IIHS: INNOVATION-INSULATED HEAT SPLATS FOR SINGLE-IMAGE REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural fields have become a dominant representation-learning paradigm: coordinate networks store an image in the weights of a continuous function, and Gaussian splatting uses explicit primitives that are fast to render and compact to store. Structural priors steer where these primitives are placed, but each primitive's footprint remains blind to image structure: an anisotropic Gaussian elongated along an edge still bleeds across it. We present Innovation-Insulated Heat Splats (IIHS), a single-image representation in which each source starts as a small Gaussian and spreads as heat through a medium that insulates at the innovations of the image, i.e., the discontinuities where it changes abruptly, so that its heat is held back at the boundary of its region. The decoder derives this medium from its own free-space render, so it costs no bits. Because the medium only needs the innovations, the same representation extends to degraded observations, where we read them with the annihilating-filter subspace of finite-rate-of-innovation sampling instead of the image gradient. At equal stored rate, IIHS is around 2 dB above the Gaussian-splat representations at low rates on Kodak-24 and DIV2K. When fitted to super-resolution and inpainting, it improves over them with the lowest LPIPS of all methods. Broadly, our work suggests that where a representation stops can matter as much as what it stores.



Figure 1: **Kodak images stored at** 0.03 **bits per pixel** (1.6 **kB each), decoded at the matched rate.** IIHS keeps object boundaries and fine structure that the other representations blur at this rate.

1 INTRODUCTION

Representing a single image as a continuous function using a neural field has become one of the most productive ideas in computer vision: one compact object that can be sampled at any resolution, differentiated, and fitted directly through the physics of an acquisition process. Two families dominate this space: coordinate networks encode the signal in the weights of a network and inherit its spectral bias (Mildenhall et al., 2022; Sitzmann et al., 2020; Tancik et al., 2020); splatting methods encode it in explicitly shaped primitives (Kerbl et al., 2023; Zhang et al., 2024) and inherit their speed. Yet, for both, faithfully reproducing high-frequency detail, such as sharp edges, fine texture and thin structures, remains a persistent struggle (Fig. 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.

More recently, approaches utilizing structural priors have been developed and shown promising advances over vanilla neural fields. In most, structure guides the fit and leaves the primitive unchanged. Recent splatting variants allocate more primitives where gradients are large (Zhang et al., 2025b; Liang et al., 2025); coordinate networks shape frequency content globally (Lindell et al., 2022; Shabanov et al., 2024), move adaptive bases toward structure (Chen et al., 2023), or carry explicit discontinuity sets when curves are given or learned on a mesh (Belhe et al., 2023; Liu et al., 2025). Here the guidance is auxiliary: it steers placement or penalizes shapes, while an anisotropic Gaussian elongated along an edge still extends across it, and a coordinate network can place an edge only by phase-aligning many frequencies. The closest step toward structure inside the renderer is contour-confined splatting (Takabe et al., 2026), which lets each Gaussian contribute only to the pixels of its own segmentation region. The boundary there is hard, and the renderer needs the segmentation map, which is obtained from an external segmenter or an annotation rather than from the representation itself.

Our approach makes structure part of how a primitive is rendered, rather than just a guide for where it is placed. For this, we propose Innovation-Insulated Heat Splats (IIHS), which represent an image as a heat field: a set of heat sources, each placed with a color and an anisotropic spread and diffused through a medium, called a "heat splat." In free space, the field of every source is an anisotropic Gaussian, and the representation reduces to a plain Gaussian splat. IIHS then derives the medium from the sources themselves: the decoder renders the sources in free space, estimates the innovation field of that render, and builds a medium that conducts everywhere except at the innovations it finds. The heat of each source is then held back at the walls of its region, so less of it is blended into the other side of a boundary. Unlike a segmentation map, these walls are soft, and no region labels are supplied to the decoder. Because the decoder derives the medium from the sources it already holds, it costs no parameters and no bits. One innovation field serves encoder and decoder: the encoder estimates it on the image to seed the sources, and the decoder on its own free-space render to place the walls. On a clean image, the gradient of the image supplies this field. On a degraded observation, whose gradient is that of the degradation, we recover it with the annihilation machinery of finite-rate-of-innovation sampling (Vetterli et al., 2002; Pan et al., 2014; Ongie & Jacob, 2016).

Our work contributes the following:

- **The heat splat (Sec. 3.1).** A Gaussian source rendered by diffusing it through a medium, so that its footprint follows the medium rather than its own parameters. It is stored with the parameters of a Gaussian splat, reduces to one exactly in free space, and any number of sources diffuse in one linear chain.
- **An innovation-insulated medium (Sec. 3.2).** The medium insulates at the discontinuities that carry the image structure; its wall strength depends on the scale being diffused, and under degradation the discontinuities are found by an annihilating-filter estimator.
- **Decode-derived guidance (Sec. 3.3).** The structure that shapes every footprint is never stored: the decoder estimates it on its own free-space render of the sources, and a handover of the walls and a free-space fidelity loss during fitting keep that render faithful.

2 Related Work

Implicit neural representations. Coordinate networks fit a signal into the weights of a deep net (Sitzmann et al., 2020; Tancik et al., 2020; Saragadam et al., 2023; Müller et al., 2022; Liu et al., 2024), with band-limited, adaptive-basis, spatially adaptive and discontinuity-aware variants (Lindell et al., 2022; Chen et al., 2023; Feng et al., 2026; Belhe et al., 2023; Liu et al., 2025). As image codecs, they quantize the weights or per-image modulations and, in later work, entropy-code them (Dupont et al., 2021; Strümpler et al., 2022; Dupont et al., 2022). Hybrids add explicit latent grids for a better rate–distortion trade-off at a low decoding cost (Ladune et al., 2023; Kim et al., 2024). These representations carry image structure only implicitly: an edge is placed by the joint action of many weights, the fit is slow, and the decoder evaluates a network at every pixel. Thera (Becker et al., 2025) coined *neural heat fields* for a corpus-trained super-resolution field whose activations decay by an analytic isotropic heat law. We are inspired by its heat-time axis and depart in what diffuses: Thera evaluates one network under a closed-form isotropic blur, whereas IIHS diffuses explicit sources through a medium whose walls shape each footprint.

Image representation with 2D Gaussians. 3D Gaussian splatting represents a scene with explicit Gaussian primitives and a fast rasterizer (Kerbl et al., 2023). GaussianImage brought the idea to single images, with a compact 2D primitive and a codec built on its quantized attributes (Zhang et al., 2024), and later work extends it to editable and to very large images (Waczyńska et al., 2025; Zhu et al., 2025; Pan et al., 2026) and to fast or feed-forward initialization (Zeng et al., 2025; Wang et al., 2026). Most recent work decides where the Gaussians go: it allocates them where the distortion is largest (Zhang et al., 2025b; Li et al., 2026; Chen et al., 2026) or where a structure prior points (Liang et al., 2025), or it confines their blending to segmentation regions (Takabe et al., 2026). Other primitives replace the Gaussian, with soft anisotropic diagrams (SAD) (Iinbor et al., 2026), wavelets (Zhang et al., 2025a) and the softly gated kernels of steered mixture-of-experts coding (Verhack et al., 2020; Li et al., 2025). In all of them, the footprint of a primitive is fixed by its own parameters. IIHS keeps the stored parameters of a Gaussian splat and changes the footprint instead: it is the result of diffusing the source through a medium that the decoder derives from the image structure.

Diffusion and structure-aware propagation. Letting values spread across an image while stopping them at its edges has long been established: as a filter, it smooths inside each region but not across the boundaries (Perona & Malik, 1990; Weickert, 1998); as a codec, it keeps a few pixel values and lets the decoder spread them into the space between the edges, refilling the image (Galić et al., 2008; Mainberger et al., 2011a; Schmaltz et al., 2014). Later work chooses which pixels to keep and optimizes their values so that diffusing them best reproduces the image (Mainberger et al., 2011b; Hoeltgen et al., 2013; Alt et al., 2022). In all of these, what spreads is the image itself, or a label for it, and the edges that stop it are read from the image as it is being restored. In contrast, IIHS spreads each stored source through a medium the decoder derives itself.

Structure detection under degradation. On a degraded observation, the image gradient mostly reproduces the degradation. To obtain high-fidelity edges in this setup, prior work has benefited from spectral data with concentration kernels (Gelb & Tadmor, 1999; Churchill et al., 2019), or from a field-of-junctions prior (Verbin & Zickler, 2021). Finite-rate-of-innovation sampling offers another robust route: a signal that is smooth between a few jumps, its *innovations*, is determined by where those jumps are, and a short filter that annihilates a low-frequency band of the measurements recovers them (Vetterli et al., 2002; Maravić & Vetterli, 2005; Dragotti et al., 2007; Blu et al., 2008; Pan et al., 2014). IIHS adopts the annihilation estimate as its innovation field under degradation and the plain gradient when the observation is clean.

3 METHOD

Figure 2 shows the overall proposed pipeline. IIHS stores an image y as N sources $\Theta = \{\theta_i\}_{i=1}^{N}$, with $\theta_i = (\boldsymbol{\mu}_i, \Sigma_i, \mathbf{c}_i)$ containing: a position $\boldsymbol{\mu}_i$, a covariance Σ_i, and a color $\mathbf{c}_i$, following an anisotropic Gaussian splat. Rendering diffuses each source as heat through a medium D that insulates at the innovations of the image and reads the image out from the diffused heat (Sec. 3.1). The medium is derived from an innovation field, a map of the discontinuities of an image, estimated by a gradient on clean images and by an annihilating-filter estimator on degraded observations (Sec. 3.2). The decoder derives the medium D from the innovation field of its own free-space render, so no structure is transmitted, and the encoder fits Θ by gradient descent through that same render (Sec. 3.3).

3.1 HEAT SPLAT

Primitives. A source i stores $\theta_i = (\boldsymbol{\mu}_i, \Sigma_i, \mathbf{c}_i)$, and $g_i(\mathbf{x}) = \exp\big(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^\top \Sigma_i^{-1}(\mathbf{x} - \boldsymbol{\mu}_i)\big)$ is its Gaussian at a pixel $\mathbf{x}$ of the image grid Ω, with $|\Omega|$ pixels. Drawn directly, the sources give a normalized Gaussian splat,

$$f_0(\mathbf{x}) \;=\; \frac{\sum_i g_i(\mathbf{x})\, \mathbf{c}_i}{\max(\sum_i g_i(\mathbf{x}),\, \delta)}, \tag{1}$$

called the *free-space render*, with g_i the weight of the color $\mathbf{c}_i$ and δ a small constant. This is needed because the stored sources cannot directly infer image structures, so the initial structure must be generated from f_0.

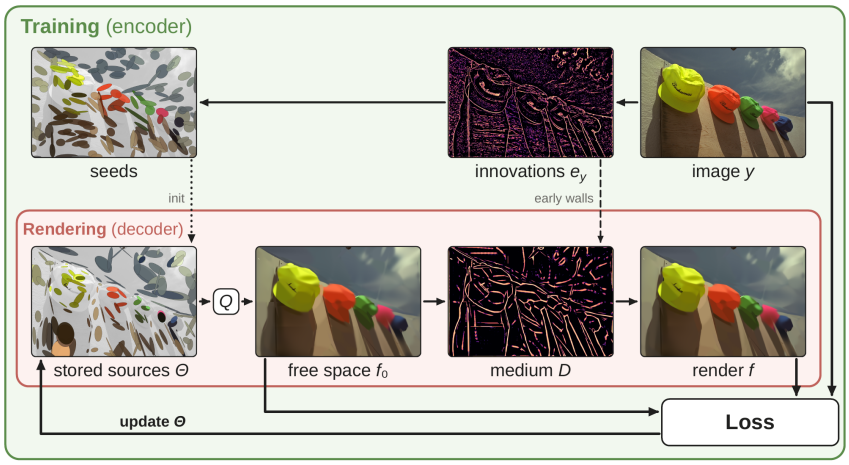

Figure 2: **Pipeline of IIHS.** The encoder estimates the innovation field e_y of the image y and seeds the sources on it. Rendering evaluates the free-space render f_0 of the sources, derives the medium D from the innovations of f_0, and diffuses the sources through D into the render f. The same render is the forward pass of every training iteration, so the sources Θ, the only thing stored, are fitted through the decoder that will later read them. Early in training, the walls are those of e_y (dashed); the medium is then handed over to the decoder's own. Q quantizes the sources, straight-through at the end of training and in the stored stream.

Shrink and diffuse. To shape the splats' footprint to follow the image structure, we must overcome two issues. First, an initial splat g_i is allowed to sit on top of the image structure, and second, tuning a Gaussian splat's parameters (position and covariance) cannot directly shape its footprint to follow the image structure. To address this, IIHS first reduces the size of the splat in an effort to pick a source that does not overlap with image innovations; then it grows the sources via heat diffusion along with the insulating medium to shape the splat's footprint. In free space, T explicit steps of size $\mathrm{d}t$ only widen a Gaussian,

$$\Sigma \;\mapsto\; \Sigma + V_T\,\mathbf{I}, \qquad V_T = 2\,\mathrm{d}t\,T \tag{2}$$

Source i therefore starts from the *starting Gaussian* g_i^0, which is g_i with Σ_i replaced by

$$\Sigma_i^0 \;=\; \Sigma_i - V_{T_i}\,\mathbf{I}, \qquad V_{T_i} = \min\big(\max(\sigma_{i,\min}^2 - 1,\,0),\; V_{T_{\max}}\big), \tag{3}$$

and diffuses for $T_i = V_{T_i}/(2\,\mathrm{d}t)$ steps, after which in free space it is g_i in the continuous limit: the stored Σ_i is the footprint a source reaches when nothing holds it back. Here $\sigma_{i,\min}$ is the standard deviation of g_i along its narrower axis (in pixels), so that g_i^0 stays at least one pixel wide, and $T_{\max}$ is the largest number of steps a source diffuses for (Appendix A.1). Each source releases heat $u_i \geq 0$ from g_i^0 and diffuses it through the *medium* $D\colon \Omega \to [0,1)$ (see Sec. 3.2), one explicit step at a time:

$$u_i^{(t+1)} \;=\; u_i^{(t)} + \mathrm{d}t\,\nabla\cdot\big((1 - D^{(t)})\,\nabla u_i^{(t)}\big), \qquad u_i^{(0)} = g_i^0. \tag{4}$$

After T_i steps, the heat a source delivers at a pixel takes the place of g_i as its weight, and we obtain the image

$$f(\mathbf{x}) \;=\; \frac{\sum_i u_i^{(T_i)}(\mathbf{x})\;\mathbf{c}_i}{\max\Big(\sum_i u_i^{(T_i)}(\mathbf{x}),\;\delta\Big)}. \tag{5}$$

When D is high, the heat is held back, so the source contributes little color beyond that point (Fig. 3); with $D \equiv 0$, $f = f_0$ up to discretization. A source rendered this way is a *heat splat*; f_Θ and $f_{0,\Theta}$ denote f and f_0 as functions of the sources.

3.2 Innovation-insulated medium

Medium. We first define the medium D, as it determines the heat splat's shape by insulating the splat's diffusion at locations where the image changes abruptly. Given an *innovation field* $e \in [0,1]$

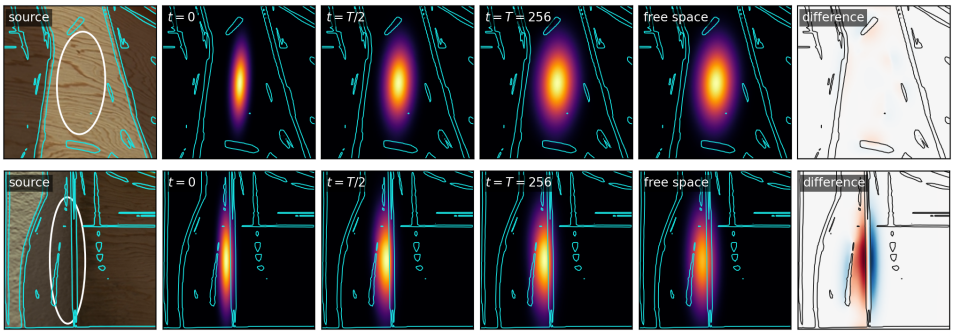

Figure 3: **A heat splat during diffusion**, with $D \equiv 0$ (top) and through a medium with a wall of high D (bottom); the last column shows the heat held back across the innovation.

of an image, close to 1 on its innovations, we define the medium D for an image as

$$D \;=\; s\,(1-\varepsilon)\,e, \qquad s(\tau) \;=\; \frac{V_\tau}{V_\tau + w^2}, \tag{6}$$

which conducts where the image is smooth and insulates at an innovation, scaled by a strength factor s and a conductance floor $\varepsilon = 0.01$ that keeps every wall slightly permeable. Here, $\tau = T_i - t$ denotes the steps a source has left in Eq. (4), and $V_\tau = 2\,\mathrm{d}t\,\tau$ is the variance those steps add. The strength of a wall s is near one while the remaining spread exceeds the wall width and fades as the source approaches its final size. A large source therefore gets insulated by full-strength walls, while a source only a few pixels wide passes through the medium almost as if it were free space. We set $w = 2.5$ px, the width of a typical wall band.

Innovation field. We introduce two ways to obtain the innovation field e for an image. We let e_y denote the field derived from the original image, and e_{f_0} the field derived from the free-space render. First, on a clean image, the gradient localizes innovations, and we use the *gradient estimator* $e = \min(|\nabla(G_1 * Y)|/\kappa,\, 1)$, i.e., the gradient magnitude of the image luma Y after a Gaussian blur G_1 of one pixel, scaled by a threshold κ. Second, on a degraded observation, the gradient is that of the degradation, making it challenging to compute the innovation field accurately. In this case, finite-rate-of-innovation sampling offers a robust route: the derivative of a piecewise-smooth image is annihilated by filters whose zeros lie on its innovations (Vetterli et al., 2002; Pan et al., 2014; Ongie & Jacob, 2016), and a low-frequency band of the observation, where the signal still dominates the degradation, determines them. On overlapping patches, we localize and blend the innovations through the annihilating subspace, and call this the *annihilation estimate* of e (Appendix A.2).

3.3 Decode-derived guidance

Decoder. The stored stream holds Θ and a header; everything else is computed. The decoder renders f_0 by Eq. (1), estimates e_{f_0} on it, derives the medium from the free-space render (denoted D_{f_0}, and D_y when derived from e_y) by Eq. (6), and runs the diffusion given in Eq. (4): nothing about the medium is transmitted.

Encoder. The encoder estimates e_y once on the image y, seeds the source positions on it, and fits the sources via the Adam optimizer (Kingma & Ba, 2015) through the rendering by minimizing

$$\mathcal{L}(\Theta) = \frac{1}{3|\Omega|}\big\|f_\Theta - y\big\|_2^2 \;+\; \frac{\gamma}{3|\Omega|}\big\|f_{0,\Theta} - y\big\|_2^2 \;+\; \frac{\lambda}{|\Omega|}\big\|(1-e_y)\,\nabla Y(f_\Theta)\big\|_1, \tag{7}$$

with weights γ and λ, the medium held constant under differentiation, and $Y(f_\Theta)$ the luma of the rendered image. The first term is the fidelity of the decode f_Θ, the image the decoder produces from the stored sources. The second term keeps the free-space render $f_{0,\Theta}$ close to the image, because the decoder derives its walls from f_0. Therefore, to render without the true innovations, we need this term to guide the free-space render to reproduce the image innovation as much as it can. The third, a total-variation (TV) term, penalizes the gradient of the decoded luma $Y(f_\Theta)$ everywhere except at the innovations of the image, where $1 - e_y$ vanishes, so that the sources stay smooth inside their regions without being pulled flat across a boundary.

Table 1: **Single-image representations on Kodak-24 at a matched parameter budget.** fp32 attributes, no quantization in training or storage, 24-image means; N = primitives the budget buys. Train and render times and peak GPU memory on kodim03 on one idle GPU; best per quality column in bold, second underlined.

Floats	Method	N	PSNR	SSIM	MS-SSIM	LPIPS	Train (s)	Render (ms)	GPU (MB)
1,573	GaussianImage++	196	21.98	0.540	0.668	0.780	48	0.13	780
	SAD	157	22.23	0.580	0.697	0.565	2	0.92	628
	Image-GS	196	<u>23.41</u>	**0.606**	<u>0.761</u>	**0.561**	25	0.36	838
	IIHS (ours)	196	**23.92**	<u>0.603</u>	**0.773**	<u>0.564</u>	507	12.60	1,952
56,623	GaussianImage++	7,075	33.46	0.875	<u>0.973</u>	0.183	46	0.14	782
	SAD	5,662	<u>33.79</u>	<u>0.883</u>	0.970	<u>0.152</u>	1	1.39	628
	Image-GS	7,077	32.04	0.876	0.972	0.195	24	0.41	842
	IIHS (ours)	7,077	**34.17**	**0.899**	**0.978**	**0.139**	424	13.00	2,804

Fitting initialization and schedule. Sources are seeded on e_y, colored from the image and oriented along the local edge tangent; every 400 iterations, the sources that own at most three pixels are moved to the pixels of largest residual, at a fixed count, so the bitstream is set before training. The fit runs $25{,}000$ iterations of Adam (schedule in Appendix Fig. 8): the walls are off for the first 10%, ramp in over the next 10%, and are handed over to D_{f_0} at 30%, re-estimated every 100 iterations. The last $5{,}000$ iterations are quantization-aware, with straight-through quantization, and the stored state is the best fresh decode of the quantized sources. The stream sorts, delta-codes, and entropy-codes the sources with asymmetric numeral systems (ANS) (Duda, 2013) (Appendix A.3).

4 Experiments

4.1 Setup

All experiments use Kodak-24 (Eastman Kodak, 1993) (24 images, 768×512) and the DIV2K validation set (Agustsson & Timofte, 2017) (100 images at full resolution). Baselines run from their official code at their published settings and are checked against the reported numbers with one scorer computing PSNR, SSIM, MS-SSIM, and LPIPS for all methods. Training details can be found in Appendix B; other settings are described in Appendix C, and code will be released upon publication.

4.2 Image Representation

We first compare IIHS with the Gaussian-splat representations at a matched parameter budget, fp32 attributes before any quantization, in Table 1. The budget fixes how many primitives each method stores; the table reports the fidelity of the decode and the cost to train and to render. At both budgets, IIHS has the highest representation fidelity compared to other methods. Since each IIHS source stores the same eight floats as a Gaussian, the medium that shapes its footprint is derived at decode time; gaining in fidelity comes without extra parameters, at the cost of training and rendering time.

4.3 Image Compression

Figure 4 shows the rate–distortion curve on the coded IIHS stream with the other per-image representations, the classical codecs, and the corpus-trained codecs, all against stored bits and under one scorer following the convention of prior works. The curve is generated from runs at six target rates on Kodak-24 and five on DIV2K, each with a fixed number of sources per image. Because the medium is derived from the sources themselves, this comes at the bit cost of a Gaussian splat: the stream holds only the sources, and the walls that shape them cost nothing. Fig. 5 shows the effect at matched rate: boundaries and thin structures stay sharp in IIHS, where the other splats blend across them. Additional rendered images (Figs. 9–12) and quantitative results (Tables 7 and 8) are provided in Appendix D.

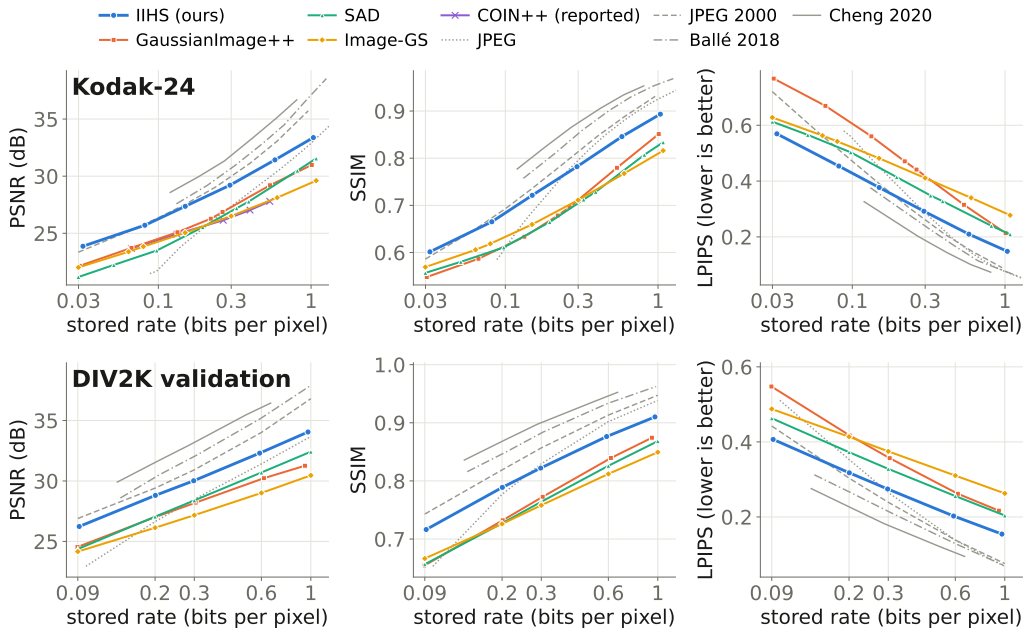

Figure 4: **Rate–distortion.** Mean PSNR, SSIM, and LPIPS of the decoded images against stored bits per pixel (log axis), one curve per method. Top: Kodak-24, bottom: DIV2K validation.

Table 2: **Degraded observations on Kodak-24.** PSNR / SSIM / LPIPS against the clean image, 24-image means; best per column and metric in bold, second underlined. IIHS is the best per-image method on every metric in both tasks and the best of all methods on LPIPS.

	Method	SR 2×	Inpainting 75%
Classical	Bicubic (Keys, 1981)	**29.39** / **0.876** / 0.178	n/a
	Fast marching (Telea, 2004)	n/a	24.49 / 0.765 / 0.300
	Biharmonic (Damelin & Hoang, 2018)	n/a	**26.08** / **0.828** / 0.230
Per-image	GaussianImage++ (Li et al., 2026)	25.68 / 0.716 / 0.417	23.20 / 0.700 / 0.355
	SAD (Iinbor et al., 2026)	26.30 / 0.780 / 0.190	21.44 / 0.648 / 0.381
	Image-GS (Zhang et al., 2025b)	27.44 / 0.765 / 0.338	25.74 / 0.804 / 0.210
	IIHS (ours)	28.02 / 0.833 / **0.139**	25.87 / 0.821 / **0.154**

4.4 Image Restoration

Gaussian-splat representations are seldom fitted to degraded observations, where the structure a fit relies on must be read from the observation itself. To extend to image restoration tasks, we fit every method through the forward operator A, and the data term of Eq. (7) becomes $\|Af_\Theta - y\|_2^2$. We evaluate $2\times$ super-resolution and inpainting with 75% of the pixels missing on Kodak-24; every method sees the same observation, stores $100{,}000$ floats, and picks its checkpoint by the error on held-out measurements (Appendix C). Table 2 shows that IIHS is the best per-image method on every metric in both tasks and has the lowest LPIPS of all methods, and Figs. 6 and 13 show that IIHS, compared against the classical references, preserves edges and structure better.

4.5 Ablations

Table 3 removes the components of IIHS one by one at 0.15 bpp on all 24 Kodak images over three seeds: we first remove the medium at render, then refit without the innovation field at training, and lastly refit with relocation disabled. Every component contributes, and the medium improves

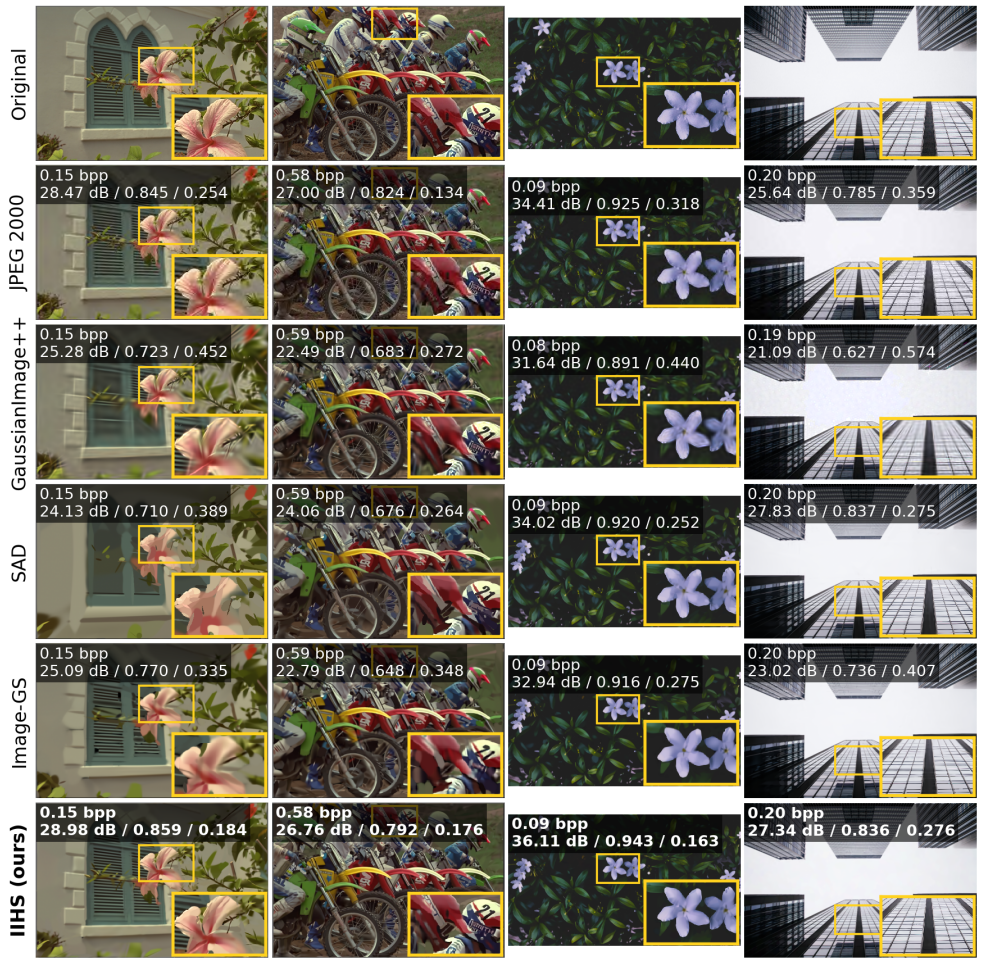

Figure 5: **Decodes at the target rates.** Two Kodak images at 0.15 and 0.6 bpp and two DIV2K images at 0.09 and 0.2 bpp. From the same number of bits, the walls keep each source's heat on its side of a boundary, so IIHS keeps edges and thin structures that the other splats blur at the low rate.



Figure 6: **Decodes from degraded observations.** kodim08 under 2× super-resolution (top) and inpainting with 75% of the pixels missing (bottom): original, input, the classical reference, IIHS, and a per-pixel map of which reconstruction is closer to the original; blue indicates IIHS is better.

the decoding of the stored bits significantly across seed variations (Fig. 7). Table 4 shows when the annihilation estimate contributes to image restoration. Bicubic decimation keeps the low frequencies that the annihilating filters are fitted to, and there the annihilation estimate gives stronger restoration; with 75% of the pixels missing, both estimators read the same fast-marching fill and the two fields perform alike.



Figure 7: **Decoding with and without the medium.** Same stored sources decoded in free space (f_0) and through the medium; blue indicates lower error with the medium.

Table 3: **Component ablation on Kodak-24.** Components removed cumulatively from IIHS; at 0.15 bpp, PSNR / SSIM / LPIPS of the stored-bit decode, 24-image means over three seeds ($\pm$ the standard deviation of the 24-image mean across seeds; per image).

Variant	PSNR	SSIM	LPIPS	step (dB)
IIHS (ours)	27.36±0.01	0.721±0.000	0.376±0.001	
− medium (free-space decode)	27.06±0.00	0.715±0.000	0.387±0.001	−0.30
− innovation field (no seeding or gated TV)	26.73±0.03	0.709±0.001	0.383±0.001	−0.33
− relocation	26.36±0.01	0.705±0.000	0.389±0.001	−0.37

Table 4: **Choice of the innovation field on Kodak-24.** PSNR / SSIM / LPIPS, 24-image means, on clean images at 0.15 bpp (stored-bit decode, one seed) and on the degraded tasks. Both fields at the encoder and decoder, same budget, selection rule, and scorer as Table 2.

Task	gradient	annihilation
clean, 0.15 bpp	27.36 / 0.721 / 0.377	27.01 / 0.715 / 0.375
SR 2×	27.70 / 0.822 / 0.155	28.02 / 0.833 / 0.139
inpainting 75%	26.03 / 0.822 / 0.153	25.87 / 0.821 / 0.154

5 Conclusion

We introduced IIHS, a single-image representation whose primitive is the heat splat: a Gaussian source diffused through a medium that insulates at the innovations of the image. Three components drive it: a heat splat, whose footprint follows the structure of the image at the bit cost of a Gaussian; an innovation-insulated medium that helps shape the heat splats' footprint, and a decoder that rebuilds the medium from its own free-space render, so the structure costs no bits. On Kodak and DIV2K, IIHS improves rate–distortion performance over the per-image representations we compare and extends to rates as low as 0.03 bits per pixel. Fitted through the forward operator, IIHS improves super-resolution and inpainting over the Gaussian-splat representations, with a lower LPIPS than the classical references. Overall, we hope this work opens a path towards primitives that carry image structure in how they are rendered without having to store it.

6 Limitations

Like other splatting methods, IIHS still lags behind corpus-trained neural codecs and JPEG 2000. For image restoration tasks, our implementation is a direct implementation: stronger regularization or guidance, such as learned priors on the sources or the medium, a context-adaptive entropy model or feed-forward initialization, and diffusing only the sources that reach a wall, are likely needed for the heat splat to match the classical and corpus-trained references. Lastly, IIHS is fitted to still images; carrying heat splats and their medium over to video and to 3D scenes is future work.

AI USE STATEMENT

In this work, we used generative AI tools (Claude Code) for the scripts that launch and run the experiment campaigns, for executing them, and for verifying the baseline code against its official releases. We also used them for literature search and for drafting and editing parts of the manuscript text. We have reviewed all AI-assisted work and ensured that the method and every ablation were designed and interpreted by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017.

Tobias Alt, Pascal Peter, and Joachim Weickert. Learning sparse masks for diffusion-based image inpainting. In *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 2022.

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 (ICLR)*, 2018.

Alexander Becker, Rodrigo Caye Daudt, Dominik Narnhofer, Torben Peters, Nando Metzger, Jan Dirk Wegner, and Konrad Schindler. Thera: Aliasing-free arbitrary-scale super-resolution with neural heat fields. *Transactions on Machine Learning Research*, 2025.

Jean Bégaint, Fabien Racapé, Simon Feltman, and Akshay Pushparaja. CompressAI: A PyTorch library and evaluation platform for end-to-end compression research. *arXiv preprint arXiv:2011.03029*, 2020.

Yash Belhe, Michaël Gharbi, Matthew Fisher, Iliyan Georgiev, Ravi Ramamoorthi, and Tzu-Mao Li. Discontinuity-aware 2D neural fields. *ACM Transactions on Graphics*, 42(6):1–11, 2023.

Josef Bigun and Gösta H Granlund. Optimal orientation detection of linear symmetry. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1987.

Thierry Blu, Pier Luigi Dragotti, Martin Vetterli, Pina Marziliano, and Lionel Coulot. Sparse sampling of signal innovations. *IEEE Signal Processing Magazine*, 25(2):31–40, 2008.

Jiacong Chen, Qingyu Mao, Xiandong Meng, Shuai Liu, Chao Li, Fanyang Meng, Youneng Bao, and Yongsheng Liang. Locality-aware density control for efficient Gaussian-based image representation. In *Proceedings of the ACM International Conference on Multimedia (MM)*, 2026.

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

Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized Gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Victor Churchill, Rick Archibald, and Anne Gelb. Edge-adaptive ℓ_2 regularization image reconstruction from non-uniform Fourier data. *Inverse Problems and Imaging*, 13(5):931–958, 2019.

Steven B Damelin and Nguyen S Hoang. On surface completion and image inpainting by biharmonic functions: Numerical aspects. *International Journal of Mathematics and Mathematical Sciences*, 2018(1):3950312, 2018.

Pier Luigi Dragotti, Martin Vetterli, and Thierry Blu. Sampling moments and reconstructing signals of finite rate of innovation: Shannon meets Strang–Fix. *IEEE Transactions on Signal Processing*, 55(5):1741–1757, 2007.

Jarek Duda. Asymmetric numeral systems: Entropy coding combining speed of Huffman coding with compression rate of arithmetic coding. *arXiv preprint arXiv:1311.2540*, 2013.

Emilien Dupont, Adam Goliński, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet. COIN: Compression with implicit neural representations. In *ICLR Workshop on Neural Compression: From Information Theory to Applications*, 2021.

Emilien Dupont, Hrushikesh Loya, Milad Alizadeh, Adam Goliński, Yee Whye Teh, and Arnaud Doucet. COIN++: Neural compression across modalities. *Transactions on Machine Learning Research*, 2022.

Eastman Kodak. Kodak lossless true color image suite (PhotoCD PCD0992). `http://r0k.us/graphics/kodak/`, 1993.

Haoan Feng, Diana Aldana, Tiago Novello, and Leila De Floriani. SASNet: Spatially-adaptive sinusoidal networks for INRs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Irena Galić, Joachim Weickert, Martin Welk, Andrés Bruhn, Alexander Belyaev, and Hans-Peter Seidel. Image compression with anisotropic diffusion. *Journal of Mathematical Imaging and Vision*, 31(2–3):255–269, 2008.

Anne Gelb and Eitan Tadmor. Detection of edges in spectral data. *Applied and Computational Harmonic Analysis*, 7(1):101–135, 1999.

Laurent Hoeltgen, Simon Setzer, and Joachim Weickert. An optimal control approach to find sparse data for Laplace interpolation. In *Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*. Springer, 2013.

Laki Iinbor, Zhiyang Dou, and Wojciech Matusik. Soft anisotropic diagrams for differentiable image representation. *ACM Transactions on Graphics*, 45(4):158:1–158:19, 2026.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, 2023.

Robert Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1153–1160, 1981.

Hyunjik Kim, Matthias Bauer, Lucas Theis, Jonathan Richard Schwarz, and Emilien Dupont. C3: High-performance and low-complexity neural compression from a single image or video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

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

Théo Ladune, Pierrick Philippe, Félix Henry, Gordon Clare, and Thomas Leguay. Cool-chic: Coordinate-based low complexity hierarchical image codec. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

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*, 2026.

Yi-Hsin Li, Mårten Sjöström, Sebastian Knorr, and Thomas Sikora. Rasterized steered mixture of experts for efficient 2D image regression. *arXiv preprint arXiv:2510.05814*, 2025.

Huanxiong Liang, Yunuo Chen, Yicheng Pan, Sixian Wang, Jincheng Dai, Guo Lu, and Wenjun Zhang. Structure-guided allocation of 2D Gaussians for image representation and compression. *arXiv preprint arXiv:2512.24018*, 2025.

David B Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. BACON: Band-limited coordinate networks for multiscale scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Chenxi Liu, Siqi Wang, Matthew Fisher, Deepali Aneja, and Alec Jacobson. 2D neural fields with learned discontinuities. *Computer Graphics Forum*, 44(2):e70023, 2025.

Zhen Liu, Hao Zhu, Qi Zhang, Jingde Fu, Weibing Deng, Zhan Ma, Yanwen Guo, and Xun Cao. FINER: Flexible spectral-bias tuning in implicit neural representation by variable-periodic activation functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Markus Mainberger, Andrés Bruhn, Joachim Weickert, and Søren Forchhammer. Edge-based compression of cartoon-like images with homogeneous diffusion. *Pattern Recognition*, 44(9):1859–1873, 2011a.

Markus Mainberger, Sebastian Hoffmann, Joachim Weickert, Ching Hoo Tang, Daniel Johannsen, Frank Neumann, and Benjamin Doerr. Optimising spatial and tonal data for homogeneous diffusion inpainting. In *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer, 2011b.

Irena Maravić and Martin Vetterli. Sampling and reconstruction of signals with finite rate of innovation in the presence of noise. *IEEE Transactions on Signal Processing*, 53(8):2788–2805, 2005.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65 (1):99–106, 2022.

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

Greg Ongie and Mathews Jacob. Off-the-grid recovery of piecewise constant images from few Fourier samples. *SIAM Journal on Imaging Sciences*, 9(3):1004–1041, 2016.

Hanjie Pan, Thierry Blu, and Pier Luigi Dragotti. Sampling curves with finite rate of innovation. *IEEE Transactions on Signal Processing*, 62(2):458–471, 2014.

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 (CVPR)*, 2026.

Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. Revising densification in Gaussian splatting. In *European Conference on Computer Vision (ECCV)*. Springer, 2024.

Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G Baraniuk. WIRE: Wavelet implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Christian Schmaltz, Pascal Peter, Markus Mainberger, Franziska Ebel, Joachim Weickert, and Andrés Bruhn. Understanding, optimising, and extending data compression with anisotropic diffusion. *International Journal of Computer Vision*, 108(3):222–240, 2014.

Akhmedkhan Shabanov, Shrisudhan Govindarajan, Cody Reading, Lily Goli, Daniel Rebain, Kwang Moo Yi, and Andrea Tagliasacchi. BANF: Band-limited neural fields for levels of detail reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Athanassios Skodras, Charilaos Christopoulos, and Touradj Ebrahimi. The JPEG 2000 still image compression standard. *IEEE Signal Processing Magazine*, 18(5):36–58, 2001.

Yannick Strümpler, Janis Postels, Ren Yang, Luc Van Gool, and Federico Tombari. Implicit neural representations for image compression. In *European Conference on Computer Vision (ECCV)*. Springer, 2022.

Masaya Takabe, Hiroshi Watanabe, Sujun Hong, Tomohiro Ikai, Zheming Fan, Ryo Ishimoto, Kakeru Sugimoto, and Ruri Imichi. Contour information aware 2D Gaussian splatting for image representation. In *IEEE International Conference on Consumer Electronics (ICCE)*, 2026.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Alexandru Telea. An image inpainting technique based on the fast marching method. *Journal of Graphics Tools*, 9(1):23–34, 2004.

Dor Verbin and Todd Zickler. Field of junctions: Extracting boundary structure at low SNR. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Ruben Verhack, Thomas Sikora, Glenn Van Wallendael, and Peter Lambert. Steered mixture-of-experts for light field images and video: Representation and coding. *IEEE Transactions on Multimedia*, 22(3):579–593, 2020.

Martin Vetterli, Pina Marziliano, and Thierry Blu. Sampling signals with finite rate of innovation. *IEEE Transactions on Signal Processing*, 50(6):1417–1428, 2002.

Joanna Waczyńska, Tomasz Szczepanik, Piotr Borycki, Sławomir Tadeja, Thomas Bohné, and Przemysław Spurek. MiraGe: Editable 2D images using Gaussian splatting. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.

Gregory K Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 34(4):30–44, 1991.

Hao Wang, Ashish Bastola, Chaoyi Zhou, Wenhui Zhu, Xiwen Chen, Xuanzhao Dong, Siyu Huang, and Abolfazl Razi. Fast 2DGS: Efficient image representation with deep Gaussian prior. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, 2026.

Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *Asilomar Conference on Signals, Systems & Computers*, 2003.

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.

Joachim Weickert. *Anisotropic Diffusion in Image Processing*. Teubner, Stuttgart, 1998.

Zhaojie Zeng, Yuesong Wang, Tao Guan, Chao Yang, and Lili Ju. Instant GaussianImage: A generalizable and self-adaptive image representation via 2D Gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

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

Wenhao Zhang, Hao Zhu, Delong Wu, Di Kang, Linchao Bao, Xun Cao, and Zhan Ma. WIPES: Wavelet-based visual primitives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025a.

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 (ECCV)*. 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 *ACM SIGGRAPH Conference Papers*, 2025b.

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*, 2025.

A Implementation details

This appendix follows the order of Sec. 3: the heat-splat decoder (Sec. 3.1), the innovation estimators behind the medium (Sec. 3.2), and the stored stream (Sec. 3.3).

A.1 Heat-splat decoder

Numerical scheme. The step is explicit and conservative, $\mathrm{d}t$ inside the stability limit of 0.25; $\delta = 10^{-4}$, and the other constants of the decoder are in Table 6. The normalized readout replaces opacity. The flux across a pixel face uses the conductivity $1 - \max(D_a, D_b)$, with D_a and D_b the insulation of the two pixels that share it, so heat is conserved up to rounding, and a wall one pixel thin closes all four of its faces; a pixel inside a wall band is then nearly isolated, which lowers fidelity inside the bands at high rates, where the sources are only a few pixels wide.

One chain for all sources. Diffusing each source on its own would cost N chains of up to $T_{\max}$ steps. Every step of Eq. (4) is linear, so $u_i^{(T_i)} = \mathcal{P}_{T_i}\, g_i^0$, with $\mathcal{P}_T$ the last T steps of one chain of $T_{\max}$ steps: all sources share that chain, g_i^0 is injected when T_i steps remain, and the numerator and denominator of Eq. (5) ride it as four channels. A decode therefore costs one chain of $T_{\max}$ steps and one splatting pass per diffusion time, independent of N: the render takes 12.6 ms with 196 sources and 13.0 ms with $7{,}077$ (Table 1).

Free-space footprint. Each g_i is a rotated anisotropic Gaussian, $\Sigma_i = R(\phi_i)\,\mathrm{diag}(\sigma_{i,1}^2, \sigma_{i,2}^2)\,R(\phi_i)^\top$ with orientation ϕ_i and standard deviations $\sigma_{i,1} \geq \sigma_{i,2} = \sigma_{i,\min}$, sampled on a window whose radius is $2.5\,\sigma_{i,1}$ rounded up to a bucket, normalized to unit mass on the window and scaled by its continuous mass with a one-pixel floor, $2\pi\sqrt{(\sigma_{i,1}^2 + \frac{1}{2\pi})(\sigma_{i,2}^2 + \frac{1}{2\pi})}$, so a broad source peaks near 1 and a source thinner than a pixel keeps the weight of one pixel.

Scale cap. For an $H \times W$ image, $\sigma_{\max} = \mathrm{clip}\big(2\sqrt{HW/(\pi N_\sigma)},\, 16,\, 64\big)$ px, rounded, with $N_\sigma = \lfloor \mathrm{floats}/9 \rfloor$ (54 px at the lowest Kodak rate, 16 px at the highest); it bounds the scale quantizer and is stored in the header.

Diffusion times. V_{T_i} of Eq. (3) is rounded down to one of 17 times between 0 and $T_{\max} = 256$ steps, spaced by about $\sqrt{2}$, so the shared chain needs 17 splatting passes; the remainder stays in g_i^0. $V_T = 0.4\,T\,\mathrm{px}^2$ at $\mathrm{d}t = 0.2$, measured from the stencil's impulse response.

Diffusion canvas and coarse grid. The diffusion runs on a padded canvas whose margin is folded back onto the border, as Eq. (1) clamps its windows, so sources near the border spread as in open space. Segments of 32 steps or more run on a $2\times$ coarser grid (sum-pooled field, max-pooled insulation so that walls stay closed, a quarter of the steps, bilinear upsampling).

Step size. At $\mathrm{d}t = 0.2$, the highest grid frequency is multiplied by $1 - 8\,\mathrm{d}t = -0.6$ per step; it decays with alternating sign, which matters only for the shortest diffusion times.

A.2 Innovation estimators

IIHS uses two estimators of the innovation field $e \in [0, 1]$ (Sec. 3.2), both on the BT.601 luma Y, returned without binarization.

Gradient estimator. $e = \min(|\nabla(G_1 * Y)|/\kappa,\, 1)$, with G_1 a Gaussian blur of $\sigma = 1$ px and κ the threshold of Table 6.

Annihilation estimator. On a degraded observation, the estimator runs on the stand-in that every fitted method reads (Appendix C: the bicubic lift of the low-resolution image for super-resolution, the fast-marching fill for inpainting), with the annihilating-filter model of finite-rate-of-innovation sampling (Vetterli et al., 2002; Pan et al., 2014; Ongie & Jacob, 2016). On overlapping patches of side S (luma edge-padded by $S/2$), the derivative spectra of patch p along both axes form the annihilation matrix $\mathbf{M}_p$ of an $n_f \times n_f$ filter. We keep the $|\mathcal{A}|$ eigenvectors of $\mathbf{M}_p^*\mathbf{M}_p$ with the smallest eigenvalues; read as the spectrum of a trigonometric polynomial $\psi_m^{(p)}$ on the patch, each vanishes on the innovations inside the patch, so their sum of squares ρ_p is near zero on the edges; the field is one minus its normalized, Hann-weighted average (sizes in Table 6),

$$\begin{aligned} \rho_p &= \textstyle\sum_{m \in \mathcal{A}} \big|\psi_m^{(p)}\big|^2, \\ e &= \max\!\left(1 - \frac{\sum_p h_p\, \rho_p / \bar{\rho}_p}{\sum_p h_p},\, 0\right), \end{aligned} \tag{8}$$

with h_p the Hann window of patch p and $\bar{\rho}_p$ the patch mean of ρ_p; the field is then rescaled to unit maximum.

A.3 Bitstream

A source is stored as the eight attributes of Table 5: its position, its covariance as two log-scales and an orientation, and its color. The sources are then sorted by quantized position; y is delta-coded, and x is coded in absolute when y changes and as a delta otherwise. Each field gets one parametric model (uniform, quantized Gaussian, quantized Laplace, or a dense histogram, chosen by coded size including side information), and everything goes into one ANS stream (Duda, 2013) with a 37-byte header (magic, version, flags, N, H, W, $\sigma_{\max}$, bit widths, color range) and the model parameters. The coder reproduces the fitter's quantizer bit for bit and asserts a round-trip decode. The decoder's constants (estimator, diffusion times, ε, $\mathrm{d}t$) are the same for every image and rate and are not transmitted. A source costs 64 bits at 196 sources and 58 at 7,077 (kodim03; 72 raw), the delta-coded positions shrinking as the count grows.

Table 5: **The stored source.** Its free-space footprint: eight attributes, 72 bits before entropy coding (DIV2K: 14-bit positions).

field	symbols	bits	quantization range
position	μ_y, μ_x	$12 + 12$	fixed, $[0, H-1]$, $[0, W-1]$
log-scales	$\log \sigma_1, \log \sigma_2$	$8 + 8$	fixed, $[\log 0.35, \log \sigma_{\max}]$
orientation	ϕ	8	$[0, \pi)$, wrapped
color	r, g, b	8×3	fixed, $[-1, 2]$

Table 6: **Settings.** The same for every image and rate (DIV2K: Appendix C).

Innovation estimators	
patch side, stride, window	32, 16, Hann
filter size n_f, subspace size $\lvert\mathcal{A}\rvert$	17, 202
Medium and diffusion	
conductance floor ε, wall width w	0.01, 2.5 px
gradient estimator threshold κ	0.05
step $\mathrm{d}t$, chain length $T_{\max}$, diffusion times	0.2, 256, 17
narrowest starting Gaussian (Sec. 3.1)	1 px
Training	
optimizer, iterations	Adam (full batch), 25,000
learning rates (position, orientation; scale, color)	10^{-1}; 2×10^{-2}
weights γ (free-space fidelity), λ (TV)	0.25, 10^{-3}
walls off; ramp; walls of the image until	10%; 10%; 30% of training
innovation estimator, clean images	gradient (Sec. 3.2)
innovation estimator, degraded observations	annihilation (Sec. 3.2)
medium re-estimated every	100 iterations (20 during QAT)
relocation of idle sources	every 400 iterations, until 80%
quantization-aware training	last 5,000 iterations, learning rates $\times 0.1$
stored state	best fresh decode, every 250 iterations
sources seeded on e_y (rest uniform)	50% below 0.1 bpp, 100% above

B TRAINING AND SETTINGS

Table 6 lists the settings. All fits and decodes run on one NVIDIA RTX PRO 6000 Blackwell GPU (96 GB) with an Intel Core Ultra 9 285K CPU (24 threads) and 64 GB of RAM, with PyTorch 2.10 and CUDA 13.0.

Schedule (Fig. 8)**.** e_y is used three times: to seed the positions (drawn with probability proportional to e_y), to gate the total-variation term, and as the walls early in training. The walls then ramp in, and the medium is handed over from D_y to D_{f_0}, re-estimated on the current free-space render; quantization-aware training uses straight-through quantization at the widths of Table 5. Without the free-space fidelity term, the free-space render of sources trained inside walls falls 1.6 dB below that of sources trained without walls (26.15 against 27.75 dB, kodim03, lowest rate).

Initialization and relocation. Each source starts with the color of the image at its position and the scale $\min\big(\frac{1}{2}\sqrt{HW/(\pi N)},\ \frac{1}{2}\sigma_{\max}\big)$, and it is oriented along the local edge tangent with an aspect set by the structure-tensor coherence (Bigun & Granlund, 1987). At each relocation step, the sources that own at most three pixels (top-1 assignment of the pixels to the free-space Gaussians) move to the pixels of largest 3×3-averaged residual and restart at $\sigma = 1$ px, with a random orientation and the image color there: the error-driven allocation of Li et al. (2026) and Rota Bulò et al. (2024) at a fixed count, so the bitstream size is set before training.

Decoder consistency. The training medium lags the sources by up to one re-estimation interval, so the training render is not the decode. At a fixed interval, the encoder decodes the quantized

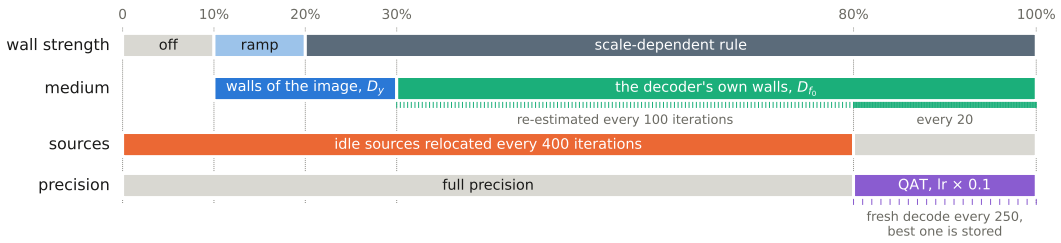

Figure 8: **Training schedule.** The 25,000 iterations of a fit.

sources from scratch, medium included, and keeps the best; every reported compression number is that decode, and a fresh-process decode of the bitstream on the same GPU and software stack reproduces it.

C Evaluation protocol and baselines

Scoring. One scorer for every decode: PSNR, SSIM (Wang et al., 2004) (luma, 11×11 Gaussian window, σ=1.5), MS-SSIM (Wang et al., 2003) (RGB) and LPIPS (Zhang et al., 2018) (AlexNet).

Rate accounting. Every curve is drawn against the bits stored: for IIHS, the entropy-coded payload with every header; for the codecs, the file size; for Image-GS, 16 bits per attribute without entropy coding (128 bits per Gaussian), the convention of its paper; for SAD, which stores fp32 sites without a quantizer, 128 bits per site as in its paper.

GaussianImage++. GaussianImage++ (Li et al., 2026) runs its official code with its compression recipe; every GaussianImage++ number in the paper is our run. It reproduces the fidelity track of its paper to 0.2 dB (35.24 vs. 35.41 dB at 10,000 Gaussians) and the two Kodak-24 compression points it prints to 0.12 dB (25.37 vs. 25.3 dB at 0.15 bpp, 31.22 vs. 31.1 at 1.08, our runs of Table 7 interpolated in log-rate).

SAD. SAD (Iinbor et al., 2026) runs its official code at a pinned commit with its 4,000-iteration schedule, within 0.41 dB of its paper under the paper's protocol (45.59 vs. 46.00 dB at 50,000 sites).

Image-GS. Image-GS (Zhang et al., 2025b) runs its official code and configuration; its paper reports no Kodak-24 numbers, and at 50,000 Gaussians our run reaches 39.09 dB.

COIN++. COIN++ (Dupont et al., 2022) enters with the Kodak-24 curve released with its code (its trained models are not public). Its rate counts only the entropy-coded modulations of 384 patches of 32×32, not the corpus-trained network, so its row is a lower bound on its rate.

Classical and learned codecs. JPEG (Wallace, 1991) (libjpeg, 4:2:0) and JPEG 2000 (Skodras et al., 2001) (OpenJPEG, irreversible 9/7 with the color transform) run across their quality ladders; Ballé 2018, the hyperprior model of Ballé et al. (2018), and Cheng 2020, the anchor model of Cheng et al. (2020), are the pretrained CompressAI releases (Bégaint et al., 2020), which reproduce the published CompressAI Kodak curves to 0.01 dB.

Image Restoration Setups. Every method shares one operator. Super-resolution by a factor r is bicubic resizing of the $H \times W$ image to $H/r \times W/r$ (PyTorch, align-corners off, no anti-aliasing); inpainting keeps each pixel with probability 0.25 under one fixed mask per image. A fixed 10% of the observed measurements, identical for every method, is held out. Each fitted method is rendered at seven checkpoints at the same fractions of its own schedule (1.6, 4.8, 14.4, 24, 48, 96 and 100%) and reports the one with the lowest held-out error; for Image-GS, which may stop early, its last step replaces the checkpoints it does not reach, and SAD's checkpoints count once its sites are within the budget. IIHS stores 100,000 floats (12,500 sources) with the settings of Table 6 and the annihilation estimator. The baselines run their official code at 100,000 fp32 floats: GaussianImage++ and Image-GS with 12,500 Gaussians (50,000 iterations; Image-GS: its official schedule, at most 10,000), and SAD with 10,000 sites and its 4,000-iteration schedule. A method that initializes from its target reads a stand-in with the missing and held-out pixels filled by fast marching (Telea, 2004); for super-resolution, every fitted method renders at full resolution and fits through A. The classical methods,

Table 7: **Kodak-24 at the six target rates.** Best of the per-image methods in bold, second best underlined, among the methods at most 10% above the target rate. †COIN++: the curve released by its authors.

	Method		0.03 bpp	0.08	0.15	0.3	0.6	1.08
Classical	JPEG (Wallace, 1991)	bpp	n/a	0.088	0.141	0.256	0.610	1.005
		PSNR	n/a	21.48	23.85	26.67	30.49	32.91
		SSIM	n/a	0.585	0.676	0.773	0.882	0.926
		LPIPS	n/a	0.580	0.465	0.323	0.141	0.075
	JPEG 2000 (Skodras et al., 2001)	bpp	0.030	0.099	0.149	0.239	0.599	0.958
		PSNR	23.34	26.34	27.52	29.07	33.06	35.72
		SSIM	0.586	0.692	0.735	0.788	0.888	0.931
		LPIPS	0.721	0.473	0.392	0.302	0.151	0.090
Learned	Ballé 2018 (Ballé et al., 2018)	bpp	n/a	n/a	0.131	0.320	0.669	0.939
		PSNR	n/a	n/a	27.58	30.96	34.51	36.72
		SSIM	n/a	n/a	0.757	0.860	0.931	0.954
		LPIPS	n/a	n/a	0.376	0.228	0.114	0.083
	Cheng 2020 (Cheng et al., 2020)	bpp	n/a	0.120	0.184	0.271	0.594	0.806
		PSNR	n/a	28.58	29.96	31.33	35.10	36.68
		SSIM	n/a	0.777	0.825	0.866	0.934	0.953
		LPIPS	n/a	0.326	0.261	0.201	0.101	0.074
Per-image	GaussianImage++ (Li et al., 2026)	bpp	0.031	0.067	0.133	0.263	0.538	1.010
		PSNR	22.15	23.72	25.09	26.86	29.20	30.98
		SSIM	0.548	0.587	0.633	0.696	0.780	0.851
		LPIPS	0.767	0.670	0.560	0.441	0.315	0.213
	COIN++† (Dupont et al., 2022)	bpp	n/a	0.067	0.132	0.266	0.537	n/a
		PSNR	n/a	23.68	24.93	26.14	27.80	n/a
	SAD (Iinbor et al., 2026)	bpp	0.030	0.098	0.195	0.326	0.814	1.080
		PSNR	21.17	23.49	25.54	27.19	30.48	31.55
		SSIM	0.557	0.611	0.665	0.712	0.807	0.834
		LPIPS	0.613	0.503	0.414	0.348	0.240	0.209
	Image-GS (Zhang et al., 2025b)	bpp	0.030	0.080	0.150	0.300	0.600	1.080
		PSNR	22.01	23.84	25.05	26.51	28.12	29.61
		SSIM	0.569	0.619	0.660	0.711	0.768	0.816
		LPIPS	0.628	0.542	0.482	0.411	0.340	0.278
	IIHS (ours)	sources	196	524	983	1,966	3,932	7,077
		bpp	0.032	0.082	0.150	0.295	0.580	1.035
		PSNR	**23.87**	**25.70**	**27.36**	**29.21**	**31.42**	**33.38**
		SSIM	**0.601**	**0.665**	**0.721**	**0.782**	**0.846**	**0.893**
		LPIPS	**0.569**	**0.454**	**0.377**	**0.292**	**0.210**	**0.148**

bicubic interpolation (Keys, 1981), biharmonic inpainting (Damelin & Hoang, 2018) (scikit-image), and fast marching (Telea, 2004) (OpenCV), are closed-form and use every observed pixel. PSNR is over the RGB image (inpainting: the missing pixels only), and SSIM and LPIPS are over the whole image, for inpainting with the observed pixels composited back.

D Additional results

D.1 Kodak-24 and DIV2K at the target rates

Tables 7 and 8 give the measured values behind Fig. 4. At every target rate on both sets, IIHS is the best per-image representation on all three metrics, with a stream that holds only its sources.

Table 8: **DIV2K validation at the five target rates.** Best of the per-image methods in bold, second best underlined, among the methods at most 10% above the target rate.

	Method		0.09 bpp	0.2	0.3	0.6	1.0
Classical	JPEG (Wallace, 1991)	bpp	0.098	0.200	0.301	0.600	0.997
		PSNR	22.94	26.64	28.48	31.39	33.62
		SSIM	0.650	0.775	0.827	0.900	0.936
		LPIPS	0.515	0.356	0.268	0.140	0.069
	JPEG 2000 (Skodras et al., 2001)	bpp	0.090	0.200	0.300	0.600	1.000
		PSNR	26.88	29.38	30.89	33.97	36.75
		SSIM	0.740	0.816	0.854	0.911	0.946
		LPIPS	0.445	0.307	0.242	0.141	0.076
Learned	Ballé 2018 (Ballé et al., 2018)	bpp	0.139	0.209	0.306	0.593	0.984
		PSNR	28.56	30.48	32.07	35.09	37.75
		SSIM	0.813	0.847	0.881	0.930	0.961
		LPIPS	0.316	0.265	0.215	0.133	0.072
	Cheng 2020 (Cheng et al., 2020)	bpp	0.135	0.206	0.287	0.511	0.660
		PSNR	29.91	31.66	33.00	35.46	36.44
		SSIM	0.836	0.871	0.897	0.936	0.952
		LPIPS	0.276	0.223	0.182	0.121	0.095
Per-image	GaussianImage++ (Li et al., 2026)	bpp	0.090	0.201	0.304	0.616	0.941
		PSNR	<u>24.54</u>	27.03	28.21	30.23	31.26
		SSIM	0.654	<u>0.732</u>	<u>0.773</u>	<u>0.839</u>	<u>0.874</u>
		LPIPS	0.548	0.417	0.357	0.262	0.217
	SAD (Iinbor et al., 2026)	bpp	0.090	0.200	0.300	0.600	1.000
		PSNR	24.37	<u>27.05</u>	<u>28.39</u>	<u>30.70</u>	<u>32.42</u>
		SSIM	0.657	0.727	0.764	0.826	0.868
		LPIPS	<u>0.463</u>	<u>0.373</u>	<u>0.328</u>	<u>0.256</u>	<u>0.204</u>
	Image-GS (Zhang et al., 2025b)	bpp	0.090	0.200	0.300	0.600	1.000
		PSNR	24.16	26.13	27.17	29.01	30.46
		SSIM	<u>0.666</u>	0.726	0.758	0.812	0.849
		LPIPS	0.488	0.414	0.375	0.310	0.263
	IIHS (ours)	bpp	0.091	0.201	0.298	0.589	0.971
		PSNR	**26.23**	**28.81**	**30.02**	**32.30**	**34.05**
		SSIM	**0.716**	**0.789**	**0.822**	**0.876**	**0.910**
		LPIPS	**0.407**	**0.318**	**0.275**	**0.203**	**0.154**

D.2 Kodak-24 and DIV2K decode visualizations at the target rates

Figures 9–12 extend Fig. 5 to three more images per set, at 0.08 and 0.15 bpp on Kodak-24 and 0.09 and 0.2 bpp on DIV2K: as there, IIHS keeps the boundaries and thin structures that the other splats blend across.

D.3 Degraded observations

Figure 13 extends Fig. 6 to every method on four more images; as there, IIHS differs from the classical reference mainly along edges and fine structure.



Figure 9: **Decodes at 0.08 bits per pixel** (kodim01, kodim14, kodim21). JPEG 2000 and the per-image methods, each at the stored rate of IIHS or the closest rate it reaches; each panel gives the stored rate and PSNR / SSIM / LPIPS, and the inset zooms into the boxed region.



Figure 10: **Decodes at 0.15 bits per pixel** (kodim01, kodim14, kodim21), same layout as Fig. 9.

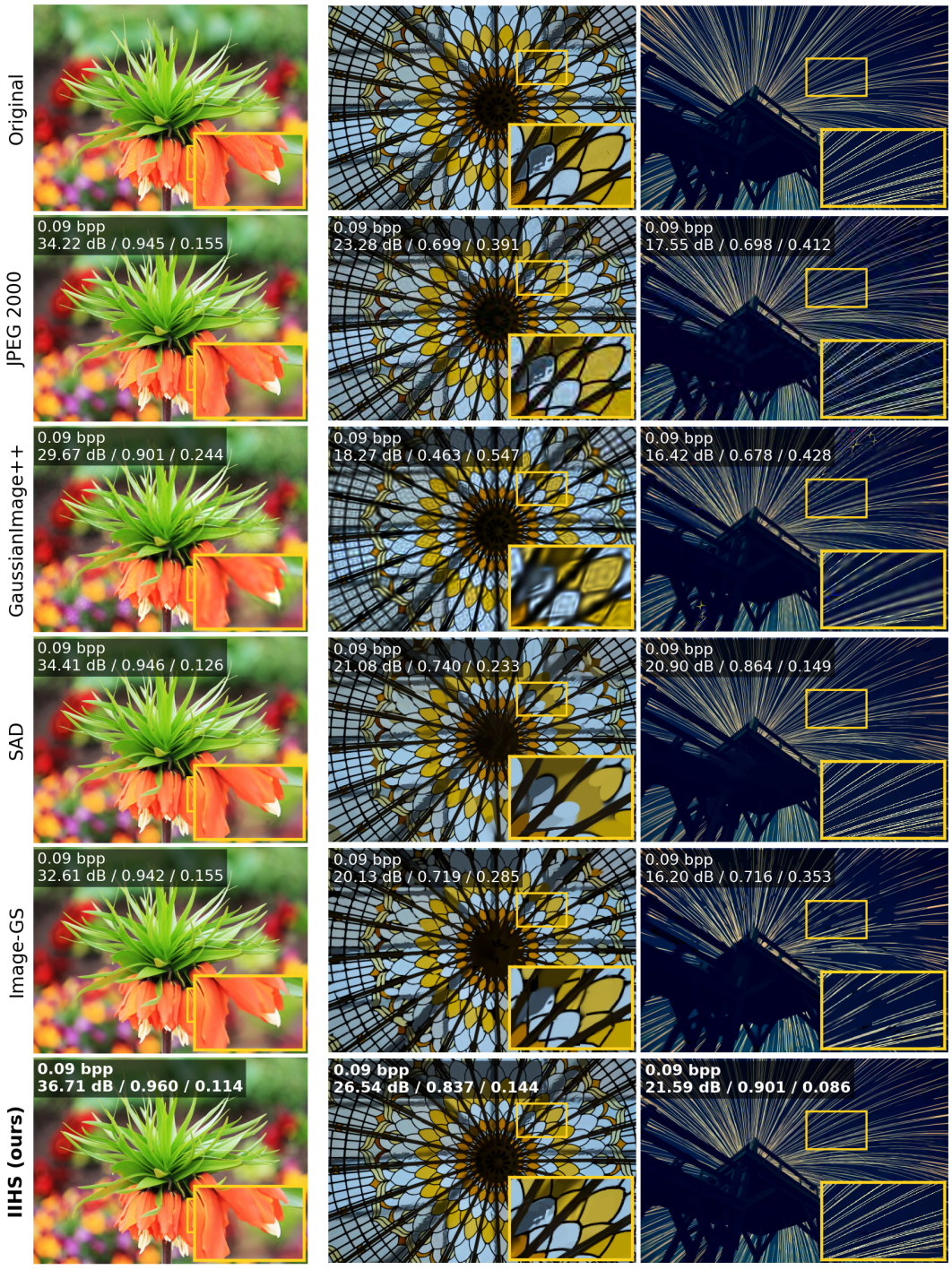

Figure 11: **DIV2K decodes at 0.09 bits per pixel** (0803, 0821, 0828), same layout as Fig. 9.

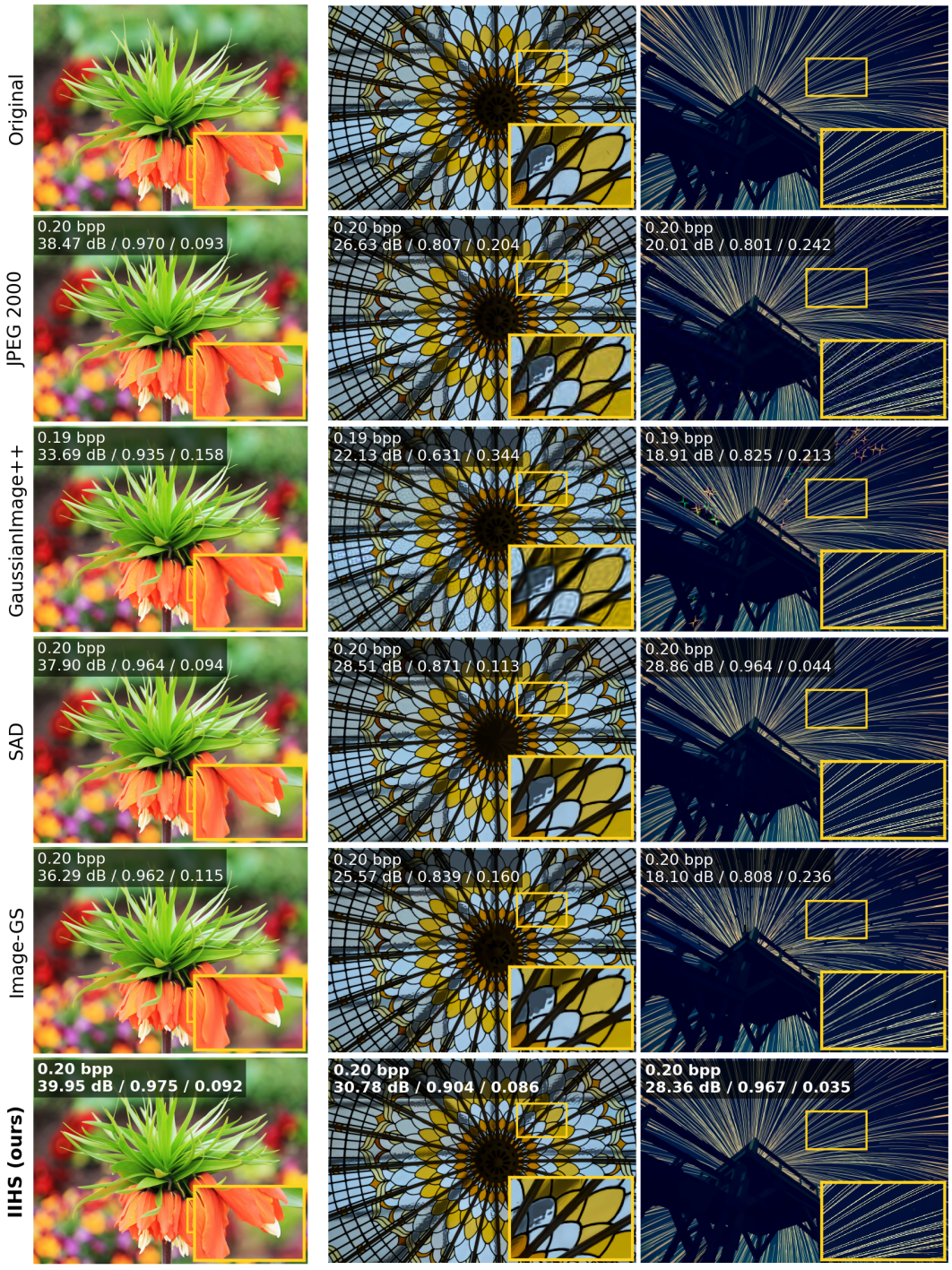

Figure 12: **DIV2K decodes at 0.2 bits per pixel** (0803, 0821, 0828), same layout as Fig. 9.



Figure 13: **Decodes from degraded observations** (SR $2\times$: kodim20, kodim16; inpainting 75%: kodim14, kodim24). Every fitted method at the checkpoint the shared selection rule picks (Sec. 4.4); the classical row is bicubic interpolation for super-resolution and biharmonic inpainting for inpainting, and inpainting decodes are composited with the observed pixels. Each panel gives PSNR / SSIM / LPIPS, and the inset zooms into the boxed region.