HyPIC: Rethinking Residual Representation for Progressive Image Compression

Anonymous authors
Paper under double-blind review

Abstract

Progressive image compression enables bandwidth-adaptive transmission by successively refining a coarse representation. These refinements can be encoded as residuals, allowing each stage to describe information not captured by the preceding approximation. Residual vector quantization (RVQ) implements this idea in learned codecs by quantizing the remaining latent residual and accumulating codewords to approximate the encoder output. We challenge latent residual approximation as the organizing principle of progressive refinement. Once a prefix establishes a base reconstruction, the remaining information should be encoded in a representation learned for improving the decoded image, without requiring faithful recovery of the original latent residual. To this end, we introduce Subspace-Projected Refinement(**SPR**), which jointly learns a compact representation of residual information and the directions through which its decoded coefficients update the VQ prefix. SPR is optimized for image rate–distortion, with a coding dimensionality chosen independently of the base latent's channel dimension. Individual coefficients adjust learned correction directions, while their synthesis jointly modifies the original latent channels. This structure leads to Hybrid Progressive Image Compression (**HyPIC**): VQ establishes the base reconstruction through shared joint patterns, and scalar quantization (SQ) encodes image-specific corrections through the learned SPR coefficients. Experimental results show that HyPIC attains competitive compression performance compared with previous methods. It achieves LPIPS BD-rate savings of up to 75.46% over Control-GIC across the four evaluated datasets. Beyond perceptual quality, HyPIC enables progressive transmission for flexibility, and also delivers faster encoding and decoding compared with ProGIC on GPUs for efficiency.

1 Introduction

Progressive image compression serves different bandwidth and quality requirements from one bitstream: a receiver decodes a coarse image and improves it as additional stages arrive. This goal connects classical successive refinement and standardized progressive coding with learned residual codecs (Equitz & Cover, 1991; CCITT, 1992; Wallace, 1992; Taubman et al., 2002). In residual vector quantization (RVQ), successive codewords approximate the encoder output by quantizing the remaining latent residual (Zhu et al., 2022; Flowers & Dey, 2024; Cao et al., 2026). More stages and context-adaptive codebooks improve this approximation (Huijben et al., 2024). A complementary question concerns the coordinates themselves: are those used to describe an image also the right coordinates for efficiently improving its reconstruction?

Once a VQ prefix establishes a base reconstruction, refinement should improve that image under an additional rate budget. Latent approximation error alone does not determine the value of a correction: its decoded effect must align with the image error left by the prefix. Our local reconstruction analysis makes this dependence explicit. It remains relevant under image-level training because the refinement representation determines which corrections the coded variables can express.

We introduce Subspace-Projected Refinement (SPR), which jointly learns compact coefficients from the latent residual and synthesis directions that update the VQ prefix. The maps are optimized for image rate–distortion, with a coefficient dimension chosen independently of the base latent width. Each scalar coefficient controls a direction spanning the original channels, so independent scalar

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

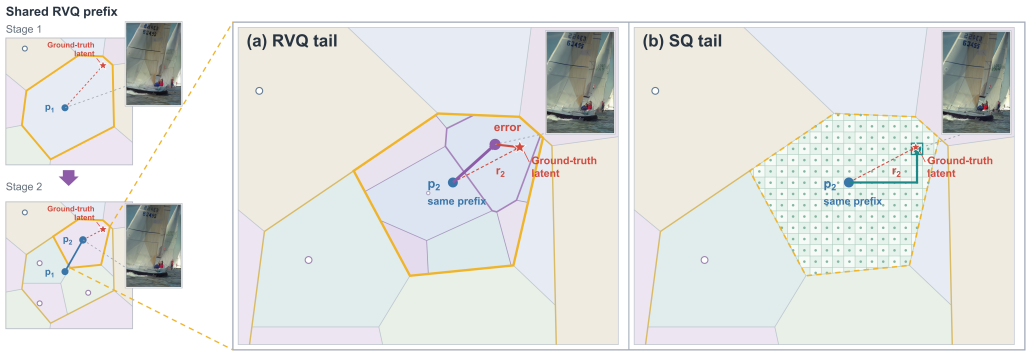

Figure 1: **Schematic comparison of RVQ and SQ refinement.** Two RVQ stages form a shared prefix. Its selected cell is enlarged to compare (a) joint codeword selection and (b) coordinate-wise scalar refinement. Gold outlines mark selected cells. Dots mark reconstruction points, red stars denote the target latent, and insets show example reconstructions.

quantization can still produce a joint latent correction. Hybrid Progressive Image Compression (HyPIC) combines these scalar-coded corrections with a VQ foundation: codewords establish shared joint patterns, then SPR coefficients adjust the learned correction directions. Figure 1 illustrates the complementary quantization operations.

HyPIC provides competitive perceptual rate–distortion performance over the evaluated ultra-low-rate range, with four progressive operating points from one model. Reconstruction diagnostics and ablations connect this performance to the learned correction representation. In particular, exact recovery of an encoder latent need not give the most useful decoder input, and fixed-dimensional reductions do not reproduce the benefit of learned directions at the tested moderate refinement payloads. Our contributions are:

- **A local design analysis of residual refinement.** We separate locally unreachable feature error from decoder-weighted coefficient error, motivating learned synthesis directions and scalar-coded amplitudes for progressive refinement.
- **SPR and hybrid progressive coding.** Independent analysis and synthesis maps produce compact scalar-coded corrections, combining a VQ foundation with image-specific refinement in HyPIC.
- **Competitive compression performance and speed.** Extensive experiments demonstrate competitive compression performance across multiple datasets, together with competitive encoding and decoding speed.

2 Related Work

Progressive image compression. Early recurrent codecs repeatedly encode residual information, enabling variable-rate reconstruction and spatially adaptive bit allocation (Toderici et al., 2015; 2017; Johnston et al., 2018). Layered residual coding also combines a conventional base with learned image refinement (Lee et al., 2019). DPICT and CTC progressively reveal latent precision through trit-plane coding and contextual refinement (Lee et al., 2022b; Jeon et al., 2023); variance-aware masking prioritizes residual latent elements (Presta et al., 2025). UQDM combines diffusion and universal quantization for progressive coding (Yang et al., 2025). RVQ instead accumulates latent codewords; ProGIC couples this structure with a lightweight generative decoder (Cao et al., 2026).

Transforms and entropy models. End-to-end learned codecs optimize analysis, quantization, and synthesis for rate–distortion (Ballé et al., 2016; Theis et al., 2017). Hyperpriors and autoregressive contexts improve probability estimates (Ballé et al., 2018; Minnen et al., 2018), with extensions to Gaussian mixtures and attention (Cheng et al., 2020), channel-wise prediction

(Minnen & Singh, 2020), and efficient spatial or grouped space–channel contexts (He et al., 2021; 2022). Mixed Transformer–CNN transforms improve representation learning (Liu et al., 2023), while multi-reference and hierarchical multiscale contexts expand entropy-model dependencies (Jiang et al., 2025; Li et al., 2025). Shallow synthesis transforms also expose the importance of encoder–decoder complexity allocation (Yang & Mandt, 2023b).

Quantization and compact representations. Scalar quantizer design and vector quantization provide complementary ways to trade reconstruction error for coding cost (Lloyd, 1982; Gray & Neuhoff, 1998). Learned discrete representations use individual or residual codebooks (Van Den Oord et al., 2017; Lee et al., 2022a), supporting vector-quantized codecs such as McQuic and NVTC (Zhu et al., 2022; Feng et al., 2023). FSQ quantizes a small number of scalar coordinates, while DAC projects audio features into a compact space for VQ (Mentzer et al., 2024; Kumar et al., 2023). QINCo adapts residual codebooks to preceding approximations (Huijben et al., 2024); HyperVQ and RDVQ couple codeword probabilities with side information or differentiable rate objectives (Yi et al., 2025; Jiang et al., 2026). SPR learns compact coordinates specifically for correction relative to a decoded prefix.

Generative and hybrid compression. Perceptual coding balances reconstruction distortion, realism, and rate (Blau & Michaeli, 2018; 2019). HiFiC and MS-ILLM use adversarial training, while multi-realism coding controls the generator's perceptual operating point (Mentzer et al., 2020; Muckley et al., 2023; Agustsson et al., 2023). Perceptually trained discrete vocabularies support generative image modeling (Esser et al., 2021), and GLC codes features within a generative VQ model (Jia et al., 2024). Diffusion codecs use conditional reconstruction, compressed global descriptions, or latent guidance (Yang & Mandt, 2023a; Careil et al., 2024; Li et al., 2024b); OSCAR provides one-step diffusion decoding across rates (Guo et al., 2026). DLF fuses VQ semantic and SQ detail branches (Xue et al., 2025). HyPIC instead assigns these quantizers successive roles: a VQ base followed by scalar-coded corrections in SPR coordinates.

3 METHOD

3.1 RECONSTRUCTION-AWARE RESIDUAL REFINEMENT

Consider an image x, its fixed encoder latent y, and a currently reconstructed latent p in the same channel space. Their residual is $r = y - p$, and a synthesis transform g_s maps p to the reconstructed image. Refinement adds a correction s to p. Its value is therefore determined by how $g_s(p + s)$ reduces error against x. To make this dependence explicit, consider a squared feature discrepancy for local analysis. Let $F = \phi \circ g_s$, where ϕ is an image feature map and may be the identity, and let $e_p = \phi(x) - F(p)$ denote the error against the original image. For corrections in a neighborhood where the first-order approximation $F(p + s) \approx F(p) + J_p s$ is appropriate, the induced local distortion model satisfies

$$\widetilde{D}_p(s) = \|e_p - J_p s\|_2^2, \qquad \widetilde{D}_p(0) - \widetilde{D}_p(s) = 2\langle e_p, J_p s\rangle - \|J_p s\|_2^2, \tag{1}$$

where J_p is the Jacobian of F at p. Thus, a correction's local reconstruction value depends on its decoded effect and its alignment with the image error remaining after the prefix. Euclidean approximation of r alone does not express these two quantities. This motivates learning coordinates for the correction task while retaining the residual as the encoder-side source of information.

This local design analysis motivates SPR's separation of learned correction directions from scalar-coded amplitudes. The directions determine the reachable decoded changes, while analysis and quantization determine how accurately coded coefficients realize them. Appendix B separates these effects, connecting the architecture to the representation, precision, and width studies in Appendix A.

3.2 HYBRID PROGRESSIVE RECONSTRUCTION

Figure 2(a) shows how HyPIC constructs a reconstruction in two complementary steps: residual vector quantization establishes a sequence of base representations, and Subspace-Projected Refinement (SPR) describes a final correction in learned coefficient coordinates. The design separates

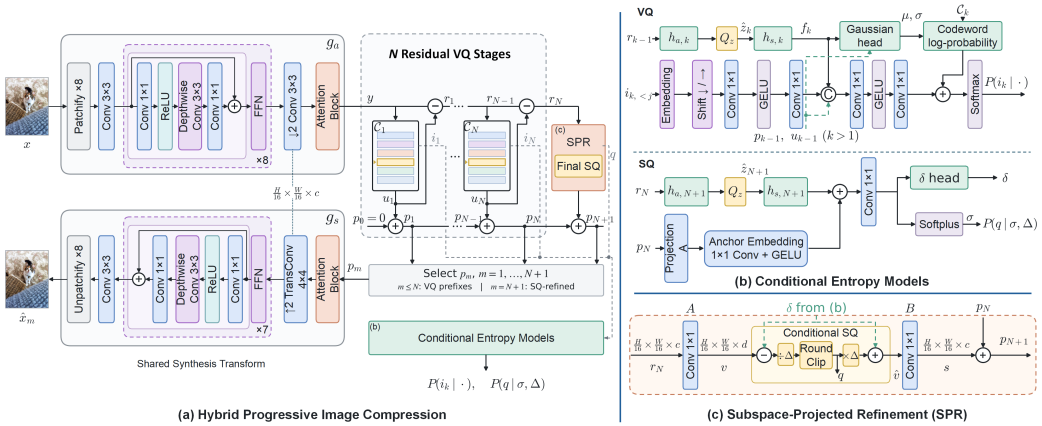

Figure 2: HyPIC architecture. (a) Residual VQ constructs progressively accumulated reconstructions, followed by a final SPR correction; every prefix uses the same synthesis transform. (b) Conditional entropy models for VQ indices and scalar refinement symbols. (c) SPR analyzes the remaining residual into compact coefficients, quantizes them conditionally, and synthesizes a correction in the original latent channel space.

the representation used to build the image from the representation used to improve an existing reconstruction. VQ contributions and the synthesized SPR correction are accumulated in the same c-channel reconstruction space and decoded by a shared synthesis transform; the encoder latent y remains fixed for a given input.

Given an image $x \in \mathbb{R}^{H \times W \times 3}$, an analysis transform g_a produces a latent y with c channels. Starting from $p_0 = 0$, VQ stage k contributes a decoded codeword tensor u_k, yielding

$$y = g_a(x), \qquad p_k = p_{k-1} + u_k, \qquad r_k = y - p_k, \qquad \hat{x}_m = g_s(p_m). \tag{2}$$

Here $k = 1, \ldots, N$, $r_0 = y$, and p_k is the accumulated prefix. SPR maps the remaining residual r_N into d coefficient channels, scalar-quantizes these coefficients, and synthesizes a correction s that gives $p_{N+1} = p_N + s$. A receiver selects any available prefix $m \in \{1, \ldots, N+1\}$ and applies the same synthesis transform g_s. The conditional entropy models in Figure 2(b) use information available at the corresponding reconstruction stage. Figure 2(c) expands the compact coefficient analysis, scalar quantization, and correction synthesis within SPR.

The transforms preserve a common $H/16 \times W/16$ latent grid throughout quantization and refinement. An 8×8 pixel rearrangement and a stride-two convolution produce this grid; repeated depthwise-convolution and feed-forward blocks operate at fixed spatial resolution. The synthesis transform reverses the spatial rearrangement and sampling; both transforms include attention at latent resolution. Sharing g_s makes the VQ prefixes and the SPR update operate in a common reconstruction space. The quantizer count determines how many intermediate reconstructions are exposed, whereas the coefficient width determines the representation budget of the final correction.

3.3 Subspace-Projected Refinement and Conditional Coding

SPR represents the remaining residual through a learned channel bottleneck. It uses independent linear analysis and synthesis maps $A \in \mathbb{R}^{d \times c}$ and $B \in \mathbb{R}^{c \times d}$, implemented by bias-free 1×1 convolutions. At every spatial position,

$$v = Ar_N = Ay - Ap_N, \qquad s = B\hat{v}, \qquad p_{N+1} = p_N + B\hat{v}, \tag{3}$$

where $\hat{v}$ is the reconstructed coefficient tensor. The maps preserve spatial resolution, and $d < c$ sets the number of coefficient channels independently of the base latent width. The columns of B are learned correction directions: changing one coefficient changes the amplitude of one direction across the original c channels. Their combination therefore produces a joint latent correction, although the coefficients themselves are quantized separately.

For a fixed prefix, the decoder acts on the correction through $J_p B\hat{v}$, with B applied position-wise. Learning A selects residual information for coding; learning B determines how the coefficients change the reconstructed latent. The maps are independent and optimized through the image objective. Their linear form preserves $A(y - p_N) = Ay - Ap_N$ and shared synthesis directions, without imposing $B\hat{v} \approx r_N$ as the reconstruction target.

The two quantizers act on different representation units. At position j, VQ stage k selects an index $i_{k,j}$ from codebook $\mathcal{C}_k$ by nearest-codeword assignment to $r_{k-1,j}$, and recovers $u_{k,j} = \mathcal{C}_k[i_{k,j}]$. Each choice supplies a jointly specified channel pattern. After the VQ prefix is established, SQ adjusts the amplitudes of the learned SPR directions independently. Given a predicted coefficient offset δ, step size $\Delta > 0$, and integer bound L, the refinement is

$$q = \mathrm{clip}_{[-L,L]}\left(\mathrm{round}\left(\frac{v - \delta}{\Delta}\right)\right), \qquad \hat{v} = \delta + \Delta q. \tag{4}$$

All operations are element-wise. The offset centers the scalar symbols; it is restored before the synthesis map B combines the coefficients. Consequently, the final stage sends a field of coefficient adjustments rather than another index into a collection of complete channel vectors. Its d coefficients per spatial position control a correction in c latent channels. This representation specifies where the quantization budget is spent; the conditional entropy model determines the rate of the resulting symbols.

Each stage supplies side information through a dedicated hyperprior:

$$z_k = h_{a,k}(r_{k-1}), \qquad \hat{z}_k = Q_z(z_k), \qquad f_k = h_{s,k}(\hat{z}_k), \quad k = 1, \ldots, N+1. \tag{5}$$

In particular, the SQ hyperanalysis observes the full-space residual r_N. The hyperlatent is modeled by a factorized Gaussian distribution with learned channel scales. Its quantized reconstruction determines f_k, so the probability parameters predicted from these features need not be represented as separate side tensors.

For VQ, the categorical model combines a Gaussian score evaluated at each codeword with learned discrete logits. If $g_{k,j,\ell}$ and $a_{k,j,\ell}$ denote these two scores for codeword ℓ, then

$$P(i_{k,j} = \ell \mid f_k, p_{k-1}, u_{k-1}, i_{k,<j}) = \frac{\exp(g_{k,j,\ell} + a_{k,j,\ell})}{\sum_t \exp(g_{k,j,t} + a_{k,j,t})}. \tag{6}$$

The Gaussian branch uses hyperfeatures and the preceding decoded prefix and codeword contribution. The discrete branch additionally embeds previously decoded indices and extracts their top and left spatial context. These causal neighbors are denoted by $i_{k,<j}$. The first stage omits the preceding-prefix inputs. Adding the two logits lets geometric codeword compatibility and contextual index statistics jointly determine the categorical distribution.

For SQ, the same analysis map A projects the decoded prefix to Ap_N. A 1×1 convolution and GELU embed this projection into anchor features, which are added to f_{N+1}. A parameter head then predicts δ and a positive scale σ. These are decoder-available conditions: the prefix comes from the VQ stages, and the hyperfeatures come from $\hat{z}_{N+1}$. The centered symbol likelihood is a discretized zero-mean Gaussian. For an interior symbol, with t indexing a scalar coefficient across spatial positions and channels,

$$P(q_t \mid \sigma_t, \Delta) = \Phi\left(\frac{\Delta(q_t + 1/2)}{\sigma_t}\right) - \Phi\left(\frac{\Delta(q_t - 1/2)}{\sigma_t}\right), \qquad -L < q_t < L, \tag{7}$$

where Φ is the standard normal cumulative distribution function. The two endpoint symbols collect the respective tail probabilities. The scale parameter controls this likelihood, while δ determines coefficient centering and restoration. The two predictions thus have distinct roles: the offset adjusts the scalar reconstruction lattice to the decoded prefix, and the scale assigns probabilities to the integer displacement on that lattice. The same offset is available when restoring the coefficients, so only q and the stage hyperlatent are counted as new refinement symbols.

For rate optimization, we define a stage-wise rate term from the main-symbol and hyperlatent negative log-likelihoods. Writing $b_k = i_k$ for VQ and $b_{N+1} = q$ for SQ,

$$R_k = \frac{-\log_2 P(\hat{z}_k) - \log_2 P(b_k \mid \text{decoded conditions})}{HW}, \qquad R_{\le m} = \sum_{k=1}^{m} R_k. \tag{8}$$

The second expression gives the cumulative rate term associated with reconstruction $\hat{x}_m$. This accounting includes each stage's side information alongside the symbols it helps predict.

3.4 Learning Across Progressive Prefixes

We train all $N+1$ reconstructions in each iteration using weighted all-prefix supervision (Cao et al., 2026). The weights allocate emphasis between the intermediate VQ prefixes and the final SPR output, giving each supported operating point a direct image loss.

Let $D_k = \alpha_k \|x - \hat{x}_k\|_1 + \beta_k \,\mathrm{LPIPS}(x, \hat{x}_k)$, allowing the photometric and perceptual weights to differ across operating points. With commitment penalty $\mathcal{L}_{\mathrm{com},k}$ and rate multiplier λ_R, the objective is

$$\mathcal{L} = \sum_{k=1}^{N+1} w_k \left[D_k + \lambda_{\mathrm{com}} \mathcal{L}_{\mathrm{com},k} + \lambda_R R_k \right], \qquad w_k = \begin{cases} \rho/N, & k \le N, \\ 1-\rho, & k = N+1. \end{cases} \tag{9}$$

Here $0 < \rho < 1$ assigns total weight ρ to the VQ reconstructions and $1-\rho$ to the refined output, independently of N. Intermediate losses train usable prefixes; the final loss trains their interaction with SPR through the shared decoder. The rate term uses marginal R_k, avoiding repeated counting of early-stage likelihoods.

For VQ, commitment penalizes the distance between the stage residual and its stop-gradient codeword contribution; codebooks are updated by exponential moving averages. For SQ, commitment is evaluated in the projected space as the mean squared difference between Ay and $\mathrm{sg}(Ap_N + \hat{v})$, where sg stops gradients. This auxiliary constraint couples projected coefficients to their quantized reconstruction without requiring the synthesized correction to recover the original full-space residual. Straight-through quantization supplies reconstruction gradients, and additive uniform noise approximates hyperlatent rounding during training. The image losses and conditional rate models jointly determine how the shared representations support the progressive operating points.

4 Experiments

4.1 Experimental Setup

Implementation Details. HyPIC's residual VQ stages can be stacked to an arbitrary depth; our main experimental comparison uses three residual VQ codebooks, followed by one scalar-quantized SPR stage. The SPR width d is likewise freely adjustable and is studied by ablation; the main experiments use $d = 10$, providing a compact coefficient representation independently of the base latent width. Each codebook contains 1,024 vectors in a 256-channel latent space, and the scalar step size is fixed to $\Delta = 1$. The three VQ prefixes and the final SPR correction provide four progressively decodable operating points from a single model, spanning approximately 0.03–0.20 bpp in our evaluation. We set $\rho = 0.2$ in Eq. (9), assigning total weights of 0.2 and 0.8 to the VQ prefixes and final reconstruction, respectively. Reported BPP is computed from the resulting entropy-coded bitstreams.

Training Details. We train HyPIC on the full ImageNet-1k training set using 256×256 random crops and random horizontal flipping. Optimization uses Adam with $(\beta_1, \beta_2) = (0.5, 0.9)$ and a constant learning rate of 10^{-4}. Training runs for 1.5 million updates with a global batch size of 32 on two NVIDIA RTX 4090 GPUs using mixed BF16 precision.

Evaluation Datasets and Metrics. We evaluate on Kodak (24 images), Tecnick (100 images at 1200×1200), DIV2K validation (100 images), and the CLIC 2020 total collection (428 images). LPIPS-VGG (Zhang et al., 2018) and DISTS (Ding et al., 2020) are our primary perceptual metrics; PSNR, MS-SSIM, FID, and NIQE results are provided in Appendix C . We compute LPIPS BD-rate relative to Control-GIC in Table 1 .

4.2 Rate–Distortion Performance

Comparison with Previous Methods. We compare with the GAN-based HiFiC (Mentzer et al., 2020) and MS-ILLM (Muckley et al., 2023), the vector-quantized Control-GIC (Li et al., 2024a),

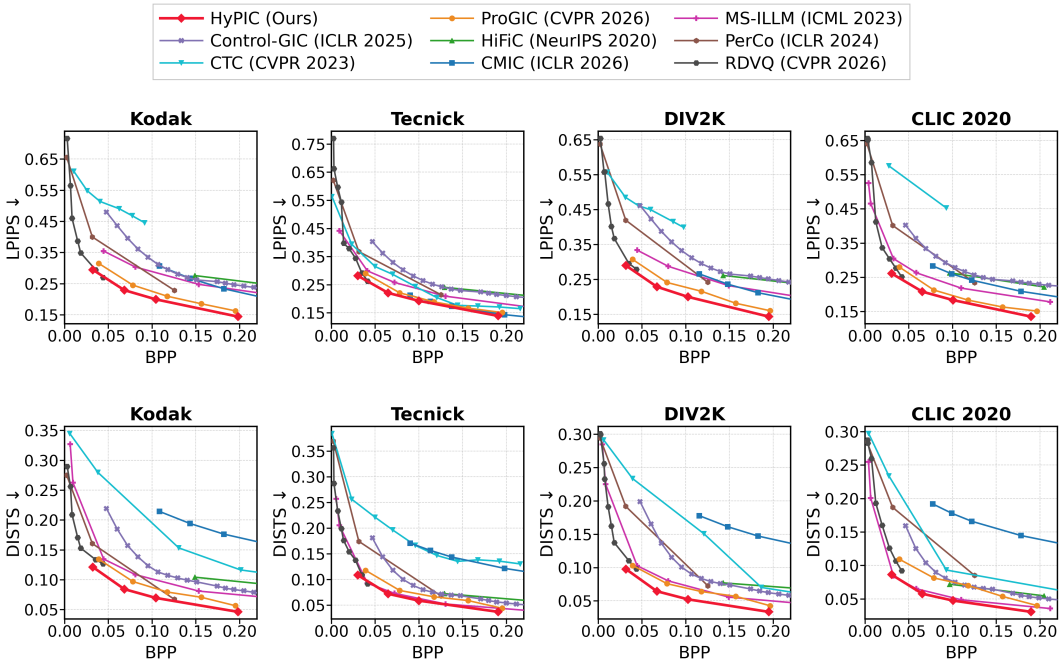

Figure 3: Rate–distortion performance on the Kodak, Tecnick, DIV2K, and CLIC2020 datasets, evaluated using LPIPS and DISTS. Curves closer to the lower left indicate better rate–distortion performance.

Table 1: Computational cost on 512×768 Kodak images using a single NVIDIA RTX 4090 GPU, and LPIPS-based BD-rate relative to Control-GIC on four datasets. The lowest and second-lowest reported values are shown in bold and underlined, respectively.

Method	Enc. (ms)	Dec. (ms)	FLOPs (G)	Params (M)	BD-rate (LPIPS)			
					Kodak	Tecnick	DIV2K	CLIC 2020
HiFiC (NeurIPS 2020)	899.21	2274.62	437.89	148.50	11.27%	3.06%	−18.03%	−11.12%
MS-ILLM (ICML 2023)	70.00	114.08	547.60	181.48	−28.48%	−51.69%	−41.88%	−62.86%
Control-GIC (ICLR 2025)	86.03	293.80	10608.47	130.36	0.00%	0.00%	0.00%	0.00%
PerCo (ICLR 2024)	53.04	2188.87	46535.01	1295.87	−50.01%	−34.50%	−38.24%	−19.36%
CTC (CVPR 2023)	676.23	1111.61	4382.19	399.00	53.08%	−39.67%	25.98%	74.11%
CMIC (ICLR 2026)	221.28	154.12	891.72	69.80	−18.85%	−60.72%	−41.65%	−23.11%
RDVQ (CVPR 2026)	414.96	417.29	3090.29	251.84	**−79.07%**	−60.97%	**−75.35%**	−69.85%
ProGIC (CVPR 2026)	7.79	6.83	343.66	34.06	−61.61%	−56.15%	−64.32%	−63.98%
HyPIC (Ours)	**5.59**	**6.39**	**245.84**	**30.68**	−72.08%	**−64.08%**	−74.53%	**−75.46%**

RDVQ (Jiang et al., 2026), and ProGIC (Cao et al., 2026), the diffusion-based PerCo (Careil et al., 2024), and the learned codecs CTC (Jeon et al., 2023) and CMIC. Figure 3 presents their perceptual rate–distortion curves. As an official ProGIC implementation was unavailable for our evaluation, we use an in-house reimplementation to measure its computational cost on the same hardware.

HyPIC achieves substantial perceptual rate savings across the four evaluated datasets. Relative to Control-GIC, its LPIPS BD-rate reductions are 72.08%, 64.08%, 74.53%, and 75.46% on Kodak, Tecnick, DIV2K, and CLIC, respectively. Table 1 reports lower Control-GIC-referenced BD-rate values for HyPIC on Tecnick and CLIC, and for RDVQ on Kodak and DIV2K. The plotted HyPIC operating points span approximately 0.03–0.20 bpp, whereas the selected RDVQ configurations span 0.0025–0.044 bpp. HyPIC covers its broader evaluated rate range with four progressively decodable operating points from one model.

Qualitative comparison. Figure 4 compares Kodak reconstructions at the indicated bitrates. In `kodim23`, HyPIC retains the parrot's eye and beak boundaries and facial feather lines; in `kodim14`, its face and clothing contours are less smoothed than those of CMIC and ProGIC.

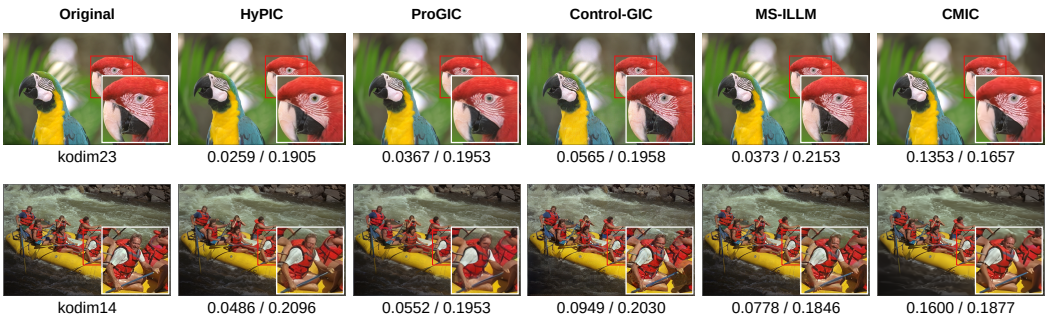

Figure 4: Qualitative comparisons for Kodak images `kodim23` and `kodim14` at the indicated bitrates. Red boxes mark the regions magnified in the insets, with the same crop used for all methods within each scene. Labels report full-image DISTS/BPP.

4.3 Ablation Studies

Same-backbone comparison and latent recovery. On Kodak, Table 2 compares five-stage RVQ with the adapted main 3RVQ+SPR model using identical encoder and decoder backbone architectures, the same 256-channel latent space, and the same training recipe. The two models are trained independently and evaluated at closely matched total rates. At these operating points, 3RVQ+SPR improves both PSNR and LPIPS over 5RVQ. Separately, the 5RVQ bypass feeds the exact encoder output to the unchanged RVQ decoder. This within-model diagnostic eliminates latent quantization error without changing the decoder, yet does not uniformly improve image quality: LPIPS improves slightly, whereas PSNR decreases. This result indicates that eliminating latent quantization error alone does not guarantee improved reconstruction quality.

Table 2: Same-backbone comparison on Kodak. Eliminating latent quantization error alone does not guarantee better image quality.

Method	BPP	PSNR (dB)	LPIPS
5RVQ	0.1953	25.56	0.1624
3RVQ+SPR	0.1982	26.02	0.1440
5RVQ (bypass)	—	25.36	0.1609

Table 3: Effect of activations on Kodak. BD-rate changes are relative to activation-free mappings.

Mapping	LPIPS-VGG	DISTS
Activation-free	0.00%	0.00%
+SiLU	+ 6.42%	+ 7.41%
+ReLU	+10.15%	+ 9.63%

Activation-free mappings. To assess the activation choice in SPR, we compare SiLU and ReLU variants with the activation-free reference on Kodak. Table 3 shows LPIPS-VGG and DISTS BD-rate increases of 6.42% and 7.41% for SiLU, and 10.15% and 9.63% for ReLU, respectively. These results support the activation-free analysis and synthesis mappings used in HyPIC.

VQ depth and scalar refinement. Figure 5 compares one- to four-VQ hybrids and pure SQ using the same image-backbone architecture and ten-channel SQ with $\Delta = 1$. In the measured four-VQ model, SPR improves LPIPS from 0.1853 to 0.1400 and DISTS from 0.0656 to 0.0450 as total rate increases from 0.1432 to 0.2344 bpp. Additional representation ablations appear in Appendix A.

Implications for refinement. The ablations support a division of labor between the VQ foundation and scalar refinement. The VQ prefixes provide complete reconstructions, while SPR spends the additional rate on amplitudes along learned synthesis directions. A scalar symbol can therefore express a coordinated correction across latent channels. The activation results favor retaining this linear combination in the tested mappings.

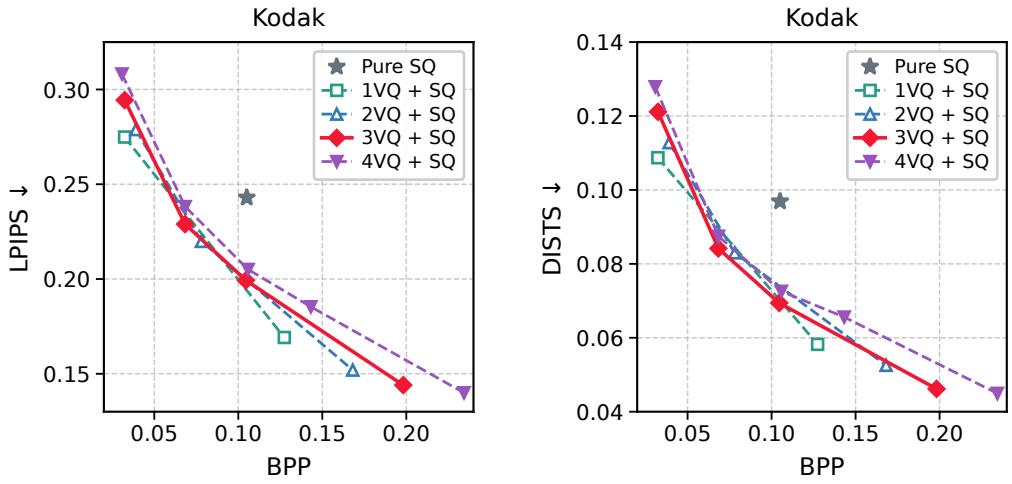

Figure 5: Effect of VQ prefix depth on Kodak.

The coded controls in Appendix A.1 further distinguish useful correction directions from compactness alone. With the same frozen decoder and correction-payload accounting, the learned SPR correction directions improve LPIPS and DISTS over the tested fixed projections and channel selectors at 0.05 and 0.075 payload bpp; the low-payload comparisons show that this advantage is rate-dependent. Choosing a refinement width thus involves both the range of corrections the representation can express and the cost of transmitting their coefficients. More dimensions provide additional freedom, but their value depends on how the decoder converts them into image changes. This motivates evaluating the refinement at matched coded payloads and in the reconstructed image, rather than selecting its dimension from latent error or captured variance alone.

5 Conclusion

HyPIC treats residual refinement as a representation-learning problem within progressive compression. A VQ foundation is followed by SPR, which learns compact scalar-coded coefficients together with their synthesis directions. The evaluated models provide competitive perceptual rate–distortion performance at ultra-low rates and progressive reconstruction from one codec. The ablations connect image-quality gains to learned correction directions; controlled coding comparisons show their advantage over the tested fixed projections and channel selectors at 0.05 and 0.075 correction-payload bpp. These findings motivate jointly designing the correction space and its coding mechanism.

Limitations. The present design fixes the coefficient width and uses spatially shared linear maps. Rate- or content-adaptive correction spaces may offer different tradeoffs. Our local analysis uses a first-order decoder approximation and quadratic feature discrepancy; it motivates the representation without providing a global optimality guarantee.

Appendix roadmap. Appendix A develops the experimental argument for SPR; Appendix B gives the local theoretical analysis; Appendix C reports additional quantitative results; and Appendix D provides further visual comparisons.

AI use statement

ChatGPT was used only to aid and polish the writing of the manuscript, and for literature retrieval and discovery to identify related work that might otherwise be overlooked. It was not used for research ideation or execution, implementation or debugging, experiments, mathematical proofs, or dataset generation. The authors take full responsibility for all research decisions, interpretations, and the final content, including any AI-assisted material.

Ethics statement

This work studies lossy image compression using existing image datasets. Perceptual reconstruction can alter fine details or produce plausible content that differs from the source; applications requiring faithful visual evidence should account for this behavior. Dataset use and any redistribution remain subject to the original terms.

Reproducibility statement

Section 3 specifies the architecture, quantizers, and training objective; Section 4 documents optimization settings, datasets, and evaluation metrics. Appendix A details the frozen-model controls, calibration data, quantization settings, and correction-payload accounting. Appendix B gives the assumptions and derivations of the local analysis, and Appendix C provides additional quantitative evaluations. We will release the code, configurations, and model weights upon acceptance.

References

Eirikur Agustsson, David Minnen, George Toderici, and Fabian Mentzer. Multi-realism image compression with a conditional generator. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22324–22333. IEEE, 2023.

Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.

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, Philip A Chou, David Minnen, Saurabh Singh, Nick Johnston, Eirikur Agustsson, Sung Jin Hwang, and George Toderici. Nonlinear transform coding. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):339–353, 2020.

Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6228–6237. IEEE, 2018.

Yochai Blau and Tomer Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff. In *International Conference on Machine Learning*, pp. 675–685. PMLR, 2019.

Hao Cao, Chengbin Liang, Wenqi Guo, Zhijin Qin, and Jungong Han. Progic: Progressive and lightweight generative image compression with residual vector quantization. *arXiv preprint arXiv:2603.02897*, 2026.

Marlene Careil, Matthew J Muckley, Jakob Verbeek, and Stéphane Lathuilière. Towards image compression with perfect realism at ultra-low bitrates. In *International conference on learning representations*, volume 2024, pp. 15544–15564, 2024.

CCITT. Information technology—digital compression and coding of continuous-tone still images—requirements and guidelines. Recommendation T.81, International Telecommunication Union, September 1992.

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.

Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE transactions on pattern analysis and machine intelligence*, 44(5):2567–2581, 2020.

William HR Equitz and Thomas M Cover. Successive refinement of information. *IEEE Transactions on information theory*, 37(2):269–275, 1991.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

Runsen Feng, Zongyu Guo, Weiping Li, and Zhibo Chen. Nvtc: Nonlinear vector transform coding. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6101–6110. IEEE, 2023.

Bryse Flowers and Sujit Dey. Bric: Bottom-up residual vector quantization for learned image compression. *IEEE Access*, 12:153105–153126, 2024.

Vivek K Goyal. Theoretical foundations of transform coding. *IEEE Signal processing magazine*, 18 (5):9–21, 2001.

Robert M. Gray and David L. Neuhoff. Quantization. *IEEE transactions on information theory*, 44 (6):2325–2383, 1998.

Jinpei Guo, Yifei Ji, Zheng Chen, Kai Liu, Min Liu, Wang Rao, Wenbo Li, Yong Guo, and Yulun Zhang. Oscar: One-step diffusion codec across multiple bit-rates. *Advances in Neural Information Processing Systems*, 38:85267–85286, 2026.

Dailan He, Yaoyan Zheng, Baocheng Sun, Yan Wang, and Hongwei Qin. Checkerboard context model for efficient learned image compression. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14766–14775. IEEE, 2021.

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.

Iris AM Huijben, Matthijs Douze, Matthew Muckley, Ruud JG Van Sloun, and Jakob Verbeek. Residual quantization with implicit neural codebooks. *arXiv preprint arXiv:2401.14732*, 2024.

Seungmin Jeon, Kwang Pyo Choi, Youngo Park, and Chang-Su Kim. Context-based trit-plane coding for progressive image compression. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14348–14357. IEEE, 2023.

Zhaoyang Jia, Jiahao Li, Bin Li, Houqiang Li, and Yan Lu. Generative latent coding for ultra-low bitrate image compression. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26088–26098. IEEE, 2024.

Shiyin Jiang, Wei Long, Minghao Han, Zhenghao Chen, Ce Zhu, and Shuhang Gu. Differentiable vector quantization for rate-distortion optimization of generative image compression. *arXiv preprint arXiv:2604.10546*, 2026.

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.

Nick Johnston, Damien Vincent, David Minnen, Michele Covell, Saurabh Singh, Troy Chinen, Sung Jin Hwang, Joel Shor, and George Toderici. Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4385–4393. IEEE, 2018.

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved rvqgan. *Advances in neural information processing systems*, 36:27980–27993, 2023.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11513–11522. IEEE, 2022a.

Jae-Han Lee, Seungmin Jeon, Kwang Pyo Choi, Youngo Park, and Chang-Su Kim. Dpict: Deep progressive image compression using trit-planes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16113–16122, 2022b.

Wei-Cheng Lee, David Alexandre, Chih-Peng Chang, Wen-Hsiao Peng, Cheng-Yen Yang, and Hsueh-Ming Hang. Learned image compression with residual coding. In *CVPR Workshops*, pp. 0, 2019.

Anqi Li, Feng Li, Yuxi Liu, Runmin Cong, Yao Zhao, and Huihui Bai. Once-for-all: Controllable generative image compression with dynamic granularity adaptation. *arXiv preprint arXiv:2406.00758*, 2024a.

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.

Zhiyuan Li, Yanhui Zhou, Hao Wei, Chenyang Ge, and Jingwen Jiang. Toward extreme image compression with latent feature guidance and diffusion prior. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(1):888–899, 2024b.

Jinming Liu, Heming Sun, and Jiro Katto. Learned image compression with mixed transformer-cnn architectures. In *2023 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 14388–14397. IEEE, 2023.

Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2): 129–137, 1982.

Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. *Advances in neural information processing systems*, 33:11913–11924, 2020.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. In *International Conference on Learning Representations*, volume 2024, pp. 51772–51783, 2024.

David Minnen and Saurabh Singh. Channel-wise autoregressive entropy models for learned image compression. In *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 3339–3343. IEEE, 2020.

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.

Matthew J Muckley, Alaaeldin El-Nouby, Karen Ullrich, Hervé Jégou, and Jakob Verbeek. Improving statistical fidelity for neural image compression with implicit local likelihood models. In *International Conference on Machine Learning*, pp. 25426–25443. PMLR, 2023.

Alberto Presta, Enzo Tartaglione, Attilio Fiandrotti, Marco Grangetto, and Pamela Cosman. Efficient progressive image compression with variance-aware masking. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 7692–7700. IEEE, 2025.

David S Taubman, Michael W Marcellin, and Majid Rabbani. Jpeg2000: Image compression fundamentals, standards and practice. *Journal of Electronic Imaging*, 11(2):286–287, 2002.

Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*, 2017.

George Toderici, Sean M O'Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar. Variable rate image compression with recurrent neural networks. *arXiv preprint arXiv:1511.06085*, 2015.

George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 5306–5314, 2017.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Gregory K Wallace. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.

Naifu Xue, Zhaoyang Jia, Jiahao Li, Bin Li, Yuan Zhang, and Yan Lu. Dlf: Extreme image compression with dual-generative latent fusion. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19227–19236. IEEE, 2025.

Ruihan Yang and Stephan Mandt. Lossy image compression with conditional diffusion models. *Advances in Neural Information Processing Systems*, 36:64971–64995, 2023a.

Yibo Yang and Stephan Mandt. Computationally-efficient neural image compression with shallow decoders. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 530–540. IEEE, 2023b.

Yibo Yang, Justus Will, and Stephan Mandt. Progressive compression with universally quantized diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 37998–38019, 2025.

Niu Yi, Xu Tianyi, Ma Mingming, and Wang Xinkun. Hypervq: Enabling hyperprior entropy modeling for vq-based generative image compression. *arXiv preprint arXiv:2512.07192*, 2025.

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

Xi Zhang and Xiaolin Wu. Lvqac: Lattice vector quantization coupled with spatially adaptive companding for efficient learned image compression. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10239–10248. IEEE, 2023.

Yue Zhao, Yuanjun Xiong, and Philipp Krähenbühl. Image and video tokenization with binary spherical quantization. In *International Conference on Learning Representations*, volume 2025, pp. 90844–90868, 2025.

Xiaosu Zhu, Jingkuan Song, Lianli Gao, Feng Zheng, and Heng Tao Shen. Unified multivariate gaussian mixture for efficient neural image compression. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17591–17600. IEEE, 2022.

A Experimental Exploration of Refinement

Transform coding couples coordinates, quantization, and bit allocation (Goyal, 2001); nonlinear codecs learn these jointly with synthesis and entropy models (Ballé et al., 2020). We ask which correction coordinates improve a progressive reconstruction, whether statistical conditioning explains their benefit, and how width changes coding cost and quality.

The first two studies freeze the three-VQ model used in the main Kodak evaluation. Alternatives share its encoder, prefix, and SPR-trained decoder, providing a mechanism diagnostic within this codec rather than a comparison of independently trained alternatives. Calibration uses 128 class-stratified ImageNet training images, resized and center-cropped to 256×256, and evaluation uses all 24 full-resolution Kodak images. The width study uses separately trained four-VQ variants.

A.1 Representation for Decoded Correction

Does latent recovery determine correction value? For prefix p_3, let $r = y - p_3$ and $v = Ar$. We compare the unrefined prefix, exact encoder latent, unquantized SPR correction, and native quantized SPR output. Holding the decoder fixed separates fidelity to the encoder latent from the image improvement delivered by the correction.

Table 4: Reconstruction references for the same frozen three-VQ model on Kodak. Continuous-latent decoding has zero latent quantization error. Image metrics are averaged over all 24 images.

Decoder input	PSNR (dB) ↑	LPIPS ↓	DISTS ↓
VQ prefix p_3	23.32	0.1993	0.0694
Continuous latent y	22.59	0.1910	0.1129
Unquantized SPR $p_3 + BAr$	26.24	0.1400	0.0450
Native SPR $p_3 + B\hat{v}$	26.02	0.1440	0.0462

Native SPR improves PSNR by 3.43 dB over decoding the exact latent y, with lower LPIPS and DISTS (Table 4). The learned correction therefore supplies a more useful input to this decoder than complete recovery of the raw residual. Appendix B explains how this distinction depends on decoded correction directions.

Do fewer coordinates suffice? Compact quantization spaces are effective in scalar image tokenization and residual audio coding (Mentzer et al., 2024; Kumar et al., 2023). To distinguish compactness from learned correction directions, we compare SPR with rank-ten PCA, three rank-ten random orthonormal projections, and two fixed ten-channel selectors, alongside the full residual. Variance-prioritized residual coding motivates the channel-selection control (Presta et al., 2025). Our selectors use calibration residual energy or a signed first-order image-loss reduction at p_3. The latter averages spatially summed negative gradient–residual products for mean $\ell_1 + 2\,\mathrm{LPIPS}_{\mathrm{VGG}}$ on raw decoder outputs. Both selections are fixed before Kodak evaluation, leave other channels uncorrected, and require no per-image index signaling.

PCA uses the leading eigenvectors of the uncentered calibration second moment. With U, Q, and T denoting its basis, a random basis, and a channel selector, the coefficient/synthesis pairs are (r, I), (Ar, B), $(U^\top r, U)$, $(Q^\top r, Q)$, and $(Tr, T^\top)$. Let $S \in \{I, B, U, Q, T^\top\}$ denote the corresponding synthesis map. Coefficient means are restored within that map, so coding-time centering leaves the chosen subspace unchanged. With calibrated step vector $\boldsymbol{\eta}$, each representation uses the unrestricted scalar quantizer,

$$q_j = \mathrm{round}\left(\frac{c_j - \mu_j}{\eta_j}\right) \in \mathbb{Z}, \qquad \hat{x} = g_s(p_3 + S(\mu + \boldsymbol{\eta} \odot q))\,, \tag{10}$$

where μ is the calibration mean and uniform η_j sweep 14 multiples of the pooled calibration standard deviation. Per-channel integer CDFs use additive 0.5 smoothing and 16-bit precision, with exact signed bypass for out-of-support symbols. Reconstruction uses symbols decoded from rANS payloads. The rate counts termination, bypass, and applicable side bytes, excluding the shared prefix and calibration parameters. This diagnostic uses a separate calibrated static correction coder so that all alternative correction representations are evaluated under the same payload accounting.

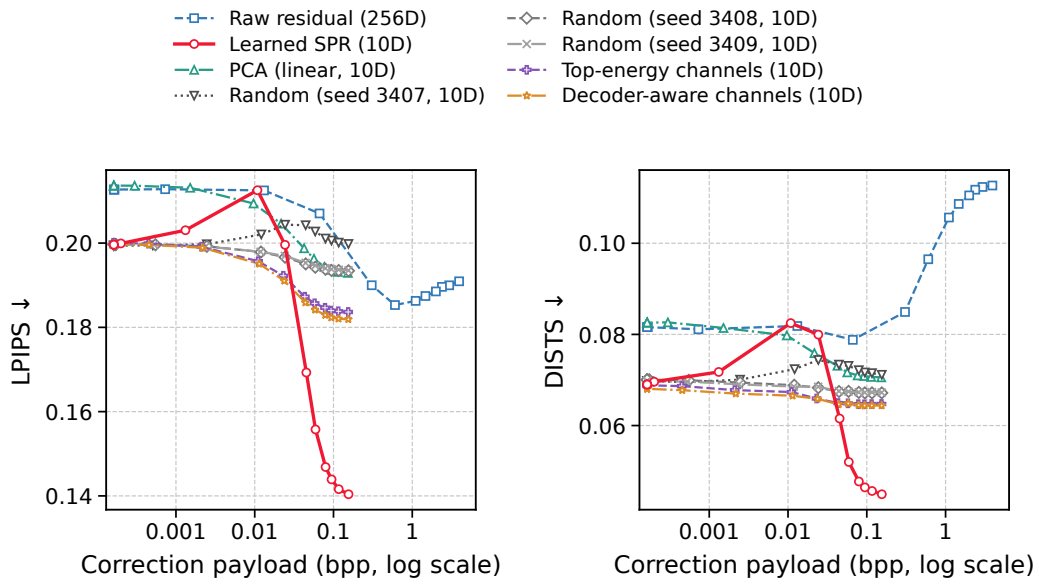

Figure 6: Correction representations under a common integer quantizer on Kodak. All 14 step settings are connected in multiplier order. Rates count actual correction payloads; every branch uses the same frozen SPR-trained decoder. Channel selections are fixed from calibration data, and all three random runs are shown.

At correction payloads of 0.05 and 0.075 BPP, SPR improves LPIPS and DISTS over every tested alternative (Figure 6). At 0.075 BPP, its LPIPS/DISTS are $0.1488/0.0486$, compared with $0.1945/0.0711$ for PCA, $0.1848/0.0647$ for energy selection, and $0.1832/0.0645$ for decoder-aware selection. Across random seeds, LPIPS ranges from 0.1939 to 0.2014 and DISTS from 0.0673 to 0.0723. Values at fixed payloads are linearly interpolated in BPP only along a unique adjacent measured segment with step multipliers of at least one. This pointwise comparison differs from BD-rate, which integrates a log-rate difference over a shared quality interval.

The value of the learned directions depends on coefficient precision. Both selectors outperform SPR at 0.01 and 0.025 BPP, and random projections also improve on SPR at 0.01 BPP. At 0.025 BPP, SPR has LPIPS close to the unrefined prefix but higher DISTS (0.0793 versus 0.0694). Thus, a useful continuous correction need not remain useful after coarse scalar quantization. Finer raw-residual quantization approaches $y = p_3 + r$; its high-rate DISTS increase is consistent with the exact-latent reference above. Together, these results connect correction value to both its decoded direction and the precision with which its amplitude is coded.

A.2 Do Statistical Treatments Explain the Benefit?

Adaptive companding and normalized quantization spaces treat local scale and geometry as coding variables (Zhang & Wu, 2023; Zhao et al., 2025). This raises a different explanation for SPR: does its benefit come from easier scalar statistics, and can additional scaling provide a substitute or further gain?

Table 5 pools 36,864 Kodak latent positions. Channel imbalance is the ratio of the 95th to 5th percentile channel standard deviations. We also report median excess kurtosis, mean absolute off-diagonal correlation, and the coefficient of variation of position-wise RMS norms. SPR has more balanced channel scales and lower excess kurtosis, while correlation and norm variation change modestly.

We apply uniform SQ, channel-wise step scaling, and local RMS normalization to raw residuals and SPR coefficients under the same actual-payload protocol. Channel scales are calibration standard deviations normalized to geometric mean one. Local norms use eight calibrated log-RMS levels;

Table 5: Coordinate statistics of the frozen three-VQ model.

Representation	Std. ratio	Excess kurtosis	Mean $\lvert\mathrm{corr}\rvert$	Norm CoV
Raw residual r (256D)	1.574	1.144	0.0457	0.314
SPR coefficients $v = Ar$ (10D)	1.045	0.315	0.0426	0.297

both quantization and reconstruction use the decoded norm, and its packed three-bit indices are counted. Table 6 compares three shared targets using the same interpolation rule. The norm indices alone cost 0.01172 BPP, making the 0.01 target unavailable for this treatment.

Table 6: Statistical treatments at matched actual correction payloads on Kodak. Each pair reports interpolated LPIPS/DISTS; lower is better. Local-norm rates include its packed side information.

		0.025 BPP		0.050 BPP		0.075 BPP	
Space	Treatment	LPIPS	DISTS	LPIPS	DISTS	LPIPS	DISTS
Raw (256D)	Uniform	0.2113	0.0812	0.2087	0.0797	0.2064	0.0790
	Channel scale	0.2134	0.0820	0.2132	0.0820	0.2121	0.0826
	Local norm	0.2119	0.0813	0.2099	0.0811	0.2068	0.0807
SPR (10D)	Uniform	0.1985	0.0793	0.1646	0.0582	0.1488	0.0486
	Channel scale	0.1986	0.0794	0.1647	0.0583	0.1487	0.0487
	Local norm	0.2132	0.0842	0.1888	0.0777	0.1556	0.0542

Uniform SQ gives lower LPIPS and DISTS than either raw-space scaling treatment at all three targets, although all three raw-space treatments yield higher distortions than the unrefined prefix. Within SPR, at the precision reported in Table 6, the LPIPS and DISTS changes from channel scaling are at most 0.0001. Counting its side information, local normalization increases both distortions: at 0.075 BPP, LPIPS changes from 0.1488 to 0.1556 and DISTS from 0.0486 to 0.0542. Thus, these treatments do not substitute for the trained correction representation under this protocol; channel rescaling adds little within SPR.

A.3 Width and Coding Efficiency

The width study uses four-VQ-plus-SPR models with $d \in \{4, 8, 16, 32\}$, identical configurations except width, and the same warm start for compatible weights. We compare Kodak validation at 100,000 updates. These models are separate from the three-VQ, ten-channel main model.

Table 7: Width comparison at 100,000 updates. LPIPS gain equals Prefix minus Final.

	Rate (BPP)			LPIPS		
Width d	VQ prefix	SQ tail	Total	Prefix ↓	Final ↓	Gain ↑
4	0.1429	0.0336	0.1765	0.1774	0.1577	0.0197
8	0.1429	0.0693	0.2122	0.1831	0.1467	0.0363
16	0.1424	0.1408	0.2832	0.1886	0.1296	0.0590
32	0.1419	0.2837	0.4256	0.1923	0.1059	0.0864

Tail cost grows approximately with width while VQ-prefix rate stays near 0.143 BPP (Table 7). Larger tail rate and lower final LPIPS occur at every shared validation step. At the matched step, prefix LPIPS increases while final LPIPS decreases, revealing a progressive-quality tradeoff under joint training. We use $d = 10$ in the three-VQ main model as a practical design choice separately from this four-VQ width comparison.

B Local Geometry of Residual Refinement

This section develops the reconstruction-aware analysis in Section 3.1. Fix a decoded prefix p and write $F = \phi \circ g_s$, $e_p = \phi(x) - F(p)$, and $J_p = \nabla F(p)$. For a sufficiently small correction s,

linearizing F gives a local feature-distance model. A fixed positive-semidefinite feature weighting H yields

$$\widetilde{D}_{p,H}(s) = (e_p - J_p s)^\top H (e_p - J_p s), \qquad M_p = J_p^\top H J_p. \tag{11}$$

The unweighted model in Eq. (1) corresponds to $H = I$. Expanding the quadratic gives

$$\widetilde{D}_{p,H}(0) - \widetilde{D}_{p,H}(s) = 2e_p^\top H J_p s - s^\top M_p s. \tag{12}$$

Thus, a correction is useful through its alignment with the current image-feature error and its decoded effect. Raw residual energy $\|y - p\|^2$ alone does not determine this improvement: the target offset e_p need not equal $J_p(y - p)$.

Reachable corrections and coefficient precision. Let $\mathcal{B} = I_{hw} \otimes B$ denote the position-wise application of the learned synthesis map to vectorized coefficients. Define $b = H^{1/2} e_p$ and $K = H^{1/2} J_p \mathcal{B}$. For $s = \mathcal{B}\hat{v}$, the local distortion is $\|b - K\hat{v}\|^2$. With $K^\dagger$ the Moore–Penrose pseudoinverse and $v^\star = K^\dagger b$, orthogonal projection gives the exact decomposition within this local model:

$$\widetilde{D}_{p,H}(\mathcal{B}\hat{v}) = \|(I - KK^\dagger)b\|^2 + (\hat{v} - v^\star)^\top K^\top K (\hat{v} - v^\star). \tag{13}$$

The first term is the local weighted feature error outside the range of K: coefficient precision alone cannot remove it for fixed B, prefix, and decoder linearization. The second term is decoder-weighted coefficient error relative to $v^\star$, including both assignment by the learned analysis map and quantization. The metric $K^\top K$ couples their decoded effects even when scalar symbols are quantized independently. Learning B changes the reachable correction space; learning A and coding its output determine which correction is realized within it. The entropy model assigns coding costs to these choices.

Quantization around a learned correction. For arbitrary coefficients v and quantization perturbation $\epsilon = \hat{v} - v$, the local distortion change is

$$\widetilde{D}_{p,H}(\mathcal{B}(v + \epsilon)) - \widetilde{D}_{p,H}(\mathcal{B}v) = -2(b - Kv)^\top K\epsilon + \epsilon^\top K^\top K \epsilon. \tag{14}$$

The linear term vanishes at a least-squares stationary point, but need not vanish for the learned analysis output. In particular, clipping and conditional mean centering do not guarantee an unbiased, independent quantization error. SPR optimizes the decoded reconstruction while retaining a coefficient-space commitment term; its training therefore need not minimize raw latent error or an unweighted coefficient error in isolation.

The fixed-direction and channel-selection comparisons in Appendix A.1 examine the correction space, while their payload sweeps examine finite-rate coefficient precision. The separate width study in Appendix A.3 examines the tradeoff between correction width, rate, and reconstruction. These experiments test the practical consequences of the local design analysis.

Scope of the local model. Equations (11)–(14) describe a first-order decoder approximation with a quadratic feature-distance surrogate. They do not identify $J_p^\top H J_p$ with the complete Hessian of the implemented ℓ_1–LPIPS objective. Even for a smooth feature loss $\ell(F(z))$, the exact Hessian also contains $\sum_i (\partial_i \ell) \nabla^2 F_i$. The local model is useful where the decoder linearization is accurate. The bias-free, activation-free maps additionally preserve the exact identity $A(y - p) = Ay - Ap$ and a shared set of synthesis directions; A and B remain independently learned rather than constrained inverses.

C Additional Quantitative Results

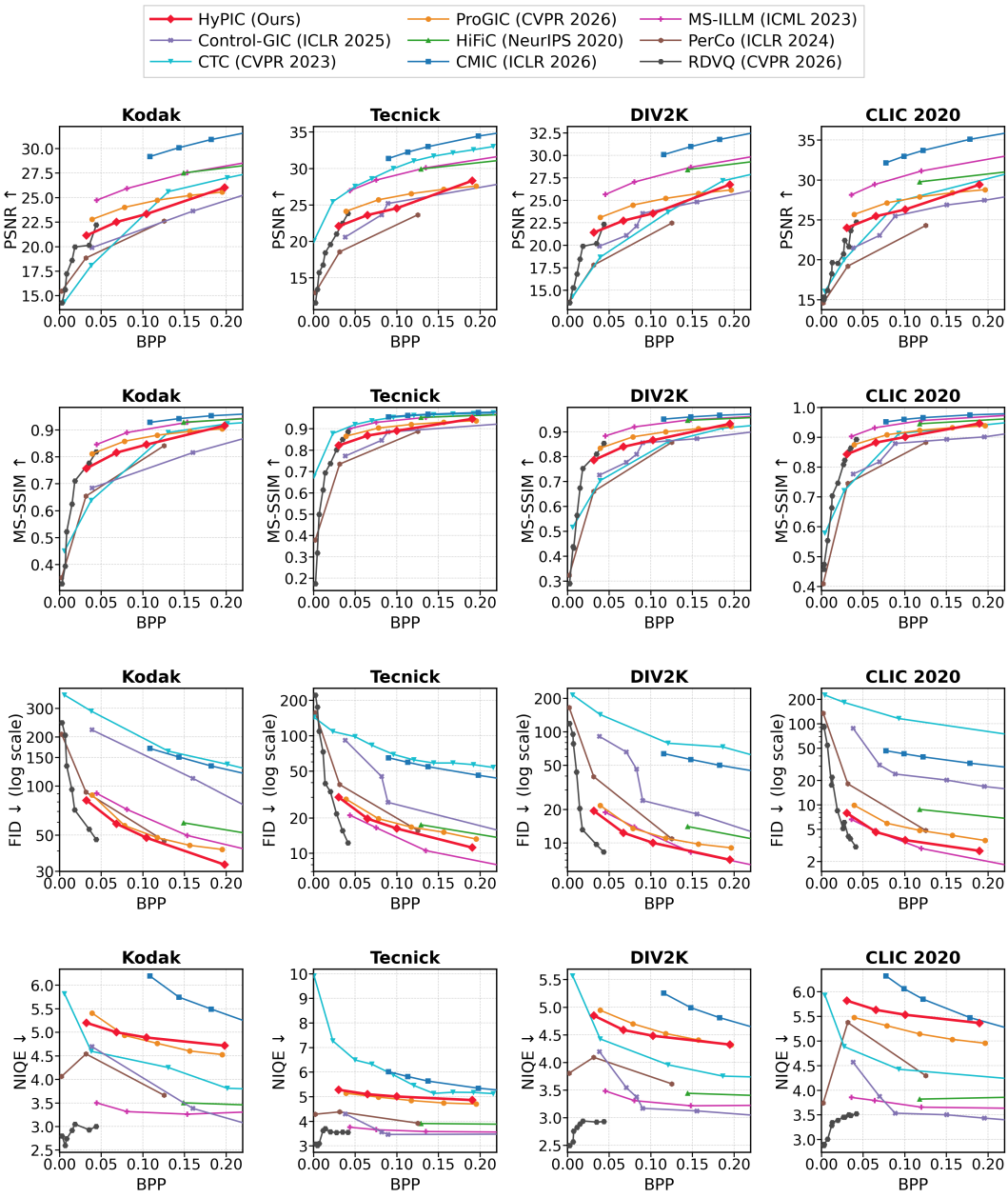

Figure 7: Additional rate–distortion metrics under the same source protocols as Figure 3. Rows show PSNR, MS-SSIM, FID, and NIQE. FID uses a logarithmic vertical scale.

D Additional Qualitative Results

Figures 8–11 extend the comparisons in Figure 4 with four additional Kodak images and one image each from Tecnick, DIV2K, and CLIC2020. The Kodak crops expose petal structure, face-paint boundaries, wheel spokes, and window divisions. HyPIC retains these structural cues, while the Control-GIC motorcycle crop exhibits pronounced texture changes. The additional scenes show transparent glass, patterned ceramics, and book lettering; the HyPIC crops preserve bowl decoration

and lettering that are softened in the CMIC outputs. In the three dataset-specific figures, the central scene combines an original upper portion with a HyPIC lower portion. The red and gold regions are compared across all methods in the left and right crop stacks, respectively, at the individually displayed operating points.

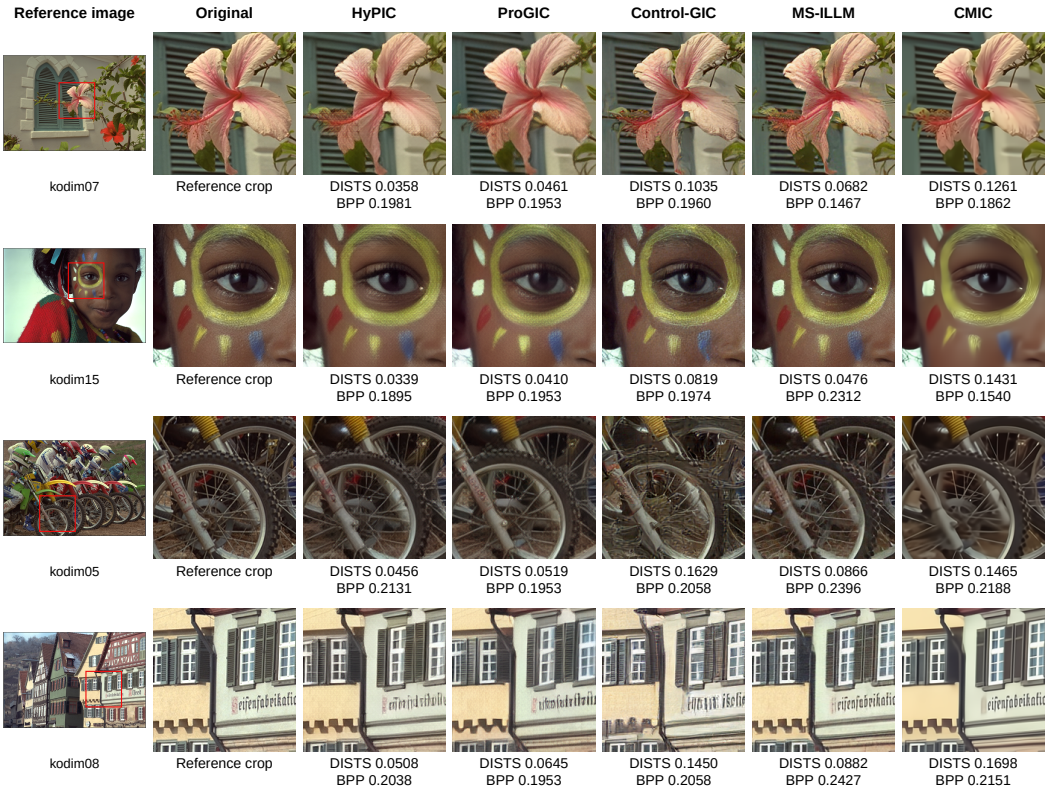

Figure 8: Additional Kodak comparisons: `kodim07`, `kodim15`, `kodim05`, and `kodim08`. The reference image locates the common crop used within each row. Labels report full-image DISTS and BPP at the displayed operating points; the metrics are not computed on the crops.



Figure 9: Qualitative comparison on Tecnick image 028. The central scene combines the original upper portion and HyPIC lower portion. Red region A is enlarged on the left and gold region B on the right, with identical crops across methods. Labels report full-image DISTS/BPP.

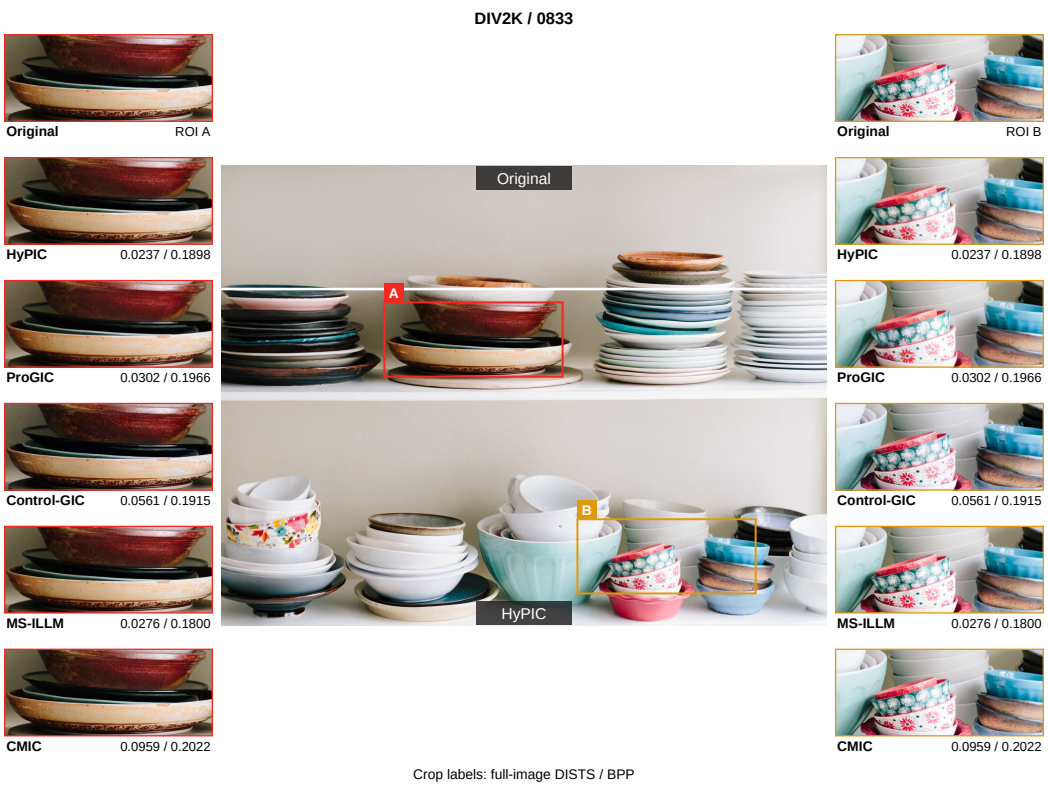

Figure 10: Qualitative comparison on DIV2K image 0833. The central scene combines the original upper portion and HyPIC lower portion; the left and right stacks compare the red and gold regions across methods. Labels report full-image DISTS/BPP at the selected operating points.

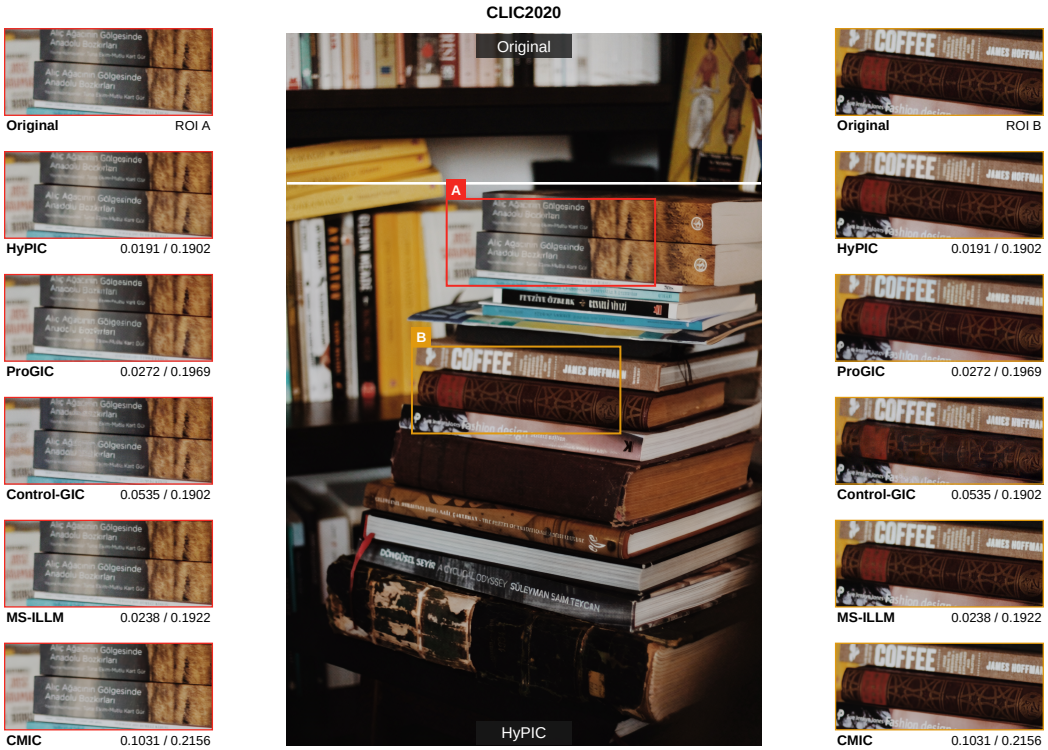

Figure 11: Qualitative comparison on a CLIC2020 book scene. The central scene combines the original upper portion and HyPIC lower portion; the left and right stacks compare the red and gold regions across methods. Labels report full-image DISTS/BPP at the selected operating points.