-image RGB pairs are bitwise identical.

		Resident memory (MiB)	
Image	Setting	Full IDs	Shared
ICB	Low	1758.1	1009.9
ICB	High	1814.0	1065.8
FGF2	Low	2277.3	1306.8
FGF2	High	2349.5	1378.9
STimage	Low	2321.3	1321.6
STimage	High	2382.8	1383.3

Table 23: **Incremental mapping construction.** Both layouts start from GPU-resident primitives and selected IDs. Three build trials and 60 paired warm renders measure extra construction cost and per-frame savings. Break-even uses unrounded timings.

Image	Setting	IDs (s)	Shared (s)	Extra (s)	Saved/render (ms)	Break-even (renders)
ICB	Low	0.009	0.096	0.087	0.268	324
ICB	High	0.006	0.110	0.104	0.233	447
FGF2	Low	0.010	0.157	0.147	0.329	447
FGF2	High	0.010	0.167	0.157	0.287	547

Selection preservation. Tests cover dictionary overflow, partial boundary blocks, nonzero output offsets, and constant regions. All tested IDs are recovered exactly, with bitwise-identical compact and full-ID renders. Rebuilding selection from a constant canvas provides a check that contributor construction is independent of reference-image content in the tested configuration. For six preselected regional streams at 0.15, 0.20, and 0.25 BPP, compilation of the 16-pass decoder selection changes PSNR by at most 1.6×10^{-5} dB. These regional checks use the low-rate decoding profile.

F.2 FULL-IMAGE RENDERING

Each image/setting is reconstructed once for quality evaluation. Three independent timing processes use the same compiled representation, with each process measuring 100 synchronized frames. A frame includes all regional launches, constant-region broadcasts, direct writes to the full RGB buffer, and GPU synchronization. Parsing, mapping construction, PNG conversion, and metric evaluation are outside this warm-rendering time. All measurements use A800 40-GB GPUs, with the same device retained for each image/setting across its three processes. SGI uses its released renderer and the reproduced decoded streams.

Table 24 reports PSNR, encoded size, and median/p95 latency. Table 25 additionally reports selection memory, resident state, map construction, and output agreement for all nine images at both capacities. Full mapping construction includes GPU transfer and contributor discovery; its reported time is the median of three full-image trials with the runtime initialized.

Table 24: **Full-image warm-rendering performance.** Median and 95th-percentile latency on one A800, pooled over three processes with 100 synchronized frames each. Decoded states and mappings are GPU-resident. Timings include complete RGB output writes and synchronization; loading, decoding, mapping construction, and image-file output are excluded. PSNR and size identify each method's operating point.

Image	Method	PSNR	Size (MiB)	Median (ms)	p95 (ms)
ICB building	SGI	40.39	31.94	74.24	74.92
ICB building	CORD Low	40.58	15.64	3.80	7.17
ICB building	CORD High	42.84	25.26	4.25	7.98
FGF2 third	SGI	37.18	46.91	79.46	79.88
FGF2 third	CORD Low	37.29	19.97	4.98	9.51
FGF2 third	CORD High	39.82	32.06	5.45	10.45

Table 25: **Rendering, construction, and memory across all image states.** Median/p95 latency pools 100 synchronized frames from each of three processes. Dictionary/dense measure selection storage; State also includes primitives, excluding RGB output. Map build is the median of three full-image trials, including GPU transfer and contributor discovery but excluding runtime initialization, entropy decoding, and parsing; a dash denotes an unreported value. Δ is rendered minus source PSNR. Overflow counts full-ID fallback blocks.

Image	Setting	MP	Median / p95 ms	Dictionary / dense MiB	State MiB	Map build s	Δ dB	Overflow
GF2_0	High	50.4	5.40 / 10.32	585.4 / 1555.9	801.8	2.72	3.4e-05	0
GF2_1	High	50.4	5.41 / 10.27	585.4 / 1555.9	801.8	2.62	-4.7e-07	0
GF2_2	High	50.4	5.45 / 10.45	585.4 / 1555.9	801.8	2.69	-1.5e-06	0
GF2_3	High	50.4	5.45 / 10.35	585.4 / 1555.9	801.8	2.64	-2.3e-06	0
ST_0	High	51.7	4.24 / 7.53	603.2 / 1602.7	791.6	—	-2.5e-05	124
ST_1	High	84.6	8.06 / 14.86	978.0 / 2599.6	1314.7	—	0.0e+00	7
ST_2	High	92.1	9.92 / 17.71	1068.0 / 2837.6	1463.2	—	-1.3e-04	244
big_building	High	39.1	4.25 / 7.98	451.2 / 1199.4	618.8	1.95	-2.6e-06	0
big_tree	High	27.7	3.14 / 5.77	319.8 / 850.0	438.7	1.48	-9.8e-07	0
GF2_0	Low	50.4	4.95 / 9.35	585.4 / 1555.9	729.6	2.26	7.8e-07	0
GF2_1	Low	50.4	4.94 / 9.32	585.4 / 1555.9	729.6	2.25	-1.3e-06	0
GF2_2	Low	50.4	4.98 / 9.51	585.4 / 1555.9	729.6	2.27	5.4e-06	0
GF2_3	Low	50.4	4.94 / 7.40	585.4 / 1555.9	729.6	2.20	-1.6e-07	0
ST_0	Low	51.7	3.91 / 7.17	603.1 / 1602.7	729.9	—	3.9e-06	56
ST_1	Low	84.6	7.44 / 8.49	978.0 / 2599.6	1202.5	—	-3.4e-06	3
ST_2	Low	92.1	8.63 / 16.42	1067.7 / 2837.6	1331.1	—	-1.8e-06	58
big_building	Low	39.1	3.80 / 7.17	451.2 / 1199.4	563.0	1.92	1.1e-05	0
big_tree	Low	27.7	2.76 / 5.22	319.8 / 850.0	399.0	1.23	-3.6e-07	0

F.3 ENCODING COST

Single-capacity encoding. For the complete `GF2_0` image, encoding 56 regions with 4,096-color palettes uses one A800 and eight CPU coding workers. The compared encoder implementations share the same fitted and QAT-refined primitives. Post-fitting processing takes 34.31 s for appearance adaptation, physical coding, regional decoded-error evaluation, replay checks, and stream assembly. Together with the separately measured fitting and QAT stages, the reported total is 146.18 s. Independent full-image validation and PNG export are excluded.

Table 26: Single-capacity encoding cost on GF2_0. Separately measured stages, in seconds.

Image	Fit	QAT	Post-fitting	Total
GF2_0	80.29	31.58	34.31	146.18

Reusable-bank construction. Constructing the seven-capacity bank for the complete ICB *big tree* image takes 406.1 s, including fitting, QAT, appearance adaptation, coding, validation, and file writing. Six regional workers share one A800, alongside eight CPU coding workers. The bank supplies operating points over 0.3–4 BPP; subsequent queries use its already encoded regional candidates.

F.4 BUDGET-QUERY COST AND REUSE

After constructing a bank once, additional target budgets reuse its encoded regional states without fitting, QAT, or appearance adaptation. Five queries per four-region problem take 23–42 ms each, including selection, candidate-file reads, and physical-stream assembly.

For complete ICB images, queries at 0.35, 0.8, 1.6, 2.8, and 3.8 BPP take 39–357 ms for *big tree* (30 regions) and 63–563 ms for *big building* (48 regions). These are per-budget medians over five repeated measurements with a prepared allocation frontier and warm file cache. Timing includes candidate reads, payload validation, container assembly, and file writing, excluding RGB rendering.