PRISM: PROGRESSIVE REGION-INFORMED STATE MODELING FOR OVERFITTED IMAGE COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Overfitted codecs compress each image by fitting a lightweight decoder and a hierarchical latent tailored to the content. Given the inherent trade-off between model expressiveness and parameter rate, existing methods largely focus on representation design and optimization, leaving explicit modeling of dependencies across the latent hierarchy underexplored. We reveal that maintaining and reusing a shared contextual state across latent grids can improve both entropy modeling and image synthesis. Motivated by this, we introduce **P**rogressive **R**egion-**I**nformed **S**tate **M**odeling (PRISM) for overfitted image compression, which progressively encapsulates cross-scale context into a compact state as entropy modeling proceeds through the latent hierarchy. Gated, region-informed updates adapt the state to spatially varying dependencies while selectively retaining information from previously decoded grids for subsequent probability estimation. The accumulated state is further supplied to the synthesis network, where the retained context complements the latent representations for image reconstruction. Experiments on Kodak and CLIC2020 demonstrate state-of-the-art rate-distortion (RD) performance among overfitted codecs, with BD-rate reductions of 9.56% and 20.35% over VTM-19.1, respectively, approaching strong codecs such as LIC-HPCM at substantially lower decoding complexity. These gains extend across diverse overfitted codecs, highlighting the potential of PRISM as a general paradigm for improving both RD performance and convergence.

1 INTRODUCTION

Learned image compression (LIC) has achieved strong rate-distortion (RD) performance through the joint optimization of neural transforms and entropy models (Ballé et al., 2018; Minnen et al., 2018). Modern neural codecs (He et al., 2022; Jiang et al., 2025; Li et al., 2025) can outperform conventional standards such as HEVC (Sullivan et al., 2012) and VVC (Bross et al., 2021), but often rely on complex transforms and advanced context networks, leading to substantial model size and decoding complexity (Jiang et al., 2025; Ladune et al., 2026). This limits their applicability when decoding resources are constrained.

Overfitted image compression offers an alternative based on image-specific optimization. Early implicit neural representation (INR)-based methods encode images through the parameters or compact modulations of coordinate-based networks (Dupont et al., 2021; 2022), while later approaches combine multi-resolution latent grids with lightweight synthesis networks and autoregressive entropy models (ARMs) (Ladune et al., 2023; Leguay et al., 2023; Philippe et al., 2025). By jointly optimizing latents and network parameters for each image, these methods achieve strong rate-distortion (RD) performance while retaining lightweight decoding (Kim et al., 2024; Wu et al., 2025).

As overfitted codecs improve, the challenge shifts from expanding model capacity to exploiting compact image-specific representations more effectively, particularly the dependencies among multi-resolution latent grids. These dependencies are important for both entropy coding and reconstruction, as also observed in (Kim et al., 2024). Previously decoded grids provide context for predicting subsequent latents, while synthesis integrates information across scales to recover image content. Recent methods increasingly exploit such hierarchical context for entropy modeling. LANCE (Benjak & Ostermann, 2026) introduces a spatial hyperprior for locally adaptive probability estimation, while Cool-chic 5.0 (Ladune et al., 2026) captures inter-grid dependencies through hyperlatents.

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.

However, existing approaches primarily reuse decoded latent values, while contextual features formed during entropy modeling are typically discarded after local probability estimation. Since these features encode cross-grid dependencies, preserving them in a shared state could benefit both subsequent entropy modeling and image synthesis. This reuse is particularly valuable in overfitted compression, where model parameters contribute directly to the coding rate and image-specific representations must be optimized under limited computation. Reusing already computed contextual features can therefore increase effective representation capacity without a proportional increase in the rate or the decoding complexity. Related context-reuse mechanisms in transform-based neural compression (Li et al., 2025) further demonstrate the value of intermediate representations beyond their immediate prediction stage, particularly under limited model capacity (Xu et al., 2020).

Motivated by these, we propose **P**rogressive **R**egion-**I**nformed **S**tate **M**odeling (**PRISM**) for overfitted image compression. PRISM progressively aggregates cross-scale context into a compact persistent state as entropy decoding proceeds through the latent hierarchy, making accumulated information available to subsequent coding stages. To regulate state evolution, a shared update network with learned memory gates is introduced to control context retention and new information incorporation, while region conditioning adapts state updates to spatially varying image statistics (Li et al., 2026). The final state is also supplied to the synthesis network, providing complementary context beyond the decoded latent values and explicitly coupling entropy modeling with reconstruction.

PRISM is built on three principles: (i) reusing intermediate context beyond local probability estimation, (ii) selectively regulating their propagation across the latent hierarchy, and (iii) adapting state updates to spatially heterogeneous content. Together, these mechanisms enable efficient context reuse within a compact architecture. PRISM achieves state-of-the-art (SOTA) RD performance among overfitted codecs with low decoding complexity and consistent gains across different codec architectures. The main contributions are summarized as follows:

- We propose PRISM, a progressive state-modeling framework that preserves and reuses intermediate representations formed during entropy modeling across the latent hierarchy. Its compact persistent state supports both subsequent entropy modeling and image synthesis, turning entropy features into reusable context for more efficient overfitted compression.
- We introduce a lightweight gated state-update mechanism that selectively propagates context across coding stages and integrates region-aware conditioning for spatial adaptation, enabling structured hierarchical context modeling with minimal model overhead.
- Extensive experiments demonstrate that PRISM achieves SOTA RD performance among overfitted codecs with low decoding complexity, substantially outperforming existing overfitted codecs and approaching strong neural codec LIC-HPCM on CLIC2020.
- We show that progressive state reuse generalizes across diverse overfitted codecs, consistently improving both RD performance and convergence, and establishing contextual state reuse as a general mechanism for improving coding efficiency.

2 Related work

2.1 Overfitted Image Compression

Overfitted image compression fits a compact neural representation to each image, using the resulting image-specific representation as the compressed code. Early approaches such as COIN and COIN++ (Dupont et al., 2021; 2022) encode images through quantized parameters of coordinate-based networks, while COMBINER (Guo et al., 2023) combines variational inference with relative entropy coding. Cool-chic (Ladune et al., 2023) establishes a latent-based framework comprising multi-resolution latent grids, a lightweight synthesis network, and an entropy model. Subsequent work (Leguay et al., 2023; Kim et al., 2024) improves this framework through advances in synthesis, quantization, and optimization. More recently, LotteryCodec (Wu et al., 2025) reduces model-rate overhead using binary masks over shared random weights, MoRIC (Li et al., 2026) introduces region-wise synthesis, and perceptual variants incorporate Wasserstein distortion objectives (Ballé et al., 2025; Wu et al.). Richer contextual information has also been exploited to improve entropy modeling. LANCE (Benjak & Ostermann, 2026) introduces a spatial hyperprior for locally adaptive probability estimation, while Cool-chic 5.0 (Ladune et al., 2026) captures inter-grid dependencies

through hyperlatents and an inter-feature context extractor. These advances highlight the potential for explicit modeling and reuse of context throughout the latent hierarchy.

2.2 Region Partitioning for implicit neural representations

INRs model signals as coordinate-dependent functions and have been widely used for image and scene representation (Sitzmann et al., 2020; Mildenhall et al., 2021; Chen et al., 2021a). To better capture spatially heterogeneous content, prior work decomposes representations into local grids or patches (Jiang et al., 2020; Tretschk et al., 2020), and conditions shared networks on region-specific latent codes or learnable modulations (Chabra et al., 2020; Mehta et al., 2021). Subsequent approaches explore multi-resolution and structured local representations (Chen et al., 2021b; Wang et al., 2023b; Yang, 2023), while complementary work improves fitting efficiency through coarse-to-fine optimization, semantic partitioning, coordinate factorization, and local-global network design (Saragadam et al., 2022; Liu et al., 2023; Liang et al., 2022; Ashkenazi & Treister, 2024). Region-aware coding has also been investigated in image compression, from shape-adaptive coding of irregular regions (Sikora et al., 1995; Li & Li, 2000; Schmalz, 2004) to INR-based approaches such as MoRIC (Li et al., 2026), which uses region-specific synthesis with shared global modulation, and Re^{2}IC (Wu et al.), which incorporates saliency-guided partitions into perceptual optimization. These works demonstrate the value of region structure for spatial adaptation, motivating its use as conditioning information for context modeling across the latent hierarchy.

2.3 Gated State and Memory Modeling

Gated recurrent architectures regulate how information is retained and updated within a compact hidden state. LSTM (Hochreiter & Schmidhuber, 1997; Gers et al., 2000) and GRU (Cho et al., 2014) establish this principle through adaptive memory and transition gates, while more recent models such as HGRN (Qin et al., 2023) and HGRN2 (Qin et al., 2024) extend gated recurrence to richer state representations and memory timescales. Related ideas also appear in linear attention, where attention can be expressed through recurrent state accumulation (Katharopoulos et al., 2020). Models including RetNet (Sun et al., 2023), gated linear attention (Yang et al., 2023), bounded-memory control (Peng et al., 2022), and gated slot attention (Zhang et al., 2024) further regulate the accumulation, forgetting, and retrieval of contextual information within compact states. Complementary work on learnable residual scaling (Bachlechner et al., 2021; Touvron et al., 2021) controls the contribution of individual updates during feature propagation. Together, these studies motivate jointly regulating context retention and update strength in progressive state modeling.

3 Methodology

3.1 Preliminary

Standard overfitted codec. Standard overfitted codecs represent an image $\mathbf{S} \in \mathbb{R}^{H\times W\times 3}$ using specific latent grids $\mathbf{z} \triangleq \{\mathbf{z}_1, \ldots, \mathbf{z}_T\}$, where $\mathbf{z}_t \in \mathbb{R}^{H_t\times W_t}$, together with a synthesis network $g_{\mathbf{W}}$. As illustrated in Fig. 1(a), the latents and network parameters are jointly optimized through an overfitting procedure $\mathcal{F}(\cdot)$, followed by quantization $\mathcal{Q}(\cdot)$ and entropy coding $\mathcal{A}(\cdot)$. Reconstruction process $\mathcal{G}(\cdot)$ generates $\hat{\mathbf{S}} = g_{\hat{\mathbf{W}}}(\hat{\mathbf{z}})$[1]. Auxiliary hyperlatents may further condition entropy decoding $\mathcal{D}(\cdot)$ (Ladune et al., 2026).

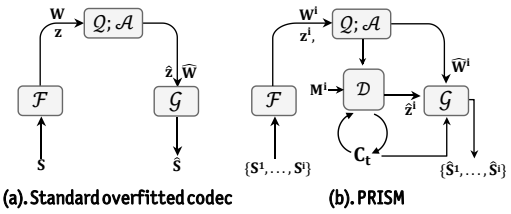


Figure 1: Standard overfitted codec vs. PRISM.

This results in an average distortion $D = \mathbb{E}_{\mathbf{S}\sim p_s}\left[d(\mathbf{S}, g_{\hat{\mathbf{W}}}(\hat{\mathbf{z}}))\right]$ at a rate $R = \mathbb{E}_{\mathbf{S}\sim p_s}\left[-\log_2 p_{\hat{\psi}}(\hat{\mathbf{z}}) - \log_2 p(\hat{\mathbf{W}}) + R_{\hat{\psi}}\right]$, where d denotes a distortion metric, $p_{\hat{\psi}}$ is typically parameterized by a lightweight autoregressive entropy model (ARM), and $R_{\hat{\psi}}$ accounts for transmitting its parameters.

[1]Throughout this paper, superscripts $\cdot^i$ denote region indices, and $\hat{\cdot}$ denotes quantized variables.

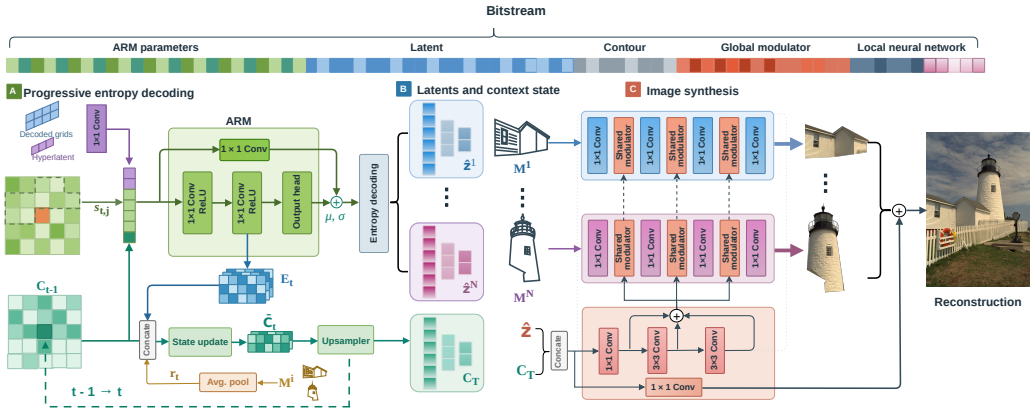


Figure 2: Overview of PRISM. (A) Context states accumulate ARM features across latent grids and condition subsequent entropy decoding. (B) Region information guides both state modeling and synthesis. (C) The final state is reused with decoded latents for image reconstruction.

PRISM. As illustrated in Fig. 1(b), PRISM augments the standard overfitted codec with region-informed state modeling and maintains a compact context state $\mathbf{C}_t$ throughout the latent hierarchy. The image is partitioned into N regions with masks $\{\mathbf{M}^i\}_{i=1}^N$. At grid t, the accumulated state $\mathbf{C}_{t-1}$ complements causal spatial context to estimate each regional latent distribution $p_{\hat{\psi}}\left(\hat{\mathbf{z}}_t^i \mid \hat{\mathbf{z}}_{<t}, \mathbf{C}_{t-1}, \mathbf{M}^i\right)$. The decoded regional latents are merged as $\hat{\mathbf{z}}_\mathbf{t} = \bigcup_{i=1}^N (\hat{\mathbf{z}}_\mathbf{t}^\mathbf{i}, \mathbf{M}^\mathbf{i})$, where $\bigcup$ denotes the region merging. The state is then updated using the current entropy-model features and region information and propagated to subsequent grids. After decoding the latent hierarchy, the final state $\mathbf{C}_T$ is reused for synthesis $\hat{\mathbf{S}} = \bigcup_{i=1}^N \mathcal{G}\left(\hat{\mathbf{W}}_i, \hat{\mathbf{z}}_i, \mathbf{C}_T, \mathbf{M}_i\right)$, allowing context formed during entropy decoding to also support reconstruction. The resulting expected distortion and rate are $D = \mathbb{E}_{\mathbf{S}\sim p_s}\left[d(\mathbf{S}, \hat{\mathbf{S}})\right]$ and $R = \mathbb{E}_{\mathbf{S}\sim p_s}\left[\sum_{i=0}^N \left[-\log_2 p_{\hat{\psi}}(\hat{\mathbf{z}}^\mathbf{i}) - \log_2 p(\hat{\mathbf{W}}^\mathbf{i}) - \log_2 p(\mathbf{M}^\mathbf{i})\right] + R_{\hat{\psi}}\right]$, where a shared ARM is employed across regions.

3.2 Progressive Region-Informed State Modeling

As shown in Fig. 2, PRISM maintains a compact region-informed context state across the latent hierarchy through three components: region-informed state representation, progressive cross-grid updates, and learned memory gating for selective context propagation.

Region-informed context-state representation. Following the low-to-high-resolution latent decoding order $t = 1, \ldots, T$, PRISM propagates a context state initialized as $\mathbf{C}_0 = \mathbf{0} \in \mathbb{R}^{H_1 \times W_1 \times c}$ with fixed channel width c. At grid t, the current state $\mathbf{C}_{t-1} \in \mathbb{R}^{H_t \times W_t \times c}$ carries context accumulated from previously decoded grids. Its update is conditioned on the current ARM features $\mathbf{E}_t \in \mathbb{R}^{H_t \times W_t \times d}$ and region information. We obtain N region masks $\{\mathbf{M}^i\}_{i=1}^N$ from SAM (Kirillov et al., 2023), with their contours transmitted via adaptive chain coding (Chang et al., 1992). At grid t, the masks are average-pooled to the latent resolution $\mathbf{r}_t = \mathrm{AvgPool}_{H_t, W_t}\left([\mathbf{M}^1; \ldots; \mathbf{M}^N]\right) \in \mathbb{R}^{H_t \times W_t \times N}$, where each channel represents the local occupancy of a region. After decoding $\mathbf{z}_t$, $\mathbf{E}_t$ and $\mathbf{r}_t$ jointly update the state to $\mathbf{C}_t \in \mathbb{R}^{H_{t+1} \times W_{t+1} \times c}$, forming a progressive region-informed representation to support subsequent probability estimation and image synthesis when $t = T$.

Progressive state updates. To update $\mathbf{C}_t$, we employ a linear layer $G_{\boldsymbol{\eta}} : \mathbb{R}^{c+d+N} \to \mathbb{R}^c$ to fuse the retained state $\mathbf{C}_{t-1}$, current ARM features $\mathbf{E}_t$ and region information at each spatial position j:

$$\bar{\mathbf{C}}_t(j) = \delta\, \mathbf{C}_{t-1}(j) + \alpha_t\, \Delta\mathbf{C}_t(j) \in \mathbb{R}^c, \quad \text{where} \quad \Delta\mathbf{C}_t(j) = G_{\boldsymbol{\eta}}\big([\mathbf{C}_{t-1}(j); \mathbf{E}_t(j); \mathbf{r}_t(j)]\big). \tag{1}$$

Here $[\cdot;\cdot]$ denotes channel-wise concatenation, δ is a shared decay coefficient, and α_t is a grid-specific trust coefficient. The updated state $\mathbf{C}_t$ is obtained as $\mathbf{C}_t = \mathcal{U}_{\boldsymbol{\kappa}}(\bar{\mathbf{C}}_t; H_{t+1}, W_{t+1})$, where

$\mathcal{U}_{\boldsymbol{\kappa}}(\cdot) : \mathbb{R}^{H_t \times W_t \times c} \to \mathbb{R}^{H_{t+1} \times W_{t+1} \times c}$ denotes learned separable upsampler shared across resolution transitions. For more details, we refer to Fig. 9.

Learned memory gating. We parameterize the gate coefficients δ and α_t from learnable scalar logitsas $\beta_\delta, \beta_{\alpha,t} \in \mathbb{R}$ as:

$$\delta = \text{sigmoid}(\beta_\delta), \qquad \alpha_t = \text{sigmoid}(\beta_{\alpha,t}), \tag{2}$$

where shared gate δ controls the strength of the retained-state branch, while the per-grid coefficient α_t regulates the contribution of the update produced by $G_{\boldsymbol{\eta}}$ at grid t. Both coefficients are shared across spatial positions and channels, providing stage-dependent control over context retention and update with only $T+1$ scalar parameters. The gates are jointly optimized for each image, quantized, and transmitted with the codec parameters.

3.3 Context-State Reuse

The context state is reused in two complementary ways: condition probability estimation during entropy decoding and complement decoded latents during image synthesis. This reuse principle is architecture-agnostic and readily extends to other overfitted codecs.

Entropy Modeling. At position j of grid t, the ARM takes the causal spatial context $\mathbf{s}_{t,j} \in \mathbb{R}^u$ together with inter-grid context $\mathbf{s}_{t-1,j} \in \mathbb{R}^u$ from previously decoded grids, where u denotes the context size. Following (Ladune et al., 2026), a lightweight MLP combines these inputs with the hyperlatent to produce an intermediate feature $\mathbf{f}_{t,j} \in \mathbb{R}^6$. We further extract the ARM feature from the final hidden layer before the output head:

$$\mathbf{E}_t(j) = h_{\boldsymbol{\psi}}\big([\mathbf{s}_{t,j}; \mathbf{f}_{t,j}]\big) \in \mathbb{R}^d. \tag{3}$$

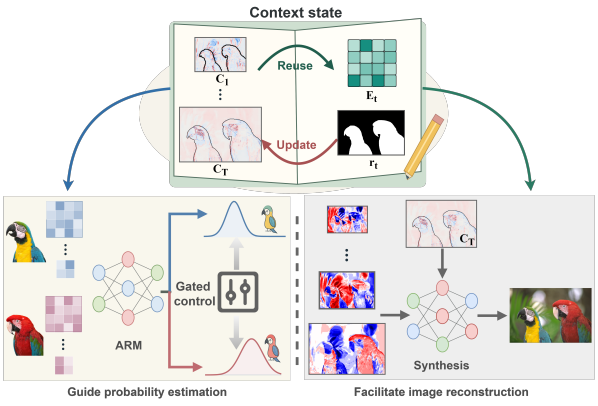


Figure 3: Context-state reuse for entropy modeling and image synthesis, where early states condition probability estimation, while the final state facilitates image synthesis.

Here $h_{\boldsymbol{\psi}}(\cdot)$ denotes the ARM hidden-feature extractor applied to the concatenated input $[\mathbf{s}_{t,j}; \mathbf{f}_{t,j}]$. Collecting all spatial positions gives $\mathbf{E}_t \in \mathbb{R}^{H_t \times W_t \times d}$. To reuse the accumulated state for entropy modeling, a shared linear projection $B(\cdot) : \mathbb{R}^{6+u+c} \to \mathbb{R}^2$ produces a residual correction to the ARM distribution prediction:

$$(\mu_{t,j}, \sigma_{t,j}) = \text{ARM}_{\boldsymbol{\psi}}\big([\mathbf{s}_{t,j}; \mathbf{f}_{t,j}]\big) + B\big([\mathbf{s}_{t,j}; \mathbf{f}_{t,j}; \mathbf{C}_{t-1}(j)]\big). \tag{4}$$

Thus, $\mathbf{C}_{t-1}$ complements local and inter-grid context with information retained from earlier decoding stages. After decoding, the collected features $\mathbf{E}_t$ update the state according to equation 1.

Image Synthesis. The state $\mathbf{C}_T$ exposes accumulated context during entropy decoding to synthesis network. Specifically, $\mathbf{C}_T$ is aligned to the image resolution and refined by a filter $\Phi(\cdot)$:

$$\Phi(\mathbf{C}_T) = \mathbf{C}_T + \mathcal{K}(\mathbf{C}_T), \tag{5}$$

where $\mathcal{K} : \mathbb{R}^{H \times W \times c} \to \mathbb{R}^{H \times W \times c}$ is a resolution-preserving symmetric convolution shared across state channels. We then concatenate it with an upsampler $\mathcal{U}_z$ to construct the input for synthesis:

$$\mathbf{X}_{\text{syn}} = [\mathcal{U}_z(\hat{\mathbf{z}}); \Phi(\mathbf{C}_T)] \in \mathbb{R}^{H \times W \times (c+T)}, \tag{6}$$

where $\mathbf{X}_{\text{syn}}$ is fed into a shared global network $f_{\hat{\boldsymbol{\theta}}_g}$ to produce modulation features $\mathbf{F}_{\text{mod}} = f_{\hat{\boldsymbol{\theta}}_g}(\mathbf{X}_{\text{syn}})$, which condition the region-specific INR $g_{\mathbf{W}^i}$ as $\hat{\mathbf{S}}^i(\mathbf{x}_i) = g_{\hat{\mathbf{W}}^i}(\mathbf{F}_{\text{mod}}, \mathbf{x}^i)$ with spatial

coordinates $\mathbf{x}^i$ as input. The regional predictions are merged and combined with a linear residual branch $L_{\boldsymbol{\theta}_{\text{res}}} : \mathbb{R}^{c+T} \to \mathbb{R}^3$ as

$$\hat{\mathbf{S}} = \bigcup_{i=1}^{N} \hat{\mathbf{S}}^i + L_{\boldsymbol{\theta}_{\text{res}}}(\mathbf{X}_{\text{syn}}). \tag{7}$$

Because $\mathbf{C}_T$ is progressively formed from ARM representations and reused in synthesis, reconstruction gradients propagate through the state pathway to couple entropy modeling and reconstruction.

3.4 Rate-Distortion Optimization

To optimize the RD trade-off, PRISM jointly fits the latent hierarchy and decoder parameters, collectively denoted by $\boldsymbol{\Omega}$, using $\mathcal{L}(\boldsymbol{\Omega}) = d(\mathbf{S}, \hat{\mathbf{S}}) + \lambda\big[R_{\text{lat}} + R_{\text{mask}} + R_{\text{model}}\big]$, where λ controls the RD trade-off. The rate terms, measured in bits per pixel, account for the main and hyperlatent grids R_{lat}, transmitted region masks R_{mask}, and shared and region-specific model parameters R_{model}, respectively. We use differentiable quantization during fitting, followed by hard-rounding refinement and RD-aware parameter quantization. Implementation details are provided in Appendix D.2.

4 Experimental results

4.1 Experiment setup

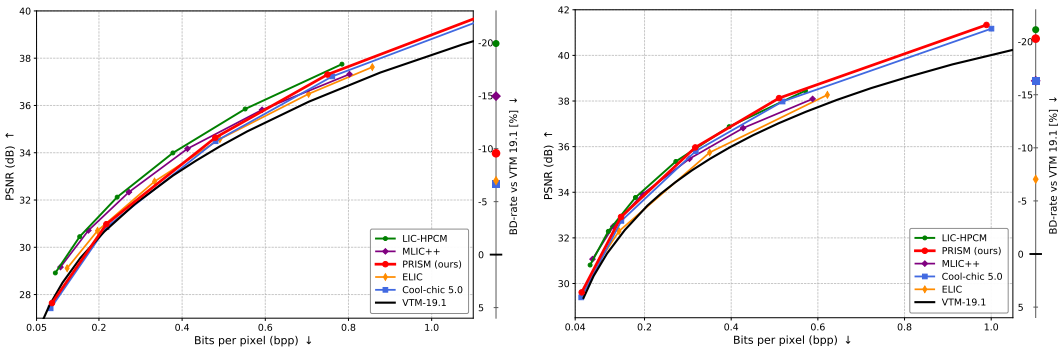
We evaluate PRISM on Kodak (Kodak, 1991) and the CLIC2020 datasets (Toderici et al., 2020), comparing it with the conventional codec VTM 19.1 (Bross et al., 2021), autoencoder-based codecs ELIC (He et al., 2022), MLIC++ (Jiang et al., 2025), and LIC-HPCM (Li et al., 2025), and overfitted codecs C3 (Kim et al., 2024), LotteryCodec (Wu et al., 2025), MoRIC (Li et al., 2026), and Coolchic 5.0 (Ladune et al., 2026). Baseline implementations and PRISM configurations are detailed in Appendices C and D, respectively. RD performance is assessed using peak signal-to-noise ratio (PSNR) and BD-rate (Gisle, 2001) relative to VTM 19.1, with bitrate reported in bpp. Decoding complexity is measured in multiply–accumulate operations per pixel (MACs/pixel), while encoding convergence is examined under different optimization budgets.

Specifically, we first present the overall RD performance, followed by an analysis of encoding convergence and decoding complexity. We then evaluate the generality of PRISM across different codec architectures and investigate its components through ablation studies and bit-allocation analysis. Finally, we present visual comparisons of reconstructed details and compression artifacts, alongside visualizations of context-state evolution and its reuse in image synthesis.

4.2 Performance analysis

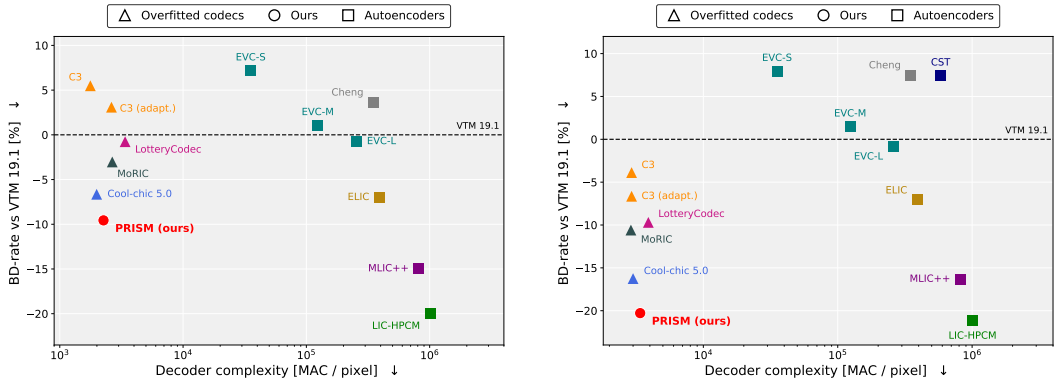
RD performance. As shown in Fig. 4, PRISM achieves state-of-the-art RD performance among the evaluated overfitted codecs. It surpasses VTM-19.1 on Kodak and CLIC2020 with BD-rate reductions of 9.56% and 20.35%, respectively. PRISM achieves a -2.95% BD-rate reduction over the strong codec COOL-CHIC 5.0 on Kodak. The advantage is more pronounced on CLIC2020, where the reduction reaches 4.99%. PRISM also outperforms ELIC (-6.98%) on kodak and achieves RD performance competitive with LIC-HPCM (-21.12%) on CLIC2020, narrowing the gap between overfitted and autoencoder-based codecs. Although MLIC++ and LIC-HPCM retain stronger RD performance on Kodak, PRISM requires significantly fewer decoder MACs per pixel than these models, preserving the decoding advantage of overfitted compression. These results establish a new RD benchmark for overfitted compression and demonstrate the effectiveness of progressive context-state reuse under a compact decoding budget. Extended RD comparisons over more baselines can be seen in Appendix E.1, Figs. 10 and 11.

Decoding efficiency. Fig. 5 compares RD performance against decoder complexity. PRISM achieves best RD performance among the evaluated overfitted codecs while maintaining comparable decoding complexity. Its decoder requires approximately 2.5k and 3.5k MACs per pixel on Kodak and CLIC2020, respectively. Compared with autoencoder-based codecs, PRISM reduces decoding complexity by over $100\times$ relative to ELIC, over $200\times$ relative to MLIC++, and approximately



(a) Kodak dataset (b) CLIC2020 dataset

Figure 4: RD performance with BD-rate results for standard image compression on (a) Kodak and (b) CLIC2020 datasets. PRISM approaches LIC-HPCM on CLIC2020, highlighting the potential of overfitted codecs to challenge leading autoencoder-based methods.



(a) Kodak dataset (b) CLIC2020 dataset

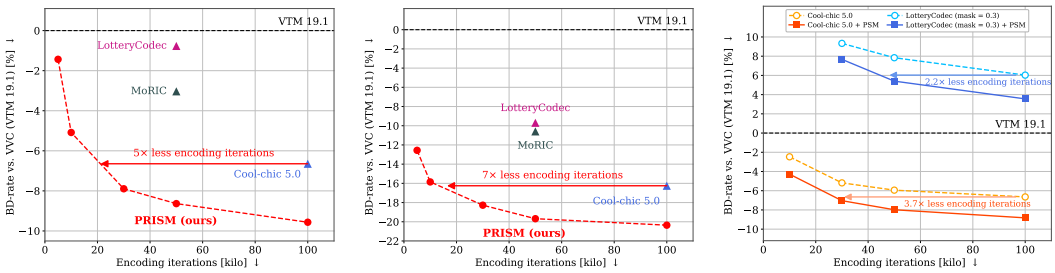
Figure 5: Trade-off between RD performance and decoding complexity on (a) Kodak and (b) CLIC2020 datasets. PRISM approaches LIC-HPCM's RD performance on CLIC2020 with orders of magnitude fewer decoder MACs, demonstrating a favorable RD-complexity trade-off.

300-400× relative to LIC-HPCM-Large. Notably, on CLIC2020, PRISM approaches the RD performance of LIC-HPCM-Large while preserving this substantial complexity advantage. These results highlight the potential of progressive context reuse to narrow the RD performance gap between overfitted and autoencoder-based codecs, while preserving the lightweight decoding advantage.

Encoding convergence. High image-specific fitting cost remains a major challenge for overfitted codecs, making convergence speed particularly important. We therefore evaluate PRISM with fitting budget from 5k to 100k iterations against other overfitted codecs (Fig. 6). With only 5k iterations, PRISM already surpasses VTM 19.1 on Kodak and CLIC2020, achieving BD-rates of -1.43% and -12.56%, respectively. At 10k iterations, it outperforms both MoRIC and LotteryCodec trained for 50k iterations. The plotted curves further indicate that PRISM matches the RD performance of Cool-chic 5.0 trained for 100k iterations using approximately $5\times$ and $7\times$ fewer fitting iterations on Kodak and CLIC2020, respectively. These results demonstrate that PRISM requires substantially fewer fitting iterations toward a target RD performance, helping alleviate the encoding burden of overfitted codecs. More detailed results are provided in Appendix E.2, Tables 11 and 12.

4.3 Generality across different codecs

Progressive state modeling can be integrated into different overfitted codecs. We quantitatively evaluate its impact on RD performance and convergence across different codecs.



(a) Kodak dataset (b) CLIC2020 dataset (c) Cross-codec convergence

Figure 6: Encoding convergence across datasets and codecs: (a) Kodak, (b) CLIC2020. (c) Convergence with and without progressive state modeling for different codecs on Kodak.

Consistent performance gain. To assess generality, we incorporate progressive state modeling (PSM) into Cool-chic 5.0 and LotteryCodec on Kodak. At 100k fitting iterations, PSM consistently improves approximately 2% BD performance across the evaluated configurations (see Fig. 7). The BD-rate relative to VTM 19.1 improves from -6.65% to -8.83% for Cool-chic 5.0 and from $+6.03\%$ to $+3.56\%$ for LotteryCodec, yielding gains of 2.18 and 2.47 percentage points, respectively. A similar benefit is observed within the PRISM configuration, where enabling PSM improves the BD-rate from -7.60% to -9.56%. These consistent gains across distinct codecs demonstrate that progressive state modeling can serve as a general mechanism for improving overfitted compression.

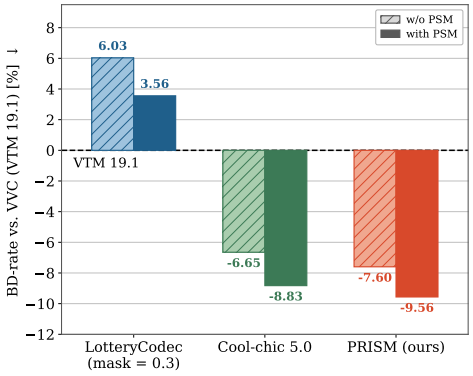


Figure 7: RD performance with and w/o PSM across different codecs on Kodak.

Consistent convergence improvement. Both backbones benefit from PSM at each evaluated fitting budget, as shown in Fig. 6(c) and Table 14 in Appendix E.3. Cool-chic 5.0 with PSM achieves a BD-rate of -7.04% after only 30k iterations, surpassing its original 100k-iteration result (-6.65%). Similarly, LotteryCodec with PSM reaches $+5.41\%$ after 50k iterations, outperforming its original result at 100k iterations ($+6.03\%$). The plotted curves indicate that PSM matches the original codecs' 100k-iteration performance using approximately $3.7\times$ and $2.2\times$ fewer fitting iterations, respectively. These consistent gains establish PSM as a reusable mechanism for improving both RD performance and optimization efficiency across overfitted codecs.

4.4 Visualization

Compression quality. Appendix F presents visual comparisons on Kodak and CLIC2020 (Figs. 13-15), where PRISM yields better reconstructions with fewer artifacts (Figs. 20 and 21). Fig. 16 further visualizes latent hierarchy, revealing how spatial information is progressively organized.

Context-state evolution. Figs. 24 and 25 visualize how the context state progressively evolves across the latent hierarchy, while Figs. 22 and 23 show how state reuse influences latent learning and reconstruction. The context states capture spatially structured and semantically meaningful information, providing complementary context for both entropy modeling and image synthesis.

Context-state contribution. We ablate the context state at selected decoding stages to assess its impact on entropy rate and reconstruction quality. Fig. 8(a) reveals increasingly detailed spatial structure from early to late states. At 0.0564 bpp, the early and middle states yield relative rate savings of 7.28% and 2.60%, whereas at 0.2338 bpp the largest saving shifts to the late state (2.31%), indicating greater reliance on finer-scale context at higher rates. State ablation also reduces PSNR

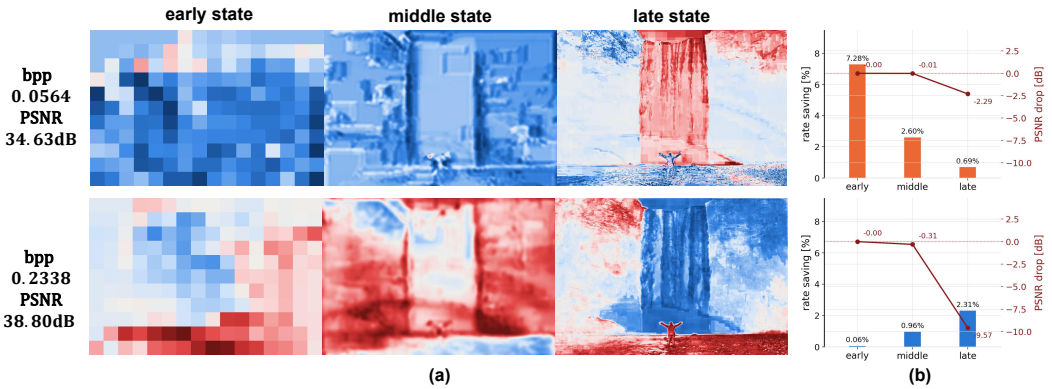


Figure 8: Progressive state visualization and contribution at two RD operating points on a CLIC2020 image. (a) Context states $\mathbf{C}_1$, $\mathbf{C}_8$, and $\mathbf{C}_{11}$. (b) Corresponding rate savings and PSNR drops when the state is ablated. Top and bottom rows use $\lambda = 4 \times 10^{-3}$ and 4×10^{-4}, respectively.

at both operating points, with later states contributing more strongly to synthesis. These results confirm that progressive state reuse provides complementary context for both entropy modeling and reconstruction. Additional examples are provided in Fig. 26.

4.5 Component ablation.

Table 1 reports component ablations on Kodak, with full PRISM as the BD-rate reference. Removing progressive state modeling causes the largest performance degradation ($+2.02\%$), confirming that cross-grid state propagation is the primary source of improvement. Disabling state reuse for synthesis incurs a comparable $+1.69\%$ performance drop, showing that context accumulated during entropy decoding remains valuable beyond probability estimation and provides complementary information for reconstruction. Learned memory gating and region conditioning further improve BD-rate by 0.81% and 0.62%, respectively, indicating that selective state regulation and spatially adaptive updates both contribute to effective context reuse. Overall, the ablations support the central design of PRISM: progressively accumulating context across the latent hierarchy, selectively refining its evolution, and reusing the resulting representation for both entropy modeling and synthesis.

Table 1: Ablation on Kodak, with full PRISM as the BD reference.

System	BD-rate ↓
PRISM (full)	0.00%
CC 5.0	$+2.95\%$
w/o State modeling	$+2.02\%$
w/o Synthesis state reuse	$+1.69\%$
w/o Learned memory gating	$+0.81\%$
w/o Region conditioning	$+0.62\%$

5 Conclusion and future work

We presented PRISM, a progressive state modeling framework that connects hierarchical entropy modeling and image synthesis in overfitted compression. PRISM accumulates contextual representations during entropy decoding, regulates their evolution through learned gating and region conditioning, and reuses the resulting state for both subsequent probability estimation and reconstruction. Experiments on Kodak and CLIC2020 demonstrate the SOTA RD performance among overfitted codecs, reducing BD-rate by 9.56% and 20.35% relative to VTM 19.1 while retaining low decoding complexity. It also converges with substantially fewer fitting iterations, while consistent gains on various codecs further demonstrate progressive state reuse as a general mechanism for improving both compression performance and convergence in overfitted compression.

Despite improved convergence, PRISM still requires per-image iterative optimization, leaving encoding cost as a practical limitation. Future work will explore faster encoding, adaptive architectures, and extensions of PRISM to video compression with temporal context.

AI USE STATEMENT

We used generative AI tools to assist with English-language polishing, grammatical correction, and improving manuscript readability. AI-assisted tools were also occasionally used for debugging experimental scripts and formatting tables.

Generative AI was not used to formulate the core research ideas, derive mathematical formulations, design model architectures, generate experimental results, or substantiate scientific claims. All AI-assisted outputs were reviewed, verified, and edited by the authors, and any generated code was independently tested and validated. The authors take full responsibility for all content, analyses, claims, and results presented in this work.

REPRODUCIBILITY STATEMENT

We will release per-image RD results, reconstructions, implementation details, and configurations for the proposed state mechanism. All code and checkpoints will be released after the review.

REFERENCES

Maor Ashkenazi and Eran Treister. Towards croppable implicit neural representations. *Advances in Neural Information Processing Systems*, 37:31473–31503, 2024.

Thomas Bachlechner, Bodhisattwa Prasad Majumder, Henry Mao, Gary Cottrell, and Julian McAuley. Rezero is all you need: Fast convergence at large depth. In *Uncertainty in artificial intelligence*, pp. 1352–1361. PMLR, 2021.

Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018.

Jona Ballé, Luca Versari, Emilien Dupont, Hyunjik Kim, and Matthias Bauer. Good, cheap, and fast: Overfitted image compression with wasserstein distortion. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23259–23268. IEEE, 2025.

Martin Benjak and Jörn Ostermann. Lance: Locally adaptive neural context estimation for overfitted image compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 2026.

Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J Sullivan, and Jens-Rainer Ohm. Overview of the versatile video coding (vvc) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021.

Rohan Chabra, Jan E Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *European Conference on Computer Vision*, pp. 608–625. Springer, 2020.

Nelson L Chang, Sanjeev R Kulkarni, and Jack Koplowitz. Adaptive chain coding for arbitrary curves. In *Curves and Surfaces in Computer Vision and Graphics III*, volume 1830, pp. 296–307. SPIE, 1992.

Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8628–8638, 2021a.

Zhang Chen, Yinda Zhang, Kyle Genova, Sean Fanello, Sofien Bouaziz, Christian Häne, Ruofei Du, Cem Keskin, Thomas Funkhouser, and Danhang Tang. Multiresolution deep implicit functions for 3d shape representation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13067–13076. IEEE, 2021b.

Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7936–7945. IEEE, 2020.

Kyunghyun Cho, Bart Van Merriënboer, Çağlar Gulçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1724–1734, 2014.

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

Emilien Dupont, Hrushikesh Loya, Milad Alizadeh, Adam Goliński, Yee Whye Teh, and Arnaud Doucet. Coin++: Neural compression across modalities. *arXiv preprint arXiv:2201.12904*, 2022.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.

Bjontegaard Gisle. Calculation of average psnr differences between rd curves. In *ITU-T SG16 Doc. VCEG-M33*, 2001.

Zongyu Guo, Gergely Flamich, Jiajun He, Zhibo Chen, and José Miguel Hernández-Lobato. Compression with bayesian implicit neural representations. *Advances in Neural Information Processing Systems*, 36:1938–1956, 2023.

Dailan He, Ziming Yang, Weikun Peng, Rui Ma, Hongwei Qin, and Yan Wang. Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5708–5717. IEEE, 2022.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3d scenes. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6000–6009. IEEE, 2020.

Wei Jiang, Jiayu Yang, Yongqi Zhai, Peirong Ning, Feng Gao, and Ronggang Wang. Mlic: Multi-reference entropy model for learned image compression. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, pp. 7618–7627. ACM, October 2023. doi: 10.1145/3581783.3611694. URL http://dx.doi.org/10.1145/3581783.3611694.

Wei Jiang, Jiayu Yang, Yongqi Zhai, Feng Gao, and Ronggang Wang. Mlic++: Linear complexity multi-reference entropy modeling for learned image compression. *ACM Transactions on Multimedia Computing, Communications and Applications*, 21(5):1–25, 2025.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020.

Hyunjik Kim, Matthias Bauer, Lucas Theis, Jonathan Richard Schwarz, and Emilien Dupont. C3: High-performance and low-complexity neural compression from a single image or video. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9347–9358. IEEE, 2024.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *2023 IEEE/CVF international conference on computer vision (ICCV)*, pp. 3992–4003. IEEE, 2023.

Kodak. Kodak dataset. 1991. URL http://r0k.us/graphics/kodak/.

Théo Ladune, Pierrick Philippe, Félix Henry, Gordon Clare, and Thomas Leguay. Cool-chic: Coordinate-based low complexity hierarchical image codec. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13469–13476. IEEE, 2023.

Théo Ladune, Pierrick Philippe, Pierre Jaffuer, Théophile Blard, Sylvain Kervadec, Félix Henry, and Gordon Clare. Cool-chic 5.0: Faster encoding and inter-feature entropy modeling for overfitted image compression. *arXiv preprint arXiv:2605.02726*, 2026.

Thomas Leguay, Théo Ladune, Pierrick Philippe, Gordon Clare, Félix Henry, and Olivier Déforges. Low-complexity overfitted neural image codec. In *2023 IEEE 25th International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6. IEEE, 2023.

Gen Li, Haotian Wu, and Deniz Gunduz. Moric: A modular region-based implicit codec for image compression. *Advances in Neural Information Processing Systems*, 38:39361–39400, 2026.

Shipeng Li and Weiping Li. Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(5):725–743, 2000.

Yuqi Li, Haotian Zhang, Li Li, and Dong Liu. Learned image compression with hierarchical progressive context modeling. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18834–18843. IEEE, 2025.

Ruofan Liang, Hongyi Sun, and Nandita Vijaykumar. Coordx: Accelerating implicit neural representation with a split mlp architecture. *arXiv preprint arXiv:2201.12425*, 2022.

Ke Liu, Feng Liu, Haishuai Wang, Ning Ma, Jiajun Bu, and Bo Han. Partition speeds up learning implicit neural representations based on exponential-increase hypothesis. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5451–5460. IEEE, 2023.

Ishit Mehta, Michaël Gharbi, Connelly Barnes, Eli Shechtman, Ravi Ramamoorthi, and Manmohan Chandraker. Modulated periodic activations for generalizable local functional representations. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14194–14203. IEEE, 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.

David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. *Advances in neural information processing systems*, 31, 2018.

Hao Peng, Jungo Kasai, Nikolaos Pappas, Dani Yogatama, Zhaofeng Wu, Lingpeng Kong, Roy Schwartz, and Noah A Smith. Abc: Attention with bounded-memory control. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7469–7483, 2022.

Pierrick Philippe, Théo Ladune, Gordon Clare, Félix Henry, Théophile Blard, and Thomas Leguay. Upsampling improvement for overfitted neural coding. In *2025 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5. IEEE, 2025.

Zhen Qin, Songlin Yang, and Yiran Zhong. Hierarchically gated recurrent neural network for sequence modeling. *Advances in Neural Information Processing Systems*, 36:33202–33221, 2023.

Zhen Qin, Songlin Yang, Weixuan Sun, Xuyang Shen, Dong Li, Weigao Sun, and Yiran Zhong. Hgrn2: Gated linear rnns with state expansion. *arXiv preprint arXiv:2404.07904*, 2024.

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

Mark S Schmalz. Techniques for region coding in object-based image compression. In *Mathematics of Data/Image Coding, Compression, and Encryption VI, with Applications*, volume 5208, pp. 11–21. SPIE, 2004.

Thomas Sikora, Sven Bauer, and Bela Makai. Efficiency of shape-adaptive 2-d transforms for coding of arbitrarily shaped image segments. *IEEE Transactions on Circuits and Systems for Video Technology*, 5(3):254–258, 1995.

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.

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

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*, 2023.

George Toderici, Wenzhe Shi, Radu Timofte, Lucas Theis, Johannes Balle, Eirikur Agustsson, Nick Johnston, and Fabian Mentzer. Workshop and challenge on learned image compression (clic2020). In *CVPR*, 2020.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 32–42. IEEE, 2021.

Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Carsten Stoll, and Christian Theobalt. Patchnets: Patch-based generalizable deep implicit 3d shape representations. In *European Conference on Computer Vision*, pp. 293–309. Springer, 2020.

Guo-Hua Wang, Jiahao Li, Bin Li, and Yan Lu. Evc: Towards real-time neural image compression with mask decay. *arXiv preprint arXiv:2302.05071*, 2023a.

Meng Wang, Yu-Shen Liu, Yue Gao, Kanle Shi, Yi Fang, and Zhizhong Han. Lp-dif: Learning local pattern-specific deep implicit function for 3d objects and scenes. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21856–21865. IEEE, 2023b.

Haotian Wu, Gen Li, Di You, Pier Luigi Dragotti, and Deniz Gunduz. Frequency-aware perceptual optimization for low-complexity implicit image compression. In *Forty-third International Conference on Machine Learning*.

Haotian Wu, Gongpu Chen, Pier Luigi Dragotti, and Deniz Gündüz. Lotterycodec: Searching the implicit representation in a random network for low-complexity image compression. *arXiv preprint arXiv:2507.01204*, 2025.

Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. A theory of usable information under computational constraints. *arXiv preprint arXiv:2002.10689*, 2020.

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

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. *arXiv preprint arXiv:2312.06635*, 2023.

Yu Zhang, Songlin Yang, Ruijie Zhu, Yue Zhang, Leyang Cui, Yiqiao Wang, Bolun Wang, Freda Shi, Bailin Wang, Wei Bi, et al. Gated slot attention for efficient linear-time sequence modeling. *Advances in Neural Information Processing Systems*, 37:116870–116898, 2024.

APPENDIX

- Appendix A: Open resources.
- Appendix B: Broader impacts.
- Appendix C: Baseline implementation.
- Appendix D: PRISM implementation.
  - Appendix D.1: Architecture configurations and parameter counts.
  - Appendix D.2: Quantization and entropy coding.
- Appendix E: Supplementary quantitative results.
  - Appendix E.1: RD performance over more baselines.
  - Appendix E.2: Performance across fitting budgets.
  - Appendix E.3: Transfer to other overfitted codecs.
  - Appendix E.4: Bit allocation.
- Appendix F: Visual comparisons and partition-boundary analysis.
  - Appendix F.1: Image visual comparisons.
  - Appendix F.2: Multiscale latent representations.
  - Appendix F.3: Local reconstruction consistency at low rates.
  - Appendix F.4: Partition-boundary artifacts.
  - Appendix F.5: Context-state reuse in synthesis.
  - Appendix F.6: Progressive evolution of the context state.

A OPEN RESOURCES

Reconstructions. We evaluate PRISM on Kodak (24 images) (Kodak, 1991) and CLIC2020 (41 images) (Toderici et al., 2020). We will release all per-image RD results and reconstructions to support reproducibility and future research.

Progressive state modeling implementation. We provide the implementation of progressive state updates, learned memory gating, cross-scale alignment, and context-state reuse for entropy modeling and synthesis, together with the detailed configurations used in our experiments. These components serve as a reference for integrating the state mechanism into other overfitted codecs.

Code and checkpoints. All codes and checkpoints will be released after the review process.

B BROADER IMPACTS

PRISM contributes to image compression through progressive context-state modeling and reuse. Its potential societal effects depend on the applications and deployment settings in which it is used.

Positive impacts. (1) **Efficiency:** Improved compression efficiency and lightweight decoding may reduce communication and decoding costs on resource-constrained devices, particularly when images are encoded once and decoded repeatedly. (2) **Content adaptation:** Region-informed modeling adapts probability estimation and reconstruction to spatially varying content, supporting visual communication under tight bitrate and computational budgets. (3) **Broad applicability:** The progressive state mechanism can be incorporated into other overfitted codecs, extending the benefits of context reuse across architectures.

Potential negative impacts. As with other image-processing techniques, compression may be misused to obscure visual details or misrepresent content in downstream applications. We encourage responsible deployment and appropriate integrity checks when visual fidelity is critical. We use publicly available datasets and release code to support transparency and reproducibility.

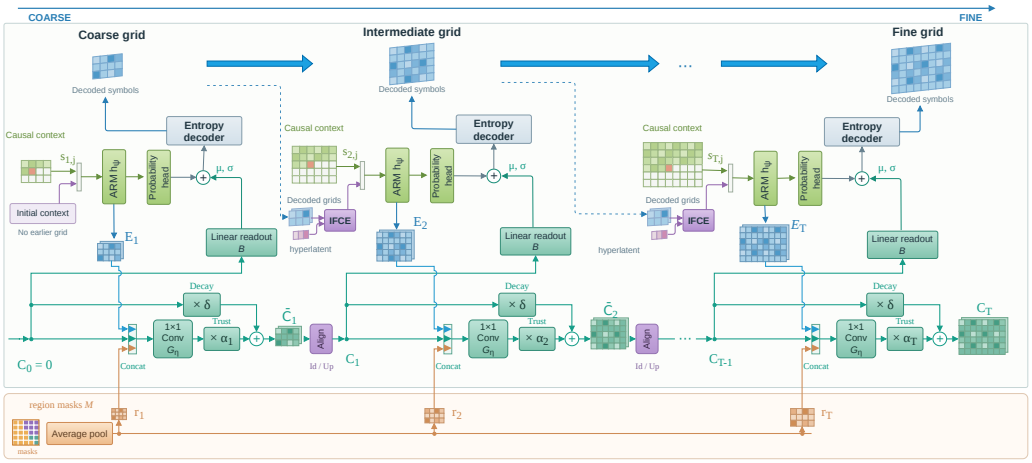


Figure 9: Progressive region-informed state modeling. Grid t reads the incoming state $\mathbf{C}_{t-1}$ for probability estimation. The shared layer $G_{\boldsymbol{\eta}}$ combines this state with ARM features $\mathbf{E}_t$ and pooled region information $\mathbf{r}_t$. The decay and trust gates control the update to $\bar{\mathbf{C}}_t$, which is mapped to $\mathbf{C}_t$ by the learned upsampler $\mathcal{U}_{\boldsymbol{\kappa}}$. The state propagates through the latent hierarchy, and the final image-resolution state $\mathbf{C}_T$ is reused for synthesis.

C Baseline implementation

We evaluate PRISM on the 24 images in Kodak (Kodak, 1991) and the 41 images in the CLIC2020 professional validation set (Toderici et al., 2020), focusing on standard full-image compression. Our comparisons include conventional, autoencoder-based, and overfitted codecs. For conventional compression, we use VTM 19.1 (Bross et al., 2021) through the benchmarking interface of CompressAI. Autoencoder-based baselines comprise Cheng (Cheng et al., 2020), also evaluated using CompressAI; EVC-S, EVC-M, and EVC-L (Wang et al., 2023a), which offer different RD–complexity trade-offs; ELIC (He et al., 2022); and MLIC++ (Jiang et al., 2025) and LIC-HPCM (Li et al., 2025), which employ advanced context modeling. Overfitted baselines include the fixed and adaptive configurations of C3 (Kim et al., 2024), Cool-chic 5.0 (Ladune et al., 2026), LotteryCodec (Wu et al., 2025), and the full-image configuration of MoRIC (Li et al., 2026). For Kodak, we use the HOP configuration of Cool-chic 5.0. The extended comparison on CLIC2020 additionally includes MLIC and MLIC+ (Jiang et al., 2023), as well as Cool-chic v4.2. LotteryCodec (mask=0.3) denotes a fixed network mask ratio of 0.3.

D PRISM implementation

PRISM builds on the hierarchical entropy model and encoding procedure of Cool-chic 5.0 (Ladune et al., 2026), together with the region-adaptive synthesis of MoRIC (Li et al., 2026). We retain the underlying optimization procedures and detail the state-related extensions and experimental configurations below.

We evaluate PRISM on the 24 Kodak images and the 41 images in the CLIC2020 professional validation set. Kodak images have a resolution of 512×768 or 768×512. Each image is fitted independently at $\lambda \in \{2 \times 10^{-2}, 4 \times 10^{-3}, 10^{-3}, 4 \times 10^{-4}, 10^{-4}\}$, with no model shared across evaluation images. We generate partitions from SAM-based automatic mask generation (Kirillov et al., 2023) for a prescribed number of regions N, accommodating diverse downstream tasks and application requirements. Region masks are transmitted as contours encoded using adaptive chain coding (Chang et al., 1992).

D.1 ARCHITECTURE CONFIGURATIONS

Table 2 summarizes the HOP configuration for Kodak and VHOP configuration for CLIC2020. CLIC2020 images below one million pixels use seven main grids; the remaining images use eight, with the same ARM configuration. The architecture is fixed for each image across λ values.

Table 2: PRISM architecture configurations. CLIC-L and CLIC-S denote the 35 images with at least one million pixels and the six smaller images, respectively. The region-feature width follows the full-N-mask formulation in the main text.

Configuration	Kodak	CLIC-L	CLIC-S
Number of images	24	35	6
Regions N	2	2	2
Main latent grids	7	8	7
Hyperlatent grids	3	4	3
Total grids T	10	12	10
Main-grid resolution levels	0–6	0–7	0–6
Hyperlatent resolution levels	4–6	4–7	4–6
State width c	2	2	2
Region-feature width N	2	2	2
Spatial ARM context size	14	20	20
IFCE output width	6	6	6
ARM hidden layers	2	2	2
ARM hidden width d	20	26	26
ARM nonlinear-branch input width	20	26	26
ARM linear-branch input width	22	28	28
Global synthesis width G	14	24	22
Local synthesis width	3	3	3
Synthesis input width	9	10	9

Each grid has one scalar channel and spatial size $\lceil H/2^\ell \rceil \times \lceil W/2^\ell \rceil$ at resolution level ℓ, with level zero at image resolution. Decoding proceeds from coarse to fine, placing each hyperlatent before its same-resolution main grid. A 512×768 Kodak image contains 526,272 latent elements across ten grids; a 1365×2048 CLIC2020 image contains 3,743,072 across twelve. The state follows this decoding order.

The ARM uses 14 spatial neighbors on Kodak and 20 on CLIC2020 within a 9×9 causal window. At each of the three finest main-grid levels, a separate IFCE affine projection maps inter-grid inputs to six channels. From finest to coarsest, its input widths are $9, 8, 7$ for ten grids and $11, 10, 9$ for twelve grids.

State modeling and reuse. We use the state update in Eq. 1 and the entropy-conditioning and synthesis mechanisms in Section 3.3. The state has $c = 2$ channels and remains fixed while decoding each grid. The ARM hidden width is $d = 20$ on Kodak and $d = 26$ on CLIC2020. The two state-input columns of B add four weights to the existing linear branch without widening the nonlinear trunk. Region features pool all N reconstructed masks to the current grid resolution.

The shared decay δ and grid-specific trust α_t are initialized to 0.999 and 0.5, respectively. The state and ARM columns of $G_{\boldsymbol{\eta}}$ use zero-mean Gaussian initialization with standard deviation $1/c^2$; its region columns and bias are zero-initialized. Only the state-input columns of B are zero-initialized, preserving its existing context weights.

The shared upsampler $\mathcal{U}_{\boldsymbol{\kappa}}$ maps $\bar{\mathbf{C}}_t$ to the next grid's resolution, with $(H_{T+1}, W_{T+1}) = (H, W)$ specifying the terminal synthesis target. Its symmetric transposed-convolution kernel has size eight, four independent taps and one bias, shared across state channels and initialized from bicubic interpolation.

For the residual state filter Φ, the convolution $\mathcal{K}$ is separable and symmetric, with kernel size seven, four independent taps and one bias, and uses Dirac initialization. As in Section 3.3, $\mathcal{U}_z$ upsamples only main latents; hyperlatents contribute to synthesis through the accumulated state.

The global modulation network uses 1×1 layers of widths $F \to G \to 3$ and two 3×3 residual layers of width three, where F is the number of main grids plus c and G is the global hidden width. Each local INR $g_{\mathbf{W}^i}$ maps two-dimensional coordinates to width three, followed by three width-three layers, totaling 45 parameters per region. Three shared aggregation layers each map $G + 12$ channels to three, combining local and global features. The predictions and linear residual are combined according to Eq. 7. Main-latent upsampling uses separable symmetric transposed convolutions of size eight, size-seven pre-concatenation filters, and final bicubic upsampling.

Parameter counts. Table 3 reports architecture-derived parameter counts, including biases, gate logits and all N region channels. The update layer and gates contain

$$P_{\text{update+gates}} = c(c + d + N) + c + (T + 1). \tag{8}$$

With $c = N = 2$, $d = (20, 26, 26)$ and $T = (10, 12, 10)$, the Kodak, CLIC-L and CLIC-S configurations require 61, 75 and 73 parameters, respectively. State upsampling and filtering add five each. The four state-input weights are counted under ARM; additional synthesis inputs are counted under synthesis. For the synthesis architecture above, the parameter count is $P_{\text{syn}} = (F + 13)G + 3F + 291 + 45N$. To accommodate the two state channels within a similar synthesis parameter budget, the global widths are adjusted from 16 to 14 for Kodak, from 27 to 24 for CLIC-L, and from 25 to 22 for CLIC-S.

Table 3: Parameter counts implied by the specified architecture with all N region channels. Latent symbols and the region partition are not network parameters. The ARM row includes B.

Module	Kodak	CLIC-L	CLIC-S
ARM, including linear residual branch	928	1,516	1,516
IFCE	162	198	162
Main-latent upsampling	60	70	60
Region-adaptive synthesis	716	963	892
State update and memory gates	61	75	73
State upsampling	5	5	5
State synthesis filter	5	5	5
Total	1,937	2,832	2,713

D.2 Quantization and entropy coding

PRISM follows the quantization-aware fitting and module-wise parameter coding of Cool-chic 5.0 (Ladune et al., 2026), including the parameters for state construction and reuse. Table 4 summarizes the settings.

Quantization-aware optimization. Let $\mathbf{z}_t$ denote a continuous latent grid in quantizer coordinates, with the encoder-side gain absorbed into this parameterization. Following Cool-chic 5.0, relaxed fitting uses

$$\tilde{\mathbf{z}}_t = \mathcal{Q}_{\text{train}}(\mathbf{z}_t) = s_\tau(s_\tau(\mathbf{z}_t) + \boldsymbol{\epsilon}_t)\,, \qquad \epsilon_{t,j} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_n^2), \tag{9}$$

where $\boldsymbol{\epsilon}_t$ matches $\mathbf{z}_t$ in shape, j indexes entries, τ is the soft-rounding temperature, and σ_n is the noise standard deviation. Elementwise soft rounding is defined as

$$s_\tau(a) = \lfloor a \rfloor + \frac{1}{2} + \frac{\tanh\big((a - \lfloor a \rfloor - \frac{1}{2})/\tau\big)}{2\tanh(1/(2\tau))}. \tag{10}$$

The final latents are $\hat{\mathbf{z}}_t = \mathcal{Q}_{\text{hard}}(\mathbf{z}_t) = \text{round}(\mathbf{z}_t)$. Refinement uses hard rounding in the forward pass and straight-through gradients for active latents.

Warm-up evaluates 8 candidates for 400 updates each, retaining four for another 400 updates. The best candidate receives up to 100,000 relaxed updates and 500 hard-rounding refinement updates. During relaxed fitting, τ decreases from 0.35 to 0.08 and σ_n from 0.22 to 0.15.

We optimize the RD objective $\mathcal{L}(\boldsymbol{\Omega})$ defined in Section 3.4. Differentiable fitting minimizes $d(\mathbf{S}, \hat{\mathbf{S}}) + \lambda R_{\text{lat}}$, with regional distortions normalized to sum to full-image mean squared error. Contour cost is fixed; model rate is included during RD-aware parameter quantization and RDOQ (Ladune et al., 2026).

Table 4: Hyper-parameter settings for PRISM. Values are shared by Kodak and CLIC2020 unless specified otherwise.

Hyper-parameter	**Initial value**	**Final value**
RD weights λ	$\{2\times10^{-2}, 4\times10^{-3}, 1\times10^{-3}, 4\times10^{-4}, 1\times10^{-4}\}$	
Candidate warm-up		
Candidates in the two rounds	8, 4	
Updates per candidate per round	400	
Learning rate / scheduling	10^{-2} / constant	
Soft-rounding temperature τ	0.35	
Gaussian-noise standard deviation	$\sigma_{n,0} = (-0.432\log_{10}\lambda + 0.747)/10$	
Stage I: relaxed-quantization fitting		
Maximum encoding updates	100,000	
Learning rate	10^{-2}	10^{-5}
Learning-rate scheduler	Cosine	
Soft-rounding temperature τ	0.35	0.08
Gaussian-noise standard deviation σ_n	0.22	0.15
Stopping patience / validation interval	5,000 / 100 updates	
Stage II: hard-rounding refinement		
Maximum encoding updates	500	
Learning rate	10^{-4}	10^{-5}
Learning-rate scheduler	Cosine	
Forward quantization / additive noise	Hard rounding / none	
Stopping patience / validation interval	10,000 / 100 updates	
Optimization		
Network optimizer / latent optimizer	SOAP / Adam	
Network moment coefficients	$(0.95, 0.95)$	
Latent moments: warm-up / main stage	$(0.725, 0.97)$	$(0.9, 0.999)$
Weight decay: networks / latents	0.01 / 0	
Gradient clipping: networks / latents	Norm 0.1 / none	
Random seed	1 for each image	
State modeling and reuse		
Regions N / state channels c	2 / 2	
Initial decay δ / trust α_t	0.999 / 0.5	
$G_{\boldsymbol{\eta}}$ state/ARM input weights	Gaussian: mean 0, standard deviation $1/c^2$	
$G_{\boldsymbol{\eta}}$ region weights and bias / state-input columns of B	Zero / zero	
State upsampler: kernel / initialization	8 / bicubic	
State filter: kernel / initialization	7 / Dirac inside the residual filter	

Learning-rate endpoints are the configured schedule minima; early stopping may end a stage sooner. The main-stage budget excludes warm-up and refinement. Dataset-dependent widths and grid counts are listed in Table 2, and encoder gains in Table 5.

Optimizer and rate-dependent settings. SOAP optimizes network parameters with gradient clipping at norm 0.1; Adam optimizes active latents without clipping. The synthesis output transform is frozen. SOAP uses maximum preconditioner dimension 256, with dimension merging, one-dimensional preconditioning and latent preconditioning disabled. Its preconditioner updates every warm-up step and every ten main-stage steps. The seed is reset to one per image. Table 5 lists the encoder gain and warm-up noise $\sigma_{n,0} = (-0.432 \log_{10} \lambda + 0.747)/10$. The gain is absorbed into the latent parameterization in Eq. 9.

Table 5: Rate-dependent fitting settings shared by Kodak and CLIC2020.

λ	Encoder gain	Warm-up noise $\sigma_{n,0}$
2×10^{-2}	16	0.1481
4×10^{-3}	16	0.1783
1×10^{-3}	16	0.2043
4×10^{-4}	20	0.2215
1×10^{-4}	24	0.2475

Latent entropy coding. The quantized grids follow the coarse-to-fine conditional factorization

$$p_{\hat{\boldsymbol{\psi}}}(\hat{\mathbf{z}} \mid \mathcal{M}) = \prod_{t=1}^{T} \prod_{j} p_{\hat{\boldsymbol{\psi}}}(\hat{z}_{t,j} \mid \mathbf{s}_{t,j}, \mathbf{f}_{t,j}, \mathbf{C}_{t-1}(j)), \tag{11}$$

where $\mathcal{M} = \{\mathbf{M}^i\}_{i=1}^{N}$ is the reconstructed partition. The ARM and B branches supply the distribution parameters using quantized network parameters (Eq. 4). Symbol probabilities integrate a Laplace density over the quantization bin:

$$p_{\hat{\boldsymbol{\psi}}}(\hat{z}_{t,j} \mid \mathbf{s}_{t,j}, \mathbf{f}_{t,j}, \mathbf{C}_{t-1}(j)) = \int_{\hat{z}_{t,j}-1/2}^{\hat{z}_{t,j}+1/2} \mathrm{Laplace}(a; \mu_{t,j}, \sigma_{t,j})\, da. \tag{12}$$

Here, a is a scalar integration variable, $\mu_{t,j}$ is the location and $\sigma_{t,j} > 0$ is the scale. These probabilities drive arithmetic coding and determine R_{lat} as the negative log probability of all main and hyperlatents, normalized by image size.

The state remains fixed within each grid and is updated only after obtaining its causal features $\mathbf{E}_t$, following the same dependencies during fitting and coding.

Network parameter quantization and coding. Weight and bias quantization steps are selected separately per module. For parameter block $\boldsymbol{\theta}_m$ and step Δ_m, the integer indices and reconstructed parameters are

$$\mathbf{q}_m = \mathrm{round}(\boldsymbol{\theta}_m / \Delta_m), \qquad \hat{\boldsymbol{\theta}}_m = \Delta_m \mathbf{q}_m. \tag{13}$$

Indices use exponential-Golomb coding. Quantization steps (Table 6) and coding orders from 0 to 12 are selected greedily per module using distortion and combined latent and model rate. RD optimized quantization (RDOQ) then tests one-integer-step changes in either direction, retaining improvements to the RD objective. This procedure includes state-update weights, gate logits, learned upsampling, the state filter and B.

Table 6: Candidate quantization steps. $2^{a:b}$ denotes $\{2^a, 2^{a+1}, \ldots, 2^b\}$.

Module	Weight step	Bias step
ARM, IFCE, state update and gates	$2^{-8:0}$	$2^{-16:0}$
Latent/state upsampling, state filter	$2^{-12:0}$	1
Synthesis	$2^{-12:0}$	$2^{-24:0}$

Let $\boldsymbol{\Theta}$ collect all decoder parameters. After quantization, the encoder resets the state and recomputes symbol probabilities using $\hat{\boldsymbol{\Theta}}$, ensuring agreement with the decoder. The model description includes the architecture, quantization settings, coding orders and parameter indices.

Region contour coding. Adaptive chain coding (Chang et al., 1992) represents region contours. The partition reconstructed before fitting supplies both pooled region features and synthesis masks; its cost is counted once. Context states are recomputed from decoded information and require no separate latent symbols.

Algorithm 1 Encoding stage of PRISM

Input: Source image $\mathbf{S}$, initial partition $\mathcal{M}_0$, trade-off λ, fitting configuration
Output: Latent description $\mathbf{b}_z$, model description $\mathbf{b}_\Theta$, contour description $\mathbf{b}_\mathcal{M}$

$\mathbf{b}_\mathcal{M} \leftarrow \mathrm{ContourEncode}(\mathcal{M}_0)$
$\mathcal{M} \leftarrow \mathrm{ContourDecode}(\mathbf{b}_\mathcal{M})$; obtain $\{\mathbf{r}_t\}_{t=1}^{T}$ by pooling all N masks
Initialize $(\mathbf{z}, \boldsymbol{\Theta})$; run candidate warm-up and select the best RD candidate
for each fitting stage s (Stage I: relaxed fitting; Stage II: STE refinement) **do**
  for each optimization step k of stage s **do**
    $\tilde{\mathbf{z}} \leftarrow \mathcal{Q}_{\mathrm{train}}(\mathbf{z}; s, k)$; $\mathbf{C}_0 \leftarrow \mathbf{0}$; $R_{\mathrm{lat}} \leftarrow 0$
    for $t = 1, \ldots, T$ **do**
      $(\mathbf{E}_t, \boldsymbol{\mu}_t, \boldsymbol{\sigma}_t) \leftarrow \mathrm{EntropyModel}_{\boldsymbol{\Theta}}(\tilde{\mathbf{z}}_{<t}, \tilde{\mathbf{z}}_{t,<\cdot}; \mathbf{C}_{t-1})$
      $R_{\mathrm{lat}} \leftarrow R_{\mathrm{lat}} + R(\tilde{\mathbf{z}}_t; \boldsymbol{\mu}_t, \boldsymbol{\sigma}_t)$
      Compute $\bar{\mathbf{C}}_t$ using Eq. 1
      Apply $\mathcal{U}_{\boldsymbol{\kappa}}$ to $\bar{\mathbf{C}}_t$ at target size (H_{t+1}, W_{t+1}) to obtain $\mathbf{C}_t$
    end for
    Form $\mathbf{X}_{\mathrm{syn}}$ from upsampled main latents and filtered $\mathbf{C}_T$ (Section 3.3)
    Reconstruct $\hat{\mathbf{S}}$ from $\mathbf{X}_{\mathrm{syn}}$ and $\mathcal{M}$ using Eq. 7
    Backpropagate $d(\mathbf{S}, \hat{\mathbf{S}}) + \lambda R_{\mathrm{lat}}$, the differentiable terms of $\mathcal{L}(\boldsymbol{\Omega})$, through the state sequence
    Update networks with SOAP (clipped gradients) and latents with Adam; update schedules
  end for
end for
$\hat{\mathbf{z}} \leftarrow \mathcal{Q}_{\mathrm{hard}}(\mathbf{z})$
Select module-wise parameter quantizers and coding orders; apply RDOQ to obtain $\hat{\boldsymbol{\Theta}}$
$\mathbf{b}_\Theta \leftarrow \mathrm{ModelEncode}(\hat{\boldsymbol{\Theta}})$
Use $\hat{\boldsymbol{\Theta}}$ for all remaining operations; reset $\mathbf{C}_0 \leftarrow \mathbf{0}$ and initialize $\mathbf{b}_z$
for $t = 1, \ldots, T$ **do**
  for each symbol position j in the causal order of grid t **do**
    $(\mathbf{E}_t(j), \mu_{t,j}, \sigma_{t,j}) \leftarrow \mathrm{EntropyModel}_{\hat{\boldsymbol{\Theta}}}(\hat{\mathbf{z}}_{<t}, \hat{\mathbf{z}}_{t,<j}; \mathbf{C}_{t-1}(j))$
    Entropy-encode $\hat{z}_{t,j}$ using $(\mu_{t,j}, \sigma_{t,j})$ into $\mathbf{b}_z$
  end for
  Compute $\bar{\mathbf{C}}_t$ using Eq. 1
  Apply $\mathcal{U}_{\boldsymbol{\kappa}}$ to $\bar{\mathbf{C}}_t$ at target size (H_{t+1}, W_{t+1}) to obtain $\mathbf{C}_t$
end for
Finalize $\mathbf{b}_z$ and **return** $(\mathbf{b}_z, \mathbf{b}_\Theta, \mathbf{b}_\mathcal{M})$

Algorithm 2 Decoding stage of PRISM

Input: Latent description $\mathbf{b}_z$, model description $\mathbf{b}_\Theta$, contour description $\mathbf{b}_\mathcal{M}$
Output: Reconstructed image $\hat{\mathbf{S}}$

Restore the decoder configuration and $\hat{\boldsymbol{\Theta}} \leftarrow \mathrm{ModelDecode}(\mathbf{b}_\Theta)$
$\mathcal{M} \leftarrow \mathrm{ContourDecode}(\mathbf{b}_\mathcal{M})$; obtain $\{\mathbf{r}_t\}_{t=1}^{T}$ by pooling all N masks
Use $\hat{\boldsymbol{\Theta}}$ for all subsequent operations; initialize $\mathbf{C}_0 \leftarrow \mathbf{0}$ and the entropy decoder
for $t = 1, \ldots, T$ **do**
  for each symbol position j in the causal order of grid t **do**
    $(\mathbf{E}_t(j), \mu_{t,j}, \sigma_{t,j}) \leftarrow \mathrm{EntropyModel}_{\hat{\boldsymbol{\Theta}}}(\hat{\mathbf{z}}_{<t}, \hat{\mathbf{z}}_{t,<j}; \mathbf{C}_{t-1}(j))$
    $\hat{z}_{t,j} \leftarrow \mathrm{EntropyDecode}(\mathbf{b}_z; \mu_{t,j}, \sigma_{t,j})$
  end for
  Compute $\bar{\mathbf{C}}_t$ using Eq. 1
  Apply $\mathcal{U}_{\boldsymbol{\kappa}}$ to $\bar{\mathbf{C}}_t$ at target size (H_{t+1}, W_{t+1}) to obtain $\mathbf{C}_t$
end for
Form $\mathbf{X}_{\mathrm{syn}}$ from upsampled main latents and filtered $\mathbf{C}_T$ (Section 3.3)
Compute shared modulation features $\mathbf{F}_{\mathrm{mod}}$ from $\mathbf{X}_{\mathrm{syn}}$ using $f_{\hat{\boldsymbol{\theta}}_{\mathrm{g}}}$
for each region $i = 1, \ldots, N$ **do**
  Predict region $\hat{\mathbf{S}}^i$ with $g_{\hat{\mathbf{W}}^i}$ using $\mathbf{F}_{\mathrm{mod}}$ and coordinates $\mathbf{x}^i$
end for
Merge regional predictions and add the linear residual to obtain $\hat{\mathbf{S}}$ using Eq. 7
return $\hat{\mathbf{S}}$

Table 7: Full-image RD performance on Kodak (24 images). BD-rate is relative to VTM 19.1. The best result within each neural codec family is bold.

Codec	BD-rate (%) ↓
VTM 19.1 (anchor)	0.00
Autoencoder-based codecs	
Cheng2020	+3.60
EVC-S	+7.16
EVC-M	+1.05
EVC-L	−0.73
ELIC	−6.98
MLIC++	−14.96
LIC-HPCM-Large	**−19.92**
Overfitted codecs	
C3	+5.49
C3 (adaptive)	+3.09
Cool-chic 4.2	+0.74
Cool-chic 5.0 (HOP)	−6.65
LotteryCodec	−0.77
MoRIC	−3.03
PRISM (ours)	**−9.56**

Table 8: Full-image RD performance on CLIC2020 (41 images). BD-rate is relative to VTM 19.1. The best result within each neural codec family is bold.

Codec	BD-rate (%) ↓
VTM 19.1 (anchor)	0.00
Autoencoder-based codecs	
Cheng2020	+7.46
CST	+7.40
EVC-S	+7.93
EVC-M	+1.45
EVC-L	−0.80
ELIC	−7.04
MLIC	−12.12
MLIC+	−15.84
MLIC++	−16.31
LIC-HPCM-Large	**−21.12**
Overfitted codecs	
C3	−3.88
C3 (adaptive)	−6.65
Cool-chic 4.2	−6.71
Cool-chic 5.0	−16.25
LotteryCodec	−9.69
MoRIC	−10.59
PRISM (ours)	**−20.35**

Encoding and decoding procedures. Algorithms 1 and 2 follow the latent decoding order in the main text. Here, $\boldsymbol{\Theta}$ collects the entropy, state, upsampling and synthesis parameters, including gate logits and regional networks. We denote the coded latent symbols by $\mathbf{b}_z$, the coded region-contour description by $\mathbf{b}_{\mathcal{M}}$, and the coded model description by $\mathbf{b}_\Theta$. The latter two recover the partition $\mathcal{M}$ and quantized decoder parameters $\hat{\boldsymbol{\Theta}}$, respectively; $\mathbf{b}_\Theta$ also includes the architecture and parameter-coding settings.

The helper $\mathrm{EntropyModel}_{\boldsymbol{\Theta}}$ returns ARM features $\mathbf{E}_t$ and probability parameters $(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t)$ from the nonlinear trunk and B. It uses causal contexts of relaxed or rounded grids during fitting and preceding decoded symbols during coding. Here, $\mathbf{z}_{t,<\cdot}$ collects within-grid contexts and $\mathbf{z}_{t,<j}$ denotes symbols preceding j. Each grid reads $\mathbf{C}_{t-1}$, updates it to $\bar{\mathbf{C}}_t$, and applies $\mathcal{U}_{\boldsymbol{\kappa}}$ to obtain $\mathbf{C}_t$.

The stage-dependent rule $\mathcal{Q}_{\mathrm{train}}$ uses noisy soft rounding in Stage I and hard rounding with a straight-through estimator (STE) in Stage II; the configuration determines active refinement parameters. The function R measures latent rate in bits per pixel. Reconstruction follows Eq. 7. All operations in the final coding passes use $\hat{\boldsymbol{\Theta}}$.

E SUPPLEMENTARY QUANTITATIVE RESULTS

Extended RD curves and VTM 19.1-relative BD-rates are provided in Figs. 10 and 11 and Tables 7 and 8, covering additional EVC configurations, MLIC variants, and Cool-chic generations on Kodak and CLIC2020. PRISM leads the evaluated overfitted codecs with BD-rates of -9.56% and -20.35%, improving upon Cool-chic 5.0 by 2.92 and 4.10 percentage points, respectively. In a direct CLIC2020 comparison, Cool-chic 5.0 has a BD-rate of $+4.99\%$ relative to PRISM. PRISM outperforms ELIC on both datasets and MLIC++ on CLIC2020, where it is within 0.77 percentage points of LIC-HPCM-Large. MLIC++ and LIC-HPCM-Large retain stronger RD performance on Kodak.

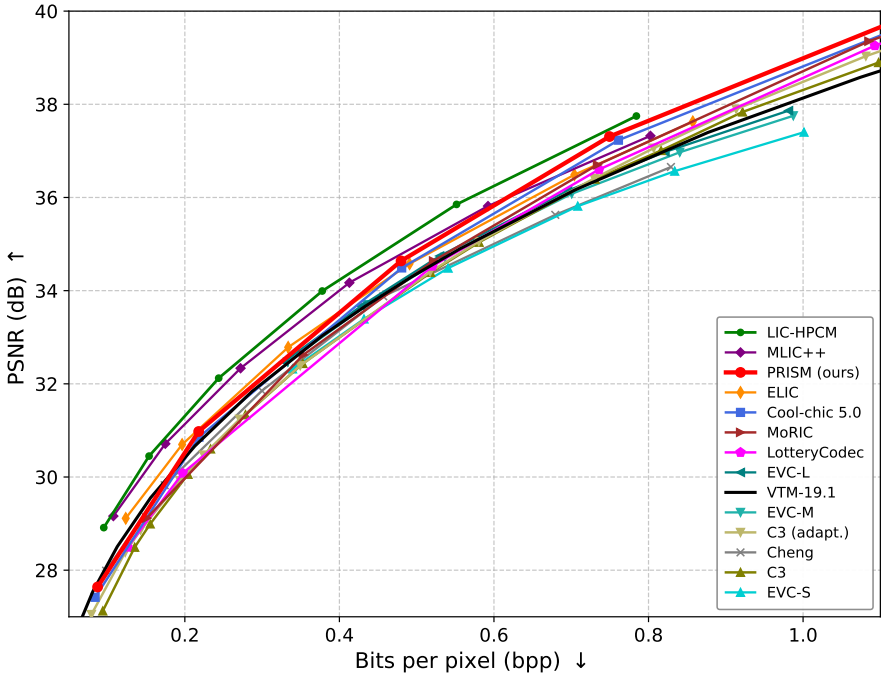


Figure 10: Extended full-image RD comparisons on Kodak.

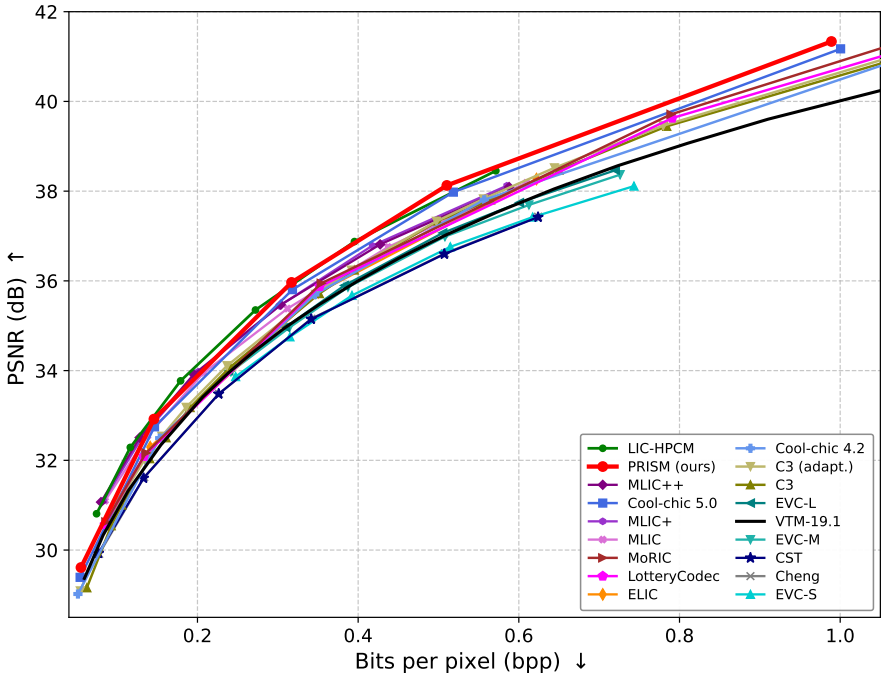


Figure 11: Extended full-image RD comparisons on CLIC2020.

E.1 EXTENDED RD COMPARISONS

Tables 9 and 10 report PRISM's rate and PSNR at the five evaluated RD operating points on Kodak and CLIC2020, respectively.

Table 9: PRISM RD results at five operating points on Kodak.

λ	Rate (bpp)	PSNR (dB)
2×10^{-2}	0.0871	27.6394
4×10^{-3}	0.2179	30.9771
1×10^{-3}	0.4798	34.6384
4×10^{-4}	0.7495	37.3077
1×10^{-4}	1.3322	41.2138

Table 10: PRISM RD results at five operating points on CLIC2020.

λ	Rate (bpp)	PSNR (dB)
2×10^{-2}	0.0552	29.6147
4×10^{-3}	0.1454	32.9146
1×10^{-3}	0.3173	35.9789
4×10^{-4}	0.5085	38.1106
1×10^{-4}	0.9886	41.3375

E.2 PERFORMANCE ACROSS FITTING BUDGETS

Tables 11 and 12 compare five PRISM fitting budgets with baseline reference points. PRISM improves consistently with additional optimization, surpassing VTM 19.1 after 5k iterations with BD-rates of -1.43% on Kodak and -12.56% on CLIC2020. On both datasets, its 10k and 30k results surpass MoRIC/LotteryCodec at 50k and Cool-chic 5.0 at 100k, respectively. Interpolation suggests that PRISM matches Cool-chic 5.0 with approximately $5\times$ fewer fitting iterations on Kodak and $7\times$ fewer on CLIC2020.

The reported budget `n_itr` covers the main optimization stage; warm-up and hard-rounding refinement are separate (Appendix D.2). Iteration reductions therefore do not directly measure reductions in total optimizer updates or wall-clock encoding time.

Table 11: PRISM BD-rate (%) relative to VTM 19.1 across the reported main-stage fitting budgets. Warm-up and refinement are accounted for separately.

Dataset	5k	10k	30k	50k	100k
Kodak	-1.43	-5.09	-7.89	-8.64	-9.56
CLIC2020	-12.56	-15.85	-18.27	-19.68	-20.35

Table 12: Overfitted reference points used in the convergence comparison. BD-rate is relative to VTM 19.1.

Codec	Reported budget	Kodak BD-rate (%)	CLIC2020 BD-rate (%)
Cool-chic 5.0	100k	-6.65	-16.25
MoRIC	50k	-3.03	-10.59
LotteryCodec	50k	-0.77	-9.69

E.3 TRANSFER TO OTHER OVERFITTED CODECS

On Kodak, PSM improves the VTM-relative BD-rate at 100k iterations by 2.18, 2.47, and 1.97 percentage points for Cool-chic 5.0, LotteryCodec, and the regional PRISM configuration, respectively (Table 13). Here, LotteryCodec uses `mask=0.3`, a controlled configuration distinct from the reference in Table 7.

PSM improves both backbones at every evaluated matched budget (Table 14). Cool-chic 5.0 with PSM reaches -7.04% at 30k iterations, surpassing its original 100k result (-6.65%); LotteryCodec with PSM similarly reaches $+5.41\%$ at 50k, outperforming its original 100k result ($+6.03\%$). Interpolation estimates approximately $3.7\times$ and $2.2\times$ fewer iterations to match the respective original codecs, supporting PSM's benefits for BD-rate and convergence.

Table 13: Generality on Kodak at a reported 100k fitting budget. Gains are differences between BD-rates measured against the same VTM 19.1 reference, in percentage points (pp). They are not direct pairwise BD-rate estimates.

Configuration	Without PSM (%)	With PSM (%)	Gain (pp)
LotteryCodec (`mask=0.3`)	+6.03	+3.56	2.47
Cool-chic 5.0	−6.65	−8.83	2.18
Regional PRISM configuration	−7.60	−9.56	1.97

Table 14: Kodak BD-rate (%) relative to VTM 19.1 for each backbone with and without PSM. Unavailable operating points are left blank.

Configuration	10k	30k	50k	100k
Cool-chic 5.0	−2.47	−5.18	−5.94	−6.65
Cool-chic 5.0 + PSM	−4.28	−7.04	−7.98	−8.83
LotteryCodec (`mask=0.3`)	—	+9.33	+7.84	+6.03
LotteryCodec (`mask=0.3`) + PSM	—	+7.67	+5.41	+3.56

E.4 Bit allocation

Fig. 12 shows how the bit budget is distributed across latent grids, network parameters, and region contours. As the total bitrate increases from approximately 0.09 to 1.33 bpp, the share assigned to array 1 rises from 8.0% to 82.7%, making the allocation increasingly concentrated in this grid. In contrast, network parameters require approximately 0.04–0.05 bpp across the operating points, so their share falls from 43.7% to 3.6%. Hyperlatents account for less than 1.8% of the estimated total rate throughout, while contour coding requires approximately 0.0013 bpp. This breakdown shows that higher bitrates primarily expand the allocation to array 1, whereas network parameter coding remains a substantial part of the budget at low rates, reinforcing the motivation for compact transmitted networks (detailed bit costs are provided in Table 15).

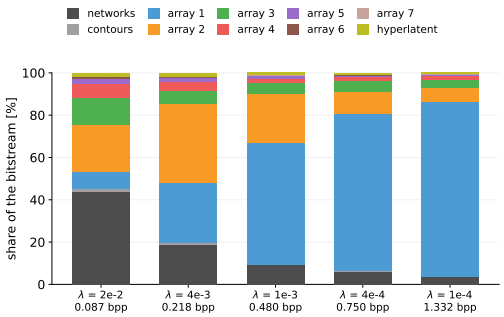


Figure 12: Estimated bit allocation across latent grids, network parameters, and region contours at different RD operating points on Kodak.

Table 15: Estimated bitrate by component on Kodak (bpp).

λ	2×10^{-2}	4×10^{-3}	1×10^{-3}	4×10^{-4}	1×10^{-4}
Array 1	.006963	.062135	.275764	.556380	1.102005
Array 2	.019415	.080886	.111411	.079230	.088881
Array 3	.011120	.013657	.024035	.036891	.048532
Array 4	.005828	.009696	.011195	.014002	.022523
Array 5	.001800	.003543	.004590	.005987	.008553
Array 6	.000789	.001269	.001868	.002390	.003130
Array 7	.000302	.000499	.000693	.000813	.001105
Hyperlatents	.001510	.003514	.004967	.006106	.007803
Networks	.038093	.041401	.044029	.046452	.048374
Contours	.001274	.001274	.001274	.001274	.001274
Total	.087093	.217873	.479825	.749524	1.332179

F VISUAL COMPARISONS AND PARTITION-BOUNDARY ANALYSIS

We provide visual comparisons on Kodak and CLIC2020, followed by an examination of multiscale latent representations, reconstruction artifacts, and context-state reuse. The final visualizations trace the evolution of the context state through selected decoding stages, connecting the intermediate representations to image synthesis.

F.1 IMAGE VISUAL COMPARISONS

Figures 13–15 compare PRISM with Cool-chic 5.0, VTM 19.1, MLIC++, and LIC-HPCM on diverse natural images. The examples cover smooth backgrounds, fine structures, and textured scenes, including buildings, vegetation, water, and object boundaries. PRISM preserves the overall scene structure while providing reconstruction quality competitive with the stronger autoencoder-based baselines. Each reconstruction is annotated with its bitrate and PSNR; the operating points are not identical across methods, so the visual comparisons should be read alongside these measurements and the full RD curves in Appendix E.1.

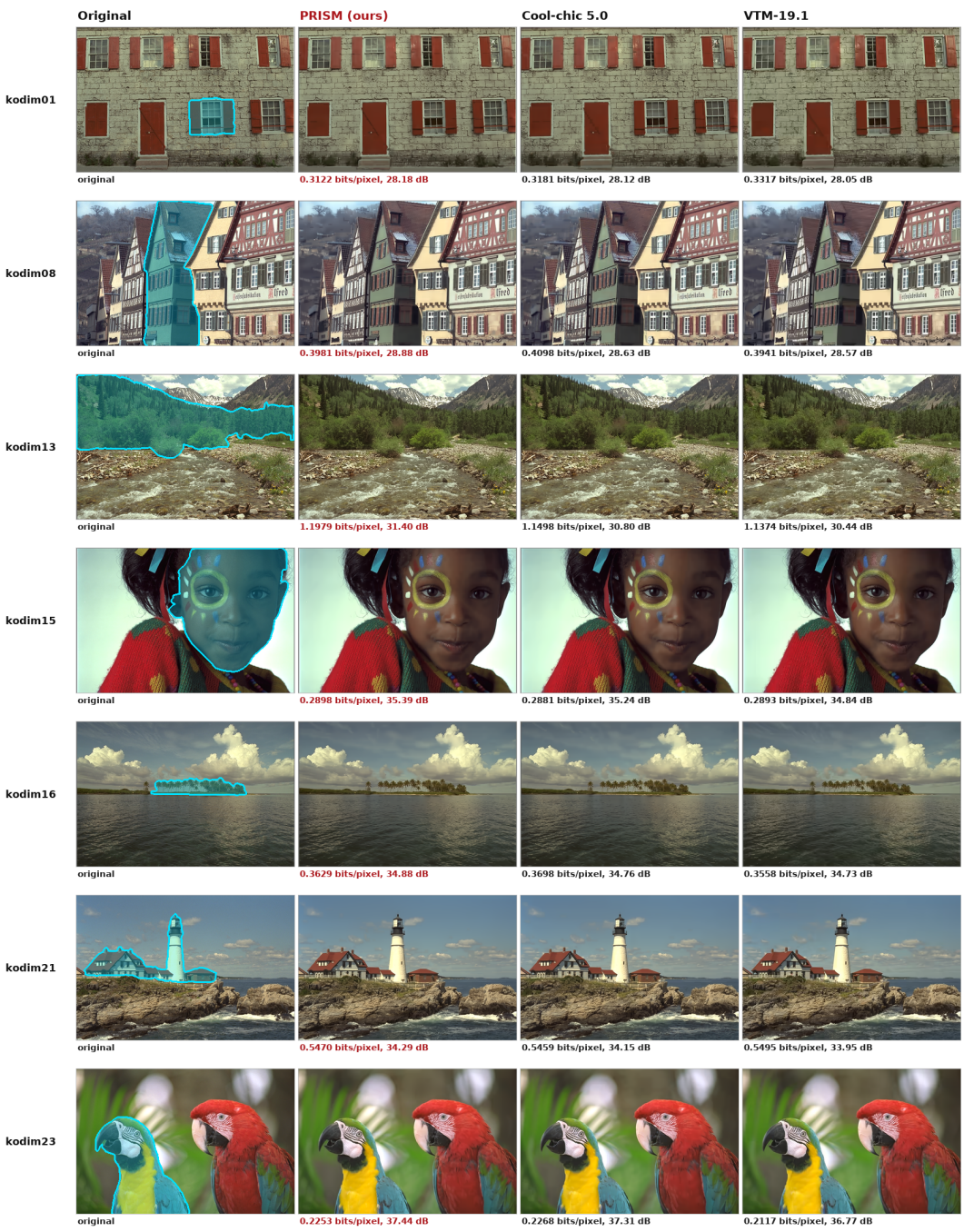


Figure 13: Full-image visual comparisons on Kodak. Columns show the original image, PRISM, Cool-chic 5.0, and VTM 19.1. The cyan overlays in the original images indicate the region partitions used by PRISM.

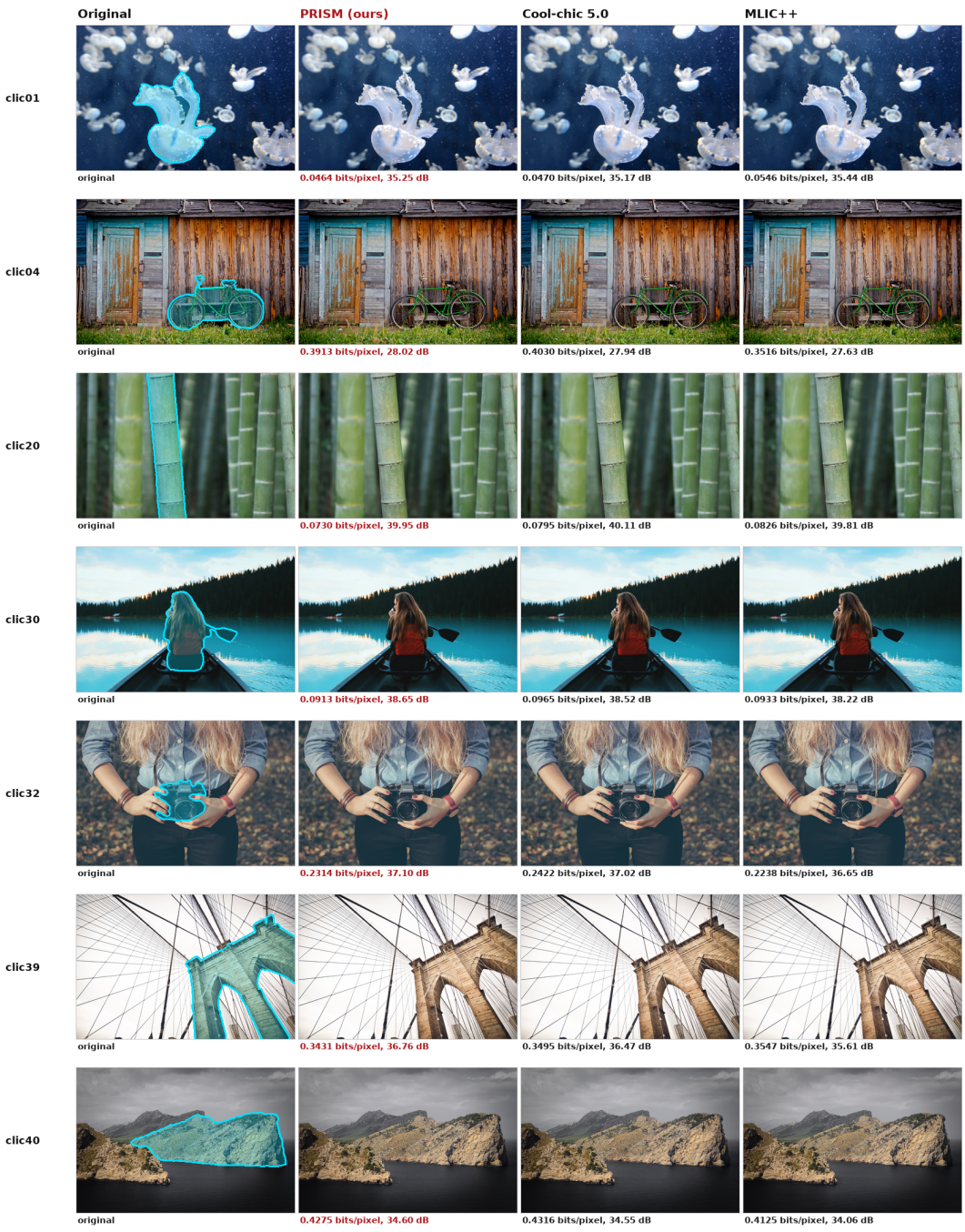


Figure 14: Full-image visual comparisons on CLIC2020 with MLIC++. Columns show the original image, PRISM, Cool-chic 5.0, and MLIC++. The cyan overlays indicate the region partitions used by PRISM.

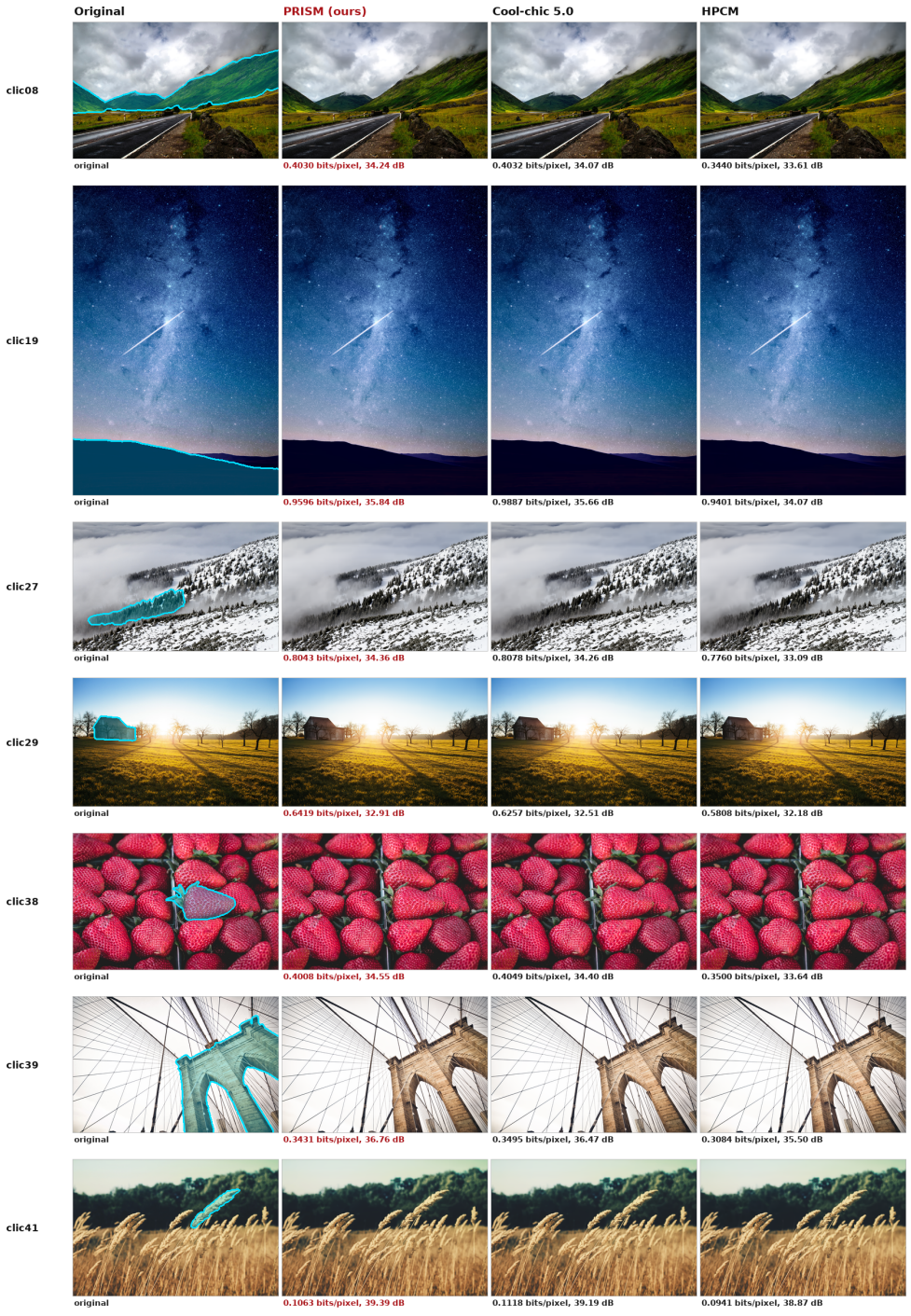


Figure 15: Full-image visual comparisons on CLIC2020 with LIC-HPCM. Columns show the original image, PRISM, Cool-chic 5.0, and LIC-HPCM.

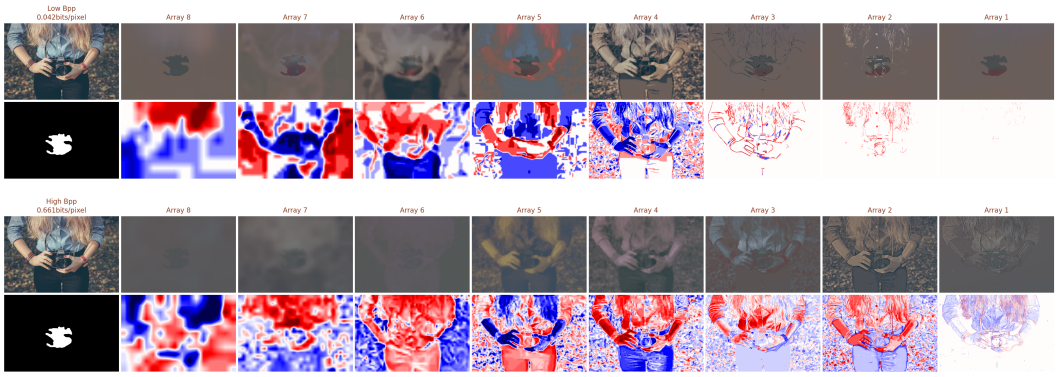


Figure 16: Multiscale latent visualizations for a CLIC2020 image at 0.042 bpp (top group) and 0.661 bpp (bottom group). Each group shows the image and region mask in the first column, followed by image-space visualizations and latent maps for Arrays 8–1. Finer arrays exhibit richer spatial detail at the higher rate.

F.2 Multiscale latent representations

Fig. 16 visualizes the latent hierarchy of a CLIC2020 image at low and high bitrates. The coarse arrays exhibit broad spatial variation, while finer arrays contain more localized structures. At the lower rate, the finest arrays display relatively sparse patterns; at the higher rate, these arrays contain richer responses around the hair, clothing, camera, and background texture. The visualizations illustrate how the spatial information represented across the hierarchy changes with the rate budget. This variation motivates maintaining context across grids: later stages can use retained information about broader structures while modeling finer spatial detail.



Figure 17: Low-rate reconstruction of Kodak image 05. The enlarged helmet crop highlights color transitions and local shading in PRISM, Cool-chic 5.0, and VTM 19.1.

F.3 Local reconstruction consistency at low rates

Figures 17–19 examine localized artifacts in low-rate reconstructions. In the motorcycle and flower examples, Cool-chic 5.0 exhibits irregular color transitions and shading in the selected crops, whereas PRISM produces more consistent color transitions around the helmet and within the petals at similar bitrates. The CLIC2020 example further shows smoother roof shading at the same reported bitrate as Cool-chic 5.0. Fine details are still lost at these operating points, but the displayed PRISM reconstructions reduce some of the localized distortions. These observations are consistent with shared context complementing the multiscale latents during synthesis; the visual comparisons alone do not establish the origin of individual artifacts.



Figure 18: Low-rate reconstruction of Kodak image 07. The enlarged flower crop highlights petal shading and localized color irregularities.

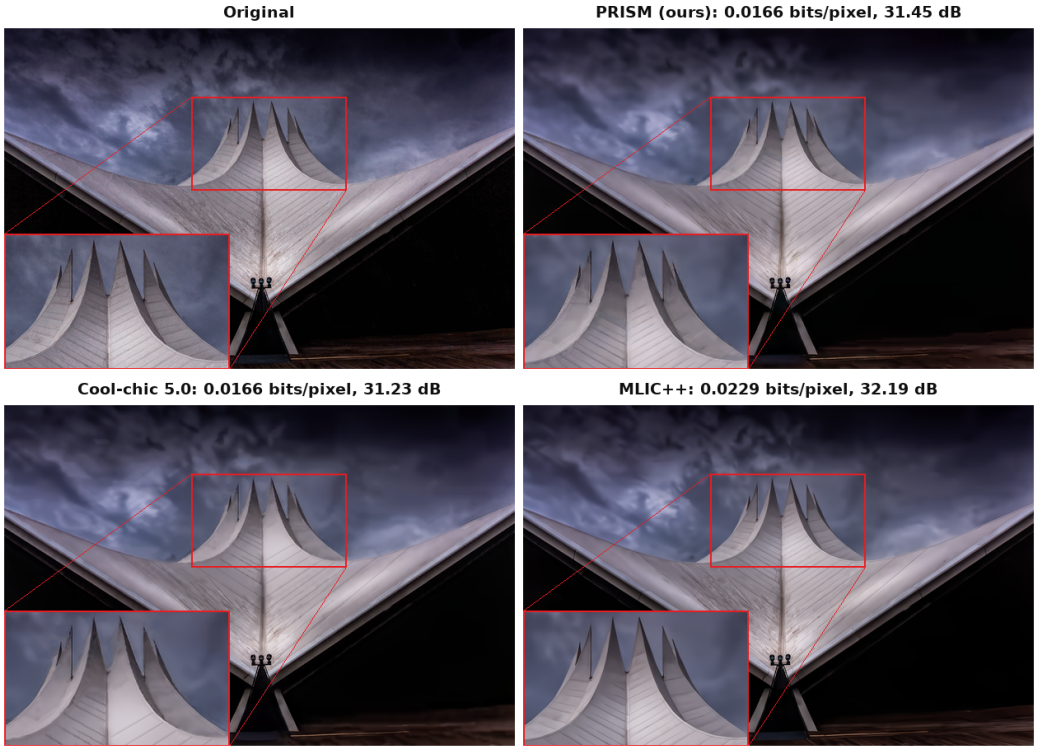


Figure 19: Low-rate reconstruction of a CLIC2020 image. The enlarged crop highlights roof shading and the roof–sky transition. PRISM and Cool-chic 5.0 have the same reported bitrate of 0.0166 bpp; MLIC++ is shown at 0.0229 bpp.

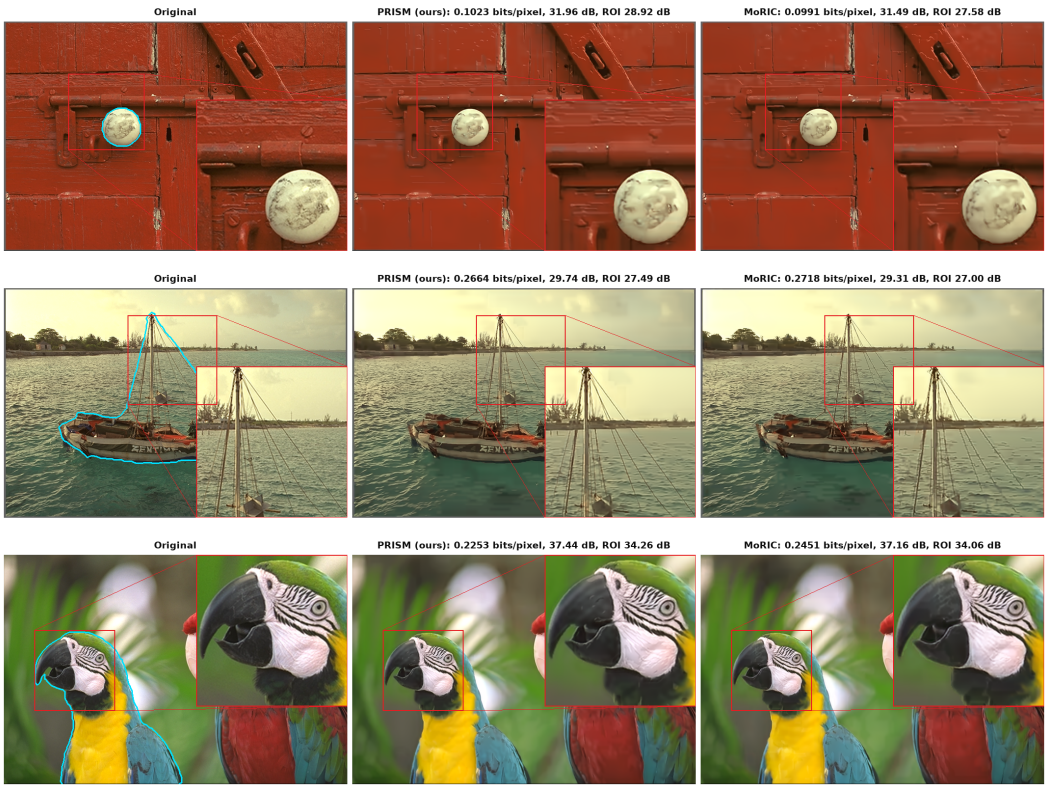


Figure 20: Partition-boundary comparisons on Kodak. Columns show the original image, PRISM, and MoRIC. Cyan contours indicate the region boundaries in the original images, and red boxes identify enlarged crops.

F.4 Partition-boundary artifacts

Figures 20 and 21 compare PRISM with MoRIC around region boundaries on Kodak and CLIC2020. The enlarged crops show fewer localized contour irregularities and more consistent transitions in the displayed PRISM reconstructions, particularly around the vegetation, laptop, and flower. PRISM also achieves higher image and ROI PSNR in all displayed examples at comparable bitrates. These results suggest that PRISM alleviates partition-related artifacts to some extent.

Shared modulation and the reused context state provide common information to the region-specific synthesis networks, which can help coordinate reconstruction across the partition. However, region selection does not explicitly enforce agreement between local networks at their boundaries. The examples therefore support improved boundary consistency without implying that such artifacts are eliminated or that the difference between the two codecs is due solely to state reuse.

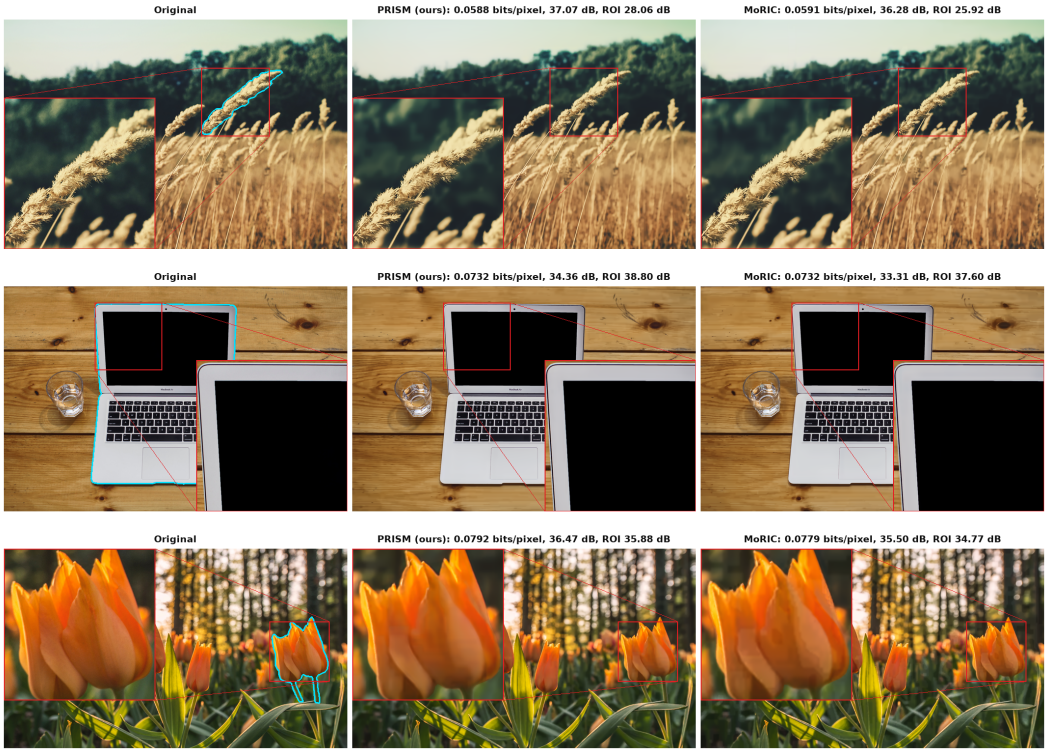


Figure 21: Partition-boundary comparisons on CLIC2020, showing vegetation, a laptop, and a flower. Columns show the original image, PRISM, and MoRIC. Cyan contours indicate region boundaries, while enlarged crops highlight local reconstruction consistency.

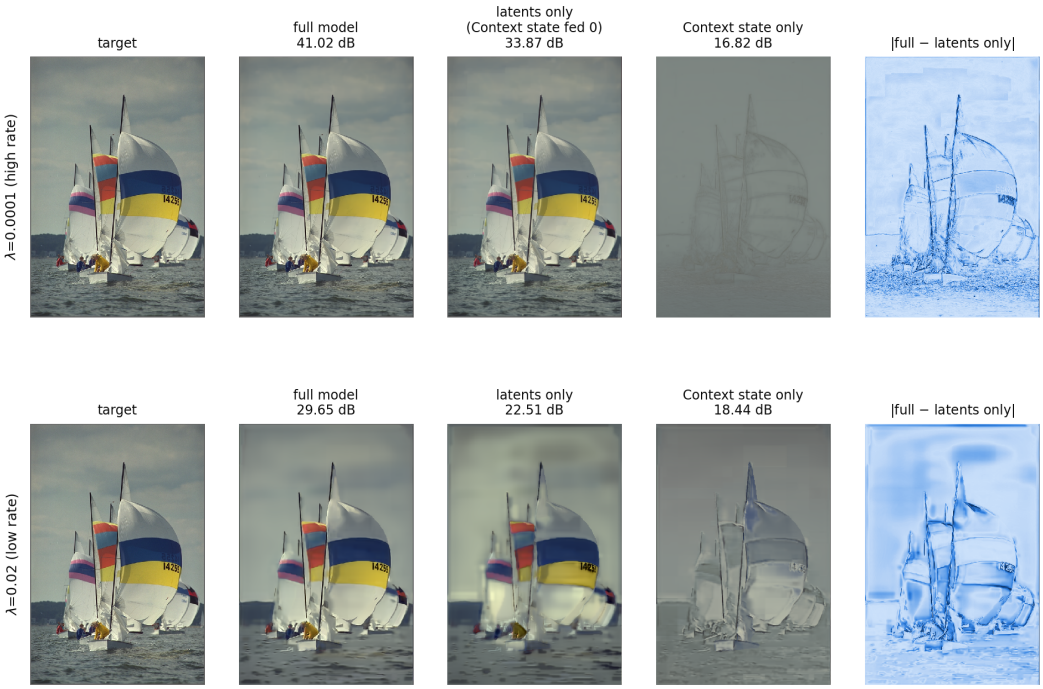


Figure 22: Context-state reuse on Kodak image 09 at $\lambda = 10^{-4}$ (top) and $\lambda = 2 \times 10^{-2}$ (bottom). Columns show the target, full reconstruction, reconstruction with the state input set to zero, the state-only diagnostic output, and the absolute difference between the full and state-zeroed reconstructions.

F.5 Context-state reuse in synthesis

Figures 22 and 23 examine how the fitted synthesis network uses the final context state $\mathbf{C}_T$ at two RD operating points. With the state input set to zero, the reconstruction retains the overall scene layout, but local contrast, object boundaries, and fine structures change visibly. The difference maps highlight the sails and water in Kodak image 09, and the lighthouse, roofs, and fence in Kodak image 19. For example, removing the state input from the low-rate Kodak 09 model reduces PSNR from 29.65 to 22.51 dB, indicating that the fitted reconstruction depends substantially on the shared context.

The state-only outputs also retain recognizable spatial structure, although their reconstruction quality is lower than that of the full model. Together, these views illustrate the complementary roles of the latent and state inputs in synthesis. They are diagnostics of an already fitted model, rather than separately optimized codec variants or an additive decomposition of its output. In particular, $\mathbf{C}_T$ is itself computed from the decoded latent hierarchy, so the state-only view does not represent compression without latents.

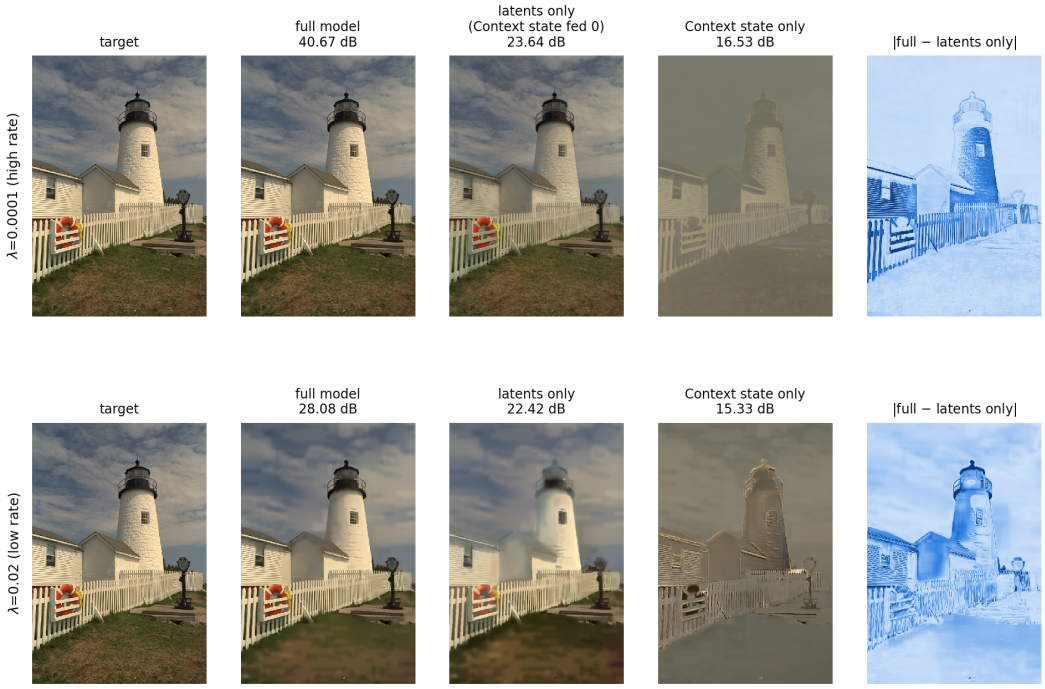


Figure 23: Context-state reuse on Kodak image 19 at $\lambda = 10^{-4}$ (top) and $\lambda = 2 \times 10^{-2}$ (bottom), following the diagnostic layout in Fig. 22. The difference maps reveal changes around the lighthouse, buildings, and fence when the synthesis state input is removed.

F.6 PROGRESSIVE EVOLUTION OF THE CONTEXT STATE

Figures 24 and 25 visualize selected decoding stages, showing the incoming state $\mathbf{C}_{t-1}$ and the updated state $\bar{\mathbf{C}}_t$ at the current grid resolution, alongside the ARM hidden-feature magnitude and the pooled region-conditioning map. The state passed to the next grid is $\mathbf{C}_t = \mathcal{U}_{\boldsymbol{\kappa}}(\bar{\mathbf{C}}_t; H_{t+1}, W_{t+1})$. Broad spatial patterns are visible at coarse resolutions, while later states exhibit more detailed responses around image structures. The differences between incoming and updated maps show that the updates can reorganize the retained context as new features become available. The CLIC2020 sequence includes both consecutive grids at the same resolution and transitions to finer grids, illustrating propagation across both types of hierarchy transition. Responses extend across the foreground and background, with the region map supplying a consistent spatial reference for the updates.

The two state channels exhibit different spatial patterns, retaining information about scene structure beyond region membership. The lower synthesis panels connect these representations to their use in reconstruction. Together, these visualizations illustrate how the compact state retains spatially structured context through the hierarchy and complements the decoded latents during synthesis. The color maps illustrate spatial patterns; activation magnitudes should only be compared when the display ranges are shared.

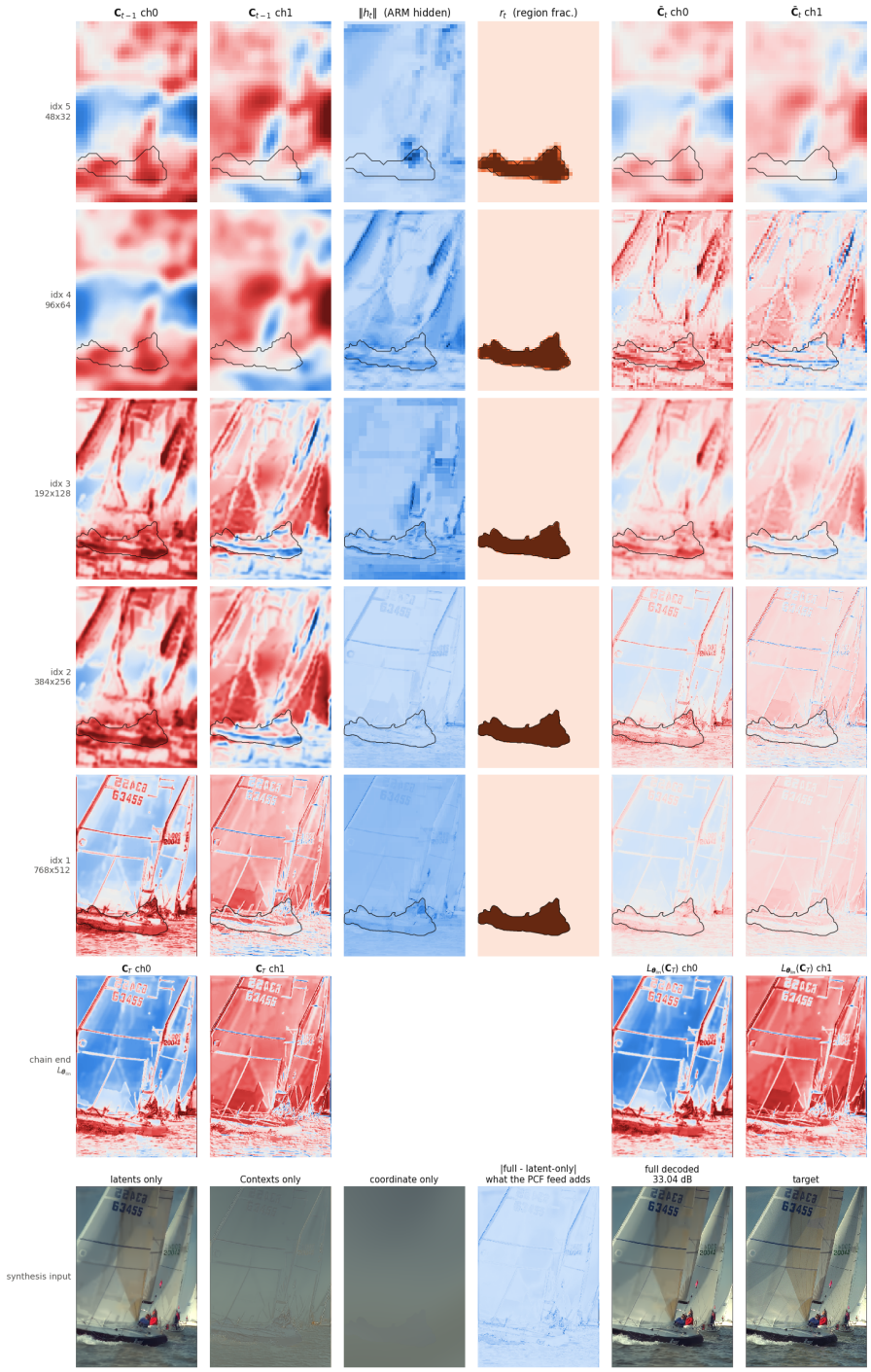


Figure 24: Selected state updates on Kodak image 10 at $\lambda = 4 \times 10^{-3}$. For each displayed stage, columns show the two channels of the incoming state $\mathbf{C}_{t-1}$, ARM hidden-feature magnitude, region conditioning, and the two channels of $\bar{\mathbf{C}}_t$ after the update and before upsampling to $\mathbf{C}_t$. Black contours indicate the region partition.

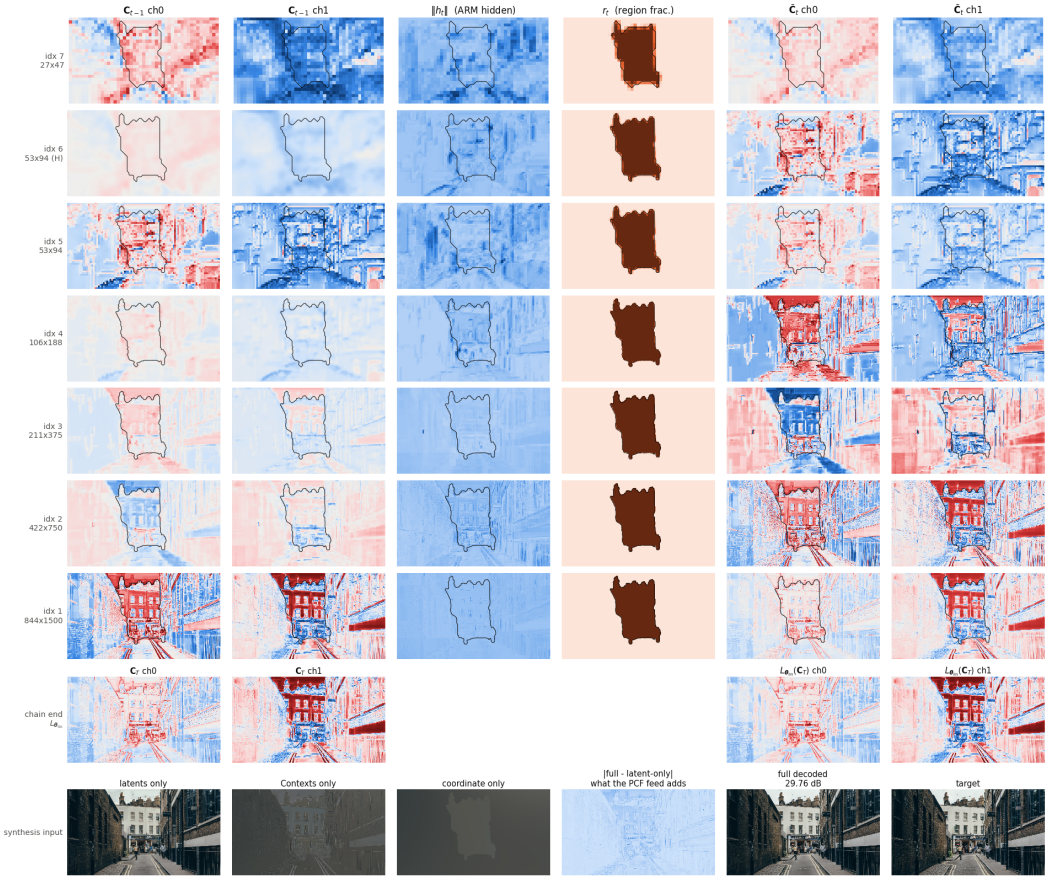


Figure 25: Selected state updates on a CLIC2020 image at $\lambda = 4 \times 10^{-3}$, following the layout in Fig. 24. The sequence includes adjacent grids at the same resolution and transitions to finer resolutions.

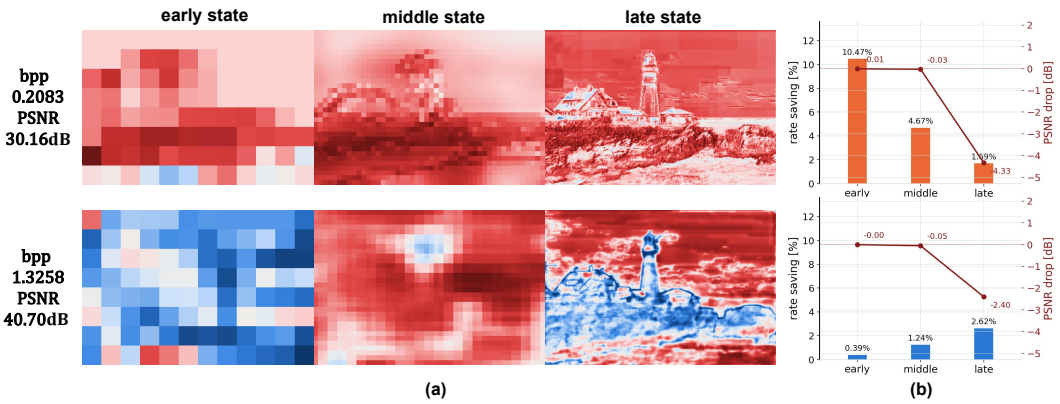


Figure 26: Progressive state visualization and contribution at two RD operating points on a kodak image. (a) Context states $\mathbf{C}_1$, $\mathbf{C}_5$, and $\mathbf{C}_9$. (b) Corresponding rate savings and PSNR drops when the state is ablated. Top and bottom rows use $\lambda = 4 \times 10^{-3}$ and 4×1^{-4}, respectively.