PREDICTING COMPRESSION GAINS BEFORE ENCODING: ATTRIBUTE MODELS FOR 2D GAUSSIAN IMAGES

Anonymous authors
Paper under double-blind review

ABSTRACT

Gaussian image representations encode an image as a set of 2D Gaussians and have become a practical format for image compression. Losslessly coding this set involves two interlinked design choices: the spatial organization of the Gaussians and the attribute model, i.e., the probability model for their attributes. These choices are usually made by building candidate coders and comparing total file size. A controlled study shows that this can mislead: among the organizations tested, a quadtree codes positions most compactly, yet produces larger files because of the attribute model paired with it. With the organization fixed, we propose a predict-before-encoding protocol for the attribute model: on a small selection split, it measures whether the information a quadtree cell carries about each attribute outweighs the cost of learning a per-cell model, and fixes each decision before any evaluation image is encoded. On 80 held-out images, the frozen decision beats coding every attribute per-cell, coding every attribute globally, and an equal-capacity non-spatial control. On Gaussian sets from a second fitting method, a codelength forecast computed from quantized symbol counts before entropy coding predicted a gain of 1.414 bits per Gaussian over per-cell coding of every attribute, against 1.361 realized, and correctly predicted the one comparison the chosen coder loses.

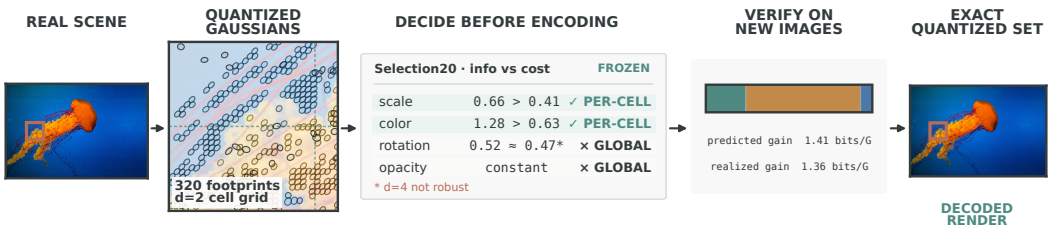

Figure 1: **Predict before encoding.** An image is fitted as a set of 2D Gaussians and quantized; the decoder reconstructs the quadtree cell of every Gaussian before it decodes any attribute (left). For each attribute we measure, on 20 selection images, how much information the cell carries against how many bits a per-cell model costs to learn, and code the attribute per-cell only when the information reliably wins: per-cell for scale and color, global for rotation and opacity (middle). The resulting routes are frozen before evaluation. On 20 new images, quantized symbol counts available before entropy coding predict a 1.414-bit/G saving in the attribute (MARKS) stream over coding every attribute per-cell. Encoding realizes 1.361 bits/G in MARKS and 1.360 bits/G in the complete file, with every byte counted (right). Exact decode refers to the quantized set, not to the source pixels.

1 INTRODUCTION

Gaussian image representations encode an image as a set of 2D Gaussians, each with a position and attributes such as scale, rotation, opacity, and color (Zhang et al., 2024b;a). They render quickly and have become a practical format for image compression, with a growing line of codecs that fit, quantize, and entropy code these Gaussians (Li et al., 2026; Gao et al., 2025; Cheng et al., 2026; Pan et al., 2026). Once the Gaussians are fitted and quantized, the reconstruction is fixed, and file size depends only on how the quantized set is losslessly coded (Figure 1).

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.

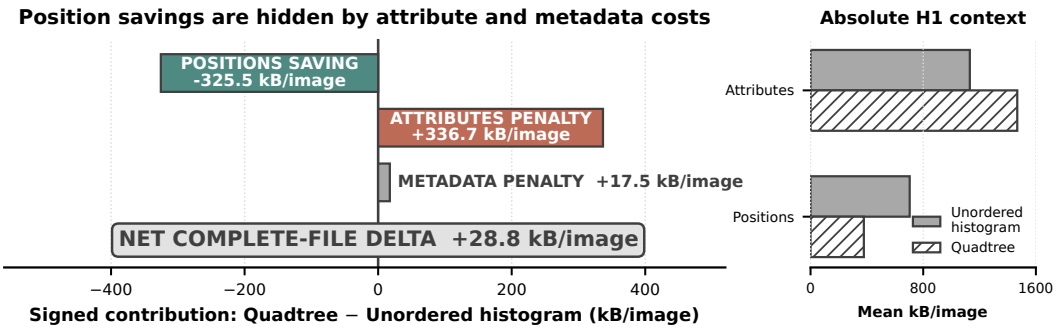

Figure 2: **Total file size hides where the bits go.** Per-segment byte changes of the quadtree coder relative to the unordered histogram coder on the same 100 images. The quadtree saves position bytes but spends more on attributes and metadata. Negative values save bytes.

Losslessly coding such a set involves two design choices. The first is the spatial organization of the Gaussians: a fixed scan order such as a Morton curve (Liu et al., 2025), an unordered treatment (Zhang et al., 2024b; Kunze et al., 2024), or a spatial hierarchy that the decoder reconstructs from coarse to fine (Huang et al., 2020; Que et al., 2021; Wang et al., 2024). The second is the attribute model, the probability model that determines how many bits each attribute value costs. The two are linked, because the spatial organization determines what context the attribute model can use. They are usually chosen by building candidate coders and comparing total file size.

Total file size can mislead, because it mixes the cost of positions with the cost of attributes. On Gaussian sets from 100 images, for example, a quadtree coder produces larger files than a simple unordered coder that transmits one histogram per attribute, so a size comparison would discard the quadtree. Yet the quadtree spends far fewer bytes on positions; its files are larger because the attribute model paired with it spends even more on attributes (Figure 2).

We therefore judge the two choices separately. Paired with the same histogram attribute model as the unordered coder, the quadtree produces files 7.42% smaller on a frozen 20-image subset (Section 6.1), so we keep it as the spatial organization for the rest of the paper. Its attribute model is harder to judge. Each attribute can be coded with a *global* model shared across the image, or with a *per-cell* model, one for each of the sixteen coarse quadtree cells, which can exploit the cell but has sixteen times as many probabilities to learn. The benefit and the cost of the per-cell model land in the same attribute bytes, so comparing file sizes shows which option won on the images tried, but not why, and therefore not whether the choice will hold on other images or other Gaussian representations.

We argue that this decision should instead be **predicted before encoding**. The gain of a per-cell model is the information the cell carries about the attribute minus the extra bits the per-cell model spends learning its probabilities. Both terms can be estimated from symbol counts on a few images, so the decision comes with its reason. We use a per-cell model only where this gain is reliably positive (Section 5).

On 20 selection images, this rule picks per-cell models for scale and color and global models for rotation and opacity. We call the resulting coder Hybrid, and the two uniform alternatives, per-cell and global for every attribute, A2 and A1. On the remaining 80 images, Hybrid achieves a lower mean attribute rate than A2, than A1, and than an equal-capacity control whose sixteen models carry no spatial information (A2′; Section 6); every bitstream decodes exactly and every byte is counted. On Gaussian sets fitted by a different method, GaussianImage (Zhang et al., 2024b), with 5k and 10k Gaussians per image instead of about 178k, a codelength forecast computed from the quantized symbol counts before entropy coding predicted an attribute-stream gain over A2 of 1.414 bits per Gaussian; encoding realized 1.361 in the attribute stream and 1.360 in the complete file. The same forecast predicted that A1 would beat Hybrid on these sparser sets, and it did, to within 1.19%.

Our contributions are:

- **A controlled study** showing that total file size can misjudge the spatial organization of a Gaussian image coder, because it mixes the costs of positions and attributes.

- **A predict-before-encoding protocol** that chooses, for each attribute, between a global and a per-cell attribute model from a few selection images, before any evaluation image is encoded.
- **Validation** on 80 held-out images and on Gaussian sets from a second fitting method, where a forecast made before entropy coding matches the realized gains and the one realized loss.

2 Related Work

Gaussian image and scene codecs. Gaussian primitives couple geometry to appearance in 3D scenes (Kerbl et al., 2023) and in 2D images. Codecs for the 2D case fit, quantize, and entropy code the primitives, using quantized position codebooks (Zhang et al., 2024a), vector quantization with a bits-back treatment of permutation symmetry (Zhang et al., 2024b), pre-trained attribute dictionaries (Gao et al., 2025), attribute-separated scalar quantizers (Li et al., 2026), clustered codebooks (Cheng et al., 2026), structured seeds (Pan et al., 2026), and amortized prediction (Zeng et al., 2026). For 3D scenes, C3DGS combines masking, vector-quantized attributes, and post-hoc entropy coding (Lee et al., 2024), and a line of work codes positions and attributes as separate streams: G-PCC positions with graph-transform attribute residuals (Yang et al., 2024), fitted-distribution arithmetic coding (Huang et al., 2025), HEVC-coded feature planes (Lee et al., 2025; Kim et al., 2026), per-file entropy-network parameters (Sridhara et al., 2025), standardized point-cloud streams (Yang et al., 2025), octree geometry with Morton-serialized attributes (Zhang et al., 2026), and separate structural and appearance streams (Hu et al., 2026). We take the representation and the quantizer as given and study only how the quantized set is losslessly coded.

Spatial organization and context for attribute coding. Three organizations recur. A fixed traversal such as a Morton curve gives an autoregressive context (Liu et al., 2025); an unordered treatment exploits exchangeability, formalized by permutation-invariant set models (Zaheer et al., 2017; Lee et al., 2019) and made operational by shuffle coding (Kunze et al., 2024); and a hierarchy such as an octree gives the decoder a coarse-to-fine spatial context (Huang et al., 2020; Que et al., 2021). For 3D Gaussians, HAC, HAC++, ContextGS, FCGS, and HEMGS build hierarchical, structured, and hybrid entropy models on such context (Chen et al., 2024; 2025b; Wang et al., 2024; Chen et al., 2025a; Liu et al., 2024), following learned hyperprior and autoregressive coding (Ballé et al., 2018; Minnen et al., 2018), and LG-HCC predicts fine-level entropy parameters from a geometry-aware anchor hierarchy (Deng et al., 2026). These works choose the organization and the attribute model together and compare total rate. We ask instead, attribute by attribute, whether the hierarchy's context is worth its cost, and we fix the answer before encoding.

Model cost and universal coding. That a richer model must repay its description cost is the minimum description length principle (Rissanen, 1978); the KT estimator and context-tree weighting quantify the cost of adapting a model online (Krichevsky & Trofimov, 1981; Willems et al., 1995). Our decision rule uses these classical quantities. The contribution is the protocol that measures them on a selection split, freezes the decision, and tests it on bitstreams that had no part in the choice.

3 Problem Formulation

Quantized Gaussian sets. Each Gaussian pairs a quantized position $\mathbf{x}_i$ with a tuple of quantized attributes $\mathbf{m}_i$ (scale, rotation, opacity, and color),

$$\mathcal{G} = \{(\mathbf{x}_i, \mathbf{m}_i)\}_{i=1}^{N}. \tag{1}$$

The set is unordered, but decoding must preserve which attributes belong to which position, including duplicate positions.

File size. For a coder C with fixed shared decoder state θ_C, we measure the bytes B_C of the complete file rather than an estimated cross-entropy, and split them into three segments,

$$L_C(\mathcal{G} \mid \theta_C) = 8|B_C| = L_{\text{geom}} + L_{\text{marks}} + L_{\text{meta}}, \qquad R_C = L_C/N. \tag{2}$$

In the bitstream, the position stream is called *geometry* and the attribute stream *marks*; MARKS denotes the attribute segment alone. L_{marks} includes the attribute payload, its framing, and any

probability table transmitted only to decode attributes; L_{meta} includes header, root, directory, and termination. Every byte is assigned to exactly one segment, and the rate R_C is in bits per Gaussian (bits/G).

Hidden cancellation. Because all compared coders reconstruct the same quantized set, the difference between the quadtree coder and Exchangeable decomposes exactly,

$$\Delta L_{\mathrm{QT-ex}} = \Delta L_{\mathrm{geom}} + \Delta L_{\mathrm{marks}} + \Delta L_{\mathrm{meta}}. \tag{3}$$

The signs of the three terms attribute the position saving to the quadtree and localize the loss in total size to the attribute model attached to it (Figure 2). Substituting one coder's attribute segment into another diagnoses the problem but is not itself a coder: syntax, state, framing, and decode order may not compose.

When does spatial context pay? Let M_h be one scalar attribute (a *head*) and C_d the depth-d quadtree cell of its Gaussian, which the decoder knows before it decodes any attribute. Each quadtree node at depth d covers a square cell of the image; at $d = 2$ the cells form a 4×4 grid. A *global* model codes M_h without C_d; a *per-cell* model keeps one model for each cell and codes M_h conditioned on C_d. Relative to the global model, the per-cell model changes the rate by

$$\Delta_h \simeq -I(M_h; C_d) + D_{1,h} - D_{0,h}, \tag{4}$$

where $I(M_h; C_d)$ is the information the cell carries about the attribute, and $D_{0,h}$ and $D_{1,h}$ are the predictive regrets of the global and per-cell models: the extra bits each spends, relative to the best fixed model of its class, because it must learn its probabilities from the data it codes. The per-cell model has more probabilities to learn, so $D_{1,h} > D_{0,h}$ in general. A coder that transmits a per-file table instead of adapting pays that table's description length $L(H_h)/N$; the adaptive coders compared below transmit no table, so this term is zero for them. Conditioning pays when

$$G_h = I(M_h; C_d) + D_{0,h} - D_{1,h} > 0. \tag{5}$$

This is classical MDL accounting (Rissanen, 1978), not a new inequality. What is new is the protocol: we estimate G_h for each attribute on a selection split, require the estimate to be robust, freeze the decision, and only then test it against physical bytes on images that played no part in it.

Data and cohorts. The main evaluation uses quantized Gaussian sets fitted by AIR (Zeng et al., 2026) to 100 DIV2K images (*Full100*), about 178k Gaussians per image on average. The diagnosis in Section 6.1 uses all of them. The decision rule is estimated on a 20-image split (*Selection20*); the remaining 80 images (*Heldout80*) are the internal test set and play no part in the decision. Five predeclared images (*Mechanism5*) serve only to check that the implementation composes as specified. A prospective study on GaussianImage representations of the DIV2K validation set uses a 5-image pilot (*Pilot5*) to authorize the run and a disjoint 20-image cohort (*Confirm20*) for the result. Finally, 24 Kodak images fitted by GaussianImage++ (*Kodak24*) test robustness to a different Gaussian producer. Every reported stream is decoded by a fresh decoder, and its bytes are reconciled segment by segment. Appendix Table 7 lists the evidentiary role of each cohort.

4 A Matched Family of Coders

All coders compared below share everything except the attribute model: quadtree-coded positions, the same canonical association of attributes to positions, 16-bit rANS entropy coding, the same attribute group schedule, the same 256-symbol block boundaries, and the same empty initial state. Adaptive coders keep symbol counts across blocks; the probability table used inside a block is frozen at its boundary, and encoder and decoder apply the same deterministic update afterward. Table 1 lists the arms.

Four causes, separated. The arms are ordered so that each step changes one thing. C3 (also called B0) transmits a dense per-file probability table for every attribute. B1 serializes the identical probabilities compactly, and B1F changes only the framing to match the adaptive arms. A1 transmits no table at all and instead keeps one backward-adaptive KT count table per attribute, so the decoder learns the probabilities as it goes; this is the global option for every attribute. A2 keeps sixteen such

Table 1: **The coders differ only in the attribute model.** Every arm codes positions with the same quadtree stream and decodes exactly. KT denotes an adaptive Krichevsky–Trofimov count table.

Arm	Attribute model	Context key	Transmitted table
C3 (B0)	one static per-file table per attribute	none	dense
B1	same probabilities as C3	none	compact
B1F	same as B1, framed like the adaptive arms	none	compact
A1 (global)	one adaptive KT table per attribute	none	none
A2 (per-cell)	16 adaptive KT tables per attribute	depth-2 quadtree cell	none
A2′	16 adaptive KT tables per attribute	geometry-blind hash	none
Hybrid	per-cell scale/color; global rotation/opacity	cell / none	none

tables per attribute, one for each cell of the depth-2 quadtree, and selects the table by the decoded cell; this is the per-cell option for every attribute. A2′ also keeps sixteen tables but selects them by a deterministic hash that carries no spatial information. Hybrid uses the A2 tables for scale and color and the A1 table for rotation and opacity, the route fixed by the rule in Section 5.

Why A2′ matters. Sixteen adaptive tables can exploit generic nonstationarity even if their key has no spatial meaning. A2 and A2′ have the same number of tables, initialization, update schedule, block framing, and symbol stream; only the key differs. The contrast A2′ versus A2 therefore isolates spatial specificity, A1 versus Hybrid measures the value of the selected spatial context, and A2 versus Hybrid measures the value of rejecting that context for rotation.

Standalone decoding. Unique integer records are sorted canonically by Morton code and then by the complete record; only afterward are duplicates expanded. Position decoding recreates the same canonical order used for attribute decoding, so duplicate positions keep their full attribute tuple. Each file serializes its profile, group identifiers, symbol counts, rANS payloads, directory, termination, and a whole-file checksum, and a fresh decoder reconstructs the same integer records from the bytes alone (Appendix A.7).

5 Measure and Freeze

Estimating the terms of Eq. 5. We estimate G_h for each attribute on Selection20 alone, by five-fold cross-fitting over images; copies of one Gaussian stay in the same fold. The per-cell predictor is a count model that backs off to a KT-smoothed training marginal,

$$q_h(m \mid c) = \frac{n_{h,c,m} + \tau q_h(m)}{n_{h,c} + \tau}, \qquad \tau \in \{1, 4, 16, 64, 256, 1024\}, \tag{6}$$

following universal estimation (Krichevsky & Trofimov, 1981). With F the fold, we report the cross-fit information $I_F = H(M_h \mid F) - H(M_h \mid C_d, F)$, the regrets $D_{0,F}$ and $D_{1,F}$ of the global and per-cell predictors on held-out folds, and the cross-fit predictive gain $G_{\mathrm{xf}} = I_F + D_{0,F} - D_{1,F}$, all as macro means over images in bits/G.

The decision rule. The operational context is the depth-2 cell ($d = 2$), and $\tau = 16$ is the smallest backoff at which the image-bootstrap lower bound of G_{xf} is positive for every nonconstant attribute. That single operating point is not enough to route an attribute: rotation's margin at $d = 2$ is positive but small (Table 2). We therefore fixed, before any target stream was encoded, a two-part rule. An attribute is coded per-cell only if

1. the 95% image-bootstrap lower bound of G_{xf} is positive at the operating point $d = 2$, $\tau = 16$; and
2. it is also positive at the finer depth $d = 4$ for some τ in the grid of Eq. 6.

Scale passes the second test at $\tau = 1024$ and color at $\tau = 256$; rotation never does, and opacity is constant in this profile. The resulting route, per-cell for scale and color and global for rotation and opacity, is Hybrid. We refer to this rule as the Gate.

The rotation rejection is a marginal call. Its depth-4 lower bound at $\tau = 1024$ is -0.0009 bits/G around a mean of 0.032; all 20 bootstrap seeds keep the bound below zero, but leaving out a single

Table 2: **The audit becomes a per-attribute decision.** I_F, $D_{1,F}$, and G_{xf} are depth-2, $\tau = 16$ Selection20 macro means; the global regret $D_{0,F} = G_{\mathrm{xf}} - I_F + D_{1,F}$ is omitted. The fifth column gives the smallest τ at which the depth-4 lower bound is positive; opacity is an exact-zero control. ✓ selects the per-cell model and × retains the global model.

Head	I_F (bits/G)	$D_{1,F}$ (bits/G)	G_{xf} (bits/G)	$d{=}4$ **first** robust τ	**Grid** Gate	**Hybrid** route
Scale	0.661	0.473	0.255	1024	✓	✓ per-cell
Rotation	0.517	0.502	0.043	none	×	× global
Opacity	0.000	0.000	0.000	constant	—	global
Color	1.283	0.690	0.650	256	✓	✓ per-cell

Selection20 image turns it positive in 6/20 cases (Appendix Table 14). We report the decision as the pre-registered rule produced it. Because rejecting rotation is the only difference between Hybrid and A2, every Hybrid-versus-A2 contrast below measures exactly this marginal decision.

What the rule does not use. The Gate is a count-only, cross-fit estimate; no Hybrid byte was observed before the route was frozen. The prospective forecast in Section 6.2 answers a different question: it uses each new image's quantized symbol counts to estimate the frozen route's gain before entropy coding, without revising the route.

Two further checks are kept separate from the Gate. A same-image calibration on Selection20 shows that the plug-in information estimate predicts the attribute-payload saving of a static per-cell table to within 0.91%; it validates the estimator, not the Gate. And because Hybrid's attribute chunks are exactly the A2 scale and color chunks and the A1 rotation and opacity chunks, its MARKS length is an identity in the A1 and A2 lengths; this is an implementation check, not evidence. Both are reported in Appendix A.2.

6 Experiments

Protocol. All arms encode identical quantized 2D Gaussian sets (the high quantization profile) of the cohorts defined in Section 3. Unless labeled MARKS, rate is eight times complete file bytes divided by Gaussian count, and a positive paired difference means the right-hand arm (Hybrid unless stated) is smaller. Hybrid gains are comparator rate minus Hybrid rate. Wins, percentile image-bootstrap intervals, and byte reconciliation are computed per image; every stream is decoded by a fresh decoder and its integer records are checked against the source by SHA-256 (seeds, image identifiers, and receipts in Appendix A.9).

6.1 Diagnose: total size picks the wrong repair

Morton, Hilbert, and Quadtree use matched learned attribute models trained for 10,000 AdamW steps; Factorized and Exchangeable are training-free empirical controls (Appendix Table 19). Table 3 shows the complete-file result. Quadtree saves 17.54% over the best fixed traversal and wins 100/100, yet its total rate exceeds Exchangeable. Segment accounting resolves the reversal: per image, the quadtree removes about 325.5 kB from positions, while its attribute model adds about 336.7 kB and metadata adds about 17.5 kB (Appendix Table 21). The loss is in the attribute model, not in the hierarchy.

C3 confirms this with a complete, decodable coder: it keeps the quadtree positions and association, transmits a dense per-file attribute table, and charges every byte. On its frozen 20-image subset it reaches 76.75 bits/G and saves 7.42% over Exchangeable and 9.00% over the original Quadtree, winning every image (Appendix Table 20; decoded examples in Appendix Figure 8). Replacing the attribute model reverses the total; the matched arms of Section 4 then ask which part of that gain is spatial context and which is model format.

Table 3: **Hierarchy beats fixed traversal on every image.** Complete-file macro rate on Full100 includes all segments; the paired effect compares Quadtree with the better of Morton and Hilbert per image.

Complete-file macro rates	
Coding model	Complete rate (bits/Gaussian)
Morton	102.24
Hilbert	102.24
Factorized	96.36
Exchangeable	82.60
Quadtree	84.29
Paired Quadtree vs. best ordered	
Mean saving	17.54%
95% CI	[17.14%, 17.92%]
Wins	100/100

Table 4: **The frozen route beats both uniform choices and the non-spatial control on held-out images.** Paired MARKS gains of Hybrid in bits/G from measured streams, with 95% image-bootstrap intervals and win/tie/loss counts. Selection20 is the split the decision was made on and is shown only for contrast.

Evaluation	**Comparator**	**MARKS gain [95% CI]**	**W/T/L**
• Selection 20	Global for all (A1)	0.471 [0.219, 0.766]	16/0/4
• Held-out 80	Global for all (A1)	0.526 [0.348, 0.710]	54/0/26
	Per-cell for all (A2)	0.190 [0.155, 0.221]	71/0/9
	Random-key (A2′)	1.721 [1.516, 1.935]	80/0/0
	Global for all (A1)	0.515 [0.361, 0.676]	70/0/30
Full 100	Per-cell for all (A2)	0.195 [0.165, 0.222]	91/0/9
	Random-key (A2′)	1.715 [1.535, 1.906]	100/0/0

6.2 VALIDATE: HELD-OUT AND PROSPECTIVE BITSTREAMS

Table 4 and Figure 3 report the frozen route on the internal test set. On Heldout80, Hybrid achieves a lower mean attribute rate than A1 (global for every attribute), A2 (per-cell for every attribute), and A2′ (equal-capacity non-spatial control); all three paired intervals exclude zero. The margin over A2′ is the largest, so the gain comes from spatial information rather than from having sixteen tables. The margin over A2 is the smallest, as expected, because it measures only the rotation decision. Full100 repeats the same pattern.

Comparison with classical attribute models. Table 5 extends the predict-before-encoding result to two predeclared alternatives: BayesMix mixes the global and per-cell models online, while B1F uses a compact static model. On Heldout80, the frozen route saves 0.0987 complete-file bits/G over BayesMix and 0.5492 over B1F, with both paired 95% intervals above zero. The BayesMix result shows that selecting attribute models before encoding can outperform mixing them during encoding. For B1F, the lower static attribute cost is outweighed by its transmitted model and framing costs. Together, these comparisons connect the advance modeling decision to complete-file savings on unseen images.

Prospective confirmation on GaussianImage. The stricter test is temporal. We fitted GaussianImage representations with 5k and 10k Gaussians to the official DIV2K validation images, quantized them with the same profile, and recorded, before any file was encoded, the cohort membership, the five arms, and the forecast attribute-bit difference between Hybrid and each adaptive arm, computed from the quantized sets' symbol counts; the record is hash-bound. Pilot5 authorized the run and Confirm20 is disjoint from it. Table 6 compares forecast and realized differences. The forecast gain over A2, 1.414 bits/G, realizes as 1.361 (3.78% error), and the gain over A2′ is within 3.11%. Both

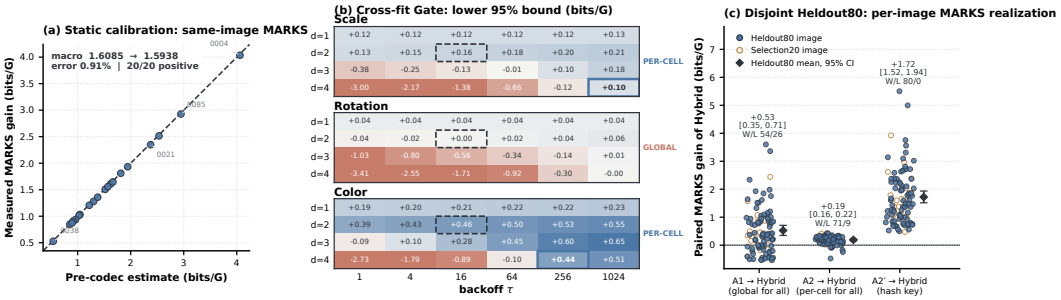

Figure 3: **Measure, decide, then realize on disjoint images.** (a) The same-image calibration on Selection20: plug-in information estimate versus measured static-table payload saving (1.6085 predicted versus 1.5938 measured bits/G; 0.91% error). (b) The Gate then selects per-cell models for scale and color and a global model for rotation. (c) Per-image MARKS gains of the frozen route on Heldout80 (filled) against A1, A2, and A2′; Selection20 images (hollow) are shown only for contrast, and diamonds mark Heldout80 means with paired 95% intervals.

Table 5: **Attribute models selected before encoding outperform classical alternatives on Heldout80.** Gains of Hybrid in bits/G; positive values favor Hybrid. Complete-file gains include probability state, selectors, framing, and syntax; 95% intervals and W/T/L are paired by image. Selection20 and Full100 provide the selection and combined-cohort results.

Cohort	Comparator	MARKS	Complete	95% CI	W/T/L
Selection20	BayesMix → Hybrid	0.1671	**0.1671**	[0.1285, 0.1982]	19/0/1
Heldout80	BayesMix → Hybrid	0.0989	**0.0987**	[0.0639, 0.1313]	63/0/17
Full100	BayesMix → Hybrid	0.1125	**0.1124**	[0.0828, 0.1404]	82/0/18
Heldout80	B1F → Hybrid	-0.1929	**0.5492**	[0.3592, 0.7497]	54/0/26
Full100	B1F → Hybrid	-0.2037	**0.5382**	[0.3735, 0.7131]	69/0/31

point counts keep the same sign (Appendix Table 12), so the result is neither a pilot-selection artifact nor a single-rate coincidence. The forecast also predicted that A1 would be *smaller* than Hybrid on these sparse sets, an A1-minus-Hybrid difference of -5.434 bits/G; the realized difference is -5.369 (1.19% error). The protocol thus forecast the sign and size of every adaptive-arm contrast before encoding, including the one it loses: the frozen route beats A2 and A2′ but is not the smallest coder on GaussianImage sets.

6.3 Secondary controls and robustness

Appendix Table 11 closes three alternative explanations. Compact serialization of the static table saves 4.738 bits/G without changing any probability, so table format is reported separately from model quality. Global online adaptation (A1) has a negative gain of -0.745 bits/G over its matched static control (B1F), so "adaptive" alone is not the contribution. Per-cell A2 beats equal-capacity A2′ by 1.788 bits/G, and the advantage grows as the block size decreases from 1024 to 64, so the gain is spatial rather than sixteen-table capacity.

General-purpose anchor. No arm is a state-of-the-art lossless coder of these records. On the same Full100 canonical integer records, xz $-9e$ reaches 68.94 bits/G and zstd -19 reaches 77.54, against 70.69 for Hybrid, 70.88 for A2, and 71.20 for A1 (Appendix Table 16). The arms are matched instruments that isolate one modeling decision; absolute rate is not the claim.

Post-hoc oracle and unordered controls. A per-file Group-MDL oracle that chooses the A1/A2 mask for each attribute after observing the file is retained only as an upper reference: it selects per-cell color and scale on most Selection20 files and never rotation or opacity (Appendix A). Practical Shuffle Coding independently confirms that exchangeability is a real coding resource (Appendix Table 10); it is orthogonal to the attribute decision.

Table 6: **Forecasts recorded before encoding match the realized bitstreams.** Confirm20, both point counts pooled (40 cells, 200 streams). Forecast and realized columns are comparator-minus-Hybrid MARKS differences in bits/G; the complete-file column charges every byte. Positive favors Hybrid. B1F is a measured static-table control, outside the pre-file forecast.

Hybrid vs.	Forecast MARKS	Realized MARKS	Complete file [95% CI]	W/T/L
A1 (global for all)	-5.434	-5.369	-5.409 [-5.949, -4.818]	0/0/40
A2 (per-cell for all)	1.414	1.361	1.360 [1.214, 1.479]	40/0/0
A2′ (hash key)	2.843	2.754	2.765 [2.479, 3.080]	40/0/0
B1F (compact static)	—	-10.394	13.782 [13.211, 14.349]	40/0/0

Robustness to a different Gaussian producer. Without retraining, we map public GaussianImage++ Kodak24 checkpoints into the five arms of Table 1. Across low and high density and two quantizers, all 480 files decode exactly. Hybrid beats A2 and A2′ on every Kodak24 image but not A1; at high density and mid quantization the gains are 1.300 and 2.794 bits/G (both 24/0/0), while A1 minus Hybrid is -1.037. The frozen route's gains over the all-per-cell and non-spatial controls transfer; superiority to the global model does not. Appendix Table 15 also identifies sparse-regime overhead: at low density the Hybrid file costs 687.4 bits/G, positions are the largest single term, and directory plus attribute framing take 30%, so the sparse-regime rate is set by full-depth quadtree occupancy coding and per-tile syntax rather than by the attribute model. Appendix Figure 5, Figure 6, and Table 18 show decoded examples, rate–quality curves, and paired intervals.

Limitations. The evaluation covers 2D Gaussian image sets across fitting methods, point counts, and quantization profiles; extending the protocol to other primitive families is a natural next step. The Gate selects attribute models, while the count-based forecast estimates their codelength differences. Decisions near the Gate threshold depend on calibration support, as quantified in Appendix Table 14. The present implementation targets exact rate measurement; throughput optimization and integration into end-to-end image codecs remain future directions.

7 Conclusion

Total file size is the natural way to compare Gaussian image coders, but it can point design in the wrong direction: a quadtree hierarchy that codes positions far more compactly looked worse overall only because of the attribute model paired with it. We showed that the decision of which attributes should use spatial context can instead be made before encoding, from the information the context provides and the cost of learning the richer model, and that a decision fixed this way holds on held-out images and in a prospective study where the predicted gain matched the realized one. The broader lesson is that a coder design choice is most credible when its expected effect is predicted, fixed, and only then tested on bitstreams that had no part in the choice.

AI use statement

Generative AI tools assisted with refining ideas and experimental plans, coding, and language polishing; the authors take full responsibility for all content.

References

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

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, 2024.

Yihang Chen, Qianyi Wu, Mengyao Li, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. Fast feedforward 3D Gaussian Splatting compression. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. HAC++: Towards 100X compression of 3D Gaussian Splatting, 2025b. arXiv:2501.12255.

Runze Cheng, Yicheng Zhan, Josef Spjut, and Kaan Akşit. Clustered codebook quantization for 2D Gaussian-based image compression. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Posters*. ACM, 2026. doi: 10.1145/3799825.3818700.

Xuan Deng, Xiandong Meng, Hengyu Man, Qiang Zhu, Tiange Zhang, Debin Zhao, and Xiaopeng Fan. LG-HCC: Local geometry-aware hierarchical context compression for 3D Gaussian Splatting, 2026. arXiv:2603.28431v3.

Yifan Gao, Xinfeng Zhang, and Gai Zhang. 2D Gaussian Splatting with pre-trained dictionaries for image compression. *IEEE Signal Processing Letters*, 32:3964–3968, 2025. doi: 10.1109/LSP.2025.3619027.

Qiang Hu, Zhenlong Wu, Lei Huang, Zihan Zheng, Xiaoyun Zhang, and Wenjun Zhang. GenSplatCodec: Feed-forward Gaussian Splatting compression via one-step diffusion, 2026. arXiv:2607.24403.

Lila Huang, Shenlong Wang, Kelvin Wong, Jerry Liu, and Raquel Urtasun. OctSqueeze: Octree-structured entropy model for LiDAR compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1313–1323, 2020.

Yuning Huang, Jiahao Pang, Fengqing Zhu, and Dong Tian. EntropyGS: An efficient entropy coding on 3D Gaussian Splatting, 2025. arXiv:2508.10227.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. doi: 10.1145/3592433.

Dong-Ha Kim, Byung-Yoon Choi, Kwan-Jung Oh, Gwangsoon Lee, and Jae-Gon Kim. Hybrid compression method for trained 3D Gaussian Splatting models based on VQ and HEVC. *Sensors*, 26(13):4125, 2026. doi: 10.3390/s26134125.

Raphail E. Krichevsky and Victor K. Trofimov. The performance of universal encoding. *IEEE Transactions on Information Theory*, 27(2):199–207, 1981. doi: 10.1109/TIT.1981.1056331.

Julius Kunze, Daniel Severo, Jan-Willem van de Meent, and James Townsend. Practical shuffle coding. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2672.

Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3D Gaussian representation for radiance field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21719–21728, 2024. doi: 10.1109/CVPR52733.2024.02052.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3744–3753. PMLR, 2019.

Soonbin Lee, Fangwen Shu, Yago Sanchez, Thomas Schierl, and Cornelius Hellge. Compression of 3D Gaussian Splatting with optimized feature planes and standard video codecs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 25496–25505, 2025.

Tiantian Li, Xinjie Zhang, Xingtong Ge, Tongda Xu, Dailan He, Jun Zhang, and Yan Wang. GaussianImage++: Boosted image representation and compression with 2D Gaussian Splatting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 6442–6449, 2026. doi: 10.1609/aaai.v40i8.37572.

Lei Liu, Zhenghao Chen, Wei Jiang, Wei Wang, and Dong Xu. HEMGS: A hybrid entropy model for 3D Gaussian Splatting data compression, 2024. arXiv:2411.18473.

Zhening Liu, Rui Song, Yushi Huang, Yingdong Hu, Xinjie Zhang, Jiawei Shao, Zehong Lin, and Jun Zhang. Feed-forward 3D Gaussian Splatting compression with long-context modeling, 2025. arXiv:2512.00877.

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

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Zizheng Que, Guo Lu, and Dong Xu. VoxelContext-Net: An octree-based framework for point cloud compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6042–6051, 2021.

Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978. doi: 10.1016/0005-1098(78)90005-5.

Shashank N. Sridhara, Birendra Kathariya, Fangjun Pu, Peng Yin, Eduardo Pavez, and Antonio Ortega. Region-adaptive learned hierarchical encoding for 3D Gaussian Splatting data, 2025. arXiv:2510.22812.

Yufei Wang, Zhihao Li, Lanqing Guo, Wenhan Yang, Alex C. Kot, and Bihan Wen. ContextGS: Compact 3D Gaussian Splatting with anchor level context model. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1631.

Frans M. J. Willems, Yuri M. Shtarkov, and Tjalling J. Tjalkens. The context-tree weighting method: Basic properties. *IEEE Transactions on Information Theory*, 41(3):653–664, 1995. doi: 10.1109/18.382012.

Qi Yang, Kaifa Yang, Yuke Xing, Yiling Xu, and Zhu Li. A benchmark for Gaussian Splatting compression and quality assessment study. In *Proceedings of the 6th ACM International Conference on Multimedia in Asia*. ACM, 2024. doi: 10.1145/3696409.3700172.

Qi Yang, Le Yang, Geert Van Der Auwera, and Zhu Li. HybridGS: High-efficiency Gaussian Splatting data compression using dual-channel sparse representation and point cloud encoder. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 71487–71507. PMLR, 2025.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep sets. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Zhaojie Zeng, Yuesong Wang, Yawei Luo, and Tao Guan. AIR: Amortized image reconstruction framework for self-supervised feed-forward 2D Gaussian Splatting, 2026. arXiv:2605.20820.

Junteng Zhang, Tong Chen, Yuxin Zhao, Yibo Shi, Jing Wang, and Zhan Ma. SpeedyGS: Content-aware 3D Gaussian Splatting compression via two-stage optimization, 2026. arXiv:2607.12656.

Pingping Zhang, Xiangrui Liu, Meng Wang, Shiqi Wang, and Sam Kwong. 2D Gaussian Splatting for image compression. *APSIPA Transactions on Signal and Information Processing*, 13(6):e501, 2024a. doi: 10.1561/116.20240025.

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*. Springer, 2024b.

A Audit-Ready Codec and Evidence Detail

The appendix makes the diagnosis-to-decision-to-validation chain auditable. It lists the role of every cohort, reports the checks kept out of the main text, specifies the bytes a fresh decoder consumes, and exposes every C3 case.

A.1 Cohorts and evidence roles

Table 7: **Every cohort has one evidentiary job.** Symbols identify measured streams (●) and count-only diagnostics (△); status labels delimit selection, held-out testing, prospective confirmation, and robustness.

Stage	Cohort	Status	Role	Claim boundary
Diagnosis	H1 Full100	● measured	complete-file segments	hidden cancellation; not a routing estimate
Decision	Selection20	△ selection-only	information–regret Gate	freezes heads; no held-out performance claim
Internal	Mechanism5 Heldout80	● measured ● held-out	implementation check disjoint physical test	exact composition, state, and decode only primary untouched validation
Transfer	Confirm20 Kodak24 grid	● prospective ● robustness-only	pre-file external test producer/profile shift	temporal confirmation after Pilot5 robustness, not selection or native RD

A.2 Static calibration and byte composition

Static same-image calibration. This check validates the count-only estimator, not the Gate. It uses the depth-2 plug-in information estimate on each Selection20 image to forecast the MARKS payload saving of a static per-cell table (global B1F minus B1F with one table per cell) subsequently encoded on the same image. Across the frozen Selection20 queue the estimate predicts 1.6085 bits/G and the measured payload saves 1.5938; the -0.0147 bits/G residual is 0.91%, with 20/20 positive cases. Estimator, depth, membership, and macro aggregation are aligned. This is empirical calibration, not an identity, and it does not validate the adaptive codec.

Exact head-byte composition. Because Hybrid's attribute chunks are the A2 scale and color chunks and the A1 rotation and opacity chunks, once the A1 and A2 streams exist the Hybrid MARKS length is

$$B_{\text{marks}}^{\text{Hybrid}} = B_{\text{scale}}^{\text{A2}} + B_{\text{color}}^{\text{A2}} + B_{\text{rotation}}^{\text{A1}} + B_{\text{opacity}}^{\text{A1}}. \tag{7}$$

This composition predicts 100/100 Full100 MARKS lengths exactly, with a maximum residual of 0 bytes; the only complete-file discrepancy is bounded directory feedback of 28 bytes when new offsets change varint widths. It verifies decode-consistent framing, not statistical generalization (Table 8, Figure 4).

Table 8: **Static calibration and byte composition have distinct targets.** The first row calibrates the same-image information estimate against 20 static-table payload reductions; the second row checks the precommitted byte composition across independently encoded groups.

Prediction	Cohort	Frozen (bits/G)	Measured (bits/G)	Residual (bits/G)	Agreement
△ plug-in → MARKS payload	Selection20	1.6085	1.5938	-0.0147	0.91%
◇ head-byte composition → Hybrid	Mechanism5	exact bytes	exact bytes	0 B	5/5

A.3 Matched baselines

Precommitted classical controls. BayesMix is a decoded online Bayesian mixture of the global and per-cell experts; B1F is a compact static coder with fully charged state and framing. Both comparator definitions were fixed before their Heldout80 streams were encoded. Table 5 separates Selection20, Heldout80, and Full100 and reports MARKS and complete-file contrasts.

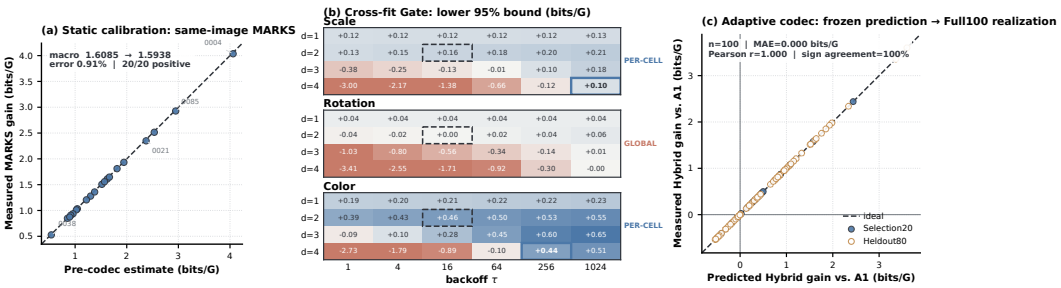

Figure 4: **Implementation identity, not validation.** Panels (a) and (b) repeat the main-text calibration and Gate; panel (c) shows the exact head-byte composition realized on Full100 (100/100 exact MARKS lengths).

On Heldout80, Hybrid saves 0.0987 complete-file bits/G over BayesMix ([0.0639, 0.1313], 63/0/17) and 0.5492 over B1F ([0.3592, 0.7497], 54/0/26). The corresponding comparator streams pass the authenticated checks in Table 9.

Table 9: **Supplemental comparator streams close physically and cryptographically.** Every count is derived from the final create-once receipt and its sealed ledgers.

Authenticated object	Coverage	Verified property	Gate
New Heldout80 comparator streams	160/160	exact integer decode	✓
New Heldout80 comparator streams	160/160	physical-byte reconciliation	✓
BayesMix adaptive streams	80/80	encoder/decoder state equality	✓
Selection20 comparator streams	40/40	source-receipt authentication	✓
Hybrid source streams	100/100	case-receipt binding	✓
Heldout80 partition seals	4/4	ledger/receipt hash equality	✓

Selection20 post-hoc oracle analysis. Group-MDL pays its selector and profile bytes but chooses an A1/A2 mask for each semantic head after observing the target file. It is therefore an intentionally advantaged oracle analysis, not held-out evidence. On Selection20, the paired oracle-minus-Hybrid complete-file contrast is -0.0545 bits/G ([-0.1008, -0.0193], 0/20 wins). The oracle selects per-cell color on 80% and scale on 60% of files, but never rotation or opacity.

Operational unordered control. U0 is the historical count-histogram proxy, U1 is a matched ordered stream over complete Gaussian records, and U2 applies Practical Shuffle Coding to the same record orbit. Table 10 charges every physical byte and keeps the operational comparison separate from H1's exchangeable model.

Practical Shuffle Coding over complete Gaussian records saves 15.9453 bits/G against its matched ordered-record stream (15.7838, 16.0942 paired 95% bounds) and wins 20/20. The virtual ANS discount agrees with the orbit discount to within 4.191e-09 bits; after tail rounding, physical payload reconciliation is exact. This establishes unordered symmetry as a real coding resource while leaving head routing to the orthogonal per-cell versus global experiment.

A.4 Additional controls

Every comparator on the prospective run. Table 12 splits the Confirm20 gains of the frozen route over A2 and A2′ by point count and reports the authorizing Pilot5; Table 13 adds A1 and B1F for both cohorts. Global A1 is smaller than Hybrid on every Pilot5 and Confirm20 cell, and the pre-file forecast predicted that sign and magnitude. The forecast column is the integer-frequency codelength difference recorded before any target file existed; B1F is a measured static-table control outside that pre-file forecast.

Gate sensitivity. Table 14 recomputes the macro lower confidence bound of the cross-fit predictive gain from the authenticated per-image audit under 20 bootstrap seeds and under leave-one-image-out

Table 10: Matched Selection20 unordered controls. U0 is the historical count-histogram proxy, U1 is the matched ordered record control, and U2 is genuine Practical Shuffle Coding over complete Gaussian records. Positive values favor the right-hand arm.

Comparison	Mean	Median	Paired 95% CI	Wins
U1 → U2, ANS payload	15.9453	16.0243	[15.7838, 16.0942]	20/20
U1 → U2, complete file	15.9453	16.0243	[15.7838, 16.0942]	20/20
U0 → U2, complete file	14.2711	14.4699	[13.9169, 14.5866]	20/20

Table 11: **Format, adaptation, spatial structure, and exactness are separately controlled.** Positive gain favors the right-hand arm. Block rows report A2 over A2′; all entries are measured Mechanism5 streams.

Factor	Matched contrast	Gain (bits/G)	n	Conclusion
Serialization	B0 → B1	4.738	5	compact format
Adaptation	B1F → A1	-0.745	5	global KT loses
Spatial key	A2′ → A2	1.788	5	space, not capacity
Block-size sensitivity: A2 gain over A2′				
Block 64	frozen-table update	1.855	5	strongest
Block 256	primary setting	1.788	5	preregistered
Block 1024	frozen-table update	1.672	5	still positive
Exactness	predicted MARKS residual	0 B	5	5/5
Framing	max directory feedback	12 B	5	bounded varint effect

Table 12: **Gains over A2 and A2′ hold at each point count.** Entries are complete-file bits/G saved by Hybrid; every cell comprises five standalone arms with exact fresh decode. The Pilot authorizes but is not pooled into the disjoint Confirm20 estimate.

Frozen cohort	Cells / streams	A2 − Hybrid	A2′ − Hybrid	W/T/L
Pilot5, both points	10/50	1.421	2.390	10/0/0
Confirm20, 5k	20/100	1.324	2.648	20/0/0
Confirm20, 10k	20/100	1.396	2.881	20/0/0
Confirm20, both	**40/200**	**1.360**	**2.765**	**40/0/0**

Table 13: **The forecast also predicts the contrast the frozen route loses.** Comparator−Hybrid differences in bits/G; positive favors Hybrid. Intervals are image-clustered percentile bootstraps over both point-count regimes.

Cohort	Comparator → Hybrid	Forecast MARKS	Realized MARKS	Complete file [95% CI]	W/T/L
Pilot5	A1 (global for all)	-6.105	-6.027	-6.062 [-6.529, -5.605]	0/0/9
Pilot5	A2 (per-cell for all)	1.477	1.416	1.418 [1.379, 1.454]	9/0/0
Pilot5	A2′ (hash key)	2.479	2.371	2.375 [2.136, 2.677]	9/0/0
Pilot5	B1F (compact static)	—	-11.159	13.419 [12.210, 15.591]	9/0/0
Confirm20	A1 (global for all)	-5.434	-5.369	-5.409 [-5.949, -4.818]	0/0/40
Confirm20	A2 (per-cell for all)	1.414	1.361	1.360 [1.214, 1.479]	40/0/0
Confirm20	A2′ (hash key)	2.843	2.754	2.765 [2.479, 3.080]	40/0/0
Confirm20	B1F (compact static)	—	-10.394	13.782 [13.211, 14.349]	40/0/0

deletion. Scale and color are robust at every cell shown. Rotation is not: its depth-4 bound sits within one thousandth of a bit of zero, and its depth-2 operating-point bound is positive for 18/20 seeds.

Kodak segment accounting. Table 15 resolves the Hybrid Kodak24 files into container segments for both densities and both quantizers. Geometry dominates at low density because full-depth

Table 14: **The rotation rejection rests on a marginal bound.** Lower 95% percentile bounds of the Selection20 macro predictive gain; "positive" counts seeds or leave-one-out subsets whose bound exceeds zero.

Head	Grid cell	Mean (bits/G)	Lower 95% seed 20260813	Lower 95% over seeds	Seeds positive	Leave-one-out positive
Scale	$d = 2, \tau = 16$	0.255	0.1637	[0.1604, 0.1656]	20/20	20/20
Scale	$d = 4, \tau = 1024$	0.164	0.1040	[0.1010, 0.1043]	20/20	20/20
Rotation	$d = 2, \tau = 16$	0.043	0.0019	[-0.0001, 0.0024]	18/20	11/20
Rotation	$d = 4, \tau = 256$	-0.233	-0.3010	[-0.3022, -0.2987]	0/20	0/20
Rotation	$d = 4, \tau = 1024$	0.032	-0.0009	[-0.0021, -0.0000]	0/20	6/20
Color	$d = 2, \tau = 16$	0.650	0.4632	[0.4572, 0.4651]	20/20	20/20
Color	$d = 4, \tau = 256$	0.701	0.4407	[0.4289, 0.4457]	20/20	20/20

Quadtree occupancy coding pays for every level of a sparse tree, and directory plus framing syntax scale with tile count rather than with Gaussians.

Table 15: **Sparse sets pay for tree depth and syntax, not for marks.** Mean Hybrid segment cost in bits/G across the 24 Kodak images of each regime; "non-payload" excludes geometry and mark rANS payload.

Hybrid segment (bits/G)	low/high	high/high	low/mid	high/mid
Mean Gaussians	986	4761	986	4761
Geometry	394.1	121.6	317.3	96.7
Mark rANS payload	85.8	58.2	76.5	49.7
Mark framing	76.7	16.5	76.7	16.5
Directory	127.7	27.3	127.4	27.3
Root	1.8	0.5	1.8	0.5
Provider/state	0.5	0.1	0.5	0.1
Header + termination	0.9	0.2	0.9	0.2
Complete file	687.4	224.4	601.0	190.8
Non-payload share	30%	20%	34%	23%

General-purpose compressor anchor. Table 16 compresses the same Full100 canonical integer records (Morton order, then full record; nine 16-bit fields) with zstd and xz, taking the best of row-major, column-major, and position-delta layouts per image, and lists the measured complete-file macro rate of every arm. B1F is available on the Selection20 subset only.

Table 16: **Absolute rate is not the claim.** xz on canonically ordered records is smaller than every arm; the arms differ from one another by the single modeling decision the protocol isolates.

Coder on the same canonical integer records	Macro bits/G	Role
Fixed-width 16-bit fields	144.0	raw serialization
zstd -19 (best of row, column, delta layouts)	77.54	general-purpose anchor
xz $-9e$ (best of row, column, delta layouts)	68.94	general-purpose anchor
B1F (compact static)	71.29	measured standalone streams
A1 (global for all)	71.20	measured standalone streams
A2′ (hash key)	72.40	measured standalone streams
A2 (per-cell for all)	70.88	measured standalone streams
Hybrid (frozen route)	70.69	measured standalone streams

A.5 GAUSSIANIMAGE++ KODAK24 TRANSFER

Table 17 reports complete physical files after adapting the independently produced covariance checkpoints into the common GPC-high integer alphabet. These rates are not the upstream method's native bpp and are not used for a cross-paper RD ranking. Their purpose is stricter: every one of the 120 files must carry its own probability state, reconcile all seven byte segments, and recover the source integer hash in an isolated decoder.



Figure 5: **Real cross-generator streams preserve the transferred Gaussian scene.** We visually inspected all 24 source/decode pairs before selecting these same-aspect examples, which span hard texture, faces, and saturated color. The lower row renders the canonical integer set recovered from the physical Hybrid stream. Selection affects presentation only; all 120 streams pass the same hash and state checks.

Table 17: **The five-arm codec closes on an independent Gaussian producer.** Image-macro complete bpp and bits/G include every container segment on Kodak24. "Exact" counts fresh-process integer and adaptive-state agreement; no native GaussianImage++ rate estimate is mixed into these columns.

Codec arm	Complete bpp	bits/G	Exact
A1 (global for all)	1.7232	687.25	✓ 24/24
A2 (per-cell for all)	1.7261	688.40	✓ 24/24
A2′ (hash KT)	1.7281	689.22	✓ 24/24
Hybrid	1.7238	687.49	✓ 24/24
B1F (static)	1.8156	724.11	✓ 24/24

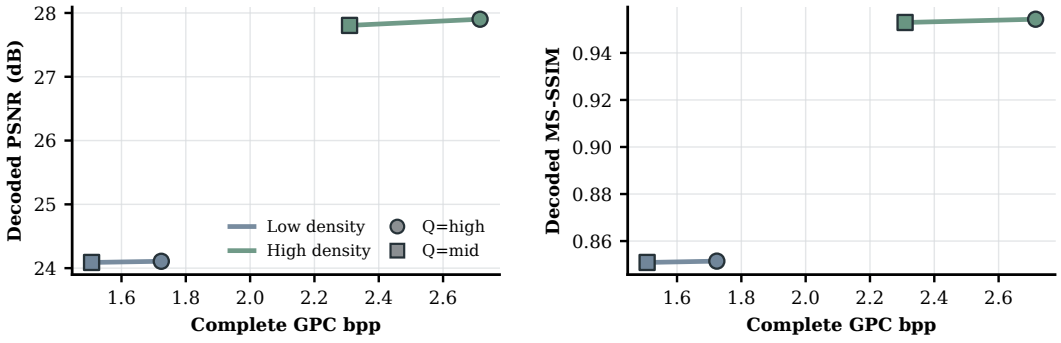

Figure 6: **Complete physical rate–quality across density and quantization.** Each point is the image-macro Hybrid rate and quality from 24 freshly decoded Kodak streams; circles/squares denote high/mid GPC quantization. Lines connect quantizers at fixed GaussianImage++ density. Rates are complete GPC bytes per source pixel, never upstream native payload.

A.6 H1 PROTOCOL AND COMPLETE C3 OUTCOME

Table 19 lists the matched implementations behind the diagnosis of Section 6.1, Table 20 the complete C3 outcome on its frozen 20-image cohort, and Figure 9 the per-image form of the hidden cancellation. Figure 8 shows decoded examples.

Table 18: **Gains over A2 and A2′ survive two representation densities and two quantizers.** The upper panel reports freshly decoded Hybrid quality and complete-file rate. The lower panel gives paired bits/G $[95\%\ \mathrm{CI}]$; W/T/L follows each interval and positive means Hybrid is smaller. The negative A1 column is retained as the sparse-regime boundary.

Density	Q	$\overline{N}$	Complete GPC bpp	PSNR	MS-SSIM
Low	high	986	1.724	24.11	0.8515
High	high	4760	2.715	27.90	0.9543
Low	mid	986	1.507	24.09	0.8509
High	mid	4760	2.309	27.80	0.9529

Density	Q	A1 − Hybrid	A2 − Hybrid	A2′ − Hybrid
Low	high	-0.241 [-0.453, -0.035]; 9/0/15	0.910 [0.847, 0.971]; 24/0/0	1.731 [1.599, 1.858]; 24/0/0
High	high	-1.983 [-2.190, -1.753]; 0/0/24	1.666 [1.603, 1.724]; 24/0/0	3.260 [3.045, 3.486]; 24/0/0
Low	mid	-0.916 [-1.120, -0.717]; 0/0/24	0.950 [0.867, 1.035]; 24/0/0	2.121 [1.936, 2.307]; 24/0/0
High	mid	-1.037 [-1.230, -0.827]; 1/0/23	1.300 [1.240, 1.357]; 24/0/0	2.794 [2.585, 3.020]; 24/0/0

Table 19: **Matched H1 implementation protocol.** Parameter counts and context controls are generated from authenticated bundles; a zero step count denotes deterministic fitting rather than missing optimization.

Codec	Architecture / probability context	Parameters	Optimizer / fit	Steps
Morton	Morton; 16-record decoded prefix; 64 buckets; two width-28 layers	953,688	AdamW, $\eta = 10^{-4}$	10,000
Hilbert	Hilbert; 16-record decoded prefix; 64 buckets; two width-28 layers	953,688	AdamW, $\eta = 10^{-4}$	10,000
Factorized	one smoothed global histogram per scalar field	89,344	Histogram MLE	0
Exchangeable	per-file empirical scalar-field histograms, transmitted in-stream	0	Training-free	0
Quadtree	decoded-tree count context + radius-2, width-32 geometry-conditioned mark MLP	440,357	AdamW, $\eta = 10^{-4}$	10,000

Table 20: **C3 closes the diagnosed gap as a complete standalone stream.** Provider bytes, paired uncertainty, exact decode, and byte reconciliation are charged and reported on the frozen diagnostic cohort.

Outcome	Result
Rates — Frozen diagnostic ($n = 20$)	
C3 complete rate (macro)	76.75 bits/Gaussian
C3 complete rate (weighted)	76.04 bits/Gaussian
Paired savings	
Mean saving vs. Exchangeable (95% CI)	7.42% ([6.22%, 8.49%])
Wins vs. Exchangeable	20/20
Mean saving vs. original Quadtree (95% CI)	9.00% ([7.76%, 10.38%])
Wins vs. original Quadtree	20/20
Provider cost	
Dense provider charged per stream	106,582 bytes
Verification	
Exact integer decode	20/20
Exact byte reconciliation	20/20

Figure 7 connects the original diagnostic target to the charged operational endpoint while keeping each source of overhead visible.

The post-hoc H1 substitution swaps mark-segment bytes between measured codecs; without provider or decoder it is never counted as C3. The bridge first isolates the H1-to-C3 cohort shift, then adds provider, framing, rANS-payload, and directory deltas on common membership. Per-image equality

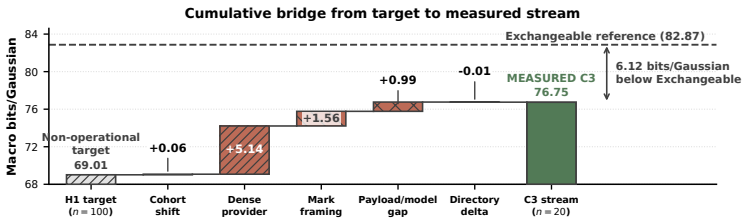

Figure 7: **From diagnostic target to charged C3 stream.** The bridge separates cohort shift, provider serialization, mark framing, payload/model gap, and directory feedback before closing to the measured complete-file rate.

checks fix geometry, root, header, and termination, so aggregate cancellation cannot pass. The terms close exactly to C3 macro rate: mark syntax remains engineering headroom, while hierarchical geometry is already retained.

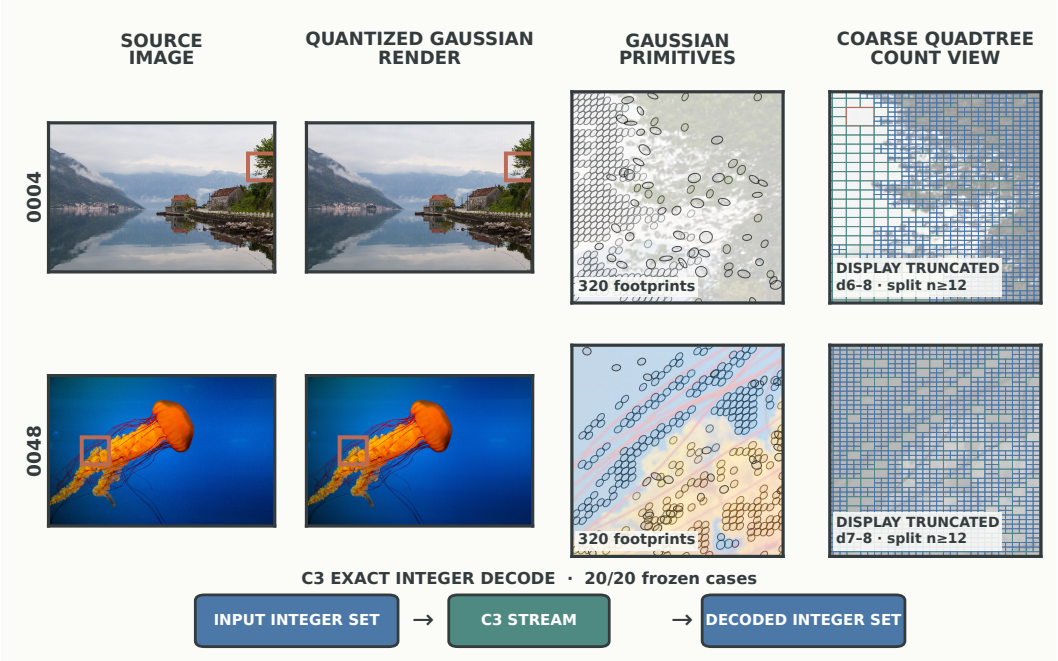

Figure 8: **The codec preserves a real Gaussian representation.** Frozen DIV2K examples show source image, quantized render, actual primitives, and a display-truncated quadtree count view. Exact integer decode concerns the quantized Gaussian set rather than source-pixel losslessness.

A.7 Normative C3 stream syntax

Complete container. A C3 file is a version-1 GPC1 container. Its physical order is header, hyperprior, root, directory, geometry chunks, mark chunks, and a four-byte whole-file CRC-32 termination. The header binds image dimensions, model and probability-profile identifiers, quantizer profile, macro-tile depth, Morton traversal, 16-bit rANS precision, segment/chunk counts, and the three fixed-segment lengths. Unsigned integer syntax is canonical ULEB128; strings are length-prefixed UTF-8. Directory entries are sorted by $(\text{kind}, \text{tile}, \text{mark group})$ and encode that key together with an absolute offset, length, and payload CRC-32. The parser rejects a non-minimal integer, unknown enum, duplicate key, gap, overlap, out-of-order span, chunk checksum failure, trailing data, or a directory/file-length mismatch before exposing any payload to the codec.

Adaptive provider. The hyperprior is the sole file-specific provider. Its canonical byte string is

$$\underbrace{\texttt{GMDP}}_{\text{magic}} \parallel \underbrace{\texttt{u8}(3)\|\texttt{u8}(1)\|\texttt{u32le}(|D|)\|D}_{C} \parallel \mathrm{SHA256}(C).$$

Type 1 denotes a global-table provider. The body D begins with a little-endian u16 head count. Each head then appears in increasing identifier order as u16(head), u32le(alphabet), and one dense little-endian float64 probability for every alphabet symbol. The seven rows are heads $0, 1, 3, 6, 9, 10, 11$: two 4096-symbol scale rows and five 1024-symbol rotation, opacity, and color rows. Zeros have a unique positive-zero representation. Every stream therefore charges the complete 106,582-byte payload. Parsing requires exact body termination, finite nonnegative normalized rows, unique valid heads, the declared SHA-256, and byte-identical canonical reserialization.

The encoder computes each row from the current file's complete scheduled symbols: $p_{h,a} = n_{h,a}/\sum_b n_{h,b}$. For rANS, only positive-probability symbols are valid. Deterministic largest-remainder normalization first reserves one count for every valid symbol, apportions the remaining $2^{16} - |S_h|$ counts by normalized mass, and resolves equal remainders by increasing symbol identifier. Hence encoder and decoder reconstruct the same integer CDF and no probability mass is silently reserved for unseen symbols.

Chunk framing and parse order. Every mark chunk begins with GM2, followed by canonical ULEB128 tile identifier, mark-group identifier, symbol count, and sub-block count. The group schedule is scale, rotation, opacity, color, with $2N, N, N, 3N$ symbols. A sub-block stores its row count and rANS byte length as ULEB128, then little-endian 32-bit rANS words. A fixed 262,144-cell table budget determines the unique sub-block partition. The decoder requires the declared rows to cover the group exactly and the rANS state to terminate exactly at each sub-block boundary.

After complete container validation, a stream-only decoder parses and canonicalizes the provider *before* constructing mark CDFs. It then decodes root and geometry, recreates the Quadtree leaf/multiplicity order, checks the exact tile/group topology and symbol counts, and finally decodes marks in the frozen group schedule. The encoder's unique records were sorted by Morton code and full integer record before multiplicity expansion; geometry and marks share that expanded index. No mark-only permutation is permitted, so duplicate centers retain their scale, rotation, opacity, and color association.

A.8 Complete byte and hash reconciliation

For each stream the authenticated all-group ledger enforces

$$\begin{aligned} |B| = {} & B_{\text{header}} + B_{\text{provider/hyperprior}} + B_{\text{root}} + B_{\text{directory}} + B_{\text{termination}} \\ & + \sum_t B_{\text{geometry},t} + \sum_{t,g} B_{\text{marks},t,g}. \end{aligned} \tag{8}$$

The mark term further reconciles to rANS payload plus GM2, identifiers, counts, lengths, and sub-block framing. Table 21 reports means generated directly from these ledgers. H1's empty hyperprior slot and C3's fully charged provider occupy the same explicitly named column; the table does not disguise provider bytes as shared state.

The receipt chain is content-addressed at three levels. Each child content receipt records the exact SHA-256 of stream.gpc and ledger.jsonl, the frozen manifest, Quadtree bundle, provider payload, construction identifier, and traversal. Its execution receipt commits to that content-receipt digest and the execution record. Each immutable batch receipt then lists its image IDs, bitstream hashes, child-receipt hashes, decoded integer hashes, provider-core hashes, and batch-ledger hash. The anonymous artifact publishes the full digests in the canonical evidence summary and provenance manifest rather than machine-specific locators.

Authentication recomputes every file digest before reading a measurement. It then requires the source and decoded canonical integer-record SHA-256 to be identical and Equation 8 to equal the physical stream length. Both gates hold for 20/20 and 20/20 frozen C3 cases. Thus exactness is not inferred from rate or rendering; it is independently hash- and byte-checked.

Table 21: **Complete accounting exposes both the mechanism and the cost of repairing it.** The upper panel separates H1 diagnostic means (Exchangeable E and Quadtree Q, $n = 100$) from the operational C3 mean ($n = 20$); every category is part of the physical file. The middle panel resolves C3 marks into payload and framing. The lower panel is an evidence-generated bridge whose diagnostic target and operational endpoint are explicitly scope-labeled.

A. Complete-file segments

Segment (mean kB/image)	H1 E ($n = 100$)	H1 Q ($n = 100$)	C3 ($n = 20$)
Header	0.10	0.10	0.10
Provider/hyperprior	0.00	0.00	106.58
Root	0.02	0.39	0.38
Directory	0.08	17.23	16.97
Geometry	703.98	378.51	376.68
Marks	1130.98	1467.71	1176.41
Termination	0.00	0.00	0.00

B. C3 mark decomposition ($n = 20$)

C3 mark group	Mean symbols	rANS kB	Framing kB	Total kB
Scale	352,907.3	392.78	13.45	406.23
Rotation	176,453.6	215.03	5.20	220.22
Opacity	176,453.6	3.26	4.46	7.72
Color	529,360.9	531.53	10.71	542.23

C. Diagnostic-to-operational bridge

Evidence-bound bridge term	bits/Gaussian
Non-operational H1 substitution target ($n = 100$)	69.0094
C3-cohort shift	+0.0644
Dense provider	+5.1374
Mark framing	+1.5606
rANS payload/model gap	+0.9872
Directory delta	-0.0112
Measured C3 stream ($n = 20$)	76.7477

Rounding: Bridge terms are shown to four decimals; signed increments reconcile the two displayed endpoints up to rounding.

A.9 Selection and paired uncertainty

All bootstrap intervals, win counts, and hash checks in the paper are computed by image identifier under the public seed 20260813. Mechanism5 uses the predeclared images 0004, 0020, 0037, 0066, and 0086.

The C3 membership was frozen before any C3 outcome was observed. The deterministic balanced_nested_rank_v1 rule uses only completed H1 fields. It stable-sorts 100 paired high-profile objects by Gaussian count and image ID, splits the rank into low/high count halves, then stable-sorts each half by the already measured Quadtree-minus-Exchangeable mark penalty and image ID. Each half is divided into low/high conditional-penalty cells of 25; ranks $0, 6, 12, 18, 24$ are selected from every cell. This yields four balanced cells and the 20 sorted IDs in Table 22. The rule spans count and mechanism-severity regimes without conditioning on the repair's rate.

All comparisons are paired by image ID. For image i, saving is $s_i = 100(R_{b,i} - R_{\mathrm{C3},i})/R_{b,i}$. The reported effect is the arithmetic mean of s_i, and a win requires $s_i > 0$ exactly. The percentile interval draws 10,000 samples of 20 paired image indices with replacement using seed 20260813, recomputes the mean saving for each draw, and returns the 0.025 and 0.975 NumPy quantiles under its linear interpolation rule. Resampling never breaks a codec pair.

Macro rate and Gaussian-weighted rate answer different summaries:

$$R_{\text{macro}} = \frac{1}{20}\sum_i \frac{8B_i}{N_i}, \qquad R_{\text{weighted}} = \frac{8\sum_i B_i}{\sum_i N_i}.$$

The paper reports both, but its paired confidence intervals concern the mean per-image saving, not the weighted aggregate. The 20-case design is a frozen mechanism and implementation diagnostic; it does not estimate universal performance over Gaussian representations or establish state of the art.

A.10 PER-IMAGE OPERATIONAL EVIDENCE

Table 22 exposes the complete paired outcome rather than only its average. Each of the exact frozen IDs appears once, and each row is generated from the authenticated stream ledger. Positive gains mean the C3 physical file is smaller than the paired baseline physical file.

Table 22: **The operational reversal holds case by case.** Complete C3 rate and paired savings against Exchangeable (E) and original Quadtree (Q) for the frozen $n = 20$ diagnostic cohort. These are standalone stream measurements, not ideal lengths or segment substitutions.

Image	Complete bytes	bits/Gaussian	Saving vs. E	Saving vs. Q
0004	1,201,669	80.30	2.72%	6.73%
0009	1,977,095	76.61	9.22%	8.29%
0014	2,049,459	74.15	9.77%	10.66%
0017	1,698,406	76.33	7.75%	9.10%
0020	1,159,140	79.65	3.80%	7.70%
0021	990,931	86.17	0.82%	3.04%
0031	1,770,333	77.62	8.43%	7.75%
0033	1,502,391	78.81	6.99%	7.01%
0037	1,860,938	71.99	9.49%	13.04%
0038	1,787,997	74.14	8.93%	10.93%
0048	1,897,062	67.17	10.15%	17.85%
0054	2,093,830	75.74	9.60%	9.23%
0066	1,853,749	76.96	8.70%	8.28%
0070	1,668,454	79.18	7.34%	6.85%
0073	1,427,289	79.94	5.21%	6.85%
0085	1,254,797	77.57	4.11%	8.97%
0086	1,750,586	77.85	8.23%	7.55%
0087	1,999,841	71.29	10.05%	13.15%
0093	2,017,820	75.86	9.52%	9.02%
0096	1,580,820	77.63	7.48%	8.03%

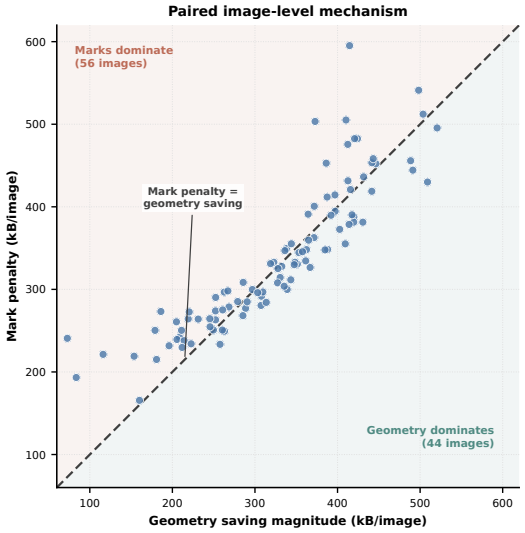

Figure 9: **The hidden cancellation repeats image by image.** Quadtree geometry-saving magnitude against its paired mark penalty for each frozen H1 image. Every point is one image, with no cross-image connection; all 100 points save geometry and penalize marks. The identity line separates images whose mark penalty exceeds the geometry saving.