RIM-HASH: RETRIEVABLE INFORMATION MAXIMIZATION BASED MULTIRESOLUTION HASH ENCODING FOR PARAMETER-EFFICIENT SIGNAL FITTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multiresolution hash encodings, popularized by Instant-NGP, provide fast and compact implicit neural representations. However, their layout is data-agnostic, level resolutions follow a fixed geometric schedule, and every spatial location accesses the hash tables identically. This uniform capacity allocation is mismatched to natural signals, whose information varies sharply across space and frequency. Consequently, it wastes capacity on low-information regions while inducing hash collisions in information-dense ones. To improve the efficiency of the multiresolution hash encoding, we recast its layout design as a Retrievable Information Maximization (RIM) problem. Specifically, for a given hash-table size, RIM selects the layout that maximizes the information that remains retrievable after hash collisions. The resulting encoder, RIM-Hash, routes low-information blocks to lightweight fallback features and concentrates hash capacity where information is both present and recoverable. Across benchmarks involving 2D images and 3D signed-distance fields, RIM-Hash consistently achieves a better parameter–fidelity trade-off than strong hash-grid baselines. On Tokyo, RIM-Hash achieves fidelity comparable to the prior state-of-the-art method using only 11.43% of its parameters, while on Girl it improves PSNR by 12.43 dB at a comparable parameter scale. Code is available in the anonymous.4open.science/r/Retrievable-Information-Maximization-FCCD.

1 INTRODUCTION

Implicit neural representations (INRs) parameterize continuous signals with neural networks that map coordinates to signal values, providing a compact and flexible representation for various signals. Early coordinate-based INRs rely primarily on multilayer perceptrons (MLPs) (Sitzmann et al., 2020; Mildenhall et al., 2021; Saragadam et al., 2023; Liu et al., 2024), but such models often exhibit spectral bias: they tend to fit low-frequency structures first while struggling with high-frequency details, making it difficult to faithfully recover fine geometric structures. A major line of work addresses this limitation by augmenting the model with external data structures, such as octrees (Takikawa et al., 2021), K-planes (Fridovich-Keil et al., 2023), and multiresolution hash tables (MRHTs). Among these approaches, Instant Neural Graphics Primitives (Instant-NGP) (Müller et al., 2022) has become a particularly influential design. It stores most of the signal representation in multiscale trainable MRHTs, while a lightweight MLP decodes the interpolated features into signal values, enabling fast convergence and high-fidelity reconstruction with a compact parameter budget. Subsequent work has further advanced Instant-NGP-style representations by strengthening the decoder (Wang et al., 2025; Hu et al., 2025; Lee et al., 2024), achieving higher signal fidelity with only a modest increase in parameter count.

Despite their success, most Instant-NGP variants largely inherit the underlying MRHT design: the level resolutions are typically determined by a data-agnostic geometric schedule, and every spatial location is allowed to access the hash tables in the same manner (Müller et al., 2022). Although this uniform design is simple and computationally efficient, it imposes a homogeneous capacity allocation across spatial locations and frequency bands that poorly matches natural signals (Proakis & Manolakis, 2007; Gonzalez, 2009), whose information distribute nonuniformly across space and frequency. Specifically, smooth regions often require little high-frequency capacity, whereas edges,

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.

textures, tissue boundaries, and geometric details require substantially more fine-scale representation. As a result, a uniform hash-grid design may allocate unnecessary hash table capacity to low-information regions while increasing collisions in information-dense regions.

This capacity-allocation issue becomes more consequential under hash collisions, in which colliding keys share the same trainable entry, limiting their independent key-specific retrievability. To make this collision-induced loss explicit, we adopt a collision–erasure surrogate (Cover et al., 1991). Under this model, each occupied hash entry is treated as preserving, in expectation, the key-specific information of only one of its colliding keys, while the downstream decoder is idealized as information-preserving. Thus, the surrogate attributes information loss exclusively to collision-induced failure of key-specific retrieval. This provides a tractable measure of how much source information remains retrievable after collisions, and leads to our central question: for a given hash-table size, how should the MRHT layout be designed to maximize retrievable information?

We address this question by proposing Retrievable Information Maximization Hash (RIM-Hash), a unified framework that jointly adapts the resolution schedule and block-wise hash access to the spatial–frequency structure of the target signal. Specifically, RIM-Hash first decomposes the signal into block–frequency components, then estimates the information associated with each component, and discounts this information according to the collision-survival factor determined by the effective load factor. The resulting retrievable information guides hash-layout selection and initializes the block-wise gates before end-to-end training.

We evaluate RIM-Hash across a broad range of signal-fitting tasks, spanning 2D images and 3D signed distance fields (SDFs). For 2D image fitting, we use the Kodak Lossless True Color Image Suite (Franzen, 1999) together with three gigapixel images (*Pluto*, *Tokyo* (Martel et al., 2021; Müller et al., 2022), and *Girl with a Pearl Earring* (Müller et al., 2022)), and all four de-identified WSIs from the BioImage Archive. For 3D SDF fitting, we use seven models from the Stanford 3D Scanning Repository (Turk & Levoy, 1994; Curless & Levoy, 1996; Gardner et al., 2003). Across all settings, RIM-Hash consistently improves the parameter fidelity trade-off over strong hash-grid baselines. In particular, it achieves fidelity comparable to previous state-of-the-art methods while using only 11.43% of the parameters, demonstrating that information-guided hash-grid layout design can substantially improve the efficiency of INR-based signal representation. Our main contributions are summarized as follows:

- To the best of our knowledge, we are the first to formulate multiresolution hash-grid design as a *retrievable information maximization* problem that explicitly accounts for information loss induced by hash collisions.
- We introduce *RIM-Hash*, a multiresolution hash encoder that jointly adapts the resolution schedule and block-wise hash-table access to the block–frequency information profile of the target signal.
- We develop an efficient RIM initialization algorithm that maximizes the proposed retrievable information surrogate by alternating between an exact level-wise gate update and a dynamic-programming resolution-schedule update.
- We extensively evaluate RIM-Hash on 2D natural images, gigapixel images, WSIs, and 3D SDFs, demonstrating consistently improved parameter fidelity trade-offs over strong INR baselines.

2 Related Work

Implicit neural representations (INRs) model signals as continuous coordinate-based functions parameterized by neural networks (Sitzmann et al., 2020; Tancik et al., 2020; Sun et al., 2024). Although conventional multilayer perceptron (MLP)-based INRs provide a flexible representation framework, they often suffer from spectral bias, whereby low-frequency components are learned more readily than high-frequency details (Sitzmann et al., 2020). To mitigate this limitation, a growing body of work has replaced purely implicit MLP mappings with explicit or hybrid representations that store learnable features in spatial data structures, query them through interpolation, and decode the resulting features using lightweight neural networks (Liu et al., 2020; Müller et al., 2022). Among these approaches, multiresolution hash-grid encoding has been particularly influential. It stores features in hash tables at multiple spatial resolutions, enabling efficient feature lookup, rapid optimization, and high-fidelity signal reconstruction (Müller et al., 2022). Only a few studies

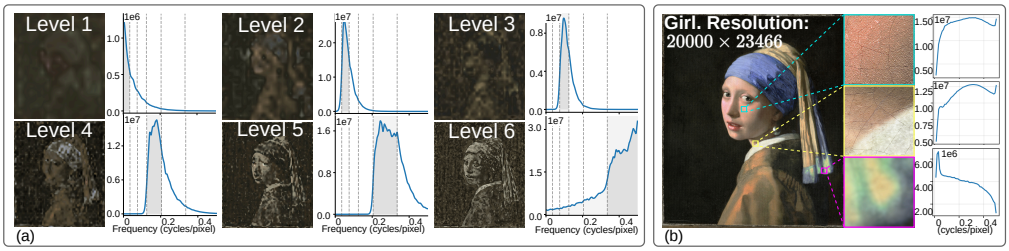

Figure 1: **Panel a**: To examine each hash-grid level, we train an Instant-NGP model with a 6-level hash grid and isolate one level at a time by retaining its features while replacing all other levels' hash-table entries with their mean feature value. The resulting reconstructions suggest spectral specialization, different levels capture distinct frequency components. This is further reflected by the empirical band-information $\mathfrak{I}(f)$ defined in Equation (7), whose frequency profile aligns with the sampling frequencies of the feature-grid levels, indicated by vertical dashed lines. **Panel b**: Different spatial blocks of the same image exhibit distinct band-information profiles, motivating RIM-Hash to allocate hash access through block-wise gates rather than uniformly across all blocks.

directly refine hash-grid design, R-MHE (Dai & Fan, 2026) rotates grid levels to reduce anisotropy and collision artifacts.

Other studies have investigated alternative spatial representations, including feature planes, sparse octrees, tensor decompositions, and hyperplane-based structures, to further improve computational efficiency, memory usage, and scalability (Reiser et al., 2023; Yang, 2023; Yu et al., 2021; Chan et al., 2022; Fridovich-Keil et al., 2023; Shue et al., 2023; Shi et al., 2024).

Despite their empirical success, existing hash-grid methods typically use data-agnostic resolution schedules (Müller et al., 2022). Consequently, capacity may be allocated to frequency bands or spatial regions with limited information, while more informative regions suffer from hash collisions. From an information-allocation perspective, we model collision-induced information degradation and, for a given hash-table size, select hash-grid resolutions to maximize retrievable information. A more comprehensive discussion of related work is provided in Appendix A.

3 Notation and Preliminaries

Notation. Scalars are denoted by non-bold letters (e.g., a, β, and L), and sets by calligraphic letters (e.g., $\mathcal{V}$ and $\mathcal{K}$). Their dimensions are inferred from context. For a positive integer n, we write $[n] \triangleq \{1, \dots, n\}$. We use $\mathbf{y}$ to denote a fixed target signal and associate it with an empirical source random variable Y, whose distribution is induced by uniformly sampling signal values from $\mathbf{y}$. All images considered in this work are 8-bit, with intensity alphabet $\mathcal{V} \triangleq \{0, 1, \dots, 255\}$. Before computing empirical distributions, intensity-valued signals are quantized using an 8-bit quantizer $Q_8(\cdot)$, while real-valued residual signals are quantized using a signed 9-bit quantizer $Q_\Delta(\cdot)$, with intensity alphabet $\mathcal{V}_\Delta \triangleq \{-255, \dots, 255\}$. For multi-channel signals, empirical entropy is computed independently for each channel and summed over channels.

We divide the spatial samples of $\mathbf{y}$ into B non-overlapping blocks of equal size, denoted by $\{\mathbf{y}_b\}_{b=1}^{B}$. For clarity, we state the notation for 2D images ($\boldsymbol{x} \in [0,1]^2$), which is the setting of our motivating visualizations and most experiments; the formulation extends directly to 3D signals.

Instant Neural Graphics Primitives. Instant-NGP represents a signal by combining a multiresolution hash-grid encoder with an MLP decoder (Müller et al., 2022). Given a coordinate $\mathbf{x} \in [0,1]^2$, the corresponding signal value is represented as

$$\mathfrak{f}_\theta(\mathbf{x}) = \mathfrak{g}_\theta(\phi(\mathbf{x})), \qquad \phi(\mathbf{x}) = [\phi_1(\mathbf{x}); \dots; \phi_L(\mathbf{x})], \tag{1}$$

where $\phi_\ell(\mathbf{x})$ is the bilinearly interpolated feature from level ℓ of the MRHT. Level ℓ uses a grid of resolution N_ℓ and a hash table $\boldsymbol{E}_\ell \in \mathbb{R}^{T_\ell \times F}$. Instant-NGP chooses the per-level resolutions using a data-agnostic geometric progression, $N_\ell = \lfloor N_{\min} b^{\ell-1} \rfloor$ for $\ell \in [L]$ and a predefined base $b > 1$. This yields grid spacing $\Delta_\ell \propto 1/N_\ell$, so higher levels can represent finer spatial variation. Therefore, concatenating features across levels provides a multiscale representation: low-resolution levels primarily capture coarse, low-frequency structure, while high-resolution levels encode finer, high-frequency details. As illustrated in the left panel of Figure 1, decoding with individual levels

suggests that higher-resolution grids recover high-frequency components, whereas lower-resolution grids represent the coarse structure of the signal.

4 METHODOLOGY

As discussed previously, MRHTs can be interpreted as a sampling-and-interpolation system (Gonzalez, 2009; Chen et al., 2024). At level ℓ, a grid with resolution N_ℓ per axis induces an effective sampling interval of $\Delta_\ell^{(x)} = w/N_\ell$ and $\Delta_\ell^{(y)} = h/N_\ell$ pixels along the horizontal and vertical directions, respectively. Under this interpretation, the corresponding Nyquist frequencies (in cycles per pixel) are

$$f_{\mathrm{Nyq},\ell}^{(x)} = \frac{N_\ell}{2w}, \qquad f_{\mathrm{Nyq},\ell}^{(y)} = \frac{N_\ell}{2h}. \tag{2}$$

Frequencies beyond these limits cannot be represented without aliasing at level ℓ and must therefore be captured by finer-resolution levels or by the lightweight MLP decoder. To quantify the information associated with a frequency f, we measure the band entropy (Shannon, 1948) between two adjacent spectral cutoffs. We consider a monotone candidate set of grid resolutions $\{N_{(k)}\}_{k=1}^{K}$ with

$$N_{(1)} < \cdots < N_{(K)} \triangleq \max(w, h), \tag{3}$$

For each candidate resolution $N_{(k)}$, let $\mathbf{y}_{\le f_{(k)}}$ denote the low-pass filtered signal obtained using the corresponding cutoff induced by $N_{(k)}$. We set $\mathbf{y}_{\le f_{(0)}} \triangleq \mathbf{0}$ and define the k-th frequency-band signal as

$$\mathbf{y}_{(k)} \triangleq \mathbf{y}_{\le f_{(k)}} - \mathbf{y}_{\le f_{(k-1)}}, \qquad k \in [K]. \tag{4}$$

Uniformly sample a spatial location from the signal, and let $B_0 \sim \mathrm{Unif}([B])$ denote its block index and $Y_{(k)}$ the corresponding quantized value of the k-th band signal $Q_\Delta(\mathbf{y}^{(k)})$. We write

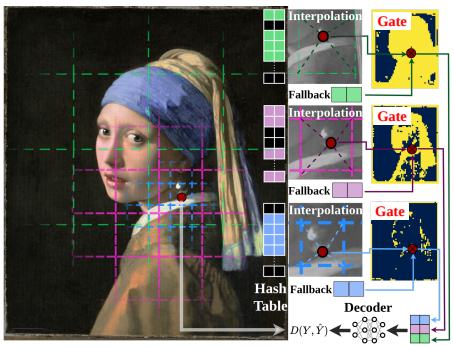

Figure 2: Overview of RIM-Hash. Given a target signal, RIM first selects the resolution schedule and initializes the block-wise soft gates by maximizing retrievable information. During subsequent signal fitting, the selected resolutions remain fixed, while the gate logits, hash-table embeddings, fallback vectors, and decoder are jointly optimized using a standard reconstruction loss.

$$\Pr(Y_{(k)} = v \mid B_0 = b) = \hat{p}_{b,k}(v), \qquad \Pr(Y_{(k)} = v) = \hat{p}_k(v) = \frac{1}{B}\sum_{b=1}^{B} \hat{p}_{b,k}(v). \tag{5}$$

For each block b, let $Y_{b,k}$ denote the block-restricted empirical source random variable, with

$$\Pr(Y_{b,k} = v) = \hat{p}_{b,k}(v). \tag{6}$$

Then, we define its empirical band-information as

$$\mathfrak{I}(f_{(k)}) \triangleq N_{(k)}^2 H(Y_{(k)}) = -N_{(k)}^2 \sum_{v \in \mathcal{V}_\Delta} \hat{p}_k(v) \log_2 \hat{p}_k(v). \tag{7}$$

where the multiplicative factor $N_{(k)}^2$ accounts for the number of spatial samples represented at resolution $N_{(k)}$. For a given L-level resolution schedule, let $\{\mathcal{F}_\ell\}_{\ell=1}^{L}$ denote a contiguous partition of $\{f_{(1)}, \ldots, f_{(K)}\}$, where $\mathcal{F}_\ell$ contains the frequency bands assigned to level ℓ.

Natural signals are spatially heterogeneous, and the same frequency band may contain substantially different amounts of information in different regions. We therefore define a block-wise counterpart of equation 7. The empirical information of frequency band $f_{(k)}$ within block b is defined as

$$\mathcal{J}_b(\mathcal{F}_\ell) \triangleq \sum_{f_{(k)} \in \mathcal{F}_\ell} \mathfrak{I}_b(f_{(k)}), \quad \mathfrak{I}_b(f_{(k)}) \triangleq -\frac{N_{(k)}^2}{B} \sum_{v \in \mathcal{V}_\Delta} \hat{p}_{b,k}(v) \log_2 \hat{p}_{b,k}(v). \tag{8}$$

Figure 1 shows that hash-grid levels capture distinct spectral components, while spatial blocks exhibit heterogeneous band-information profiles, directly motivating the design of RIM-Hash.

4.1 RIM-HASH AND RETRIEVABLE INFORMATION OBJECTIVE

Motivated by the observation in Figure 1 that different spatial regions exhibit heterogeneous band-information profiles while different hash-grid levels predominantly capture distinct frequency components, we introduce *RIM-Hash* (Figure 2). RIM-Hash jointly adapts the resolution schedule and the spatial access to each hash-grid level. Specifically, for each level ℓ and block b, RIM-Hash assigns a soft gate $\mathbf{g}_{\ell b} \in [0, 1]$, which provides a differentiable relaxation of the routing decision between the hash feature and a trainable fallback feature. The feature supplied to the decoder is

$$\widetilde{\phi}_\ell(\mathbf{x}) = \mathbf{g}_{\ell,b(\mathbf{x})}\phi_\ell(\mathbf{x}) + \big(1 - \mathbf{g}_{\ell,b(\mathbf{x})}\big)\mathbf{v}_\ell, \tag{9}$$

where $\mathbf{v}_\ell$ is a shared trainable fallback feature. Theorem 1 shows that an optimal gate is binary except for at most one block per level. Let T_ℓ denote the size of the level-ℓ hash table and $\kappa_\ell \triangleq N_\ell^2/B$ the effective number of hash keys contributed by a fully active block. The corresponding effective load factor is

$$\alpha_\ell(\mathbf{g}_\ell) = \frac{\kappa_\ell}{T_\ell}\sum_{b=1}^{B}\mathbf{g}_{\ell b}. \tag{10}$$

Increasing the total gate mass routes more block–band information through level ℓ, but also increases its hash-table occupancy and hence collision-induced information loss. To characterize this allocation–collision trade-off, fix a block b, level ℓ, and a frequency band $f_{(k)} \in \mathcal{F}_\ell$, and let

$$Y_{b,k} \triangleq Y_{(k)} \mid B_0 = b,$$

which represents a randomly sampled quantized signal value from the k-th band within block b. Under the collision–erasure surrogate, we interpret $\mathbf{g}_{\ell b}$ as the probability that a key from block b is routed through level ℓ, while a routed key survives collision with probability $q(\alpha_\ell)$. We summarize these two effects by a binary retrieval variable $R_{b\ell} \in \{0, 1\}$ such that

$$\Pr(R_{b\ell} = 1) = \mathbf{g}_{\ell b}q(\alpha_\ell). \tag{11}$$

Under the surrogate, retrieval is independent of the signal value, so $R_{b\ell}$ is independent of $Y_{b,k}$. To isolate collision-induced information loss from any additional loss introduced by the downstream decoder, we consider an ideal, information-preserving decoder that can recover the exact signal if there is no collision and full routing to the hash table. In contrast, a failed retrieval provides no key-specific information. Accordingly, we define $\widehat{Y}_{b,\ell,k}$ as the ideal post-collision signal. It equals the routed source value when $R_{b\ell} = 1$ and a fixed, distinguishable erasure state otherwise. The following proposition characterizes the source information retained under this collision–erasure surrogate.

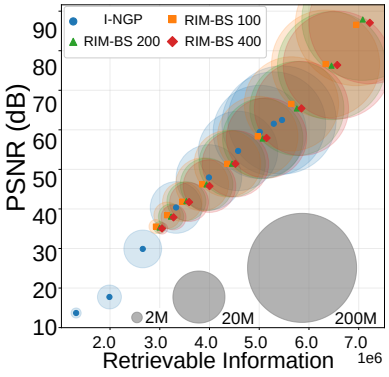

Figure 3: On the *Pluto*, reconstruction PSNR scales approximately linearly with retrievable information across different hash-table and block sizes. Marker size denotes model size, and BS denotes the block size used by RIM-Hash.

Proposition 1 (Retrievable information under the collision–erasure surrogate)**.** *For any block b, level ℓ, and frequency band $f_{(k)} \in \mathcal{F}_\ell$,*

$$I(Y_{b,k}; \widehat{Y}_{b,\ell,k}) = H(\widehat{Y}_{b,\ell,k} \mid R_{b\ell}). \tag{12}$$

The proof is provided in Appendix B. Proposition 1 suggests that, under the collision–erasure surrogate, the retained mutual information equals the conditional entropy of the ideal post-collision signal. Specifically, $R_{b\ell} = 1$ preserves the source value, whereas $R_{b\ell} = 0$ produces a fixed erasure symbol. Since $R_{b\ell}$ is independent of $Y_{b,k}$, and $\Pr(R_{b\ell} = 1) = \mathbf{g}_{\ell b}q(\alpha_\ell)$,

$$H(\widehat{Y}_{b,\ell,k} \mid R_{b\ell}) = \mathbf{g}_{\ell b}q(\alpha_\ell)H(Y_{b,k}). \tag{13}$$

Together with Proposition 1, this yields

$$I(Y_{b,k}; \widehat{Y}_{b,\ell,k}) = \mathbf{g}_{\ell b}q(\alpha_\ell)H(Y_{b,k}), \qquad q(\alpha) \triangleq \frac{1 - e^{-\alpha}}{\alpha}, \quad q(0) = 1. \tag{14}$$

Under the standard Poisson occupancy model, $q(\alpha)$ is the survival probability of a routed key; its derivation is provided in Appendix C. Hence, $q(\alpha_\ell)$ discounts the information routed through level ℓ according to the expected loss of key-specific retrievability caused by hash collisions.

Summing Equation (14) over the frequency bands assigned to level ℓ and using the block-wise band-information defined in Equation (8), we obtain the level-wise retrievable information

$$J_\ell(\mathbf{g}_\ell; \mathcal{F}_\ell) \triangleq q(\alpha_\ell(\mathbf{g}_\ell)) \sum_{b=1}^{B} \mathbf{g}_{\ell b} \mathcal{J}_b(\mathcal{F}_\ell). \tag{15}$$

To evaluate whether Equation (15) is predictive of the final fitted representation, we empirically examine the relationship between total retrievable information and reconstruction fidelity in Figure 3. Across different hash-table and block sizes on *Pluto*, retrievable information exhibits a strong approximately linear relationship with reconstruction PSNR. That is, for a fixed hash-table budget, RIM selects the frequency partition and block-wise gates by solving

$$\max_{\{\mathcal{F}_\ell\}_{\ell=1}^{L}, \{\mathbf{g}_\ell\}_{\ell=1}^{L}} \sum_{\ell=1}^{L} J_\ell(\mathbf{g}_\ell; \mathcal{F}_\ell) \qquad \text{s.t.} \quad 0 \le \mathbf{g}_{\ell b} \le 1, \quad \forall \ell, b. \tag{16}$$

4.2 Algorithm for Solving the Optimization in Equation (16)

The objective in Equation (16) jointly selects the contiguous frequency partition and the block-wise gates. Direct optimization is not amenable to an efficient closed-form solution, because the selected frequency bands determine the information assigned to each level, while the gates determine the effective load factor and, consequently, the collision-survival factor. We therefore use alternating coordinate ascent: with the partition fixed, we update the gates independently for each level; with the gates fixed, we update the ordered partition exactly by dynamic programming.

Gate update. The gate update solves

$$\max_{0 \le \mathbf{g}_{\ell b} \le 1} q\left(\frac{\kappa_\ell}{T_\ell} \sum_{b=1}^{B} \mathbf{g}_{\ell b}\right) \sum_{b=1}^{B} \mathbf{g}_{\ell b}\, \mathcal{J}_b(\mathcal{F}_\ell), \quad \forall \ell \in [L] \tag{17}$$

Theorem 1 characterizes a global maximizer of Equation (17) when the frequency partition is fixed.

Theorem 1 (Globally optimal level-wise gate)**.** *Fix a level ℓ and a frequency partition $(\mathcal{F}_\ell)$, and let π order the blocks by nonincreasing information, $\mathcal{J}_{\pi(1)}(\mathcal{F}_\ell) \ge \cdots \ge \mathcal{J}_{\pi(B)}(\mathcal{F}_\ell)$. Then there exists an optimal gate mass $G_\ell^\star \in [0, B]$ such that, with $m^\star = \lfloor G_\ell^\star \rfloor$, the gate*

$$\mathbf{g}^\star_{\ell,\pi(i)} = \begin{cases} 1, & i \le m^\star, \\ G_\ell^\star - m^\star, & i = m^\star + 1, \\ 0, & i > m^\star + 1, \end{cases} \qquad i \in [B]. \tag{18}$$

The optimal gate is therefore binary except for at most one block, the $m^\star$ most informative blocks are fully activated, the next block receives the fractional remainder $G_\ell^\star - m^\star$, and all remaining blocks are deactivated. The resulting scalar objective is globally unimodal. Hence, $G_\ell^\star$ is located by scanning the B unit intervals of $[0, B]$; if the maximizer lies in the interior of an interval, its unique stationary point is computed to any prescribed accuracy by bisection. The proof and search procedure are provided in Appendix D, with gate visualizations in Appendix J.

Resolution schedule update. With the gates fixed, the remaining subproblem is to place the resolution boundaries over the ordered candidate frequency bands. Recall that a schedule is represented by cut indices $0 = k_0 < k_1 < \cdots < k_L = K$, where level ℓ receives the segment $\mathcal{F}_\ell$ and uses resolution $N_{(k_\ell)}$. For any two indices $0 \le u < e \le K$, define the candidate segment

$$\mathcal{F}_{(u:e)} \triangleq \{f_{(u+1)}, \ldots, f_{(e)}\}. \tag{19}$$

If this segment is assigned to level ℓ, then the level is instantiated at the endpoint resolution $N_{(e)}$, which is the smallest candidate resolution whose Nyquist limit covers all bands in $\mathcal{F}_{(u:e)}$. Extending the definition of Equation (15) to a candidate segment, the corresponding level-wise retrievable information under the fixed gates $\mathbf{g}_\ell$ is

Table 1: Image fitting results on Kodak. All methods use the same training iterations and comparable parameter counts. Our method achieves the best PSNR and SSIM with fewer parameters. **Bold** and underlined denote the best and second-best results.

Method	WIRE CVPR 2023	SIREN NeurIPS 2020	SCONE AAAI 2024	I-NGP TOG 2022	NFFB CVPR 2023	NeuRBF ICCV 2023	MetricGrids CVPR 2025	R-MHE ICLR 2026	Ours
Params (K)	373	207	207	206	208	207	207	207	197
PSNR ↑	37.59	38.70	39.72	37.06	38.77	38.70	<u>40.01</u>	38.01	**41.08**
SSIM ↑	0.9596	0.9512	<u>0.9662</u>	0.9386	0.9537	0.9488	0.9568	0.9426	**0.9697**

Table 2: Quantitative results on Kodak, comparing our method with Gaussian primitive baselines.

Method	I-NGP TOG 2022	NeuRBF ICCV 2023	Gaussian Splatting TOG 2023	GaussianImage ECCV 2024	GaussianImage++ AAAI 2026		SGI CVPR 2026		Ours		
Params (K)	300	337	3540	560	379	3023	1001	2972	364	427	528
PSNR ↑	43.88	43.78	43.69	44.08	45.55	55.25	38.41	47.62	46.64	50.80	58.08
MS-SSIM ↑	0.9976	0.9964	0.9991	0.9874	0.9836	0.9987	0.9702	0.9958	0.9898	0.9997	0.9999

$$J_\ell(\mathbf{g}_\ell; \mathcal{F}_{(u:e)}) = q\left(\frac{1}{T_\ell}\sum_{b=1}^{B}\kappa_{\ell,e}\mathbf{g}_{\ell b}\right)\sum_{b=1}^{B}\mathbf{g}_{\ell b}J_b(\mathcal{F}_{(u:e)}). \tag{20}$$

For fixed gates, the total objective decomposes additively across hash levels, while the candidate frequency bands are partitioned into contiguous intervals. The resulting problem can therefore be solved exactly using dynamic programming (Cormen et al., 2022). The proof of optimality and implementation details are provided in Appendix E. We alternate between the gate update and the resolution-schedule update until the retrievable-information initialization objective converges [1]. As shown in Appendix F, RIM converges within only a few iterations and incurs negligible processing overhead. Even in the worst case, optimization takes only 1.6 s for the gigapixel Girl image and 1.2 s for the Pluto image. The target-dependent preprocessing for low-pass construction and block-wise band-information estimation is performed only once, requiring approximately 16 s on Girl and 10 s on Pluto. The resulting partition determines the fixed per-level resolutions, while the optimized soft gates initialize the trainable gate logits, which map to gate values through a sigmoid function. We then jointly optimize the gate logits, hash-table embeddings, fallback features, and decoder using the reconstruction loss, while keeping the selected resolutions fixed. In the next section, we evaluate the efficiency of RIM-Hash through a comprehensive set of experiments.

5 Experiments

We evaluate the proposed method on four benchmarks: the Kodak dataset (Franzen, 1999), three gigapixel natural images, ultra-high-resolution whole-slide images (WSIs), and the Stanford 3D Scanning Repository (Turk & Levoy, 1994; Curless & Levoy, 1996; Gardner et al., 2003). Unless otherwise specified, all models are trained with the Adam optimizer (Kingma & Ba, 2014) and cosine annealing learning-rate decay. We use an ℓ_2 reconstruction loss for 2D fitting and MAPE for SDF fitting. Across all experiments, we use the same multiresolution hash-grid architecture with feature dimension fixed to 2 at every level and the same number of levels for all methods. For a given hash-table size, all variants use the same architecture; different model scales are obtained by varying the hash-table size. For all experiments, we use anti-aliased resizing as the low-pass filter. For Kodak, we set $K = 30$ with uniform spacing in the frequency domain; for gigapixel images and WSIs, we set $K = 200$. Additional implementation details and wall-clock cost are provided in the Appendix G. When reporting the number of trainable parameters for RIM, we include the trainable gate logits, fallback vectors, hash-table embeddings, and decoder parameters.

Kodak Dataset. We first evaluate on the Kodak dataset (Franzen, 1999), which contains 24 images at a resolution of 768×512, using PSNR and SSIM (Wang et al., 2004). As shown in Table 1, RIM-Hash achieves the best average PSNR and SSIM, exceeding MetricGrids by 1.07 dB with fewer parameters. It also ranks first on 22/24 images in PSNR and 21/24 images in SSIM (Figure 12, Appendix I), demonstrating gains on the vast majority of images. We also compare against Gaussian-primitive-based representations (Kerbl et al., 2023; Zhang et al., 2024; Li et al., 2026; Pan et al., 2026), RIM-Hash reaches 58.08 dB with 528K parameters, only 17.46% of GaussianImage++ (Table 2).

[1] Each alternating update is non-decreasing; since retrievable information is bounded, the objective values converge. Any fixed point is coordinate-wise optimal, though global optimality is not guaranteed.

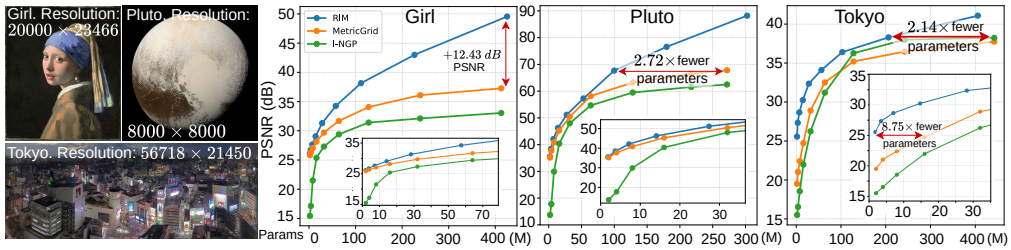

Figure 4: Gigapixel Image Fitting Performance. **Left:** Gigapixel images (Tokyo, Pluto, and Girl) **Right:** Quantitative comparison of reconstruction quality (PSNR) vs. the number of trainable parameters (M). Across all three images, our method consistently outperforms Instant-NGP (I-NGP) and MetricGrids, achieving a superior Pareto frontier and higher fidelity at every parameter scale.

Table 3: Numerical comparison on WSI fitting. We report PSNR (dB) and the number of trainable parameters. RIM-Hash achieves a stronger parameter–fidelity trade-off, showing that retrievable-information maximization yields a more efficient WSI representation.

Method	I-NGP TOG 2022			NeuRBF ICCV 2023	CINR MICCAI 2024	MetricGrids CVPR 2025			Ours		
Params (M)	95.45	234.45	402.60	161.48	239.63	95.46	247.68	390.01	10.16	92.85	237.35
PSNR ↑	40.97	41.95	42.06	40.30	42.07	41.06	42.97	<u>43.03</u>	41.07	42.63	**44.63**

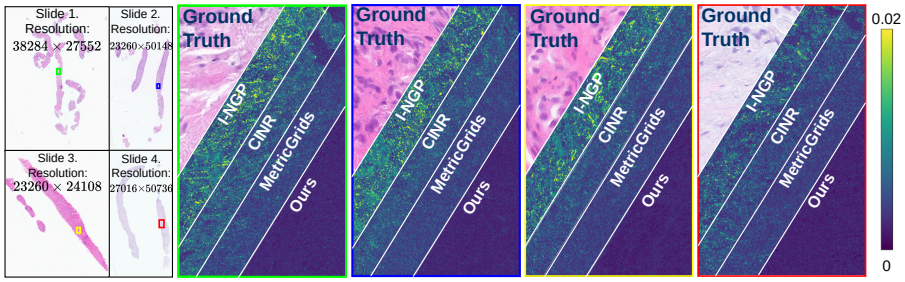

Figure 5: Qualitative reconstruction comparison on four WSIs using approximately 234M trainable parameters. **Left**: thumbnails of the four WSIs. **Right**: magnified regions showing the ground truth and residual maps of I-NGP, CINR, MetricGrids, and our method. Across all four WSIs, our method achieves the smallest reconstruction distortion, with consistently lower residuals.

Gigapixel Natural Images. To evaluate scalability, we conduct experiments on three gigapixel natural images: *Pluto*, *Tokyo*, and *Girl With a Pearl Earring* (Girl). These images require representing fine spatial details over extremely large resolutions, making them substantially more challenging than standard-resolution image fitting benchmarks. We compare against Instant-NGP (Müller et al., 2022), NeuRBF (Chen et al., 2023), and MetricGrids (Wang et al., 2025) under various parameter budgets. Figure 4 illustrates the PSNR–parameter trade-off across the three images. Our method consistently achieves a stronger Pareto frontier than the baselines, indicating that the proposed RIM-Hash improves parameter efficiency at scale. The gains are especially pronounced in the high-fidelity regime, where uniformly allocated hash grids become increasingly inefficient due to uneven frequency and spatial band-information across the image. Detailed numerical results are provided in Appendix H. On the Tokyo image, our method matches the low-budget MetricGrids result using only 11.43% of its parameters. On the Girl image, at comparable model scale, our method reaches 49.49 dB, improving over MetricGrids by 12.43 dB. These results indicate that adaptive resolution allocation becomes increasingly effective as image scale and frequency complexity increase.

Whole Slide Images. Whole-slide images (WSIs) are gigapixel histopathology images used in clinical diagnosis. As shown in Table 3, RIM-Hash provides a consistently stronger parameter–fidelity trade-off than the compared hash-grid baselines. At the smallest model scale, RIM-Hash achieves 41.07 dB with 10.16M trainable parameters, matching the approximately 95M-parameter I-NGP and MetricGrids models, which obtain 40.97 dB and 41.06 dB, respectively. At the largest evaluated scale, RIM-Hash reaches 44.63 dB with 237.35M parameters, outperforming the 390.01M-parameter MetricGrids model by 1.60 dB while using 39.1% fewer parameters. These results demonstrate that RIM-Hash improves the parameter–fidelity trade-off for spatially heterogeneous WSI signals.

3D Signed Distance Fields. We further evaluate whether the parameter-efficiency gains of RIM-Hash extend from 2D signals to 3D geometry using seven SDFs from the Stanford 3D Scanning

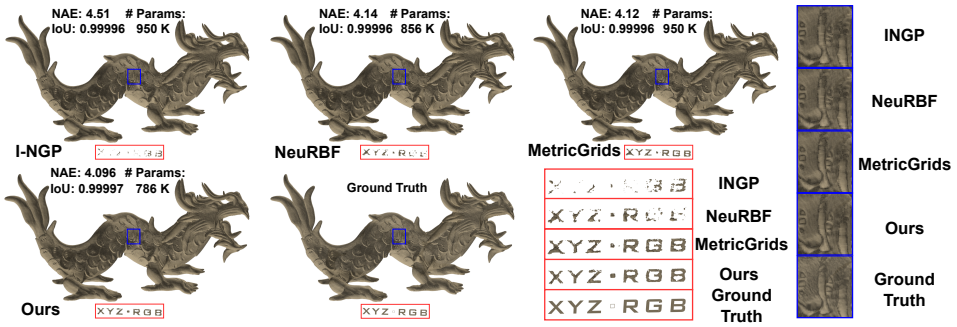

Figure 6: RIM-Hash uses 786K trainable parameters, compared with 856K for NeuRBF and 950K for I-NGP and MetricGrids, while maintaining comparable geometric fidelity in IoU and NAE. The zoomed-in regions further show that this parameter reduction does not introduce noticeable degradation in fine surface and texture details. Best viewed digitally with zoom.

Repository. SDFs provide a natural setting for this evaluation because their informative high-frequency variations are concentrated near object surfaces and high-curvature regions, where selective capacity allocation can reduce redundant hash-grid parameters. As shown in Table 4, RIM-Hash uses only 786K trainable parameters, 17.3% fewer than I-NGP and MetricGrids (950K), while maintaining comparable IoU and NAE. This quantitative result is consistent with the qualitative comparison on the XYZ Dragon in Figure 6, where the reduced-parameter RIM-Hash representation retains fine surface structures and local geometric details comparable to those of the larger baselines. These results demonstrate that RIM-guided capacity allocation reduces redundant representation capacity without sacrificing geometric fidelity, extending the parameter-efficiency benefit of RIM-Hash from 2D images to 3D implicit geometry.

Table 4: 3D SDF fitting. RIM-Hash reduces the trainable parameter count by 17.3% relative to I-NGP and MetricGrids while preserving comparable geometric fidelity in IoU and NAE.

Method	IOU ↑	NAE ↓	Params ↓
NGLOD CVPR 2021	0.99991	4.21	10.15M
I-NGP TOG 2022	0.99991	3.42	950K
NeuRBF ICCV 2023	0.99988	3.20	856K
MetricGrids CVPR 2025	0.99991	3.09	950K
Ours	0.99992	3.02	**786K**

6 Ablation Study

We conduct a component-wise ablation to isolate the effect of the two main allocation mechanisms in RIM-Hash: the adaptive resolution schedule and the block-wise gates. All variants use the same decoder, number of levels, feature dimension, and training protocol; only the specified component is removed. Figure 7 reports the PSNR–parameter trade-off on the Pluto image. The gates provide the largest gain in the small-hash-table regime, where collisions are severe, and capacity must be selectively assigned to informative spatial blocks. As the hash-table budget increases and collisions become less dominant, the adaptive resolution schedule becomes more critical, since the main limitation shifts from collision avoidance to allocating resolution across frequency bands. The full RIM-Hash model consistently gives the best Pareto frontier, showing that the two mechanisms are complementary. We defer the full ablation suite, including the collision-survival factor, fallback vector, block size, and candidate resolutions, to Appendix L.

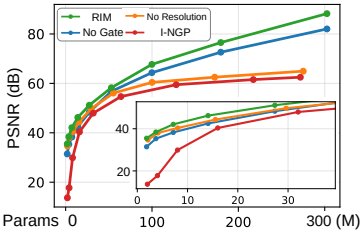

Figure 7: Controlled RIM ablation on Pluto. Geometric Schedule replaces only the RIM-selected resolutions; Uniform Gates replaces only the RIM gate allocation. All other architecture and training settings are identical.

7 Conclusion

We presented RIM-Hash, a signal-adaptive multiresolution hash encoding framework for efficient implicit neural representations. RIM-Hash estimates block-wise band-information, models collision-induced information loss through the collision-survival factor, selects the resolution schedule, and initializes trainable block-wise hash-access gates. Experiments on natural images, gigapixel images, whole-slide images, and 3D SDFs show that RIM-Hash improves the parameter–fidelity trade-off over strong INR baselines, indicating that retrievable information is an effective criterion for efficient hash-grid design. We discuss limitations and future work in the Appendix M due to space constraints.

AI USE STATEMENT

Generative AI tools, including ChatGPT, were used for language editing and to assist with code implementation. All AI-assisted code was reviewed, edited, and tested for correctness by at least one author. The authors take full responsibility for the final manuscript and implementation.

ETHICS STATEMENT

All WSI datasets used in this work are publicly available and were obtained from open-access websites. The usage of these datasets strictly follows the terms and conditions set by the dataset providers and adheres to established academic and research community standards. No personally identifiable information or sensitive patient data is involved.

REPRODUCIBILITY STATEMENT

We have taken steps to ensure our results are reproducible. All model and algorithmic details, training procedures, hyperparameters, evaluation protocols, and metrics are specified in the main text. The appendix provides complete proofs, implementation notes, ablations, and additional qualitative results. An anonymized GitHub repository (anonymous.4open.science/r/Retrievable-Information-Maximization-FCCD) contains the source code and configuration files. All datasets used in our experiments are publicly available.

REFERENCES

Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16123–16133, 2022.

Yihang Chen, Qianyi Wu, Mehrtash Harandi, and Jianfei Cai. How far can we compress instant-ngp-based nerf? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20321–20330, 2024.

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

Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.

Thomas M Cover, Joy A Thomas, and John Kieffer. *Elements of information theory*, volume 2. wiley New York, 1991.

Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pp. 303–312, New York, NY, USA, 1996. Association for Computing Machinery. ISBN 0897917464. doi: 10.1145/237170.237269. URL https://doi.org/10.1145/237170.237269.

Tianxiang Dai and Jonathan Fan. Characterizing and optimizing the spatial kernel of multi resolution hash encodings. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=q05hC1Pzkr.

Rich Franzen. Kodak lossless true color image suite, 1999. URL https://r0k.us/graphics/kodak/.

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

Andrew Gardner, Chris Tchou, Tim Hawkins, and Paul Debevec. Linear light source reflectometry. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pp. 749–758, New York, NY, USA, 2003. Association for Computing Machinery. ISBN 1581137095. doi: 10.1145/1201775.882342. URL https://doi.org/10.1145/1201775.882342.

R.C. Gonzalez. *Digital Image Processing*. Pearson Education, 2009. ISBN 9788131726952. URL https://books.google.ca/books?id=a62xQ2r_f8wC.

Yubin Hu, Xiaoyang Guo, Yang Xiao, Jingwei Huang, and Yong-Jin Liu. Ngp-rt: Fusing multi-level hash features with lightweight attention for real-time novel view synthesis. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 148–165, Cham, 2025. Springer Nature Switzerland.

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

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

Zoe Landgraf, Alexander Sorkine Hornung, and Ricardo Silveira Cabral. Pins: progressive implicit networks for multi-scale neural representations. *arXiv preprint arXiv:2202.04713*, 2022.

DongEon Lee, Chunsu Park, SeonYeong Lee, SiYeoul Lee, and MinWoo Kim. Convolutional Implicit Neural Representation of pathology whole-slide images . In *proceedings of Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*, volume LNCS 15007. Springer Nature Switzerland, October 2024.

Jason Chun Lok Li, Chang Liu, Binxiao Huang, and Ngai Wong. Learning spatially collaged fourier bases for implicit neural representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13492–13499, 2024.

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.

Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020.

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*, pp. 2713–2722, 2024.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacic, Thomas Y. Hou, and Max Tegmark. KAN: Kolmogorov–arnold networks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Ozo7qJ5vZi.

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

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, 2021.

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

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)*, pp. 12162–12172, June 2026.

J.G. Proakis and D.G. Manolakis. *Digital Signal Processing*. Prentice Hall international editions. Pearson Prentice Hall, 2007. ISBN 9780131873742. URL https://books.google.ca/books?id=H_5SAAAAMAAJ.

Sameera Ramasinghe and Simon Lucey. Beyond periodicity: Towards a unifying framework for activations in coordinate-mlps. In *European Conference on Computer Vision*, pp. 142–158. Springer, 2022.

Christian Reiser, Rick Szeliski, Dor Verbin, Pratul Srinivasan, Ben Mildenhall, Andreas Geiger, Jon Barron, and Peter Hedman. Merf: Memory-efficient radiance fields for real-time view synthesis in unbounded scenes. *ACM Transactions on Graphics (ToG)*, 42(4):1–12, 2023.

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*, pp. 18507–18516, 2023.

Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Kexuan Shi, Xingyu Zhou, and Shuhang Gu. Improved implicit neural representation with fourier reparameterized training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25985–25994, 2024.

J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20875–20886, 2023.

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

Jia-Mu Sun, Tong Wu, and Lin Gao. Recent advances in implicit representation-based 3d shape generation. *Visual Intelligence*, 2(1):9, 2024.

Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3D shapes. 2021.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.

Greg Turk and Marc Levoy. Zippered polygon meshes from range images. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '94, pp. 311–318, New York, NY, USA, 1994. Association for Computing Machinery. ISBN 0897916670. doi: 10.1145/192161.192241. URL https://doi.org/10.1145/192161.192241.

Peng-Shuai Wang, Yang Liu, Yu-Qi Yang, and Xin Tong. Spline positional encoding for learning 3d implicit signed distance fields. *arXiv preprint arXiv:2106.01553*, 2021.

Shu Wang, Yanbo Gao, Shuai Li, Chong Lv, Xun Cai, Chuankun Li, Hui Yuan, and Jinglin Zhang. Metricgrids: Arbitrary nonlinear approximation with elementary metric grids based implicit neural representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21381–21391, 2025.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

Zhijie Wu, Yuhe Jin, and Kwang Moo Yi. Neural fourier filter bank. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14153–14163, 2023.

Runzhao Yang. Tinc: Tree-structured implicit neural compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18517–18526, 2023.

Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5752–5761, 2021.

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

A COMPREHENSIVE RELATED WORK

Coordinate-based implicit neural representations. Implicit neural representations parameterize continuous signals by mapping coordinates to signal values using neural networks (Sitzmann et al., 2020; Tancik et al., 2020; Sun et al., 2024). Early INR models commonly rely on multilayer perceptrons, but such networks tend to learn low-frequency components before high-frequency details, a phenomenon often described as spectral bias (Sitzmann et al., 2020). A large body of work improves high-frequency fitting by modifying the input representation or the network architecture. Fourier feature mappings encode coordinates with sinusoidal bases to improve the learning of high-frequency functions (Tancik et al., 2020), while periodic activations and related nonlinearities further improve the representation of oscillatory signals (Sitzmann et al., 2020; Liu et al., 2024; Ramasinghe & Lucey, 2022; Saragadam et al., 2023). Other approaches modify coordinate parameterizations or increase model capacity through alternative network families (Wang et al., 2021; Landgraf et al., 2022; Liu et al., 2025). These methods improve the expressiveness and optimization behavior of coordinate-based INRs, but they do not directly address how explicit grid capacity should be allocated when the representation is stored in hash tables.

Grid-based and hybrid neural representations. Learnable feature grids provide an efficient alternative to purely implicit MLP representations. In these methods, a coordinate queries nearby grid entries by interpolation, and the resulting feature vector is passed to a lightweight decoder (Liu et al., 2020; Müller et al., 2022). Such hybrid representations substantially accelerate optimization because much of the signal information is stored explicitly in the grid. Many data structures have been proposed to improve scalability, including 2D feature planes (Reiser et al., 2023), sparse octrees (Yang, 2023; Yu et al., 2021), and hyperplane decompositions (Chan et al., 2022; Fridovich-Keil et al., 2023; Shue et al., 2023; Shi et al., 2024). Multiresolution hash-grid encoding is especially effective because it stores multiscale features in compact hash tables, allowing fast lookup while controlling memory usage (Müller et al., 2022). However, hash collisions create a capacity-sharing mechanism: multiple spatial locations may map to the same hash table entry, limiting their independent key-specific retrievability.

Hash-grid design and collision-aware capacity allocation. Most existing hash-grid methods use manually specified design choices, including the number of levels, table sizes, and geometric growth rates of grid resolutions (Müller et al., 2022; Dai & Fan, 2026). These choices are usually data-agnostic and do not explicitly account for how the target signal is distributed across resolutions or how hash collisions affect key-specific retrievability. Consequently, a fixed geometric schedule may over-allocate capacity to frequency bands with little information while under-allocating capacity to informative bands or regions. Our method addresses this gap by modeling collision-induced information degradation and using it to guide resolution selection and spatial activation in multiresolution hash encoding.

B PROOF OF PROPOSITION 1

Proof. Fix any block b, level ℓ, and frequency band $f_{(k)} \in \mathcal{F}_\ell$. Under the collision–erasure surrogate, $\widehat{Y}_{b,\ell,k} = Y_{b,k}$ when $R_{b\ell} = 1$, whereas $\widehat{Y}_{b,\ell,k}$ is a fixed erasure symbol when $R_{b\ell} = 0$. Since successful retrieval and erasure are distinguishable under the surrogate, $R_{b\ell}$ is determined by $\widehat{Y}_{b,\ell,k}$.

Hence,

$$I(Y_{b,k}; \widehat{Y}_{b,\ell,k}) = I(Y_{b,k}; \widehat{Y}_{b,\ell,k}, R_{b\ell}) \tag{21}$$

$$= I(Y_{b,k}; R_{b\ell}) + I(Y_{b,k}; \widehat{Y}_{b,\ell,k} \mid R_{b\ell}). \tag{22}$$

By Equation (12), $R_{b\ell}$ is independent of $Y_{b,k}$, so $I(Y_{b,k}; R_{b\ell}) = 0$. Moreover,

$$I(Y_{b,k}; \widehat{Y}_{b,\ell,k} \mid R_{b\ell} = 1) = H(Y_{b,k}),$$

because successful retrieval preserves $Y_{b,k}$ exactly, while

$$I(Y_{b,k}; \widehat{Y}_{b,\ell,k} \mid R_{b\ell} = 0) = 0,$$

because the erasure output is constant. Therefore,

$$I(Y_{b,k}; \widehat{Y}_{b,\ell,k}) = \mathbf{g}_{\ell b} q(\alpha_\ell) H(Y_{b,k}).$$

Similarly,

$$H(\widehat{Y}_{b,\ell,k} \mid R_{b\ell}) = \Pr(R_{b\ell} = 1) H(Y_{b,k}) + \Pr(R_{b\ell} = 0) \cdot 0 \tag{23}$$

$$= \mathbf{g}_{\ell b} q(\alpha_\ell) H(Y_{b,k}), \tag{24}$$

where independence of $R_{b\ell}$ and $Y_{b,k}$ is used again. Hence,

$$I(Y_{b,k}; \widehat{Y}_{b,\ell,k}) = H(\widehat{Y}_{b,\ell,k} \mid R_{b\ell}),$$

which proves the proposition. □

C Derivation of the Collision-Survival Factor

Let $K \sim \mathrm{Poisson}(\alpha)$ denote the number of collision keys for a routed key. Under the collision–erasure surrogate, one of the $K+1$ colliding keys remains retrievable. Hence, the survival probability of a routed key is

$$q(\alpha) = \mathbb{E}\left[\frac{1}{K+1}\right] \tag{25}$$

$$= \sum_{k=0}^{\infty} \frac{1}{k+1} e^{-\alpha} \frac{\alpha^k}{k!} \tag{26}$$

$$= \frac{e^{-\alpha}}{\alpha} \sum_{k=0}^{\infty} \frac{\alpha^{k+1}}{(k+1)!} \tag{27}$$

$$= \frac{1 - e^{-\alpha}}{\alpha}. \tag{28}$$

For $\alpha = 0$, we define $q(0) = 1$ by continuity.

D Proof of Theorem 1

Proof. The proof has two parts. Part 1 establishes the structure of the solution, *i.e.* there exists a global maximizer that is binary except possibly at one coordinate. This reduces the vector optimization problem to a one-dimensional maximization over the total gate mass. Part 2 shows how to locate this optimal mass efficiently.

Fix a level ℓ with assigned frequency segment $\mathcal{F}_\ell$ instantiated at resolution N_ℓ, and abbreviate

$$J_b \equiv J_b(\mathcal{F}_\ell), \qquad \kappa \equiv \kappa_\ell, \qquad T \equiv T_\ell.$$

Let π order the blocks by nonincreasing assigned block information and write

$$\mathcal{J}_{(i)} \equiv \mathcal{J}_{\pi(i)}, \qquad \mathcal{J}_{(1)} \geq \cdots \geq \mathcal{J}_{(B)} \geq 0.$$

Define the prefix sums

$$P_m \equiv \sum_{i=1}^{m} \mathcal{J}_{(i)}, \qquad P_0 = 0.$$

The level-wise gate objective is

$$J_\ell(\mathbf{g}_\ell) = q\left(\frac{\kappa G(\mathbf{g}_\ell)}{T}\right) \sum_{b=1}^{B} \mathbf{g}_{\ell b} \mathcal{J}_b.$$

After extending the collision factor continuously to the origin by $q(0) = \lim_{\alpha \to 0^+} q(\alpha) = 1$, the objective is continuous on the compact feasible set $[0,1]^B$. Hence a global maximizer exists.

Part 1: structure of the solution. For any fixed gate mass $G \in [0, B]$, the collision factor $q(\kappa G/T)$ is constant. Therefore, among all gates with mass G, maximizing J_ℓ is equivalent to the linear program

$$\max_{\mathbf{g} \in [0,1]^B} \sum_{b=1}^{B} \mathbf{g}_b \mathcal{J}_b \qquad \text{s.t.} \qquad \sum_{b=1}^{B} \mathbf{g}_b = G. \tag{29}$$

Equivalently, we minimize $-\sum_b \mathbf{g}_b \mathcal{J}_b$ over the same polytope. The Lagrangian is

$$\mathcal{L}(\mathbf{g}, \lambda, \boldsymbol{\mu}, \boldsymbol{\nu}) = -\sum_{b=1}^{B} \mathbf{g}_b \mathcal{J}_b + \lambda \left(\sum_{b=1}^{B} \mathbf{g}_b - G \right) - \sum_{b=1}^{B} \mu_b \mathbf{g}_b + \sum_{b=1}^{B} \nu_b (\mathbf{g}_b - 1),$$

where $\mu_b, \nu_b \geq 0$. The KKT conditions are

$$-\mathcal{J}_b + \lambda - \mu_b + \nu_b = 0, \qquad \mu_b \mathbf{g}_b = 0, \qquad \nu_b(\mathbf{g}_b - 1) = 0.$$

Thus any optimum of the fixed-mass problem satisfies

$$\mathbf{g}_b = 1 \Rightarrow \mathcal{J}_b \geq \lambda, \qquad 0 < \mathbf{g}_b < 1 \Rightarrow \mathcal{J}_b = \lambda, \qquad \mathbf{g}_b = 0 \Rightarrow \mathcal{J}_b \leq \lambda.$$

Hence, an optimal fixed-mass gate can be chosen with a threshold structure, blocks with larger J_b are activated first, with at most one fractional gate.

Because the blocks are sorted so that $\mathcal{J}_{(1)} \geq \cdots \geq \mathcal{J}_{(B)}$, an optimal fixed-mass solution is obtained by filling blocks in this order. For $G \in [0, B]$, write

$$m = \lfloor G \rfloor, \qquad \theta = G - m.$$

If $m < B$, one such optimizer is

$$\mathbf{g}_{\pi(i)}(G) = \begin{cases} 1, & i \leq m, \\ \theta, & i = m+1, \\ 0, & i > m+1. \end{cases}$$

If $m = B$, all gates are equal to one. When there are ties among the $\mathcal{J}_b$, the fixed-mass optimizer need not be unique; however, the mass assigned to tied blocks can be rearranged without changing the objective value. Therefore, there always exists an optimal fixed-mass gate with at most one fractional coordinate.

The optimal value of the fixed-mass problem is

$$H_\ell(G) = P_m + (G - m)\mathcal{J}_{(m+1)}, \qquad G \in [m, m+1], \qquad m = 0, \ldots, B-1,$$

with $H_\ell(B) = P_B$. Therefore, the original vector problem is equivalent to the scalar problem

$$G_\ell^\star \in \arg\max_{0 \leq G \leq B} \mathcal{F}_\ell(G), \qquad \mathcal{F}_\ell(G) \triangleq q\left(\frac{\kappa G}{T}\right) H_\ell(G). \tag{30}$$

The greedy fill at any maximizing mass $G_\ell^\star$ is consequently a global maximizer of the original gate-update problem.

Part 2: locating the optimal gate mass. It remains to solve the scalar problem in Equation (30). On the interval $G \in [m, m+1]$, $m = 0, \dots, B-1$, define

$$C_m \triangleq P_m - m\mathcal{J}_{(m+1)} = \sum_{i=1}^{m} \big(\mathcal{J}_{(i)} - \mathcal{J}_{(m+1)}\big) \geq 0.$$

Then

$$H_\ell(G) = P_m + (G-m)\mathcal{J}_{(m+1)} = C_m + \mathcal{J}_{(m+1)}G.$$

Pass to the load variable

$$\alpha = \frac{\kappa G}{T}, \qquad \alpha_m = \frac{\kappa m}{T}.$$

On the interval $\alpha \in [\alpha_m, \alpha_{m+1}]$, define

$$\widetilde{F}_m(\alpha) \triangleq \mathcal{F}_\ell\left(\frac{T\alpha}{\kappa}\right).$$

Using $\alpha q(\alpha) = 1 - e^{-\alpha}$, we obtain

$$\widetilde{F}_m(\alpha) = C_m q(\alpha) + \frac{T}{\kappa}\mathcal{J}_{(m+1)}(1 - e^{-\alpha}).$$

Moreover,

$$q'(\alpha) = \frac{(\alpha+1)e^{-\alpha} - 1}{\alpha^2} = -e^{-\alpha}\psi(\alpha), \qquad \psi(\alpha) \triangleq \frac{e^\alpha - 1 - \alpha}{\alpha^2},$$

with the continuous extension $\psi(0) = 1/2$. Therefore,

$$\widetilde{F}'_m(\alpha) = e^{-\alpha} D_m(\alpha),$$

where

$$D_m(\alpha) \triangleq \frac{T}{\kappa}\mathcal{J}_{(m+1)} - C_m\psi(\alpha).$$

Since $e^{-\alpha} > 0$, the sign of $\widetilde{F}'_m(\alpha)$ is exactly the sign of $D_m(\alpha)$.

We now show that the derivative sign can change at most once over the entire interval $[0, B]$. First, ψ is strictly increasing on $[0, \infty)$ because

$$\psi(\alpha) = \sum_{k \geq 2} \frac{\alpha^{k-2}}{k!}.$$

Hence, for fixed m, $D_m(\alpha)$ is nonincreasing in α.

Second, the derivative-sign term can only decrease when passing from one unit interval to the next. Indeed, for $m = 0, \dots, B-2$,

$$\mathcal{J}_{(m+2)} \leq \mathcal{J}_{(m+1)}$$

and

$$C_{m+1} = P_{m+1} - (m+1)\mathcal{J}_{(m+2)} = C_m + (m+1)\big(\mathcal{J}_{(m+1)} - \mathcal{J}_{(m+2)}\big) \geq C_m.$$

Therefore, for every $\alpha \geq 0$,

$$D_{m+1}(\alpha) = \frac{T}{\kappa}\mathcal{J}_{(m+2)} - C_{m+1}\psi(\alpha) \leq \frac{T}{\kappa}\mathcal{J}_{(m+1)} - C_m\psi(\alpha) = D_m(\alpha).$$

Thus the derivative-sign term is nonincreasing both within each interval and across adjacent intervals. Consequently, $\mathcal{F}_\ell$ is globally unimodal on $[0, B]$: it increases while the derivative sign is positive, decreases after the derivative sign becomes negative, and if the sign changes at a breakpoint, that breakpoint is a maximizer.

The optimal mass can therefore be located by scanning the intervals from left to right and performing at most one bisection. For each $m = 0, \dots, B-1$, compute the endpoint loads

$$\alpha_m = \frac{\kappa m}{T}, \qquad \alpha_{m+1} = \frac{\kappa(m+1)}{T}.$$

There are three cases. If

$$D_m(\alpha_m) \le 0,$$

then $\mathcal{F}_\ell$ is nonincreasing to the right of $G = m$. Since the scan has not found an earlier sign change, $G_\ell^\star = m$ is a global maximizer. If

$$D_m(\alpha_m) > 0 \qquad \text{and} \qquad D_m(\alpha_{m+1}) < 0,$$

then the unique maximizer lies in the interior of this interval. It is the unique root of

$$D_m(\alpha) = 0,$$

or equivalently,

$$C_m\big(e^\alpha - 1 - \alpha\big) = \frac{T}{\kappa}\mathcal{J}_{(m+1)}\alpha^2.$$

The unique root is computed to any prescribed accuracy by bisection on $[\alpha_m, \alpha_{m+1}]$, and the corresponding optimal gate mass is

$$G_\ell^\star = \frac{T}{\kappa}\alpha^\star.$$

If

$$D_m(\alpha_m) > 0 \qquad \text{and} \qquad D_m(\alpha_{m+1}) = 0,$$

then $\mathcal{F}_\ell$ is nondecreasing on $[m, m+1]$ and nonincreasing immediately after the breakpoint, so $G_\ell^\star = m + 1$ is a global maximizer.

If none of these cases occurs, then $D_m(\alpha_{m+1}) > 0$, so $\mathcal{F}_\ell$ is increasing throughout the current interval and the scan proceeds to the next interval. If the scan reaches the end without finding a sign change, then $\mathcal{F}_\ell$ is nondecreasing on all of $[0, B]$, and $G_\ell^\star = B$ is a global maximizer.

Finally, by Part 1, the greedy fill at mass $G_\ell^\star$ globally solves the original gate-update problem: with $m^\star = \lfloor G_\ell^\star \rfloor$, activate the $m^\star$ most informative blocks, assign the fractional remainder $G_\ell^\star - m^\star$ to the next block if $m^\star < B$, and deactivate the rest. If $G_\ell^\star$ is an integer, the fractional case is vacuous; if $m^\star = B$, every block is active. This proves the theorem. □

E Resolution-Schedule Update via Dynamic Programming

This section provides the complete derivation of the resolution-schedule update introduced in Section 4.2. Once the block-wise gates are fixed, selecting the resolution schedule reduces to an ordered segmentation problem over the candidate frequency bands. The resulting subproblem can be solved exactly by dynamic programming.

Recall the ordered candidate resolutions

$$N_{(1)} < N_{(2)} < \cdots < N_{(K)}, \tag{31}$$

with corresponding ordered frequency bands $f_{(1)}, \dots, f_{(K)}$. A valid L-level schedule is represented by cut indices

$$0 = k_0 < k_1 < \cdots < k_L = K. \tag{32}$$

Under this schedule, level ℓ receives the contiguous segment

$$\mathcal{F}_\ell = \mathcal{F}_{(k_{\ell-1}:k_\ell)} = \left\{ f_{(k_{\ell-1}+1)}, \dots, f_{(k_\ell)} \right\}, \tag{33}$$

and is instantiated at resolution

$$N_\ell = N_{(k_\ell)}. \tag{34}$$

For arbitrary indices $0 \le u < e \le K$, define the candidate segment

$$\mathcal{F}_{(u:e)} \triangleq \{f_{(u+1)}, \dots, f_{(e)}\}. \tag{35}$$

If this segment is assigned to level ℓ, the level is instantiated at the endpoint resolution $N_{(e)}$, which is the smallest candidate resolution whose Nyquist limit covers all bands in $\mathcal{F}_{(u:e)}$.

Let $\kappa_{\ell,e} > 0$ denote the effective number of hash keys contributed by a fully active block when level ℓ is instantiated at resolution $N_{(e)}$. Under the fixed gate vector

$$\mathbf{g}_\ell = (\mathbf{g}_{\ell 1}, \dots, \mathbf{g}_{\ell B}),$$

the resulting effective load factor is

$$\alpha_{\ell,e}(\mathbf{g}_\ell) \triangleq \frac{\kappa_{\ell,e}}{T_\ell} \sum_{b=1}^{B} \mathbf{g}_{\ell b}. \tag{36}$$

Once the final endpoint k_ℓ has been selected, the level-wise quantities used in Equations (10) and (15) satisfy

$$\kappa_\ell = \kappa_{\ell,k_\ell}, \qquad \alpha_\ell(\mathbf{g}_\ell) = \alpha_{\ell,k_\ell}(\mathbf{g}_\ell). \tag{37}$$

For block b, the information contained in $\mathcal{F}_{(u:e)}$ is

$$\mathcal{J}_b(\mathcal{F}_{(u:e)}) = \sum_{k=u+1}^{e} \mathfrak{I}_b(f_{(k)}). \tag{38}$$

Extending Equation (15) to a candidate segment, the retrievable information obtained by assigning $\mathcal{F}_{(u:e)}$ to level ℓ is

$$J_\ell(\mathbf{g}_\ell; \mathcal{F}_{(u:e)}) = q(\alpha_{\ell,e}(\mathbf{g}_\ell)) \sum_{b=1}^{B} \mathbf{g}_{\ell b}\, \mathcal{J}_b(\mathcal{F}_{(u:e)}). \tag{39}$$

Equivalently,

$$J_\ell(\mathbf{g}_\ell; \mathcal{F}_{(u:e)}) = q\left(\frac{\kappa_{\ell,e}}{T_\ell} \sum_{b=1}^{B} \mathbf{g}_{\ell b}\right) \sum_{b=1}^{B} \mathbf{g}_{\ell b}\, \mathcal{J}_b(\mathcal{F}_{(u:e)}). \tag{40}$$

The score depends on both segment boundaries: u determines which frequency bands are assigned to the level, while e additionally determines the endpoint resolution and hence the effective load factor.

With the gates fixed, the resolution-schedule update is therefore

$$\max_{0=k_0<k_1<\cdots<k_L=K} \sum_{\ell=1}^{L} J_\ell\big(\mathbf{g}_\ell; \mathcal{F}_{(k_{\ell-1}:k_\ell)}\big). \tag{41}$$

Thus, the fixed-gate subproblem partitions the K ordered candidate bands into L nonempty contiguous segments.

Let $V(\ell, e)$ denote the maximum retrievable information obtained by assigning the first e candidate bands to the first ℓ levels. The boundary conditions are

$$V(0,0) = 0, \qquad V(0,e) = -\infty \quad \text{for } e > 0. \tag{42}$$

A state (ℓ, e) is feasible only if

$$\ell \le e \le K - L + \ell, \tag{43}$$

because each of the first ℓ levels and each of the remaining $L - \ell$ levels must receive at least one candidate band.

For a feasible state (ℓ, e), let u be the final cut point. The first $\ell - 1$ levels then receive the first u candidate bands, while level ℓ receives $\mathcal{F}_{(u:e)}$. The recurrence is

$$V(\ell, e) = \max_{\ell-1 \le u < e} \big\{V(\ell-1, u) + J_\ell\big(\mathbf{g}_\ell; \mathcal{F}_{(u:e)}\big)\big\}. \tag{44}$$

The recurrence in Equation (44) is exact by Bellman's principle of optimality. To see this, consider an optimal partition associated with a feasible state (ℓ, e), and let u be its final cut point. Its objective value decomposes as

$$\underbrace{\sum_{j=1}^{\ell-1} J_j\big(\mathbf{g}_j; \mathcal{F}_{(k_{j-1}:k_j)}\big)}_{\text{contribution of the first } \ell-1 \text{ levels}} + J_\ell\big(\mathbf{g}_\ell; \mathcal{F}_{(u:e)}\big)\,. \tag{45}$$

The prefix formed by the first $\ell - 1$ levels must itself be an optimal partition of the first u candidate bands. Otherwise, replacing it with a higher-scoring prefix would improve the complete partition without changing its final segment, contradicting optimality. Therefore, for a fixed final cut u, the largest attainable objective value is

$$V(\ell - 1, u) + J_\ell\big(\mathbf{g}_\ell; \mathcal{F}_{(u:e)}\big)\,. \tag{46}$$

Conversely, every feasible predecessor u combines an optimal partition represented by $V(\ell - 1, u)$ with the nonempty final segment $\mathcal{F}_{(u:e)}$, thereby forming a feasible partition for state (ℓ, e). Maximizing over all feasible final cuts therefore searches the complete set of valid partitions and yields Equation (44). Together with the boundary conditions in Equation (42), recursive application of this argument shows that $V(L, K)$ is the optimal objective value of the fixed-gate schedule subproblem.

To recover a corresponding globally optimal schedule, store one maximizing predecessor for each feasible state:

$$P(\ell, e) \in \operatorname*{arg\,max}_{\ell-1 \le u < e} \big\{V(\ell - 1, u) + J_\ell\big(\mathbf{g}_\ell; \mathcal{F}_{(u:e)}\big)\big\}\,. \tag{47}$$

Starting from the terminal state (L, K), set $k_L = K$ and recursively recover the remaining cut indices according to

$$k_{\ell-1} = P(\ell, k_\ell), \qquad \ell = L, L-1, \ldots, 1. \tag{48}$$

The resulting assignment is

$$\mathcal{F}_\ell = \mathcal{F}_{(k_{\ell-1}:k_\ell)}, \qquad N_\ell = N_{(k_\ell)}, \qquad \ell \in [L]. \tag{49}$$

The candidate-segment scores can be evaluated efficiently using prefix sums. For each block b, define

$$S_{b,e} \triangleq \sum_{k=1}^{e} \mathfrak{I}_b(f_{(k)}), \qquad S_{b,0} = 0. \tag{50}$$

The information contained in any candidate segment is then

$$\mathcal{J}_b(\mathcal{F}_{(u:e)}) = S_{b,e} - S_{b,u}. \tag{51}$$

For each level ℓ, define the gate-weighted prefix sum

$$\Gamma_{\ell,e} \triangleq \sum_{b=1}^{B} \mathbf{g}_{\ell b} S_{b,e}, \qquad \Gamma_{\ell,0} = 0. \tag{52}$$

It follows that

$$\sum_{b=1}^{B} \mathbf{g}_{\ell b}\, \mathcal{J}_b(\mathcal{F}_{(u:e)}) = \Gamma_{\ell,e} - \Gamma_{\ell,u}. \tag{53}$$

Therefore, the candidate-segment score can be evaluated as

$$J_\ell(\mathbf{g}_\ell; \mathcal{F}_{(u:e)}) = q\left(\frac{\kappa_{\ell,e}}{T_\ell} \sum_{b=1}^{B} \mathbf{g}_{\ell b}\right) (\Gamma_{\ell,e} - \Gamma_{\ell,u})\,. \tag{54}$$

Once the block-wise prefix sums, gate-weighted prefix sums, gate masses, and endpoint-dependent key counts have been computed, each DP transition can be evaluated in constant time. Thus, there is no need to explicitly materialize all $O(LK^2)$ candidate-segment scores.

Computing the block-wise prefix sums in Equation (50) requires $O(BK)$ time. These quantities depend only on the target signal and can therefore be reused across the alternating-optimization

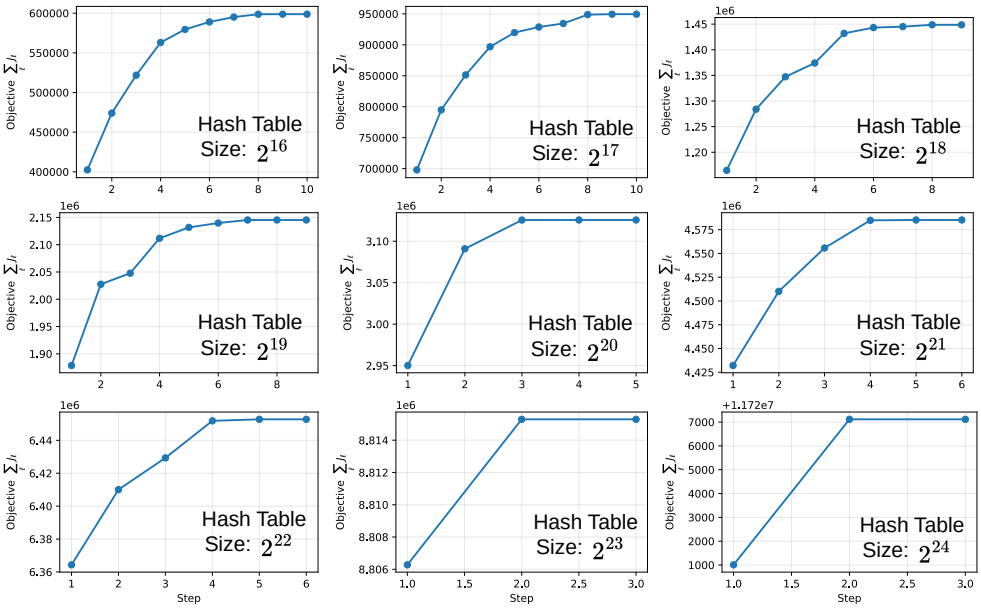

Figure 8: Convergence behavior of RIM on the Girl image under different hash-table sizes. Each subplot reports the objective value $\sum_{\ell=1}^{L} J_\ell$ across RIM optimization steps for a fixed hash-table size, ranging from 2^{16} to 2^{24}. Consistent with the monotonic ascent property of the alternating optimization, the objective is non-decreasing across iterations and rapidly reaches a plateau in all settings, demonstrating fast empirical convergence.

iterations. For one fixed-gate schedule update, constructing the gate-weighted prefix sums in Equation (52) requires $O(LBK)$ time.

The dynamic program has $O(LK)$ feasible states. Each state considers at most K predecessor indices, and each transition is evaluated in constant time using Equation (54). Therefore, the dynamic-programming search requires

$$O(LK^2) \tag{55}$$

time, and the complete fixed-gate schedule update requires

$$O(LBK + LK^2). \tag{56}$$

The value and predecessor tables require $O(LK)$ memory, while the block-wise and gate-weighted prefix sums require $O(BK + LK)$ memory. Hence, the total memory complexity is

$$O(BK + LK). \tag{57}$$

Backtracking requires $O(L)$ additional time. Explicitly storing all candidate-segment scores would increase the memory cost to $O(LK^2)$, but this is unnecessary because each score can be evaluated on demand.

F RIM-Hash Convergence

In this section, we verify the convergence behavior of the RIM algorithm by plotting the objective (RI) value across optimization steps under different hash-table sizes, as shown in Figure 8 and Figure 9. The objective is monotonically non-decreasing for all tested settings and rapidly stabilizes after a small number of iterations. This confirms that, regardless of the memory budget, each RIM update consistently improves or preserves the objective, which is consistent with the monotonic ascent property of the proposed optimization procedure.

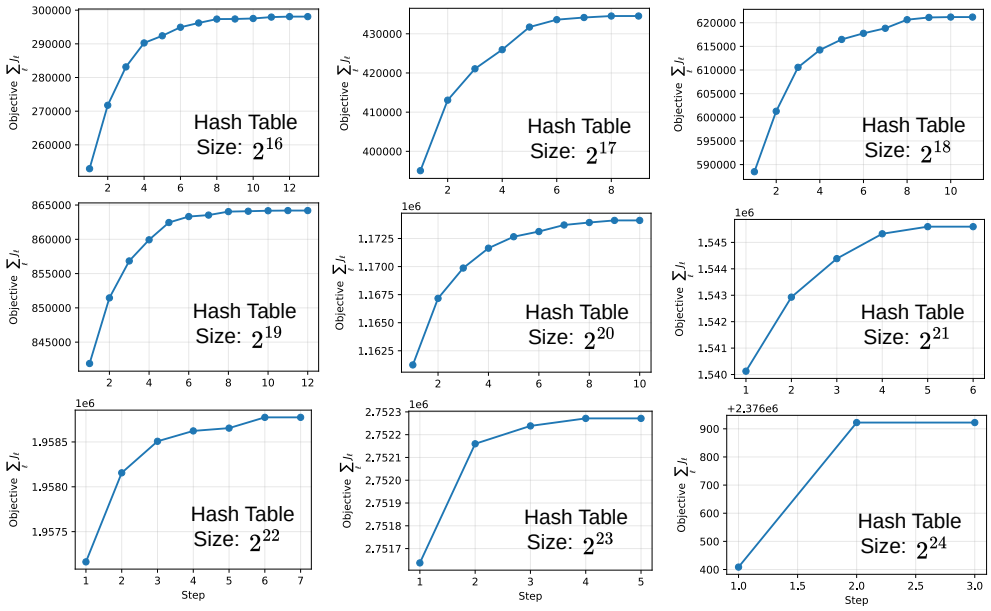

Figure 9: Convergence behavior of RIM on the Pluto image under different hash-table sizes. Each subplot reports the objective value $\sum_{\ell=1}^{L} J_\ell$ across RIM optimization steps for a fixed hash-table size, ranging from 2^{16} to 2^{24}. Consistent with the monotonic ascent property of the alternating optimization, the objective is non-decreasing across iterations and rapidly reaches a plateau in all settings, demonstrating fast empirical convergence.

G Experimental Setup and Wall-Clock Cost

Our implementation follows the training and evaluation protocol of MetricGrids (Wang et al., 2025) unless otherwise specified. All controlled hash-grid experiments use $L = 16$ resolution levels with a feature dimension of $F = 2$ per level.The interpolated features from all levels are concatenated and decoded by a three-layer MLP with two hidden layers of width 64 and ReLU activations, followed by a linear output layer. Consequently, the decoder maps the $LF = 32$ dimensional hash-grid feature to RGB values for 2D image fitting and to a scalar value for SDF fitting. The same decoder architecture is used for all controlled hash-grid variants.

Following MetricGrids, we optimize the network using Adam with $(\beta_1, \beta_2, \epsilon) = (0.9, 0.99, 10^{-15})$ and zero weight decay. For Kodak, gigapixel images, and WSIs, the initial learning rate is 10^{-3} and is cosine annealed to 10^{-5} over the full optimization horizon. We use the ℓ_2 reconstruction loss for all 2D image-fitting experiments. For 3D SDF fitting, following the MetricGrids protocol, the initial learning rate is 10^{-4}, cosine annealed to 10^{-5}, and MAPE is used as the reconstruction loss. Each optimization step uses 2^{18} sampled coordinates. Kodak and SDF models are trained for $20{,}000$ optimization steps, while gigapixel images are trained for $35{,}000$ steps; WSIs use the same optimization protocol as the gigapixel-image experiments. The coordinate-sampling, interpolation, optimizer, learning-rate scheduling, and evaluation procedures otherwise follow MetricGrids. For every controlled comparison at a fixed parameter budget, the same sampled data, number of optimization steps, decoder, and evaluation points are used across methods.

For a given hash-table size, the decoder architecture, number of levels, feature dimension, optimizer, and training schedule are therefore held fixed across the controlled hash-grid variants. Different model scales are obtained only by changing the hash-table size. This protocol isolates the effect of the hash-grid layout from changes in decoder capacity or optimization.

Table 5: Wall-clock optimization time, in seconds, for our method and the baseline methods on Kodak, gigapixel-image, and SDF fitting. All timings are measured on an NVIDIA RTX A6000 GPU.

Data	Kodak	Girl	Tokyo	Pluto	SDF
I-NGP	220s	2798s	4073s	1676s	463s
MetricGrids	1428s	5630s	7942s	2384s	1626s
Ours	377s	3073s	4525s	1743s	677s

For RIM-Hash, the resolution schedule selected during initialization remains fixed during signal fitting, whereas the hash-table embeddings, gate logits, level-wise fallback features, and decoder parameters are optimized jointly. For Kodak, we use $K = 30$ candidate resolutions uniformly spaced in the frequency domain. For gigapixel natural images and WSIs, we use $K = 200$ candidates to provide a sufficiently fine search space at substantially larger spatial resolutions.

For each candidate resolution $N_{(k)}$, we construct the corresponding low-pass signal $\mathbf{y}_{\leq f_{(k)}}$ using PyTorch resize operations. Specifically, the target image is first resized to the spatial resolution associated with $N_{(k)}$ using anti-aliased resizing and is then resized back to its original spatial resolution. The adjacent-band residual used in Equation (7) is subsequently computed as $\mathbf{y}_{\leq f_{(k)}} - \mathbf{y}_{\leq f_{(k-1)}}$, so that all band residuals are evaluated on the same pixel grid. All resizing operations are implemented as PyTorch tensor operations and executed directly on the GPU, making the band-information estimation procedure computationally lightweight.

We alternately update the block-wise gates and the resolution schedule until the retrievable-information objective converges. The resulting gates initialize the trainable gate logits used during reconstruction optimization.

Unless otherwise specified, we use a block side length of 16 pixels for Kodak and 400 pixels for gigapixel natural images and WSIs. Accordingly, each large 2D image is partitioned into non-overlapping 400×400 spatial blocks, with one hash-access gate for each block–level pair.

For the 3D experiments, the 2D blocks are replaced by volumetric cubes with a default side length of 8, corresponding to $8 \times 8 \times 8$ voxels. Each cube has an independent gate at each hash-grid level. Hash features are queried using trilinear interpolation, and the remaining RIM operations—band-information estimation, collision-aware gate initialization, resolution selection, and joint reconstruction optimization—are extended dimension-wise to three dimensions.

We evaluate 2D fitting on the 24 images of the Kodak Lossless True Color Image Suite, the *Pluto*, *Tokyo*, and *Girl* gigapixel images, and all four ultra-high-resolution WSIs. Image coordinates are normalized to $[0, 1]^2$. Reconstruction fidelity is evaluated using PSNR and SSIM. For Kodak, we report averages over all 24 images; for gigapixel images and WSIs, we report either individual results or the average over the corresponding WSI set, as specified in each table. PSNR and SSIM are computed directly from the floating-point reconstructions without quantizing the reconstructed signal.

We evaluate 3D fitting on seven objects from the Stanford 3D Scanning Repository. Coordinates are normalized to $[0, 1]^3$. Following the MetricGrids SDF protocol, all compared methods use the same sampled training coordinates and evaluation points. We report intersection over union (IoU) and normal angular error (NAE).

For RIM-Hash, the reported model size includes all trainable parameters: hash-table embeddings, gate logits, level-wise fallback features, and decoder parameters. Quantities used only for RIM initialization, such as the estimated block-wise band-information and dynamic-programming tables, are non-trainable and are therefore excluded from the parameter count.

We report wall-clock optimization time on NVIDIA RTX A6000 GPUs in Table 5. RIM-Hash incurs moderate additional optimization cost relative to I-NGP because of its adaptive routing, but remains substantially faster than MetricGrids. Specifically, RIM-Hash reduces optimization time relative to MetricGrids by approximately 74%, 45%, 43%, 27%, and 58% on Kodak, Girl, Tokyo, Pluto, and SDF, respectively. Thus, the improved parameter–fidelity trade-off does not rely on prohibitively expensive optimization.

Table 6: Numerical comparison on gigapixel Girl image fitting. We report PSNR (dB), SSIM, and the number of trainable parameters. RIM-Hash maintains competitive fidelity in both low- and high-parameter regimes. **Bold** and underlined denote the best and second-best results.

Method	I-NGP TOG 2022			NeuRBF ICCV 2023	MetricGrids CVPR 2025			Ours			
Params (M)	8.01	32.01	412.80	115.40	8.02	32.02	412.80	3.71	31.18	111.58	423.86
PSNR ↑	21.51	27.30	33.05	31.56	27.03	29.70	37.06	27.04	31.01	38.15	**49.49**
SSIM ↑	0.579	0.896	0.959	0.960	0.817	0.957	0.994	0.842	0.974	0.998	**0.999**

Table 7: Numerical comparison on gigapixel Pluto image fitting. We report PSNR (dB), SSIM, and the number of trainable parameters. RIM-Hash maintains competitive fidelity in both low- and high-parameter regimes. **Bold** and underlined denote the best and second-best results.

Method	I-NGP TOG 2022			NeuRBF ICCV 2023	MetricGrids CVPR 2025			Ours			
Params (M)	16.01	64.01	217.30	57.46	8.02	128.02	271.98	7.15	99.83	179.68	302.57
PSNR ↑	40.37	54.66	61.54	53.35	41.51	61.20	62.49	42.08	67.42	76.53	**88.23**
SSIM ↑	0.9584	0.9983	0.9996	0.9924	0.9585	**0.9999**	**0.9999**	0.9722	**0.9999**	**0.9999**	**0.9999**

H GIGAPIXEL IMAGE RESULTS

We present the full quantitative and qualitative results on the three gigapixel images, namely, *Tokyo*, *Pluto* and *Girl*. These three images stress the resolution schedule in different ways. *Tokyo* and *Girl* contain dense, spatially varying high-frequency structures, whereas *Pluto* is smoother at the global scale but still requires accurate modeling of subtle multiscale boundaries and texture variations. Together, they provide a useful test bed for evaluating whether a resolution schedule generalizes across different gigapixel images.

Table 7 shows that RIM-Hash also yields consistent gains on *Pluto*. In the low-parameter regime, RIM-Hash attains 42.08 dB with 7.15M parameters, outperforming both I-NGP (40.37 dB, 16.01M) and MetricGrids (41.51 dB, 8.02M). In the high-fidelity regime, RIM-Hash reaches 76.53 dB with 179.68M parameters, surpassing MetricGrids' 62.49 dB at 271.98M parameters and I-NGP's 61.54 dB at 217.3M parameters. The residual maps in Figure 10 are consistent with the quantitative results: RIM-Hash produces the smallest high-error regions and the lowest overall reconstruction error. Unlike *Tokyo*, where the weakness of non-adaptive schedules is especially pronounced, *Pluto* shows that even when a strong baseline performs well, collision-aware adaptation still leads to a better fidelity–parameter trade-off.

I PER-IMAGE RESULTS ON KODAK

To complement the dataset-level averages reported in Table 1, we provide the per-image PSNR and SSIM results for all 24 Kodak images in Figure 12. We compare RIM-Hash with I-NGP and MetricGrids under comparable parameter budgets. Specifically, RIM-Hash contains 197K trainable parameters, whereas I-NGP and MetricGrids contain 206K and 207K parameters, respectively.

As shown in Figure 12, RIM-Hash achieves the highest PSNR on 22 of the 24 images and the highest SSIM on 21 of the 24 images. Thus, the improvements in the average Kodak results are not driven by only a small subset of images; instead, RIM-Hash consistently improves reconstruction fidelity across images with diverse visual content. These results further demonstrate that retrievable-

Table 8: Numerical comparison on gigapixel Tokyo image fitting. We report PSNR (dB), SSIM, and the number of trainable parameters. RIM-Hash maintains competitive fidelity in both low- and high-parameter regimes. **Bold** and underlined denote the best and second-best results.

Method	I-NGP TOG 2022			NeuRBF ICCV 2023	MetricGrids CVPR 2025			Ours			
Params (M)	8.01	128.00	442.40	131.70	16.01	128.10	442.60	1.83	60.33	103.00	405.60
PSNR ↑	18.56	36.26	38.45	34.32	24.72	35.19	37.75	25.56	34.39	36.40	**41.07**
SSIM ↑	0.3212	0.9281	0.9362	0.8936	0.8173	0.9578	0.9423	0.8344	0.9562	0.9703	**0.9824**



Figure 10: Qualitative comparison on three gigapixel images. The leftmost panel presents the ground-truth image, while the three panels on the right visualize the per-pixel ℓ_2 error maps of I-NGP, MetricGrids, and the proposed method, respectively.

information-guided resolution and spatial-capacity allocation provide a stronger parameter–fidelity trade-off than the uniform hash-grid allocation used by I-NGP and MetricGrids.

J Visualization of the RIM-Hash Mask

We further visualize the learned RIM Masks after initialization on three gigapixel images, namely *Girl*, *Pluto*, and *Tokyo*. For each image, we present the gate masks of all 16 hash-grid levels using hash table sizes of 2^{16}, 2^{20}, and 2^{24}. The levels are ordered from the coarsest resolution at Level 1 to the finest resolution at Level 16. Yellow regions indicate activated hash-grid features, whereas purple regions are suppressed by the gate.

We can observe two consistent patterns. First, a smaller hash table results in substantially sparser gates. When the table size is 2^{16}, the limited capacity and the resulting severe hash collisions make many spatial features difficult to retrieve reliably. Consequently, RIM suppresses a large fraction of the activations, particularly at the middle and fine-resolution levels. Increasing the table size to 2^{20} and 2^{24} alleviates the collision pressure and allows more spatial regions to access the corresponding hash levels. The gates therefore become progressively denser as the hash-table capacity increases. This trend is consistently observed for the *Girl*, *Pluto*, and *Tokyo* images.

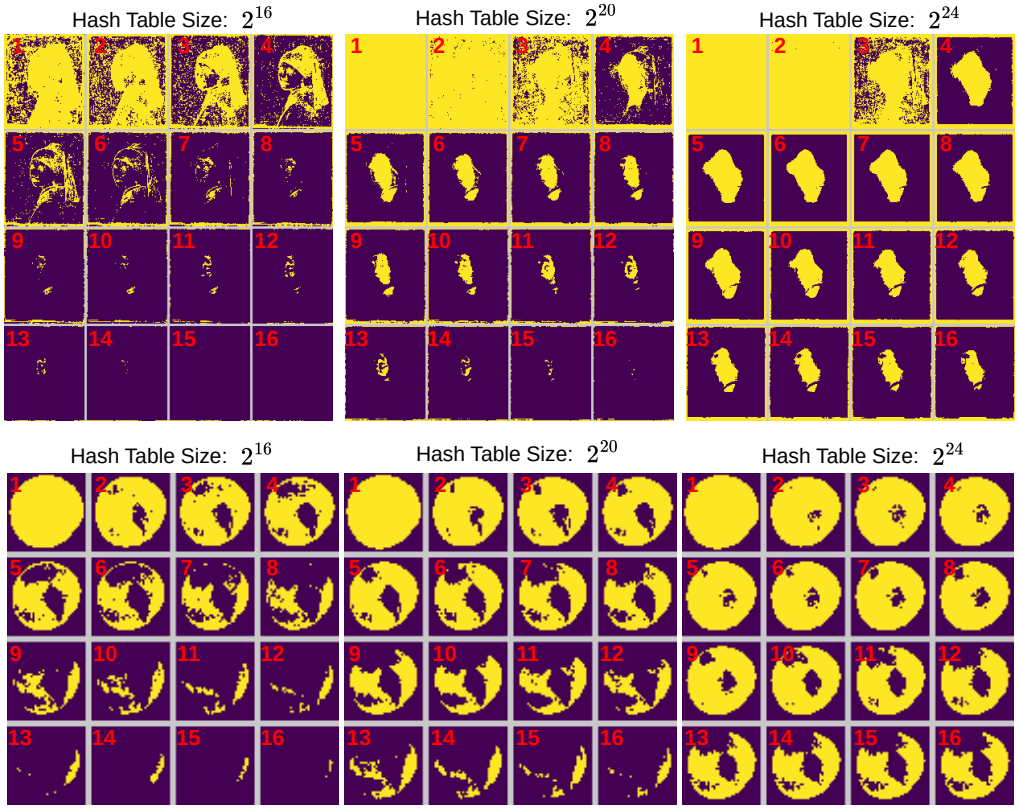

Figure 11: Visualization of the learned RIM gates for the *Girl* and *Pluto* gigapixel images with hash table sizes of 2^{16}, 2^{20}, and 2^{24}. Each group contains the gate masks of 16 hash-grid levels ordered from coarse to fine. Yellow and purple denote activated and suppressed regions, respectively. Larger hash tables produce denser gates by reducing hash collisions, whereas the gates become progressively more localized and sparse toward finer-resolution levels.

Second, for a given hash-table size, the gates generally become increasingly sparse and spatially localized at higher levels. The coarse levels are activated over broad regions because they represent low-frequency structures that contribute to the reconstruction of a large number of pixels. In contrast, the finer levels primarily encode localized high-frequency details and are therefore activated only in regions with sufficiently rich image content. For example, the fine-level gates concentrate around the foreground silhouette and facial structures in *Girl*, the planetary body and its boundary in *Pluto*, and the building structures and textured regions in *Tokyo*. Even with a large table of size 2^{24}, the finest levels remain selective rather than being uniformly activated.

These visualizations demonstrate that RIM adapts hash-grid usage jointly across spatial locations and resolution levels. Under a restricted hash-table budget, it aggressively suppresses collision-prone or weakly informative activations. As the available capacity increases, more features become reliably retrievable and are consequently retained. Meanwhile, the progressive sparsification toward finer levels prevents the model from spending high-resolution capacity on smooth or weakly informative regions. The learned gates therefore provide an interpretable realization of retrievable-information maximization: hash-table capacity is assigned only where its expected contribution to reconstruction outweighs the loss caused by collisions.

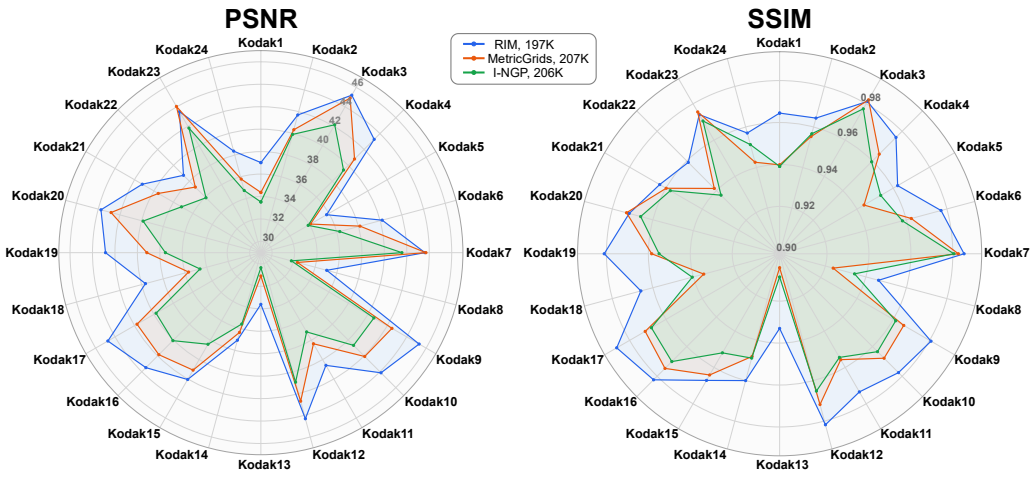

Figure 12: **Per-image reconstruction performance on the Kodak dataset.** We report PSNR (*left*) and SSIM (*right*) for RIM-Hash, MetricGrids, and I-NGP under comparable parameter budgets. Despite using only 197K trainable parameters, compared with 207K for MetricGrids and 206K for I-NGP, RIM-Hash achieves the best PSNR on 22 of the 24 images and the best SSIM on 21 of the 24 images. These results demonstrate consistent per-image improvements over the previous state-of-the-art MetricGrids while using fewer parameters.

K Relationship Between Retrievable Information and Reconstruction Quality

To assess whether retrievable information is a meaningful indicator of reconstruction quality, we plot retrievable information against final reconstruction PSNR in Figure 14. We evaluate I-NGP and RIM-Hash with block sizes of 100, 200, and 400 on the Pluto, Girl, and Tokyo gigapixel images across multiple model capacities. Marker size denotes the corresponding number of trainable parameters.

Across all three images, PSNR increases approximately monotonically with retrievable information, following an image-specific linear trend with Pearson correlation above 0.992. Importantly, retrievable information is not determined solely by model size: at comparable parameter counts, RIM-Hash consistently attains higher retrievable information and PSNR than I-NGP. This indicates that optimizing the hash-grid layout can improve the amount of recoverable signal information without materially increasing model capacity.

These results empirically support retrievable information as a proxy objective for multiresolution hash-grid design. In particular, higher retrievable information is consistently associated with higher reconstruction fidelity across architectures, gate configurations, and model scales.

L Comprehensive Ablation Study

We provide comprehensive ablations of the main design choices in RIM-Hash. The main paper isolates the two core components—the adaptive resolution schedule and the block-wise gates—on the Pluto image. We further examine the collision-survival factor, gate-selection strategy, fallback feature, block size, candidate-resolution set, and number of RIM optimization steps. Unless otherwise specified, all variants use the same decoder, number of hash-grid levels, feature dimension, optimizer, learning-rate schedule, reconstruction loss, number of training iterations, and trainable gate logits during reconstruction fitting. Each ablation modifies only the corresponding RIM initialization component while keeping the subsequent joint-training protocol unchanged.

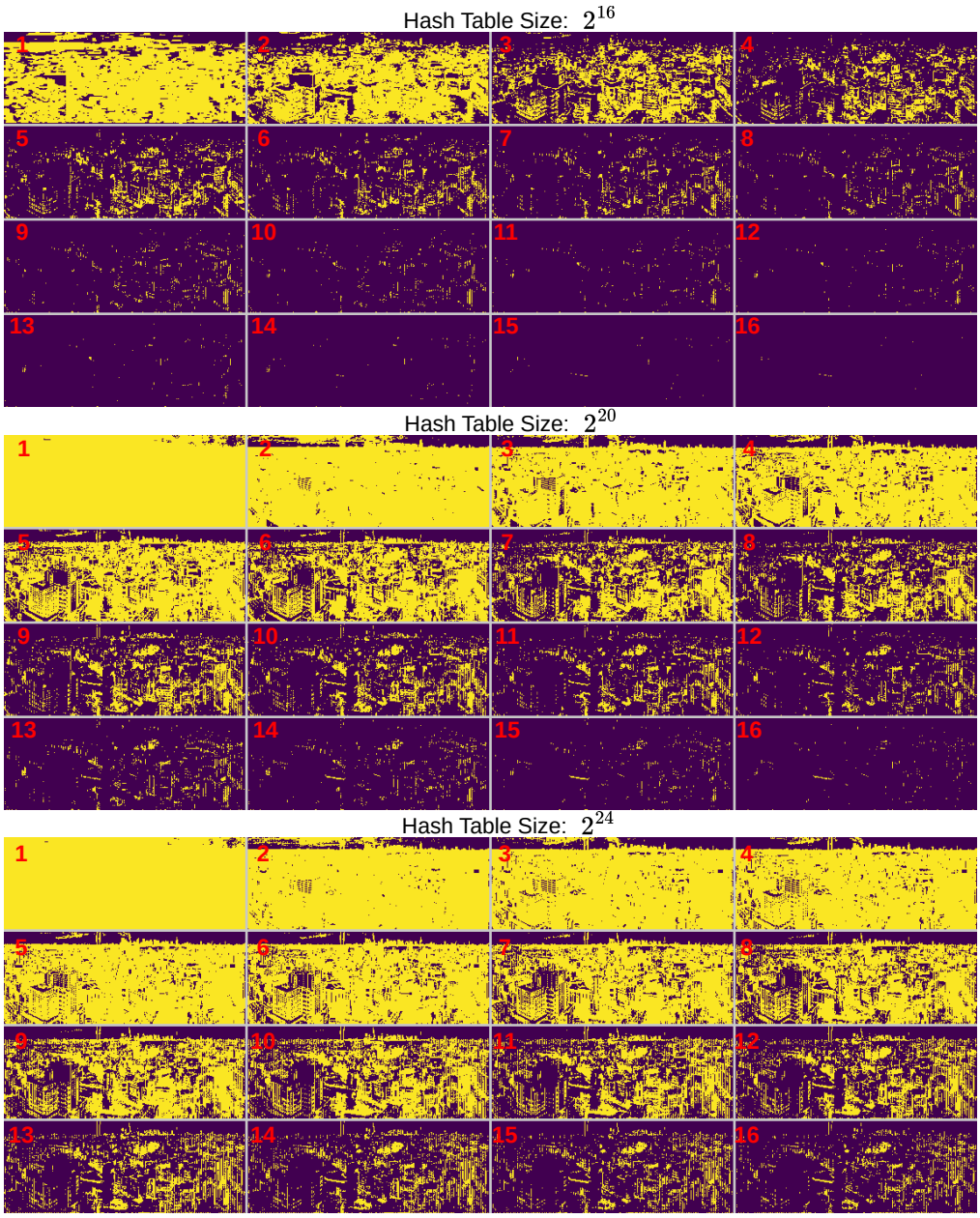

Figure 13: Visualization of the learned RIM gates for the *Tokyo* gigapixel images with hash table sizes of 2^{16}, 2^{20}, and 2^{24}. Each group contains the gate masks of 16 hash-grid levels ordered from coarse to fine. Yellow and purple denote activated and suppressed regions, respectively. Larger hash tables produce denser gates by reducing hash collisions, whereas the gates become progressively more localized and sparse toward finer-resolution levels.

L.1 Collision-survival Factor

We examine the collision model underlying the RIM objective. The full model uses the occupancy-based survival factor $q(\alpha) = (1 - e^{-\alpha})/\alpha$, which discounts the band-information assigned to each

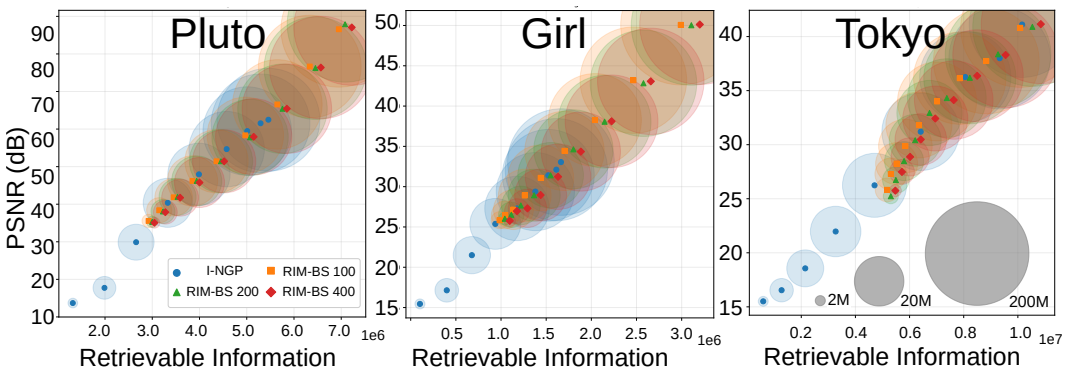

Figure 14: Relationship between retrievable information and reconstruction PSNR on the *Pluto*, *Girl*, and *Tokyo* images. We evaluate I-NGP and RIM-Hash with different block sizes. Despite their different hash-table configurations, resolution schedules, and gating strategies, all variants follow a consistent approximately linear relationship between retrievable information and PSNR. The circle size indicates the number of trainable parameters.

Table 9: Ablation of the collision model on the Pluto and Kodak dataset. The occupancy-based survival factor gives the strongest layout across all parameter budgets.

Dataset		W. collision factor			W.O. collision factor		
Pluto	Params (M)	7.15	99.83	179.68	10.32	110.43	182.64
	PSNR	42.08	67.42	76.53	40.95	66.53	75.12
	SSIM	0.9722	0.9999	0.9999	0.9624	0.9998	0.9999
Kodak	Params (K)	69.5	197	528	97.4	214	532
	PSNR	33.70	41.08	58.08	31.37	40.26	57.95
	SSIM	0.8877	0.9697	0.9999	0.8501	0.9592	0.9999

level according to its effective load factor. This factor converts the gate-assigned information into retrievable information by accounting for collision-induced information loss.

As shown in Table 9, the full collision-aware objective achieves the best performance. Removing the survival factor causes the optimizer to consider only the amount of information assigned to each level, without accounting for how much remains retrievable after hashing. This leads to a larger performance degradation in the low-parameter regime, where limited hash-table capacity produces more severe collisions. These results show that the survival factor is an essential mechanism for guiding capacity allocation across a broad range of effective load factors.

L.2 Gate Selection

We evaluate whether the gains of RIM-Hash arise merely from reducing hash-table access or from identifying the spatial locations where hash features are most useful. To isolate the effect of gate allocation, we compare the RIM-selected gates with two matched-mass controls. At each level, all strategies use the same total gate mass. The *random-gate* baseline randomly assigns this mass to spatial blocks, whereas the *uniform-gate* baseline distributes it evenly across all blocks.

As shown in Table 10, both control strategies perform substantially worse than the RIM-selected gates across all parameter budgets. For example, at 302.57M parameters, RIM-Hash achieves 88.23 dB, compared with 69.32 dB for random gates and 71.02 dB for uniform gates. These results demonstrate that matching the overall gate mass is insufficient. The benefit of RIM-Hash comes from its retrievable information-guided spatial allocation, which selectively enables hash access in regions where it contributes most to reconstruction fidelity.

Table 10: Ablation of gate-selection strategies on Pluto. All strategies use the same total gate mass at each level. The substantial advantage of RIM-Hash shows that performance depends on the spatial allocation of hash access rather than only on its overall amount.

Gate strategy	99.83M params	179.68M params	302.57M params
RIM-Hash	**67.42**	**76.53**	**88.23**
Random gates	56.83	63.57	69.32
Uniform gates	58.46	65.14	71.02

Table 11: Ablation of the fallback-feature design on Pluto. A trainable fallback consistently improves reconstruction quality for spatial blocks whose hash features are suppressed by small gate values, while adding only a negligible number of parameters.

Fallback vector	99.83M params	179.68M params	302.57M params
Trainable fallback feature	**67.42**	**76.53**	**88.23**
Zero fallback	65.89	74.93	86.98
Fixed random feature	66.12	75.38	87.34

L.3 Fallback Feature

We ablate the design of the fallback feature used when hash access is suppressed by the learned gates. Specifically, we compare three alternatives: a trainable fallback feature, a fixed random feature, and a fixed zero feature. As shown in Table 11, the trainable fallback consistently achieves the best reconstruction quality across all parameter budgets.

Although the fallback feature is shared globally within each level and introduces only a negligible number of additional trainable parameters, it provides substantial performance gains. At 99.83M parameters, for example, the trainable fallback improves PSNR from 65.89 dB with a zero fallback and 66.12 dB with a fixed random fallback to 67.42 dB. This result indicates that spatial blocks with limited hash access still benefit from a compact learned representation rather than a fixed substitute.

We also evaluated a separate fallback feature for each spatial block. While this variant achieves comparable reconstruction fidelity, it introduces additional parameters and complicates the routing mechanism. We therefore adopt a shared trainable fallback feature for each level in the main model.

L.4 Block Size

We study the sensitivity of RIM-Hash to the spatial block size on Pluto under three parameter budgets. As shown in Table 12, increasing the block size slightly reduces the total number of trainable parameters because fewer block-wise gate logits are required. However, reconstruction quality does not improve monotonically with block size. Across all three model scales, both PSNR and SSIM increase as the block size grows from 50 to 400, with a block size of 400 consistently achieving the best performance.

This trend reflects a trade-off between spatial adaptivity and partition granularity. Small blocks provide fine-grained routing decisions, but they introduce more gate parameters and may fragment spatially coherent structures into many independently optimized regions. In contrast, moderately larger blocks aggregate sufficient local information to produce more stable estimates of block-wise band-information while retaining enough spatial resolution to distinguish informative and low-information regions. Consequently, a block size of 400 provides the most effective allocation of hash-table capacity.

When the block size is further increased to 800, both PSNR and SSIM decrease despite the marginal reduction in parameter count. At this scale, each block covers a large and potentially heterogeneous image region, forcing pixels with different spatial-frequency characteristics to share the same gate decisions. This overly coarse partition weakens the spatial adaptivity of RIM-Hash and limits its ability to concentrate capacity on informative regions. The consistent behavior across all parameter budgets indicates that the preferred block size is not specific to a particular model scale. We therefore use a block size of 400 for the Pluto experiments.

Table 12: Sensitivity to spatial block size on Pluto across three model scales. Larger blocks reduce the number of trainable gate parameters, while reconstruction quality improves up to a block size of 400 and decreases when the partition becomes overly coarse.

Block size	50			100			200		
Params (M)	101.21	183.06	304.95	100.41	181.75	303.64	99.97	181.03	303.07
PSNR ↑	66.84	75.92	87.58	67.16	76.24	87.92	66.98	76.04	87.56
SSIM ↑	0.99891	0.99964	0.99991	0.99903	0.99970	0.99993	0.99910	0.9990	0.99995

Block size	400			800		
Params (M)	99.83	179.68	302.57	99.04	177.65	300.54
PSNR ↑	67.42	76.53	88.23	66.21	75.31	84.02
SSIM ↑	0.99912	0.99975	0.99995	0.99908	0.99972	0.99991

Table 13: Sensitivity to the number of candidate frequency bands K on Pluto across three model scales. Performance saturates at $K = 200$, while increasing K to 400 provides no consistent improvement despite substantially higher schedule-optimization cost.

Candidate bands K	50			100			200			400		
Params (M)	99.76	179.59	302.44	99.88	179.63	302.52	99.83	179.68	302.57	99.91	179.75	302.61
PSNR ↑	65.00	73.20	83.60	66.60	75.40	86.70	**67.42**	**76.53**	88.23	67.30	76.40	**88.25**
SSIM ↑	0.99979	0.99987	0.99991	0.99983	0.99990	0.99993	0.99986	0.99992	0.99994	0.99985	0.99992	0.99994

L.5 Candidate Resolution Set

We examine the sensitivity of RIM-Hash to the number of candidate frequency bands K used in resolution-schedule optimization. As shown in Table 13, increasing K from 50 to 200 consistently improves reconstruction quality across all three model scales. A small candidate set provides only a coarse discretization of the frequency domain, restricting the schedules available to the dynamic program. Consequently, $K = 50$ produces substantially lower PSNR and SSIM, particularly in the high-capacity regime, where an inaccurate resolution schedule prevents the model from fully utilizing its available parameters. Increasing K from 50 to 100 yields a clear improvement, and increasing it further to 200 provides another consistent gain across all model scales. This result indicates that a sufficiently dense candidate set is important for aligning the hash-grid resolutions with the spatial-frequency distribution of the target image. The improvement is especially pronounced at the largest parameter budget, where PSNR increases from 83.60 dB at $K = 50$ to 88.23 dB at $K = 200$. Increasing K further to 400 provides no consistent benefit. Relative to $K = 200$, the PSNR decreases by 0.12 and 0.13 dB at the two smaller model scales and increases by only 0.02 dB at the largest scale. The corresponding SSIM values are either slightly lower or unchanged. In contrast, the computational cost of the schedule update grows quadratically with K, as the dynamic program has complexity $O(LK^2)$. Doubling K from 200 to 400 therefore approximately quadruples the schedule-search cost without providing a meaningful or consistent improvement in reconstruction fidelity. The parameter counts vary slightly across candidate-set sizes because different values of K can produce different resolution schedules, consequently, different numbers of explicitly stored grid entries at unsaturated levels. However, these variations are small relative to the total model size and do not account for the observed differences in reconstruction quality. Overall, performance largely saturates at $K = 200$, which provides a favorable balance between schedule granularity and optimization cost. We therefore use $K = 200$ for the gigapixel-image and WSI experiments.

M Limitations and Future Work

This work focuses on optimizing the encoder while retaining a lightweight decoder. Consequently, the decoder does not explicitly exploit the spatial correlations inherent in natural signals. Future work will investigate more expressive, spatially aware decoder architectures to further improve reconstruction fidelity. Moreover, directly extending RIM-Hash to NeRF-based novel-view synthesis is nontrivial. The proposed objective requires measuring the band-information of the target signal, whereas the appearance and band-information of unseen viewpoints are not directly observable during training. A promising direction is to estimate view-consistent band-information from the available training views or learn predictive proxies for the RIM objective across viewpoints.

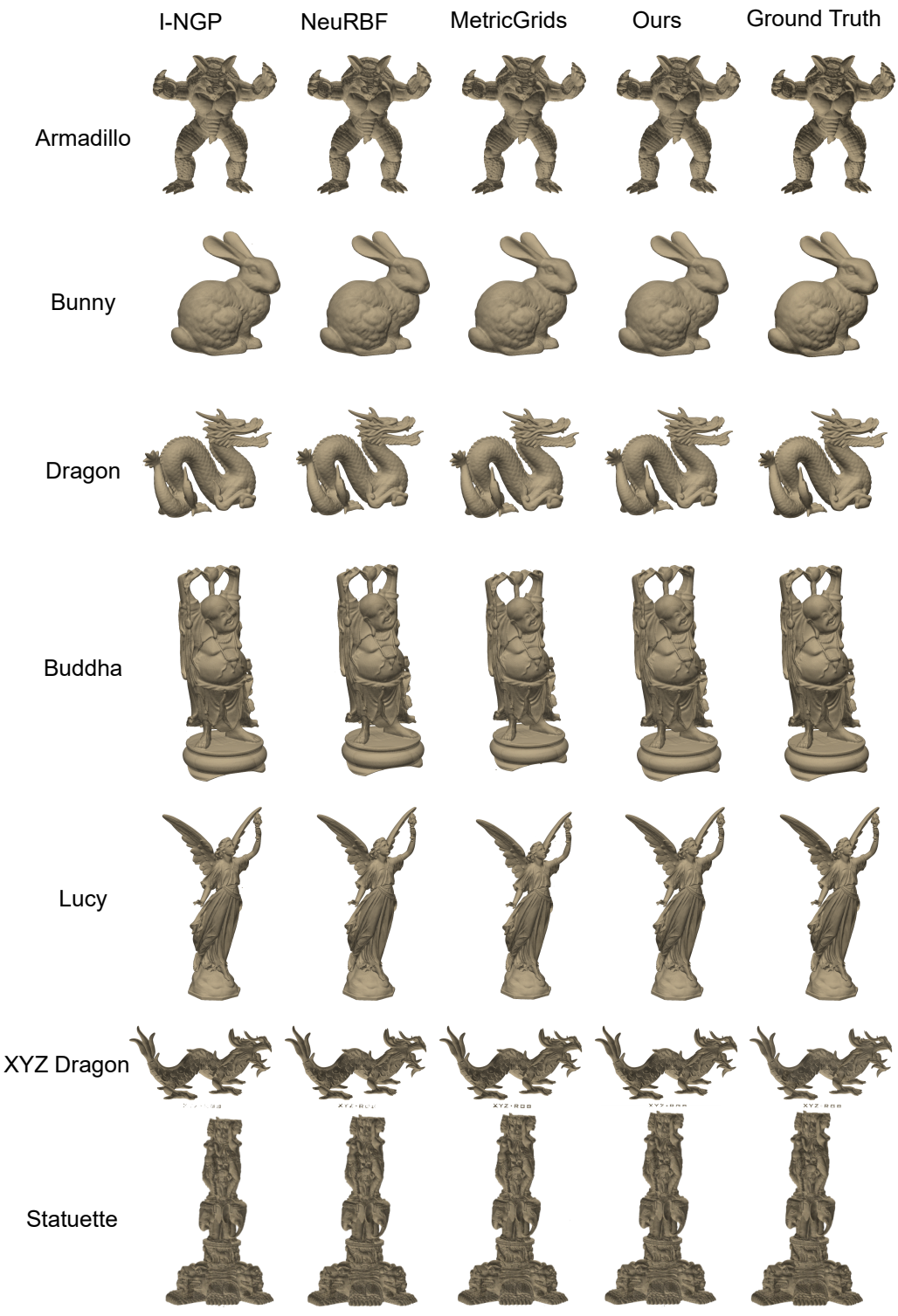

Figure 15: Qualitative comparison across seven SDF datasets. Our method achieves comparable reconstruction quality using only 786K trainable parameters, compared with 856K for NeuRBF and 950K for I-NGP and MetricGrids.