HyperVQ: Enabling Hyperprior Entropy Modeling for VQ-Based Generative Image Compression

Yi Niu **Tianyi Xu** **Mingming Ma*** **Xinkun Wang**
Xidian University
{25171111417,25171213993}@stu.xidian.edu.cn
niuyi@mail.xidian.edu.cn, mamingming@xidian.edu.cn

Abstract

Vector Quantization (VQ) based generative image compression has achieved remarkable perceptual quality. However, existing VQ codecs suffer from two fundamental limitations. First, they lack efficient content-adaptive entropy modeling and rely on static frequencies, leading to low coding efficiency. Second, efficient differentiable index-probability modeling remains challenging for joint Rate-Distortion (RD) optimization. To resolve these issues, we propose HyperVQ, a principled framework that establishes a high-performance hyperprior entropy foundation for VQ-based codecs. The core insight of HyperVQ is to shift probability modeling entirely into the continuous embedding space. Instead of directly predicting probabilities for discrete symbols, HyperVQ predicts a high-dimensional continuous multivariate Gaussian distribution for the continuous latents. By treating the discrete codebook entries as fixed "anchors" in this space, we convert the continuous Gaussian density into categorical index probabilities based on relative distances. This elegant formulation provides a powerful, spatially-adaptive entropy engine and renders the cross-entropy rate objective fully differentiable, empowering the network to actively and dynamically optimize the RD trade-off during training. To ensure practicality, we design the lightweight H Block and the Probability Estimation Engine (PEE) to facilitate highly parallel, millisecond-level inference. Experiments demonstrate that HyperVQ acts as a universal module across diverse VQ architectures (single-scale, large-codebook, RVQ), achieving an average bitrate saving of 18.5%, which is 7.28$\times$ the saving achieved by conventional Huffman coding. This establishes a robust, RD-controllable foundation for next-generation generative image compression.

1 Introduction

In recent years, deep learning-based image compression has made significant progress [1]. End-to-end optimized autoencoders with powerful entropy models [2, 3] have become the dominant paradigm. Typically based on Scalar Quantization (SQ), they surpass classical codecs like VVC [4] on pixel-level metrics (e.g., PSNR). However, optimizing for pixel-level metrics often leads to perceptually poor results, particularly at low bitrates.

To address this, Generative Image Compression (GIC) models (e.g., GANs [5], VQ-VAEs [6]) optimize the Rate-Distortion-Perception (RDP) trade-off, enabling highly realistic reconstructions at extremely low bitrates. Among these, Vector Quantization (VQ) based paradigms are particularly favored for discretizing features into semantically rich codebook indices.

Despite VQ's excellent performance in perceptual reconstruction, it faces a dual bottleneck that severely limits its potential. First, existing VQ-based codecs lack efficient, content-adaptive entropy

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

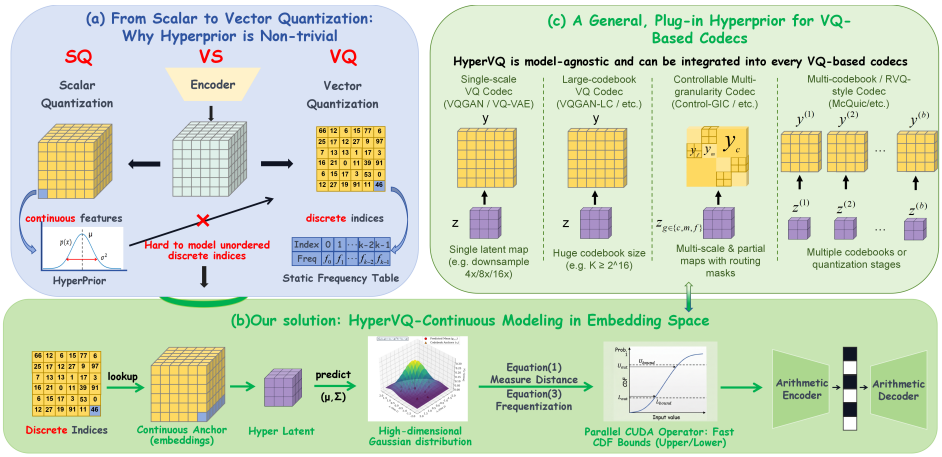

Figure 1: **HyperVQ: A Plug-and-Play Entropy Model and RD Optimizer for VQ-Based Codecs.** We present HyperVQ, which enables high-dimensional Gaussian modeling in the continuous embedding space of VQ indices to replace static frequency tables, bridging the gap between VQ and advanced entropy modeling. As a model-agnostic plug-in module, it supports every VQ-based codec variants out of the box.

modeling. By employing static frequency tables, they assume a stationary global distribution that ignores the recurring local patterns and spatial redundancies present in diverse image contents. Second, and more critically, efficient differentiable probability modeling of discrete, unordered VQ indices remains challenging for joint Rate-Distortion (RD) optimization. Because prior codecs lack a differentiable, content-adaptive probability model for the indices, the index-rate term remains unavailable during training. Consequently, the bitrate is often steered by manual heuristics (e.g., codebook size, downsampling factor) rather than actively optimized by the network.

In this paper, we introduce **HyperVQ**, a principled hyperprior entropy framework designed to resolve these bottlenecks. The core insight of HyperVQ lies in shifting the probability modeling entirely into the continuous embedding space. Instead of directly predicting probabilities for discrete symbols, HyperVQ predicts a high-dimensional continuous multivariate Gaussian distribution for the continuous latents at each spatial location. By treating the discrete codebook entries $\{\mathbf{e}_k\}$ as fixed "anchors" in this space, we mathematically convert this continuous Gaussian density into categorical index probabilities based on their relative distances (§3.2). This formulation not only provides a powerful, spatially-adaptive entropy engine, but crucially, it renders the cross-entropy rate objective fully differentiable. By establishing this differentiable gradient force field in the embedding space, HyperVQ empowers the network to actively and dynamically optimize the RD trade-off during training, effectively shifting RD scheduling from hand-tuned heuristics to the network itself.

As a universal, plug-and-play module, HyperVQ seamlessly integrates across diverse VQ scenarios, including single-scale, large-codebook, residual quantization, and multi-granularity settings (§4.2). By successfully adapting to varied structural constraints, including the entropy modeling of hierarchical scales and the handling of partial feature maps caused by arbitrary rate control (§3.3.3), HyperVQ rigorously proves its adaptive capability and potential to serve as a foundational component for the entire VQ family.

Our main contributions are summarized as follows:

- We propose **HyperVQ**, the first principled hyperprior entropy model that models discrete index probabilities differentiably by shifting probability modeling entirely into the continuous embedding space.
- We mathematically derive a fully differentiable cross-entropy rate objective that maps continuous Gaussian densities to discrete categorical probabilities. This establishes an elegant gradient force field that brings true end-to-end, dynamic Rate-Distortion (RD) optimization to VQ-based generative compression.
- To ensure practicality, we design the lightweight **H Block** and the **Probability Estimation Engine (PEE)** to facilitate highly parallel, millisecond-level inference (§3.3). We demon-

strate that HyperVQ acts as a universal module, achieving an average bitrate saving of 18.5%, which is 7.28× the saving achieved by conventional Huffman coding.

- We instantiate HyperVQ on the most demanding controllable multi-granularity setting, termed **HyperVQ-Controllable Generative Image Compression (HVQ-CGIC)**, achieving state-of-the-art performance across multiple metrics; notably, it obtains over 61% bit reduction on Kodak while maintaining the same LPIPS as the previous SOTA (§4.3).

2 Related Work

2.1 Learned image compression

Grounded in Shannon's rate-distortion theory [7], Ballé et al. [1] established the dominant paradigm of end-to-end optimization based on Scalar Quantization (SQ). Subsequent research has advanced this paradigm by exploring powerful architectures and accurate probability models.

Architectural advancements include sophisticated nonlinear transforms [8–11], content-adaptive mechanisms [12–14], and Transformer [15, 16] or Diffusion-based models [17, 18].

For entropy modeling, the Hyperprior [2] parameterizes a conditional distribution $p(\hat{\mathbf{y}}|\mathbf{z})$ for continuous latents. Combined with autoregressive priors [3, 19, 20], these learned methods finally outperformed BPG in PSNR, inspiring numerous subsequent R-D improvements [5, 21–34].

While effective, these foundational works are almost exclusively designed for SQ. Vector Quantization (VQ) offers superior theoretical rate-distortion properties but has been less common. Beyond early soft-to-hard VQ [35] and recent Lattice VQ (LVQ) frameworks [36–39], effectively modeling the entropy of discrete, unordered VQ indices remains a significant gap.

2.2 VQ-Based generative image compression

Recently, VQ-based generative compression has evolved to optimize the Rate-Distortion-Perception (RDP) trade-off across diverse architectures. These include standard single-scale VQ codecs [6, 40], large-codebook designs for higher fidelity [41], controllable multi-granularity codecs for variable bitrates [42], and multi-codebook or Residual VQ (RVQ) models utilizing cascading quantization stages [43].

Across all these families, efficient entropy modeling remains a critical bottleneck. Existing codecs estimate index entropy using static, global frequency tables, failing to exploit content-specific recurring patterns and strong spatial redundancies. We address this fundamental gap by shifting probability modeling entirely into the continuous embedding space. HyperVQ introduces a general, codebook-size-independent hyperprior entropy model serving as a universal plug-in for all aforementioned VQ families. It enables true end-to-end Rate-Distortion optimization via a fully differentiable cross-entropy objective, completely bypassing the parameter explosion of discrete classification. NVTC [37] applies ECVQ [44] within learned transforms. Jiang et al. [45] combine differentiable codebook assignments with autoregressive priors (Appendix A.3).

3 Method

3.1 Framework overview

In this section, we present HyperVQ, a general entropy modeling framework. As illustrated in Figure 2, HyperVQ is designed as a universal component that can be integrated into diverse vector quantization (VQ) based codecs. Our design aims to resolve the inefficiency of static frequency tables by introducing content-adaptive, learned entropy modeling. The framework consists of three core components.

1) Base VQ backbone. This part establishes the system foundation, using a pre-trained or jointly trained encoder to map the input image x into discrete codebook indices across specified feature branches $b \in \mathcal{B}$. 2) HyperVQ module. Serving as the analytical core, this module predicts content-adaptive probability distributions for the VQ indices by operating within the continuous embedding space, bridging the gap between continuous hyperpriors and discrete symbols. 3) Parallel arithmetic

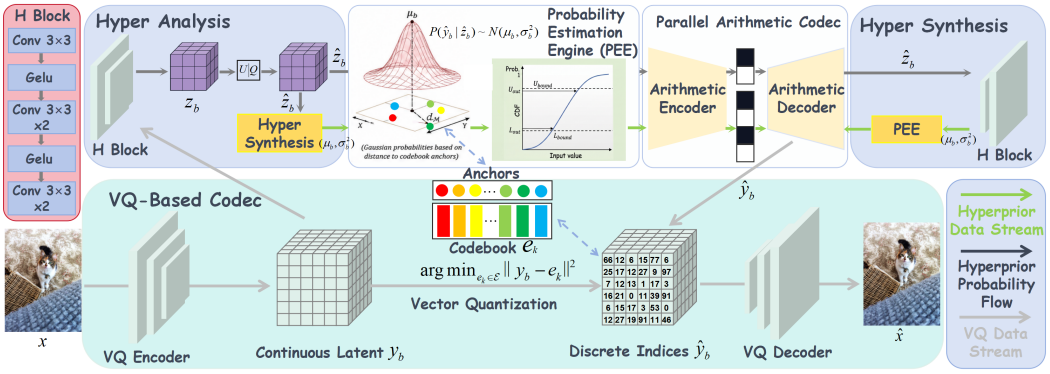

Figure 2: Overall framework of HyperVQ. The architecture consists of a base VQ codec, a generic HyperVQ module that predicts embedding-space distributions, and a parallel arithmetic codec. The Probability Estimation Engine (PEE) converts the predicted continuous density into discrete frequencies for efficient transmission.

codec. To ensure inference efficiency, we develop an accelerated engine that utilizes custom CUDA kernels to perform parallel density-to-probability mapping and arithmetic coding.

3.2 Principle of hyperprior for VQ indices

3.2.1 Problem Setting: Bridging continuous priors and discrete Indices

The success of hyperpriors in Scalar Quantization (SQ) [2, 3] arises from predicting parameters (μ, σ) for a continuous latent variable. However, VQ indices are discrete and unordered symbols, lacking the continuous spatial structure inherent in SQ latents. Consider an input image $x \in \mathbb{R}^{3\times H\times W}$. A generic VQ encoder outputs continuous feature maps $\{y_b\}_{b\in\mathcal{B}}$, where b denotes a specific branch, scale, or quantization stage. This multi-branch formulation enables HyperVQ to support diverse VQ architectures, ranging from multi-granularity (multi-scale) representations to multi-codebook (e.g., residual) quantization stages. The VQ step maps the D-dimensional feature vector $(y_b)_{ij}$ at each spatial location (i, j) to a discrete index $(\hat{y}_b)_{ij} \in \{1, \dots, K\}$. Constructing a content-adaptive probability model $P(\hat{y}_b \mid \hat{z}_b)$ for these indices is the critical challenge addressed by HyperVQ.

3.2.2 Core Idea: Gaussian prior in the embedding Space

Our key insight is that although the indices $(\hat{y}_b)_{ij}$ are discrete, their corresponding codebook embedding vectors $\mathbf{e}_k \in \mathbb{R}^D$ reside in a continuous D-dimensional embedding space. Consequently, rather than modeling the index k directly, we predict a conditional probability distribution within this continuous embedding space.

Specifically, for each branch $b \in \mathcal{B}$, a hyperprior branch processes the feature y_b to generate a continuous hyper-latent z_b. After quantization to $\hat{z}_b$, a HyperSynthesis network predicts the parameters of a multivariate Gaussian distribution, denoted as $\Theta_{b,ij}(\hat{z}_b) = (\boldsymbol{\mu}_{b,ij}, \boldsymbol{\Sigma}_{b,ij})$. Here, $\boldsymbol{\mu}_{b,ij} \in \mathbb{R}^D$ represents the expected feature vector (expected content) at that location, and $\boldsymbol{\Sigma}_{b,ij} \in \mathbb{R}^{D\times D}$ captures the prediction uncertainty.

3.2.3 From continuous Gaussian to discrete probabilities via Mahalanobis Distance

The next step is to bridge the continuous Gaussian parameters $(\boldsymbol{\mu}_{b,ij}, \boldsymbol{\Sigma}_{b,ij})$ to the probabilities $P((\hat{y}_b)_{ij} = k \mid \hat{z}_b)$ of the discrete indices k. We treat the codebook embeddings $\mathbf{e}_k$ as fixed "anchors" in the embedding space. As illustrated in Figure 3, we hypothesize that the probability of selecting index k at location (i, j) is related to the squared Mahalanobis Distance between its anchor $\mathbf{e}_k$ and the predicted Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_{b,ij}, \boldsymbol{\Sigma}_{b,ij})$, expressed as

$$d^2_{\mathcal{M}}(\mathbf{e}_k, \boldsymbol{\mu}_{b,ij}) = (\mathbf{e}_k - \boldsymbol{\mu}_{b,ij})^\top (\boldsymbol{\Sigma}_{b,ij})^{-1} (\mathbf{e}_k - \boldsymbol{\mu}_{b,ij}) \tag{1}$$

The categorical probability distribution is obtained via a Softmax function as follows

$$P((\hat{y}_b)_{ij} = k \mid \hat{z}_b) = \frac{\exp\left(-\frac{1}{2} d^2_{\mathcal{M}}(\mathbf{e}_k, \boldsymbol{\mu}_{b,ij})\right)}{\sum_{\ell=1}^{K} \exp\left(-\frac{1}{2} d^2_{\mathcal{M}}(\mathbf{e}_\ell, \boldsymbol{\mu}_{b,ij})\right)} \tag{2}$$

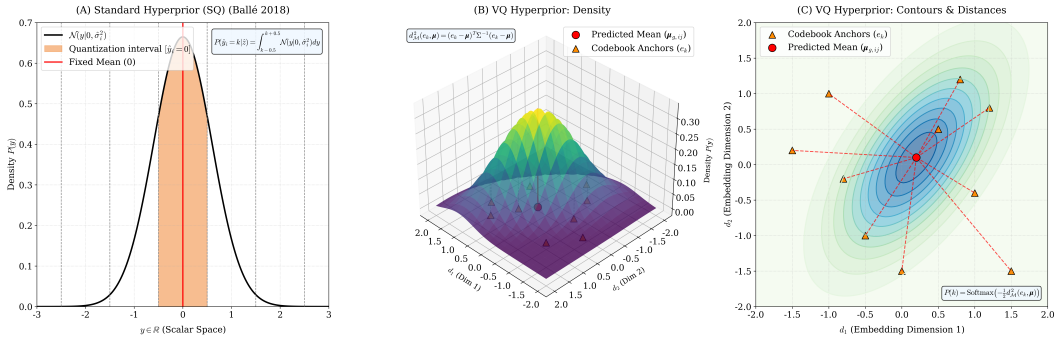

Figure 3: (a) Standard scalar quantization (SQ) hyperprior. (b)(c) Our proposed HyperVQ models a Gaussian probability cloud in the D-dim embedding space. The codebook entries act as anchors, and their probabilities are determined by their relative distances to the predicted distribution center.

For computational efficiency and stability, we assume the covariance matrix is diagonal and isotropic, i.e., $\mathbf{\Sigma}_{b,ij} = (\sigma_{b,ij})^2\mathbf{I}$. Under this simplification, the probability reduces to the scaled squared Euclidean distance, given by

$$P((\hat{y}_b)_{ij} = k \mid \hat{z}_b) = \frac{\exp\left(-\frac{\|\mathbf{e}_k - \boldsymbol{\mu}_{b,ij}\|^2}{2(\sigma_{b,ij})^2}\right)}{\sum_{\ell=1}^{K} \exp\left(-\frac{\|\mathbf{e}_\ell - \boldsymbol{\mu}_{b,ij}\|^2}{2(\sigma_{b,ij})^2}\right)} \tag{3}$$

Intuitively, while SQ hyperpriors predict a 1D Gaussian, our method predicts a high-dimensional Gaussian probability "cloud" in the embedding space. Codebook anchors closer to the cloud's center receive higher probability, while the spread $\sigma_{b,ij}$ reflects the model's uncertainty, effectively controlling the concentration of the distribution and the resulting bitrate.

3.2.4 Implications for efficient entropy coding and RD optimization

The core value of Equation (3) lies in providing a complete categorical probability distribution $P_{b,ij} = \{p_{b,ij,k}\}_{k=1}^{K}$ over the VQ index $(\hat{y}_b)_{ij}$ at (i, j). According to information theory, arithmetic coding a symbol using this distribution achieves an expected code length close to its Shannon entropy. The expected bitrate for encoding all VQ indices at branch b is formulated as

$$\begin{aligned} R_{\hat{y}_b} &= \mathbb{E}_x \left[\sum_{i,j} H(P^{\star}_{b,ij}, P_{b,ij}) \right] \\ &= \mathbb{E}_x \left[\sum_{i,j} \left(H(P^{\star}_{b,ij}) + D_{\mathrm{KL}}(P^{\star}_{b,ij} \parallel P_{b,ij}) \right) \right] \end{aligned} \tag{4}$$

where $P^{\star}_{b,ij}$ denotes the real but unknown true distribution of the index given image x. Crucially, by modeling this true discrete distribution (derived from the deterministic nearest-neighbor VQ assignment) as a Dirac delta function, we directly minimize this cross-entropy via backpropagation. This mathematical formulation elegantly establishes a differentiable gradient force field in the embedding space (fully derived in Appendix D). Specifically, the gradients actively pull the predicted mean $\boldsymbol{\mu}_{b,ij}$ toward the target anchor while repelling it from other anchors based on their assigned probabilities. This empowers the network to dynamically optimize the RD trade-off during training.

3.3 Module design and parallel implementation

This section maps the VQ hyperprior principles onto the network modules of HyperVQ, following the overall pipeline illustrated in Figure 2.

3.3.1 Base VQ backbone and HyperVQ branches

A generic VQ encoder distills hierarchical representations from the input image x into continuous latents $\{y_b\}_{b\in\mathcal{B}}$. The VQ module maps each continuous feature vector to its nearest entry in a shared

codebook $\mathcal{E}$, producing discrete index maps $\hat{y}_b$. For each branch $b \in \mathcal{B}$, an independent hyperprior branch consisting of a HyperAnalysis network H_a compresses y_b into a hyper-latent z_b. During training, z_b is quantized via additive uniform noise $\hat{z}_b = z_b + \mathcal{U}(-0.5, 0.5)$; during inference, we use rounding $\hat{z}_b = \text{round}(z_b)$. Subsequently, a HyperSynthesis network H_s generates the Gaussian parameters $(\boldsymbol{\mu}_b, \sigma_b) = H_s(\hat{z}_b)$.

3.3.2 Parallel probability estimation and entropy coding

A defining feature of our framework is its inference efficiency, which is rooted in the spatial independence of the index probabilities. Given the isotropic assumption, the categorical distribution for each spatial location (i, j) depends only on the locally predicted parameters $(\boldsymbol{\mu}_{b,ij}, \sigma_{b,ij})$. This property enables massive parallelization across the feature map.

First, the Probability Estimation Engine (PEE) computes the frequencies simultaneously across all locations by parallelizing the distance measurements to the codebook anchors. Second, to interact with the arithmetic coder, the PEE does not need to transmit the entire probability table sequentially. Instead, for a target index k at location (i, j), it computes and transmits the lower and upper bounds of the Cumulative Distribution Function (CDF) as follows

$$[F_{\text{lower}}, F_{\text{upper}}] = \left[\sum_{\ell=1}^{k-1} P((\hat{y}_b)_{ij} = \ell \mid \hat{z}_b), \sum_{\ell=1}^{k} P((\hat{y}_b)_{ij} = \ell \mid \hat{z}_b) \right] \tag{5}$$

By framing the problem this way, we implement a custom CUDA kernel to parallelize this CDF bound construction and the subsequent interval updates in the arithmetic codec, ensuring high-speed encoding and decoding without sequential bottlenecks (see Appendix B for the algorithmic pseudo-code and latency micro-benchmarks).

3.3.3 Instantiation on a Controllable Multi-Granularity Codec

We evaluate the robustness of HyperVQ by instantiating it on a controllable multi-granularity codec, Control-GIC [42], which presents a highly demanding latent environment (the complete architectural flowchart is illustrated in Appendix E). The primary challenge stems from its arbitrary rate control mechanism that dynamically routes features across hierarchical scales via spatial routing masks. Note that although these spatial routing masks are not entropy-coded, their bit overhead is properly calculated and included in the total reported bitrate.

Unlike traditional VQ codecs that process complete and spatially uniform feature maps, this routing mechanism yields sparse and fragmented latent representations. Each granularity level contains only partial information and exhibits non-contiguous spatial distributions that shift based on content complexity. Such irregularity poses a significant obstacle for entropy modeling because standard methods rely on dense and stable spatial structures to predict probabilities reliably.

HyperVQ addresses these challenges through a unified hyperprior modeling strategy across all granularities. Since the framework utilizes a shared codebook, our module can effectively parameterize the categorical distributions of these sparse indices by leveraging the aggregated multi-scale context. This enables the framework to maintain high efficiency despite the fragmented nature of the latent space. The Rate-Distortion performance for this challenging instantiation is detailed in the experimental section to demonstrate the robustness and flexibility of our method.

3.4 Objective and optimization

The entire framework is jointly optimized to minimize a unified Rate-Distortion (RD) objective. By introducing the differentiable cross-entropy rate term derived in §3.2, HyperVQ allows the network to explicitly balance reconstruction quality and bitrate consumption according to the following formulation

$$\mathcal{L} = \mathbb{E}[D(x, \hat{x})] + \sum_{b \in \mathcal{B}} (\lambda_{y,b} R_{\hat{y}_b} + \lambda_{z,b} R_{\hat{z}_b}) \tag{6}$$

$$D(x, \hat{x}) = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{percep}} + \lambda_{\text{GAN}} \mathcal{L}_{\text{GAN}} + \lambda_{\text{VQ}} \mathcal{L}_{\text{VQ}} \tag{7}$$

where $\mathcal{L}_{\text{rec}} = \|x - \hat{x}\|_2^2$ is the MSE reconstruction loss, and $\mathcal{L}_{\text{percep}}$ is the LPIPS perceptual loss. The adversarial loss is defined as

$$\mathcal{L}_{\text{GAN}} = -\log \mathcal{D}(\hat{x}) \tag{8}$$

Table 1: Bitrate savings of HyperVQ across different VQ-based models. Gain is computed relative to the theoretical bpp. For Residual Vector Quantization (RVQ) baselines, stages are cumulatively combined (+).

Dataset	Method	Codebook	Scale	Theoretical bpp	Huffman (Gain)	HyperVQ (Gain)
ImageNet-1K Val	VQGAN-LC (NeurIPS'24)	16384	16×	0.0547	0.0541 (1.1%)	0.0458 (16.3%)
		100K	16×	0.0641	0.0642 (-0.2%)	0.0564 (12.0%)
		100K	8×	0.2595	0.2516 (3.0%)	0.2072 (20.2%)
	VQGAN (CVPR'21)	1024	4×	0.6250	0.5844 (6.5%)	0.4703 (24.8%)
		2048	4×	0.6875	0.6473 (5.8%)	0.5119 (25.5%)
Kodak	Fine-tuned VQ (DCC'24)	1024	4×	0.6250	0.6040 (3.4%)	0.5621 (10.1%)
		1024	8×	0.1563	0.1522 (2.6%)	0.1335 (14.6%)
		1024	16×	0.0391	0.0389 (0.5%)	0.0337 (13.8%)
	Mo-VQGAN (NeurIPS'22)	1024	4×	0.6250	0.5947 (4.8%)	0.4984 (20.3%)
	Control-GIC (ICLR'25)	1024	4×	0.6250	0.5943 (4.9%)	0.4930 (21.1%)
		1024	8×	0.1562	0.1533 (1.9%)	0.1381 (11.6%)
		1024	16×	0.0391	0.0389 (0.5%)	0.0327 (16.4%)
	McQuic (CVPR'22)	2×8192	16×	0.1016	0.1005 (1.1%)	0.0766 (24.6%)
		$+2 \times 2048$	+32×	0.1230	0.1216 (1.1%)	0.0945 (23.2%)
		$+2 \times 512$	+64×	0.1277	0.1263 (1.1%)	0.0983 (23.0%)
				Average Gain	**2.54%**	**18.5%**

where $\mathcal{D}$ is a PatchGAN discriminator. To stabilize the embedding space, the VQ commitment loss is formulated across all branches $b \in \mathcal{B}$ by

$$\mathcal{L}_{\text{VQ}} = \sum_{b \in \mathcal{B}} \left(\|\text{sg}[y_b] - \mathbf{e}_{\hat{y}_b}\|_2^2 + \beta \|\text{sg}[\mathbf{e}_{\hat{y}_b}] - y_b\|_2^2 \right) \tag{9}$$

where $\text{sg}[\cdot]$ denotes the stop-gradient operator, $\mathbf{e}_{\hat{y}_b}$ represents the quantized features via codebook lookup.

The rate term consists of the cross-entropy for VQ indices ($R_{\hat{y}_b}$) and the bitrate of hyper-latents ($R_{\hat{z}_b}$). The bitrate weights $\lambda_{y,b}$ and $\lambda_{z,b}$ control the exact RD curve positioning. Notably, this formulation hands over the RD scheduling directly to the gradient descent process, enabling the VQ backbone and the hyperprior to co-adapt perfectly for optimal rate-distortion-perception (RDP) trade-offs. Implementation details and the progressive joint-training protocol are provided in Appendix C.

4 Experiments

4.1 Experimental protocol

Evaluated Codecs. To evaluate generalizability, we test HyperVQ across diverse representative architectures. These include single-scale codecs (VQGAN [6], Fine-tuned VQ [46], Mo-VQGAN [40]), large-codebook models (VQGAN-LC [41]), controllable multi-granularity codecs (Control-GIC [42]), and multi-codebook RVQ frameworks (McQuic [43]).

Datasets and Metrics. Model evaluation utilizes ImageNet-1K Val [47] and the standard Kodak [48] dataset for comprehensive benchmarking. Additional high-resolution results on DIV2K [49] and the CLIC 2020 professional set [50] are provided in Section 4.4. To comprehensively assess both objective fidelity and subjective realism, we measure bitstream size via bits per pixel (bpp) and employ a robust suite of metrics including LPIPS and DISTS for reference-based perceptual quality, along with FID, NIQE, and IS for no-reference generative realism, and PSNR for pixel-level distortion.

Implementation Details. Training configurations including optimizers and schedules follow the original papers for each respective backbone. Detailed integration protocols for equipping them with HyperVQ are provided in Appendix C.

4.2 Cross-Model generalization across VQ-Based codecs

To verify universal applicability, we integrate HyperVQ into established VQ codecs, comparing the resulting coding efficiency against theoretical bitrate and Huffman coding.

As demonstrated in Table 1, HyperVQ consistently yields significant bitrate savings across diverse architectures, codebook sizes, and scales. HyperVQ effectively mines latent redundancies by leveraging high-dimensional context within the embedding space (corroborated in Appendix H). This

Table 2: Comprehensive Comparison of Efficiency and RD Performance. *Params* indicates the number of network parameters in Millions. *Time* reflects the average Encoding/Decoding latency in Seconds (s). *BD-Rate Savings (%)* are calculated relative to M&S [2]. **Bold** indicates the best result, underline indicates the second best, and N/A indicates data not available or applicable.

Method	Efficiency			BD-Rate Savings (%) of Ours vs. Method				
	Params (M)	Enc. (s)	Dec. (s)	LPIPS ↓	DISTS ↓	FID ↓	NIQE ↓	IS ↓
Traditional Codecs								
BPG [51]	N/A	0.196	0.342	4.51	-52.02	-12.60	23.60	N/A
VVC [4]	N/A	46.902	0.153	-19.54	-56.07	-13.92	19.98	N/A
Learned Generative								
HiFiC [5]	184.27	0.537	1.453	-72.90	-65.09	-67.18	-75.54	-66.46
CDC [17]	68.66	0.114	1.307	-56.74	-71.50	-65.06	-40.12	-20.17
SCR [52]	11.88	**0.029**	**0.032**	37.26	-47.84	-6.76	-60.51	-9.71
CTC [53]	399.00	0.526	0.754	108.83	-49.61	-20.11	-82.06	-32.73
Control-GIC [42]	130.69	0.097	0.115	-60.42	-54.12	-51.22	-85.56	-43.12
HVQ-CGIC (Ours)	132.30	0.091	0.101	**-87.30**	**-71.68**	**-77.20**	**-92.22**	**-67.63**

content-adaptive modeling particularly benefits large-codebook and complex hierarchical paradigms, validating HyperVQ as a plug-and-play module for the entire VQ family.

4.3 Detailed results of HVQ-CGIC

We further analyze our framework's most challenging instantiation, namely the controllable multi-granularity setting (HVQ-CGIC). Unlike standard codecs, this setting dynamically routes features to generate multi-scale and partial feature maps. Modeling the entropy of these sparse indices rigorously tests the hyperprior's robustness.

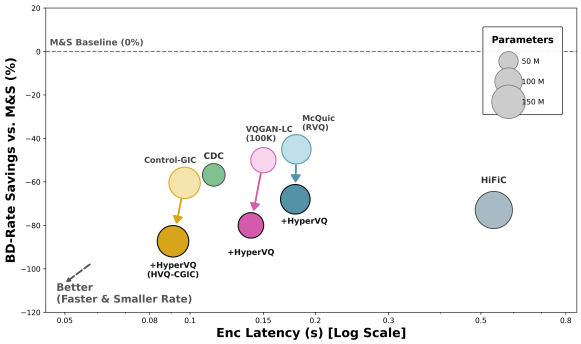

Figure 4: Visualizing the efficiency-performance trade-off. Bubble size represents parameter count. Our HyperVQ significantly pushes VQ-based codecs toward the "Better" quadrant (lower-left), achieving superior RD performance with negligible latency overhead.

Baseline Methods. We compare our method against traditional codecs (BPG [51], VVC [4]), classical neural models (M&S [2]), variable-rate/progressive frameworks (SCR [52], CTC [53]), and state-of-the-art generative approaches including HiFiC [5] and CDC [17]. Additionally, expanded comparisons with recent non-VQ continuous-latent generative paradigms, such as MS-ILLM [54], CRDR [55], and EGIC [56], are provided in Appendix A.1.

Quantitative and Qualitative Performance. Figure 5 shows HVQ-CGIC significantly outperforms baselines across perceptual metrics (LPIPS, NIQE, DISTS, FID, IS). Visual comparisons in Figure 6 confirm it preserves sharper structures and more realistic textures at highly constrained bitrates (see Appendix I for extended comparisons).

Efficiency and BD-Rate Evaluation. Table 2 details the trade-off between performance gains and computational cost. HVQ-CGIC achieves the highest BD-Rate savings against the M&S anchor while maintaining highly competitive inference speeds.

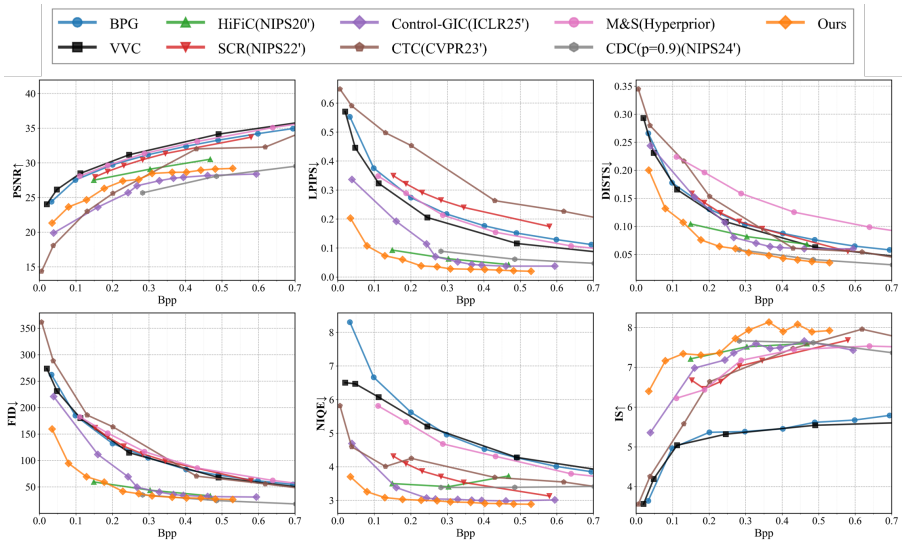

Figure 5: Rate-distortion and perceptual comparisons on Kodak. HVQ-CGIC pushes the Pareto frontier across multiple metrics.

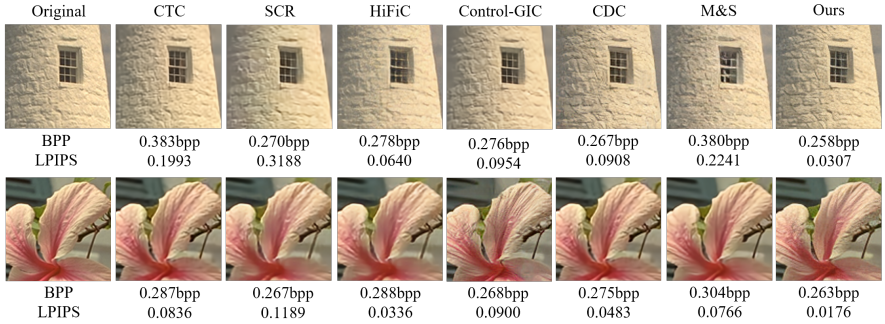

Figure 6: Qualitative comparisons of reconstruction fidelity between HVQ-CGIC and baseline methods at extremely low bitrates.

Visualizing Coding Efficiency. As visualized in Figure 4, in VQ-based frameworks, encoding speed is inherently constrained by symbol count; larger codebooks or multi-scale architectures typically lead to higher computational overhead during entropy estimation. This bottleneck is exacerbated when high-order context models are employed (detailed in Appendix F).

By shifting probability modeling into the continuous embedding space, our HyperVQ module enables the parallel output of probability tables and simultaneous arithmetic coding. Consequently, even when accounting for the entropy coding stage, HyperVQ-equipped models not only achieve a better Rate-Distortion (RD) frontier but also exhibit faster end-to-end inference compared to original backbones using static or sequential coding.

4.4 Performance on High-Resolution Benchmarks

We evaluate the generalization capability of HVQ-CGIC on high-resolution benchmarks: the CLIC 2020 Professional Set [50] and DIV2K [49]. The Rate-Distortion curves are presented in Figure 7.

Performance on CLIC 2020. As shown in the top row of Figure 7, HVQ-CGIC maintains significant advantages in perceptual metrics (LPIPS, FID, KID) on the professional photography dataset. This confirms that our entropy-based routing strategy generalizes well to complex lighting and composition without requiring dataset-specific retraining. **Performance on DIV2K.** On the 2K-resolution DIV2K dataset (Figure 7, bottom row), HVQ-CGIC consistently outperforms the baselines across the

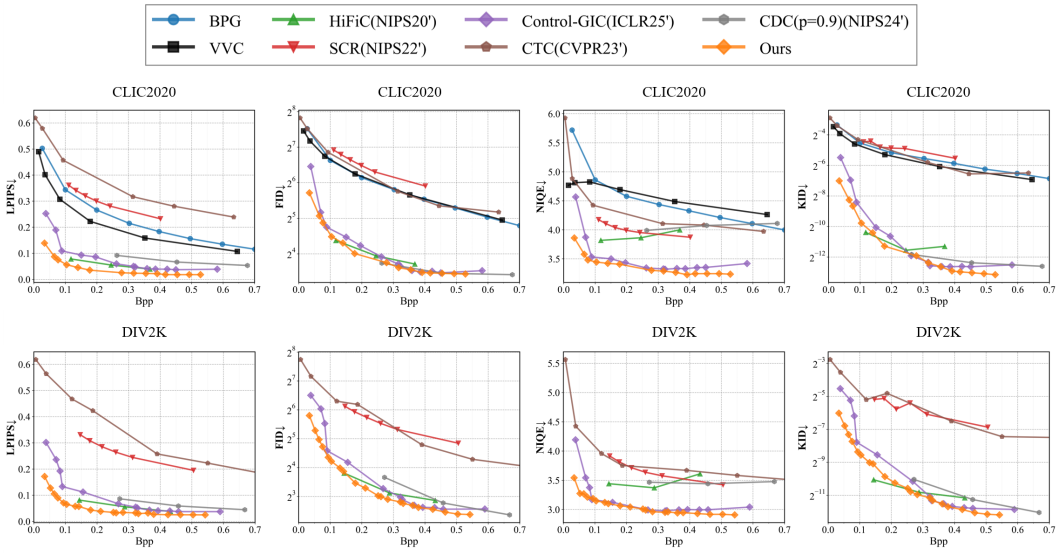

Figure 7: **High-Resolution Generalization.** Comparison of methods on the CLIC 2020 professional set (top) and DIV2K (bottom).

measured metrics. This demonstrates the robustness of our hyperprior modeling in handling rich high-frequency details at scaling resolutions.

4.5 Rate decomposition and covariance

Content-adaptive coding must transmit the hyperprior as well as the indices. Table 3 decomposes this cost for Control-GIC on Kodak ($K = 1024$): total bitrate remains below Huffman coding at all three scales (Table 1), with hyper-latents accounting for approximately 3%.

The covariance ablation tests the value of dimension-specific variances. Diagonal covariance saves 0.3–1.1% bitrate at 3.5–4.8× encoding time. Thus, a compact spatially adaptive prior provides strong coding gains in this setting.

Table 3: Bitrate decomposition and covariance ablation. Rates are in bpp; encoding times are in ms.

	Isotropic rate			Diagonal rate	Encoding time	
Scale	R_y	R_z	$R_y + R_z$	$R_y + R_z$	Isotropic	Diagonal
4×	0.4789	0.0141	0.4930	0.4877	2.71	12.87
8×	0.1346	0.0035	0.1381	0.1376	2.06	8.63
16×	0.0315	0.0012	0.0327	0.0326	1.03	3.58

5 Limitation, discussion and conclusion

HyperVQ relies on isotropic covariance for efficiency, theoretically limiting cross-dimensional correlation modeling. Its shift to continuous-space probability enables fully differentiable end-to-end RD optimization. This plug-and-play framework provides a foundational entropy-coding component for the VQ family, paving the way for precise and dynamic Rate-Distortion-Perception control in next-generation generative image compression.

Supplementary material. Appendices A–I cover extended comparisons, CUDA/CDF construction, integration and training, theoretical derivations, HVQ-CGIC architecture, context-model comparisons, K-way classification, spatial bit allocation, and visual reconstructions, respectively.

Acknowledgments and Disclosures

This work is supported by the National Natural Science Foundation of China (No.62371356 and No.62405238), in part by China Postdoctoral Science Foundation (No.2024M752512).

The authors declare no competing interests.

References

[1] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," *arXiv preprint arXiv:1611.01704*, 2016.

[2] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *arXiv preprint arXiv:1802.01436*, 2018.

[3] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," *Advances in neural information processing systems*, vol. 31, 2018.

[4] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, "Overview of the versatile video coding (vvc) standard and its applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, 2021.

[5] F. Mentzer, G. D. Toderici, M. Tschannen, and E. Agustsson, "High-fidelity generative image compression," *Advances in neural information processing systems*, vol. 33, pp. 11 913–11 924, 2020.

[6] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 873–12 883.

[7] C. E. Shannon *et al.*, "Coding theorems for a discrete source with a fidelity criterion," *IRE Nat. Conv. Rec*, vol. 4, no. 142-163, p. 1, 1959.

[8] O. Rippel and L. Bourdev, "Real-time adaptive image compression," in *International conference on machine learning*. PMLR, 2017, pp. 2922–2930.

[9] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. V. Gool, "Generative adversarial networks for extreme learned image compression," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 221–231.

[10] N. Johnston, D. Vincent, D. Minnen, M. Covell, S. Singh, T. Chinen, S. J. Hwang, J. Shor, and G. Toderici, "Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4385–4393.

[11] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang, "Learning convolutional networks for content-weighted image compression," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3214–3223.

[12] C. Shin, H. Lee, H. Son, S. Lee, D. Lee, and S. Lee, "Expanded adaptive scaling normalization for end to end image compression," in *European Conference on Computer Vision*. Springer, 2022, pp. 390–405.

[13] G. Pan, G. Lu, Z. Hu, and D. Xu, "Content adaptive latents and decoder for neural image compression," in *European Conference on Computer Vision*. Springer, 2022, pp. 556–573.

[14] M. Li, S. Gao, Y. Feng, Y. Shi, and J. Wang, "Content-oriented learned image compression," in *European Conference on Computer Vision*. Springer, 2022, pp. 632–647.

[15] Y. Zhu, Y. Yang, and T. Cohen, "Transformer-based transform coding," in *International conference on learning representations*, 2022.

[16] J. Liu, H. Sun, and J. Katto, "Learned image compression with mixed transformer-cnn architectures," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 14 388–14 397.

[17] R. Yang and S. Mandt, "Lossy image compression with conditional diffusion models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 64 971–64 995, 2023.

[18] M. Careil, M. J. Muckley, J. Verbeek, and S. Lathuilière, "Towards image compression with perfect realism at ultra-low bitrates," in *The Twelfth International Conference on Learning Representations*, 2023.

[19] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool, "Conditional probability models for deep image compression," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4394–4402.

[20] J. Lee, S. Cho, and S.-K. Beack, "Context-adaptive entropy model for end-to-end optimized image compression," *arXiv preprint arXiv:1809.10452*, 2018.

[21] Y. Choi, M. El-Khamy, and J. Lee, "Variable rate deep image compression with a conditional autoencoder," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 3146–3154.

[22] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized gaussian mixture likelihoods and attention modules," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 7939–7948.

[23] C. Lin, J. Yao, F. Chen, and L. Wang, "A spatial rnn codec for end-to-end image compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 269–13 277.

[24] D. Minnen and S. Singh, "Channel-wise autoregressive entropy models for learned image compression," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 3339–3343.

[25] X. Zhang and X. Wu, "Attention-guided image compression by deep reconstruction of compressive sensed saliency skeleton," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 354–13 364.

[26] D. He, Y. Zheng, B. Sun, Y. Wang, and H. Qin, "Checkerboard context model for efficient learned image compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 771–14 780.

[27] F. Yang, L. Herranz, Y. Cheng, and M. G. Mozerov, "Slimmable compressive autoencoders for practical neural image compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4998–5007.

[28] G. Gao, P. You, R. Pan, S. Han, Y. Zhang, Y. Dai, and H. Lee, "Neural image compression via attentional multi-scale back projection and frequency decomposition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 677–14 686.

[29] J.-H. Kim, B. Heo, and J.-S. Lee, "Joint global and local hierarchical priors for learned image compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5992–6001.

[30] X. Zhang and X. Wu, "Multi-modality deep restoration of extremely compressed face videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 2024–2037, 2022.

[31] D. He, Z. Yang, W. Peng, R. Ma, H. Qin, and Y. Wang, "Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 5718–5727.

[32] W. Jiang, J. Yang, Y. Zhai, P. Ning, F. Gao, and R. Wang, "Mlic: Multi-reference entropy model for learned image compression," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 7618–7627.

[33] G.-H. Wang, J. Li, B. Li, and Y. Lu, "Evc: Towards real-time neural image compression with mask decay," *arXiv preprint arXiv:2302.05071*, 2023.

[34] Z. Jia, J. Li, B. Li, H. Li, and Y. Lu, "Generative latent coding for ultra-low bitrate image compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 26 088–26 098.

[35] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. V. Gool, "Soft-to-hard vector quantization for end-to-end learning compressible representations," *Advances in neural information processing systems*, vol. 30, 2017.

[36] X. Zhang and X. Wu, "Lvqac: Lattice vector quantization coupled with spatially adaptive companding for efficient learned image compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 10 239–10 248.

[37] R. Feng, Z. Guo, W. Li, and Z. Chen, "Nvtc: Nonlinear vector transform coding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6101–6110.

[38] E. Lei, H. Hassani, and S. S. Bidokhti, "Approaching rate-distortion limits in neural compression with lattice transform coding," *arXiv preprint arXiv:2403.07320*, 2024.

[39] X. Zhang and X. Wu, "Learning optimal lattice vector quantizers for end-to-end neural image compression," *Advances in Neural Information Processing Systems*, vol. 37, pp. 106 497–106 518, 2024.

[40] C. Zheng, T.-L. Vuong, J. Cai, and D. Phung, "Movq: Modulating quantized vectors for high-fidelity image generation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 23 412–23 425, 2022.

[41] L. Zhu, F. Wei, Y. Lu, and D. Chen, "Scaling the codebook size of vq-gan to 100,000 with a utilization rate of 99%," *Advances in Neural Information Processing Systems*, vol. 37, pp. 12 612–12 635, 2024.

[42] A. Li, F. Li, Y. Liu, R. Cong, Y. Zhao, and H. Bai, "Once-for-all: Controllable generative image compression with dynamic granularity adaption," *arXiv preprint arXiv:2406.00758*, 2024.

[43] X. Zhu, J. Song, L. Gao, F. Zheng, and H. T. Shen, "Unified multivariate gaussian mixture for efficient neural image compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 612–17 621.

[44] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-constrained vector quantization," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 1, pp. 31–42, 1989.

[45] S. Jiang, W. Long, M. Han, Z. Chen, C. Zhu, and S. Gu, "Differentiable vector quantization for rate-distortion optimization of generative image compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[46] Q. Mao, T. Yang, Y. Zhang, Z. Wang, M. Wang, S. Wang, L. Jin, and S. Ma, "Extreme image compression using fine-tuned vqgans," in *2024 Data Compression Conference (DCC)*. IEEE, 2024, pp. 203–212.

[47] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[48] E. K. Company, "Kodak lossless true color image suite," http://r0k.us/graphics/kodak/, 1993, original images, commonly used as a test set for image compression.

[49] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 126–135.

[50] G. Toderici, L. Theis, N. Johnston, E. Agustsson, F. Mentzer, J. Ballé, W. Shi, and R. Timofte, "Clic 2020: Challenge on learned image compression," *Retrieved March*, vol. 29, p. 2021, 2020.

[51] F. Bellard, "BPG image format," https://bellard.org/bpg/, 2018.

[52] J. Lee, S. Jeong, and M. Kim, "Selective compression learning of latent representations for variable-rate image compression," *Advances in Neural Information Processing Systems*, vol. 35, pp. 13 146–13 157, 2022.

[53] S. Jeon, K. P. Choi, Y. Park, and C.-S. Kim, "Context-based trit-plane coding for progressive image compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14 348–14 357.

[54] M. J. Muckley, A. El-Nouby, K. Ullrich, H. Jégou, and J. Verbeek, "Improving statistical fidelity for neural image compression with implicit local likelihood models," in *International Conference on Machine Learning*. PMLR, 2023, pp. 25 426–25 443.

[55] S. Iwai, T. Miyazaki, and S. Omachi, "Controlling rate, distortion, and realism: Towards a single comprehensive neural image compression model," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 2900–2909.

[56] N. Körber, E. Kromer, A. Siebert, S. Hauke, D. Mueller-Gritschneder, and B. Schuller, "Egic: enhanced low-bit-rate generative image compression guided by semantic segmentation," in *European Conference on Computer Vision*. Springer, 2024, pp. 202–220.

Supplementary Material

This supplementary material serves to substantiate the findings presented in the main paper, offering deep theoretical analysis, detailed architecture design, granular ablation data, and extended visual comparisons. The content is organized as follows:

- **Appendix A: Extended Quantitative RD Comparisons.** Provides extended Rate-Distortion curves, including comparisons against non-VQ generative methods and evaluations on high-resolution benchmarks (CLIC 2020, DIV2K).
- **Appendix B: CUDA-Accelerated Entropy Preparation (PEE).** Provides the mathematical definition and pseudo-code for the GPU-based Cumulative Distribution Function (CDF) construction, along with our bounded tail-floor approximation for ultra-large codebooks.
- **Appendix C: Integration Protocol.** Details a rapid adaptation method to equip any VQ backbone with HyperVQ, alongside an ablation study justifying our joint-training recommendation.
- **Appendix D: Theoretical Derivation.** Elucidates the complete mathematical derivation from the rate-distortion objective to the differentiable loss function, including a formal justification for the centroid approximation.
- **Appendix E: Detailed Architecture of HVQ-CGIC.** Presents the complete flowchart of the instantiation on the controllable multi-granularity codec.
- **Appendix F: Comparison with Traditional High-Order Context Modeling.** Evaluates HyperVQ against manually designed spatial context arithmetic coders in terms of compression bounds and inference latency.
- **Appendix G: Analysis: Why Not a Direct K-Way Classifier?** Details why our continuous embedding-space Gaussian modeling is vastly superior to predicting probabilities via a K-way classifier.
- **Appendix H: Entropy Coding and Spatial Bit Usage Analysis.** Visually demonstrates how HVQ-CGIC exhausts coding redundancy in sparse latent spaces.
- **Appendix I: Extended Qualitative Visual Comparisons.** Offers extensive high-resolution visual comparisons across multiple datasets to demonstrate the superior perceptual fidelity of our framework.

A Extended Quantitative RD Comparisons

In this section, we present additional quantitative comparisons to further validate the superiority of the proposed framework. First, we benchmark HVQ-CGIC against recent state-of-the-art *non-VQ* continuous-latent generative codecs. Second, we summarize the method's high-resolution generalization capability on the CLIC 2020 and DIV2K datasets.

A.1 Comparison with Non-VQ Generative Codecs

To provide a comprehensive evaluation across different generative paradigms, we compare HVQ-CGIC against MS-ILLM[54], CRDR[55], and EGIC[56]. As shown in Figure S8, HVQ-CGIC establishes a highly competitive Pareto frontier against these advanced continuous-latent models. This strongly validates the potential of VQ-based compression when equipped with powerful, content-adaptive hyperprior entropy modeling.

A.2 Performance on High-Resolution Benchmarks

See Section 4.4.

A.3 Comparison with Differentiable VQ

Following Jiang et al. [45], we retain their $16\times$ backbone and replace $32\times/64\times$ branches with HyperVQ. Table S4 reports Kodak results; latency includes forward inference, entropy modeling, and coding.

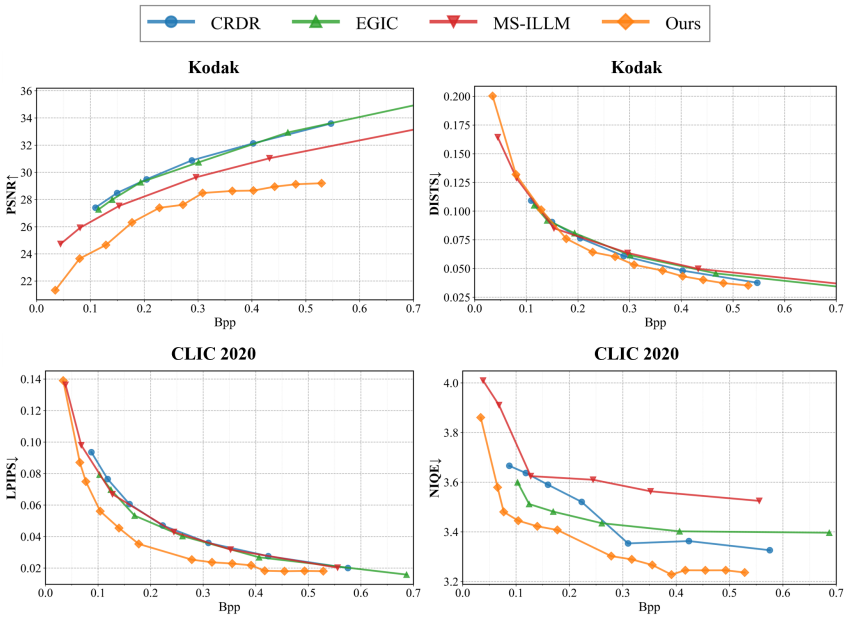

Figure S8: **Comparison with non-VQ generative methods** on Kodak and CLIC 2020 test sets.

Table S4: Comparison with differentiable VQ.

Method	bpp	PSNR	LPIPS	Enc. (ms)	Dec. (ms)
Jiang et al. [45]	0.0352	21.8995	0.1603	429.1	424.1
HyperVQ	0.0336	23.2361	0.1417	137.5	164.4

B CUDA-Accelerated Entropy Preparation

To reduce the entropy coding latency in high-resolution image compression, we migrate the Cumulative Distribution Function (CDF) preparation process from the CPU to the GPU by designing a custom CUDA-accelerated entropy preparation operator. This operator does not replace the final arithmetic coder; rather, it efficiently transforms the network-output probability parameters into the integer CDFs, lower bounds, and frequencies directly consumable by the arithmetic coder.

B.1 Categorical CDF Preparation for Index Stream

Let $\mathbf{l}_n \in \mathbb{R}^K$ denote the logits at symbol position n, where K is the alphabet size. We first convert logits into probabilities:

$$\pi_{n,k} = \frac{\exp(l_{n,k})}{\sum_{j=0}^{K-1} \exp(l_{n,j})}, \quad k = 0, \ldots, K-1. \tag{S1}$$

Given an entropy precision of p bits, the integer scale is $S = 2^p$. To guarantee a valid integer CDF with a minimum increment of one for each symbol, we use monotonic quantization:

$$\hat{C}_{n,0} = 0, \quad \hat{C}_{n,K} = S, \tag{S2}$$

and for $k = 1, \ldots, K-1$,

$$\hat{C}_{n,k} = \min\left(\left\lfloor (S-K) \sum_{j=0}^{k-1} \pi_{n,j} \right\rfloor + k,\ S - (K-k) \right). \tag{S3}$$

For the target symbol index $i_n \in \{0, \ldots, K-1\}$, the arithmetic encoder consumes the lower bound $L_n^{(idx)}$ and frequency $F_n^{(idx)}$:

$$L_n^{(idx)} = \hat{C}_{n,i_n}, \quad F_n^{(idx)} = \hat{C}_{n,i_n+1} - \hat{C}_{n,i_n}. \tag{S4}$$

Algorithm 1 CUDA-Accelerated Entropy Preparation

Require: Categorical logits $\mathbf{l} \in \mathbb{R}^{N \times K}$, target indices $\mathbf{i} \in \{0, \dots, K-1\}^N$; hyper-latent symbols $\mathbf{z}$, channel-wise scales $\boldsymbol{\sigma}$, precision p.

Ensure: Integer lower bounds $\mathbf{L}^{(idx)}, \mathbf{L}^{(hyp)}$, frequencies $\mathbf{F}^{(idx)}, \mathbf{F}^{(hyp)}$, and CDFs $\hat{\mathbf{C}}, \tilde{\mathbf{C}}$.

Set integer scale $S \leftarrow 2^p$

// Stage 1: Categorical CDF Preparation (Index Stream)

for each spatial position $n \in \{1, \dots, N\}$ **in parallel do**

  $\pi_{n,k} \leftarrow \frac{\exp(l_{n,k})}{\sum_j \exp(l_{n,j})}$ for $k = 0, \dots, K-1$ ▷ Softmax

  Compute integer CDF $\hat{C}_{n,:}$ via monotonic quantization to scale S

  $L_n^{(idx)} \leftarrow \hat{C}_{n,i_n}$

  $F_n^{(idx)} \leftarrow \hat{C}_{n,i_n+1} - \hat{C}_{n,i_n}$

end for

// Stage 2: Gaussian CDF Preparation (Hyper-latent Stream)

for each channel c **in parallel do**

  $q_c(v) \leftarrow \exp(-v^2/2\sigma_c^2)$ for $v \in [v_{\min}, v_{\max}] \cap \mathbb{Z}$

  $p_c(v) \leftarrow q_c(v) / \sum_u q_c(u)$ ▷ Normalize PMF

  Quantize $p_c(v)$ into integer CDF $\tilde{C}_{c,:}$ using monotonic rule

end for

for each hyper-latent symbol $z_{c,m}$ **in parallel do**

  $r_{c,m} \leftarrow z_{c,m} - v_{\min}$ ▷ Calculate table offset

  $L_{c,m}^{(hyp)} \leftarrow \tilde{C}_{c,r_{c,m}}$

  $F_{c,m}^{(hyp)} \leftarrow \tilde{C}_{c,r_{c,m}+1} - \tilde{C}_{c,r_{c,m}}$

end for

return $\mathbf{L}^{(idx)}, \mathbf{F}^{(idx)}, \hat{\mathbf{C}}$ and $\mathbf{L}^{(hyp)}, \mathbf{F}^{(hyp)}, \tilde{\mathbf{C}}$

B.2 Gaussian CDF Preparation for Hyper-latents

For the hyper-latent stream, we discretize the Gaussian distribution on the support $v \in [v_{\min}, v_{\max}] \cap \mathbb{Z}$, and compute the unnormalized mass by $q_c(v) = \exp\left(-\frac{v^2}{2\sigma_c^2}\right)$. After normalization:

$$p_c(v) = \frac{q_c(v)}{\sum_{u=v_{\min}}^{v_{\max}} q_c(u)}. \tag{S5}$$

The corresponding integer CDF $\tilde{C}_{c,:}$ is constructed using the same monotonic quantization rule. For symbol $z_{c,m}$, let $r_{c,m} = z_{c,m} - v_{\min}$. The arithmetic encoder consumes the corresponding lower bound $L_{c,m}^{(hyp)}$ and frequency $F_{c,m}^{(hyp)}$:

$$L_{c,m}^{(hyp)} = \tilde{C}_{c,r_{c,m}}, \quad F_{c,m}^{(hyp)} = \tilde{C}_{c,r_{c,m}+1} - \tilde{C}_{c,r_{c,m}}. \tag{S6}$$

B.3 Bounded Tail-Floor for Ultra-Large Codebooks

For standard codebooks ($K \leq 8192$), evaluating the exact Euclidean distance to all anchors is highly parallelizable and practically instantaneous. However, for ultra-large codebooks (e.g., VQGAN-LC with $K = 100,000$), calculating a full exact Softmax introduces unnecessary memory and latency overheads.

To resolve this, our PEE utilizes a **Bounded Tail-Floor Approximation**. Since our logits are strictly monotonically decreasing with geometric distance (Equation 3), anchors positioned far from the predicted mean $\boldsymbol{\mu}_{b,ij}$ contribute almost zero mass to the true categorical distribution. Therefore, during the CDF construction phase in the CUDA kernel, we evaluate exact distances only for a set of candidate anchors within a threshold radius. For all remote, "pruned" anchors, we bypass the exponentiation step and directly assign them the minimum valid integer frequency ($\hat{F}_k = 1$) in the arithmetic integer scale S.

This strategy ensures three critical advantages: 1) It maintains a strictly legal and monotonic CDF, guaranteeing 100% mathematical decodability without any bitstream format changes; 2) The decoder

replicates the exact same deterministic tail-floor rule based on the decoded $\boldsymbol{\mu}$; 3) It slashes the large-codebook entropy preparation latency to a mere $\sim$ 10ms penalty. We emphasize that the compression performance reported in Table 1 of the main paper is obtained using this Bounded Tail-Floor Approximation within our PEE. As shown in Table S5, the average gap between this efficient bounded estimation and the exact dense Softmax is less than 0.2 percentage points in bitrate gain, justifying the use of the bounded version as our default practical configuration.

Table S5: Ablation Study of Bounded Tail-Floor Approximation on VQGAN-LC (ImageNet-1K Val). The results show that the average bitrate gap between the proposed approximation and the exact dense Softmax is negligible (within 0.2 percentage points of gain), justifying our efficient tail-floor design.

Method	**Codebook (K)**	**Scale**	**Theo. bpp**	**HyperVQ (Approx.)**	**HyperVQ (Exact Softmax)**
VQGAN-LC	16384	16×	0.0547	0.0458 (16.3%)	0.0458 (16.3%)
VQGAN-LC	100K	16×	0.0641	0.0564 (12.0%)	0.0562 (12.5%)
VQGAN-LC	100K	8×	0.2595	0.2072 (20.2%)	0.2071 (20.2%)

B.4 Acceleration Analysis and Timing Disambiguation

It is important to clarify the relationship between the system-level timings reported in the subsequent tables (e.g., Table S6) and the kernel-level efficiency of our proposed module.

The ultra-low encoding and decoding latencies (e.g., 2.71 ms and 4.17 ms) presented in the context ablation study represent the per-image throughput measured under a **batched parallel inference** setting on the Kodak dataset, which fully saturates the GPU resources and demonstrates the theoretical peak pipeline speed.

In contrast, the micro-benchmark analysis below isolates the computational acceleration of the custom CDF preparation kernel itself on a **single, unbatched** high-resolution (1920×1080) image. In this kernel-level test, we measure exclusively the CDF preparation cost, deliberately excluding neural-network forward passes, process creation, and file I/O overheads. By offloading this high-dimensional density-to-probability mapping to the GPU, our custom CUDA implementation achieves a massive **39.8× speedup** over the baseline sequential CPU execution. In terms of processing throughput, this corresponds to a dramatic increase from 4.60 MPix/s to 183.0 MPix/s.

Furthermore, at the system-prototype level, replacing the original Huffman coding bottleneck with our GPU-accelerated CDF preparation improves the entire end-to-end entropy pipeline by $1.93\times$ for encoding and $2.13\times$ for decoding over a serial CPU pipeline. This validates that our complex content-adaptive entropy modeling introduces negligible latency overhead while unlocking substantial Rate-Distortion gains.

C Integration Protocol: How to Apply HyperVQ to Existing VQ Codecs

Because HyperVQ operates exclusively in the continuous embedding space, it can be rapidly deployed to enhance the entropy modeling of almost any existing VQ-based architecture. In this section, we provide a practical, three-stage progressive integration recipe.

It is important to note that while **Stage B** (Rapid Adaptation) provides a highly effective and quick method to slash the bitrate of a pre-trained VQ model, we strongly recommend completing **Stage C** (Joint End-to-End Training) to achieve the optimal Rate-Distortion balance. We use the HVQ-CGIC instantiation as a case study to ablate and validate this strategy.

Stage A: Baseline VQ Training (or Loading Pre-trained Weights).
We first obtain a high-quality VQ backbone consisting of an encoder, decoder, vector quantizer, and discriminator. Only the distortion and commitment terms are optimized: $\mathcal{L}_{\text{A}} = D(x, \hat{x})$. At this stage, the hyperprior networks are absent, establishing a strong perceptually-aligned reconstruction baseline. (For existing models, one can simply load the pre-trained weights).

Stage B: Rapid Hyperprior Adaptation (Quick Application).
We freeze all the parameters of the base VQ codec. The HyperVQ networks (Analysis and Synthesis transforms) are randomly initialized and trained *exclusively* by minimizing the differentiable rate term: $\mathcal{L}_{\text{B}} = R$. This forces the hyperprior to rapidly learn efficient, content-adaptive probability predictions for the fixed VQ indices produced by the frozen encoder.

Stage C: Joint RD Fine-tuning (Recommended).
Finally, we unfreeze the entire framework and fine-tune it end-to-end with a reduced learning rate using the full Rate-Distortion objective: $\mathcal{L}_{\text{C}} = D(x, \hat{x}) + R$. This allows the VQ encoder's feature representations and the hyperprior's probability models to co-adapt, actively restructuring the latent space for compressibility.

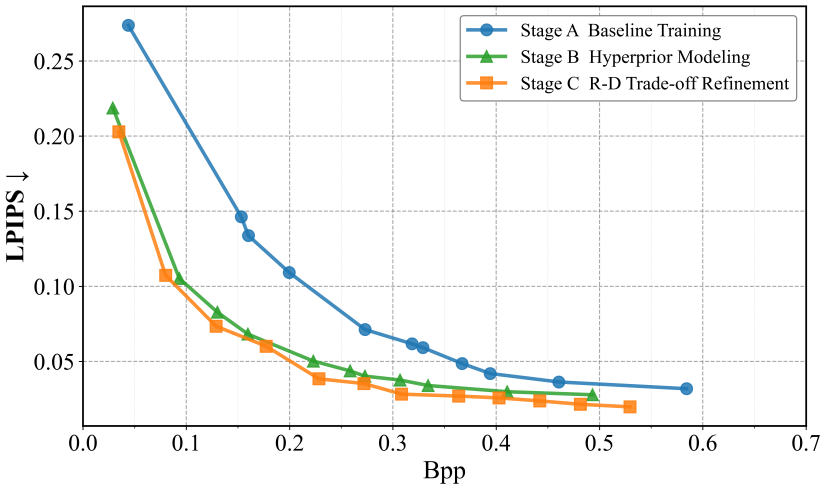

Figure S9: **Ablation study of the three training stages on Kodak.** Stage B (Rapid Adaptation) significantly drops the bitrate with minimal quality compromise, while Stage C (Joint Fine-tuning) pushes the Pareto frontier further to the lower-left, optimizing the ultimate RD trade-off.

Ablation Analysis. The effectiveness of this protocol is validated in Figure S9. The transition from Stage A to Stage B proves that integrating HyperVQ directly yields massive bitrate savings without altering the pre-trained latent representations. This makes the "Quick Application" highly valuable for rigid, legacy VQ systems. However, Stage C further shifts the RD frontier downwards and to the left. By allowing the network to actively optimize the feature embeddings under the guidance of the learned entropy model, joint training unlocks the true potential of Rate-Distortion-Perception control.

D Theoretical Derivation and Loss Formulation

In this section, we provide a rigorous, step-by-step mathematical derivation of the HyperVQ objective function. We aim to demonstrate the complete logical chain from the fundamental Rate-Distortion definition to the final differentiable loss term.

D.1 Derivation of the VQ Rate Objective

Let x denote the input image and $\hat{z}_b$ denote the quantized hyper-latent at granularity b. Our goal is to minimize the bitrate $R_{\hat{y}_b}$ required to transmit the VQ indices $\hat{y}_b$. Let $P^{\star}_{b,ij}(k)$ be the true (unknown) posterior probability that the index at location (i, j) is k, and let $P_{\theta,b,ij}(k)$ be the predicted probability parameterized by θ.

The rate $R_{\hat{y}_b}$ is defined as the expected code length. We expand the cross-entropy term step-by-step:

$$R_{\hat{y}_b} = \mathbb{E}_x \left[\sum_{i,j} H(P^\star_{b,ij}, P_{\theta,b,ij}) \right] \tag{S7}$$

$$= \mathbb{E}_x \left[\sum_{i,j} \sum_{k=1}^{K} -P^\star_{b,ij}(k) \log P_{\theta,b,ij}(k) \right] \tag{S8}$$

$$= \mathbb{E}_x \left[\sum_{i,j} \sum_{k=1}^{K} -P^\star_{b,ij}(k) \log \left(\frac{P^\star_{b,ij}(k)}{P^\star_{b,ij}(k)} P_{\theta,b,ij}(k) \right) \right] \tag{S9}$$

$$= \mathbb{E}_x \Bigg[\sum_{i,j} \Bigg(\underbrace{\sum_{k=1}^{K} -P^\star_{b,ij}(k) \log P^\star_{b,ij}(k)}_{H(P^\star_{b,ij})} + \underbrace{\sum_{k=1}^{K} P^\star_{b,ij}(k) \log \frac{P^\star_{b,ij}(k)}{P_{\theta,b,ij}(k)}}_{D_{\mathrm{KL}}(P^\star_{b,ij} \| P_{\theta,b,ij})} \Bigg) \Bigg] \tag{S10}$$

Since the entropy of the true distribution $H(P^\star_{b,ij})$ is independent of the network parameters θ, minimizing the rate is strictly equivalent to minimizing the KL divergence. In our deterministic VQ framework, the "true" index $k^\star_{b,ij}$ is obtained via nearest-neighbor search:

$$k^\star_{b,ij} = \arg\min_k \|(y_b)_{ij} - \mathbf{e}_k\|^2 \tag{S11}$$

Thus, the true distribution is a Dirac delta function: $P^\star_{b,ij}(k) = \delta(k, k^\star_{b,ij})$. Substituting this into the rate equation simplifies the objective to the Negative Log-Likelihood (NLL):

$$R_{\hat{y}_b} \equiv \mathbb{E}_x \left[\sum_{i,j} \sum_{k=1}^{K} -\delta(k, k^\star_{b,ij}) \log P_{\theta,b,ij}(k) \right] \tag{S12}$$

$$= \mathbb{E}_x \left[\sum_{i,j} -\log P_{\theta,b,ij}(k^\star_{b,ij}) \right] \tag{S13}$$

D.2 From Continuous Hyperprior to Discrete Probability

The hyperprior network outputs the parameters of a continuous Gaussian distribution $\mathcal{N}(\mathbf{h}; \boldsymbol{\mu}_{b,ij}, \boldsymbol{\Sigma}_{b,ij})$ in the embedding space. By modeling the discrete probability mass as being proportional to the likelihood density at the quantization centroids, we convert the continuous density into a categorical distribution:

$$P_{\theta,b,ij}(k) \approx \frac{p(\mathbf{e}_k | \hat{z}_b)}{\sum_{l=1}^{K} p(\mathbf{e}_l | \hat{z}_b)} \tag{S14}$$

$$= \frac{\exp\left(-\frac{1}{2} d^2_{\mathcal{M}}(\mathbf{e}_k, \boldsymbol{\mu}_{b,ij})\right)}{\sum_{l=1}^{K} \exp\left(-\frac{1}{2} d^2_{\mathcal{M}}(\mathbf{e}_l, \boldsymbol{\mu}_{b,ij})\right)} \tag{S15}$$

where $d^2_{\mathcal{M}}$ is the squared Mahalanobis distance.

D.3 Expansion under Isotropic Assumption

To ensure computational efficiency and numerical stability, we assume an isotropic covariance $\boldsymbol{\Sigma}_{b,ij} = (\sigma_{b,ij})^2 \mathbf{I}$. This reduces the Mahalanobis distance to a scaled squared Euclidean distance.

Substituting this probability model back into Eq. (S13), the loss for a single spatial location expands to:

$$\mathcal{L}_{rate} = -\log P_\theta(k^\star) \tag{S16}$$

$$= -\log\left(\frac{\exp\left(-\frac{\|\mathbf{e}_{k^\star}-\boldsymbol{\mu}\|^2}{2\sigma^2}\right)}{\sum_{l=1}^{K}\exp\left(-\frac{\|\mathbf{e}_l-\boldsymbol{\mu}\|^2}{2\sigma^2}\right)}\right) \tag{S17}$$

$$= -\left[\log\left(\exp\left(-\frac{\|\mathbf{e}_{k^\star}-\boldsymbol{\mu}\|^2}{2\sigma^2}\right)\right) - \log\left(\sum_{l=1}^{K}\exp\left(-\frac{\|\mathbf{e}_l-\boldsymbol{\mu}\|^2}{2\sigma^2}\right)\right)\right] \tag{S18}$$

$$= \underbrace{\frac{\|\mathbf{e}_{k^\star}-\boldsymbol{\mu}\|^2}{2\sigma^2}}_{\text{Anchor Alignment}} + \underbrace{\log\left(\sum_{l=1}^{K}\exp\left(-\frac{\|\mathbf{e}_l-\boldsymbol{\mu}\|^2}{2\sigma^2}\right)\right)}_{\text{Log-Partition (Uncertainty penalty)}} \tag{S19}$$

D.4 Centroid Approximation Bound

In Equation (3) of the main paper, we evaluate the density precisely at the codebook centroids $p_k \propto \mathcal{N}(\mathbf{e}_k; \boldsymbol{\mu}, \sigma^2\mathbf{I})$ rather than performing an exact probability mass integral over the entire Voronoi cell $q_k = \int_{\mathcal{V}_k}\mathcal{N}(x; \boldsymbol{\mu}, \sigma^2\mathbf{I})dx$. This is a deliberate centroid-density approximation designed to maintain computational tractability in high-dimensional embedding spaces. Empirically, it provides stable, differentiable gradients without the prohibitive computational cost of high-dimensional cell integration.

To mathematically justify this approximation, we provide a theoretical bound demonstrating that it is highly accurate when codebook cells are relatively small and regular. Let the Voronoi cell $\mathcal{V}_k$ have a bounded radius r_k, such that for any $x \in \mathcal{V}_k$, $\|x - \mathbf{e}_k\| \le r_k$. For the isotropic Gaussian density $\phi(x) = \mathcal{N}(x; \boldsymbol{\mu}, \sigma^2\mathbf{I})$, the variation in log-density for any $x \in \mathcal{V}_k$ compared to the centroid $\mathbf{e}_k$ can be evaluated by expanding the squared distance $\|x-\boldsymbol{\mu}\|^2 = \|(x-\mathbf{e}_k)+(\mathbf{e}_k-\boldsymbol{\mu})\|^2$. The difference is bounded by:

$$|\log\phi(x) - \log\phi(\mathbf{e}_k)| = \left|-\frac{\|x-\boldsymbol{\mu}\|^2}{2\sigma^2} - \left(-\frac{\|\mathbf{e}_k-\boldsymbol{\mu}\|^2}{2\sigma^2}\right)\right| \tag{S20}$$

$$\le \frac{\|\mathbf{e}_k-\boldsymbol{\mu}\|r_k + r_k^2/2}{\sigma^2} \tag{S21}$$

This inequality formally demonstrates that when the Voronoi cells are relatively small (small r_k), the codebook is regular, and the predicted uncertainty spread σ is not too small, the centroid density closely approximates the true cell mass. Mismatches predominantly occur only when the codebook cell volume is highly non-uniform or when $\sigma \to 0$. In practice, the network learns to adapt its predicted σ and $\boldsymbol{\mu}$ to account for these quantization cells, making the empirical coding loss negligible while completely avoiding the $\mathcal{O}(D^3)$ or numerical integration costs associated with exact Voronoi mass computation.

D.5 Gradient Analysis for Differentiability

To demonstrate exactly how this formulation enables end-to-end differentiability, we apply the chain rule to Eq. (S19). The gradient with respect to the predicted mean $\boldsymbol{\mu}$ splits into two terms:

$$\begin{aligned}
\frac{\partial \mathcal{L}_{rate}}{\partial \boldsymbol{\mu}} &= \frac{\partial}{\partial \boldsymbol{\mu}} \left(\frac{\|\mathbf{e}_{k^\star} - \boldsymbol{\mu}\|^2}{2\sigma^2} \right) \\
&\quad + \frac{\partial}{\partial \boldsymbol{\mu}} \log \left(\sum_{l=1}^{K} \exp\left(-\frac{\|\mathbf{e}_l - \boldsymbol{\mu}\|^2}{2\sigma^2} \right) \right) && \text{(S22)} \\
&= -\frac{(\mathbf{e}_{k^\star} - \boldsymbol{\mu})}{\sigma^2} \\
&\quad + \frac{\sum_{l=1}^{K} \exp\left(-\frac{\|\mathbf{e}_l - \boldsymbol{\mu}\|^2}{2\sigma^2} \right) \frac{(\mathbf{e}_l - \boldsymbol{\mu})}{\sigma^2}}{\sum_{j=1}^{K} \exp\left(-\frac{\|\mathbf{e}_j - \boldsymbol{\mu}\|^2}{2\sigma^2} \right)} && \text{(S23)} \\
&= -\frac{(\mathbf{e}_{k^\star} - \boldsymbol{\mu})}{\sigma^2} \\
&\quad + \sum_{l=1}^{K} \underbrace{\left(\frac{\exp(-\frac{\|\mathbf{e}_l - \boldsymbol{\mu}\|^2}{2\sigma^2})}{\sum_{j=1}^{K} \exp(-\frac{\|\mathbf{e}_j - \boldsymbol{\mu}\|^2}{2\sigma^2})} \right)}_{P_\theta(l)} \frac{(\mathbf{e}_l - \boldsymbol{\mu})}{\sigma^2} && \text{(S24)} \\
&= \frac{1}{\sigma^2} \left(\sum_{l=1}^{K} P_\theta(l)(\mathbf{e}_l - \boldsymbol{\mu}) - (\mathbf{e}_{k^\star} - \boldsymbol{\mu}) \right) && \text{(S25)}
\end{aligned}$$

The final gradient acts as an elegant, differentiable force field: it pulls the predicted mean $\boldsymbol{\mu}$ towards the target codebook anchor $\mathbf{e}_{k^\star}$ (first term), while pushing it away from all other codebook vectors $\mathbf{e}_l$, weighted by their currently predicted probabilities $P_\theta(l)$ (second term).

E Detailed Architecture of HVQ-CGIC

While HyperVQ serves as a universal plug-in for any VQ-based codec, we specifically instantiate it on a controllable multi-granularity scenario to rigorously prove its robustness. The complete flowchart of this instantiation, termed HVQ-CGIC, is illustrated in Figure S10.

The Challenge of Arbitrary Rate Control. Standard VQ codecs typically process fixed-scale, full-resolution latent maps. In contrast, the multi-granularity framework dynamically routes features across multiple scales (coarse, medium, and fine) to achieve continuous rate adaptation. This mechanism inherently produces sparse, dynamically masked, and partial feature maps. Traditional statistical entropy models fail to adapt to these non-uniform distributions efficiently.

Overall Architecture Design. As depicted in Figure S10, the HVQ-CGIC framework begins with a **Multi-Granularity Encoder** that extracts hierarchical representations. These features are then quantized using a unified, shared **Codebook** $\mathcal{E}$.

To tackle the sparse latent challenge, our **Hyperprior Analysis & Synthesis** module processes the quantized hyper-latents $\hat{z}_b$ to dynamically predict the continuous Gaussian parameters $(\boldsymbol{\mu}, \sigma^2)$ for the active regions defined by the routing masks. Specifically, the **Hyper Synthesis block** is structurally positioned to directly parameterize the probability mapping, seamlessly adapting to the varying scales and accurately modeling the probabilities of the masked indices.

Finally, the **Probability-Adaptive Decoders** utilize these multi-scale indices, modulated by the spatial masks m_i, to reconstruct the final image $\hat{x}$. By unifying the multi-scale VQ embeddings under a single hyperprior probability model, HVQ-CGIC unlocks precise Rate-Distortion optimization even for arbitrarily masked, sparse latent representations.

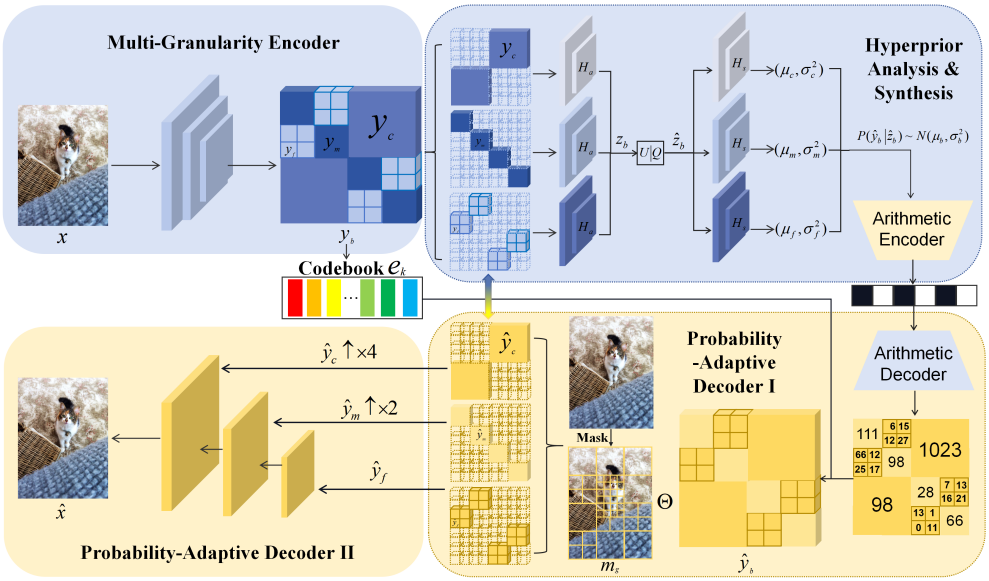

Figure S10: **Detailed Flowchart of HVQ-CGIC.** The framework employs a Multi-Granularity Encoder and shares a unified codebook across granularities. The HyperVQ module robustly predicts distributions for the sparse, dynamically masked indices. $\otimes$ represents element-wise multiplication with spatial routing masks.

F Comparison with Traditional High-Order Context Modeling

A traditional method for improving baseline statistical entropy coding relies on designing adaptive arithmetic coders enriched with high-order spatial contexts (as illustrated in Figure S11). These strategies incrementally build upon basic probability tables by dynamically updating symbol encoding frequencies in real-time based on local neighbor combinations observed during inference.

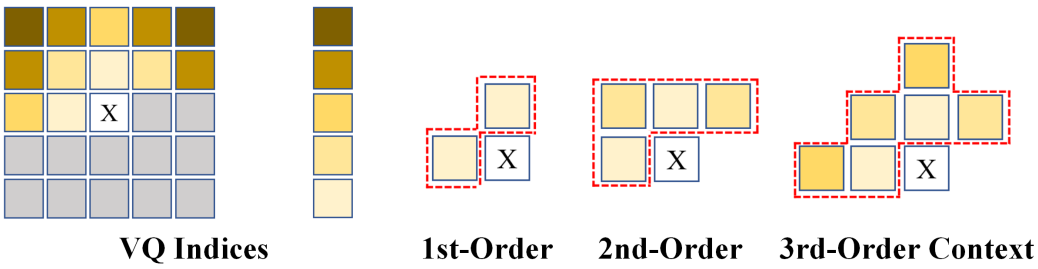

Figure S11: Illustration of spatial context designs. Building upon standard statistical entropy coding, these high-order context strategies perform adaptive arithmetic coding by dynamically updating symbol encoding frequencies in real-time based on neighboring test data.

However, as presented in Table S6, manually tracking index-based spatial contexts demonstrates severe diminishing marginal returns. While a 3rd-order context does improve the BPP marginally over the baseline Control-GIC (from 0.594 down to 0.583), the sequential nature of maintaining and querying these complex frequency states balloons the encoding time to an unacceptable 145.2 ms.

To evaluate our method against these traditional approaches, we present two variants of our model corresponding to the progressive integration protocols detailed in Appendix C. Specifically, HVQ-CGIC (BPP) refers to the model after Stage B (Rapid Hyperprior Adaptation), where the hyperprior is trained exclusively to minimize the bitrate while the VQ backbone remains frozen. Conversely, HVQ-CGIC (RD) denotes the model after Stage C (Joint RD Fine-tuning), representing the fully co-adapted, end-to-end optimized framework that balances both rate and distortion.

Table S6: Ablation study on the Kodak dataset (downsampling factor 4). Percentages in green indicate relative improvements over the theoretical uncompressed limit (No Compression). The Enc. and Dec. times represent the batched inference latency (sum of hyperprior network inference and arithmetic coding time), highlighting the massive throughput advantage of parallel execution.

Method	**BPP** (↓)	**LPIPS** (↓)	**Enc.** (ms)	**Dec.** (ms)
No Compression	0.625	-	-	-
Control-GIC[42]	0.594 (-4.9%)	0.0248	8.43	18.1
1st-Order Context	0.587 (-6.0%)	0.0248	4.66	9.04
2nd-Order Context	0.584 (-6.5%)	0.0248	32.9	48.4
3rd-Order Context	0.583 (-6.7%)	0.0248	145.2	190.2
HVQ-CGIC (RD)	0.530 (-15.2%)	**0.0197**	2.78	4.98
HVQ-CGIC (BPP)	**0.493** (**-21.1%**)	0.0278	**2.71**	**4.17**

In sharp contrast to traditional methods, our HyperVQ approach implicitly learns an ultra-high-dimensional context representation via its lightweight parametric network, entirely bypassing the need for sequential frequency table updates. Consequently, our Stage B model, HVQ-CGIC (BPP), extracts far more latent correlation than traditional 3rd-order contexts, achieving an impressive 0.493 BPP. Furthermore, the parallel capabilities of our Probability Estimation Engine (discussed in Appendix B) drastically reduce the batched encoding latency to just 2.71 ms. Meanwhile, the fully fine-tuned HVQ-CGIC (RD) achieves an optimal balance, significantly improving the perceptual quality (LPIPS drops to 0.0197) while maintaining a highly competitive bitrate and ultra-low inference latency.

G Analysis: Why Not a Direct K-Way Classifier?

When attempting to establish content-adaptive entropy modeling for discrete Vector Quantization (VQ) indices, the most intuitive approach might be to utilize a neural network as a direct K-way classifier (where K is the codebook size) to output a categorical distribution directly. However, in practice, we found that such non-parametric probability models fail to converge efficiently or scale properly.

1. Parameter Explosion ($\mathcal{O}(K)$ vs $\mathcal{O}(D)$): A direct K-way classifier requires outputting a logit for every single codebook entry at every spatial location. For a latent map of size $W \times H$, the output tensor shape is $W \times H \times K$. For a standard codebook size of $K = 1024$, this represents a massive prediction space. In contrast, our HyperVQ module is a **parametric model** that only requires predicting the parameters $(\boldsymbol{\mu}, \sigma)$ of a Gaussian distribution in the D-dimensional continuous embedding space. Assuming $D = 4$, our network only outputs a tensor of shape $W \times H \times (D + 1)$. This means the K-way classifier requires over $\mathbf{200}\times$ more parameters at the final prediction layer, making it highly memory-intensive and prone to overfitting.

2. Codebook-Size-Independent Scalability: Modern advancements in VQ-based generation are rapidly moving towards ultra-large codebooks ($K = 100,000$ or larger). A K-way classifier becomes mathematically intractable to optimize for such large vocabularies due to the vanishing gradients in the immense Softmax calculation. Our HyperVQ framework fundamentally bypasses this limit: since the network only predicts the D-dimensional mean feature $\boldsymbol{\mu}$, its complexity is entirely **codebook-size-independent**.

H Entropy Coding and Spatial Bit Usage Analysis

To intuitively understand why HyperVQ excels under dynamic constraints, we visualized the spatial bit usage distribution of $Kodim21$ at a $\times 4$ downsampling ratio compared to the Control-GIC baseline. As shown in Figure S12, our framework almost exhausts every bit of coding redundancy in the latent feature space. By capturing the underlying structural correlations via the embedding-space Gaussian

prior, HVQ-CGIC saves over 21.4% bitrate compared to Control-GIC on this single image, without any loss in textural fidelity.

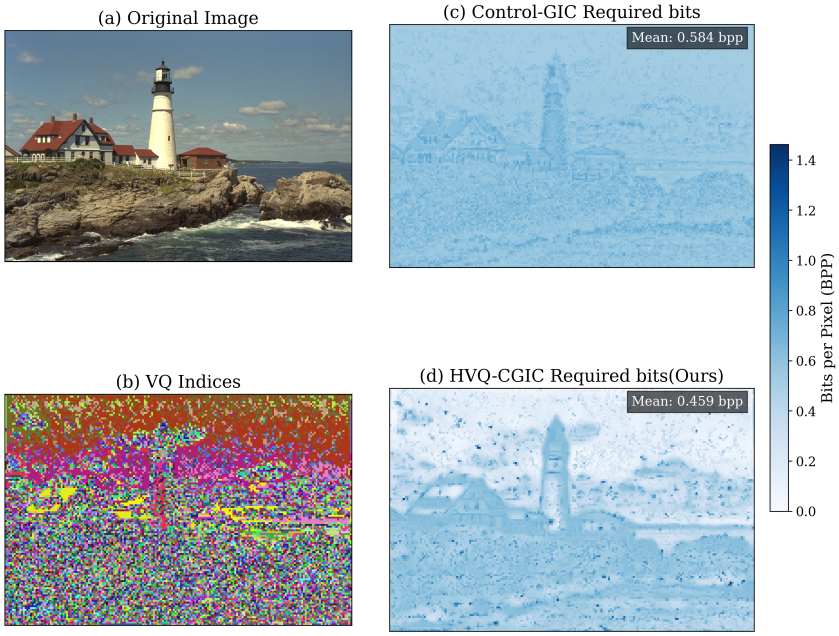

Figure S12: **Visualization of spatial bit usage distribution on** $Kodim21$**.** HVQ-CGIC almost exhausts every bit of coding redundancy in the latent feature space.

I Extended Qualitative Visual Comparisons

This section provides extensive visual comparisons to demonstrate the perceptual superiority of HVQ-CGIC in complex scenes, high-frequency textures, and structured regions. We compare our framework against SOTA baselines (HiFiC, Control-GIC, CDC) on the Kodak, DIV2K, and CLIC 2020 datasets.

Structure Preservation (Kodak). Figure S14 visualizes performance on structured regions (e.g., text, geometric lines). HVQ-CGIC preserves sharp edges and accurate geometry, avoiding the blurring or ringing artifacts observed in baselines.

Texture Synthesis (DIV2K). Figure S15 demonstrates the reconstruction of dense, stochastic textures (e.g., foliage, urban details). While previous VQ methods tend to over-smooth fine grain, HVQ-CGIC faithfully synthesizes high-frequency details, yielding a photorealistic appearance.

Complex Scenes (CLIC 2020). In complex real-world scenes (Figures S13 and S16), HVQ-CGIC exhibits superior global consistency. It strongly suppresses generative artifacts and color shifts, maintaining fidelity to the original content supported by the precise probability estimation of the VQ indices.

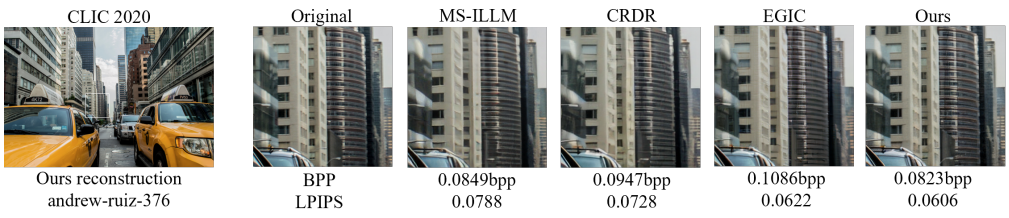

Figure S13: **Qualitative comparison at extremely low bit rates on CLIC 2020.** HVQ-CGIC reconstructs faithful textures with far fewer artifacts.

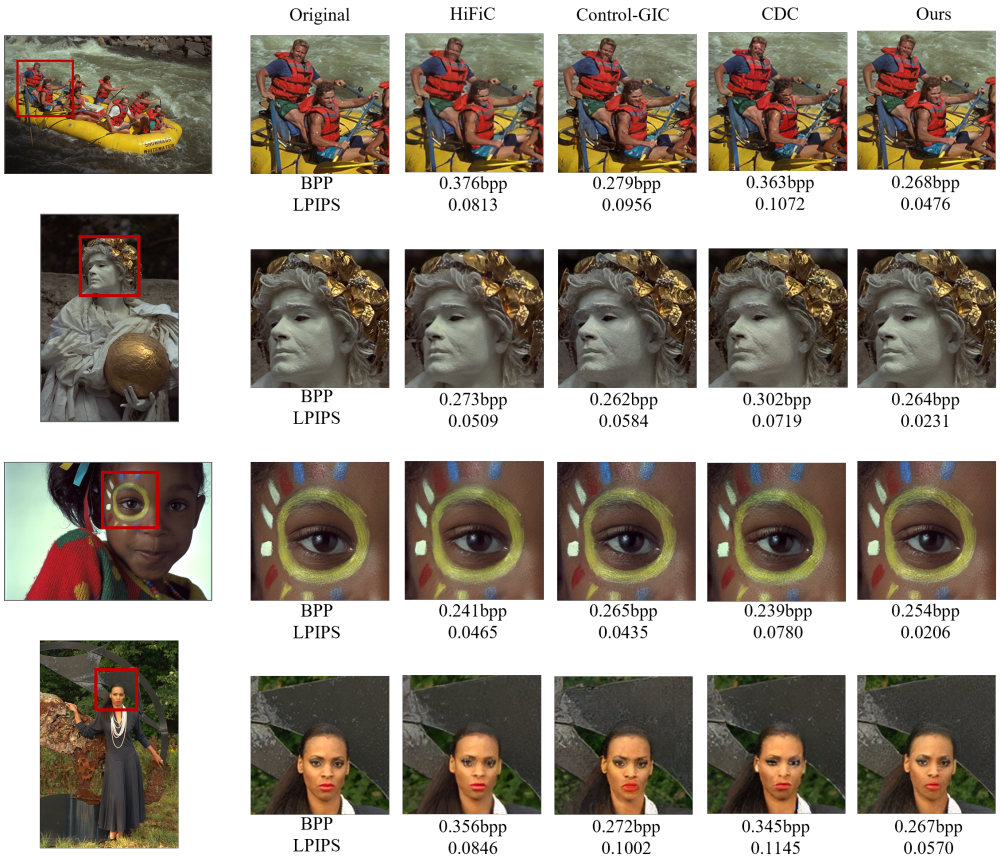

Figure S14: **Qualitative Comparison on Kodak Dataset.** Best viewed when zoomed in. HVQ-CGIC preserves sharp structural edges and suppresses artifacts better than baselines in structured regions.

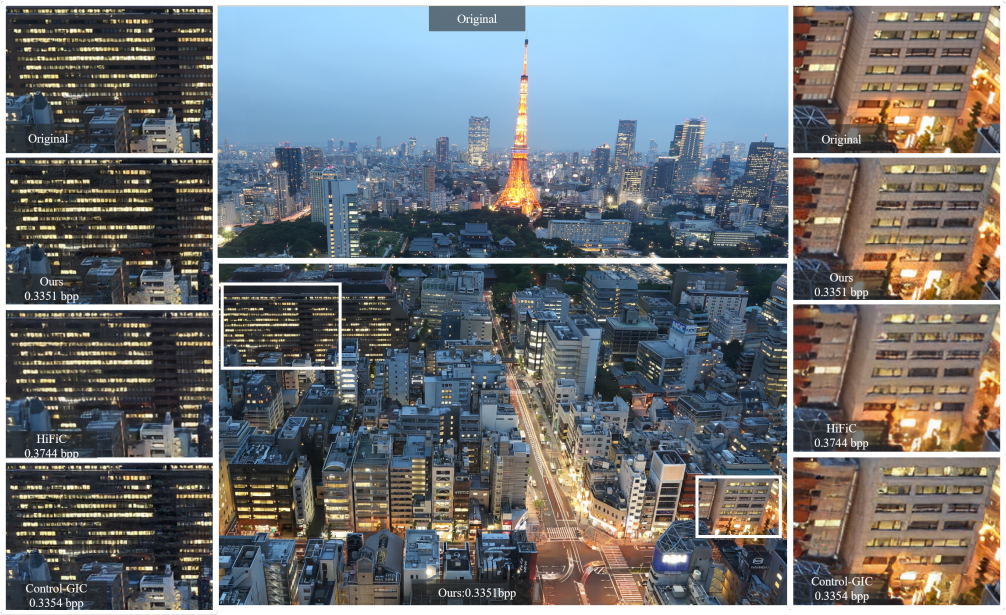

Figure S15: **Qualitative Comparison on DIV2K Dataset.** Comparison of fine-grained texture reconstruction.

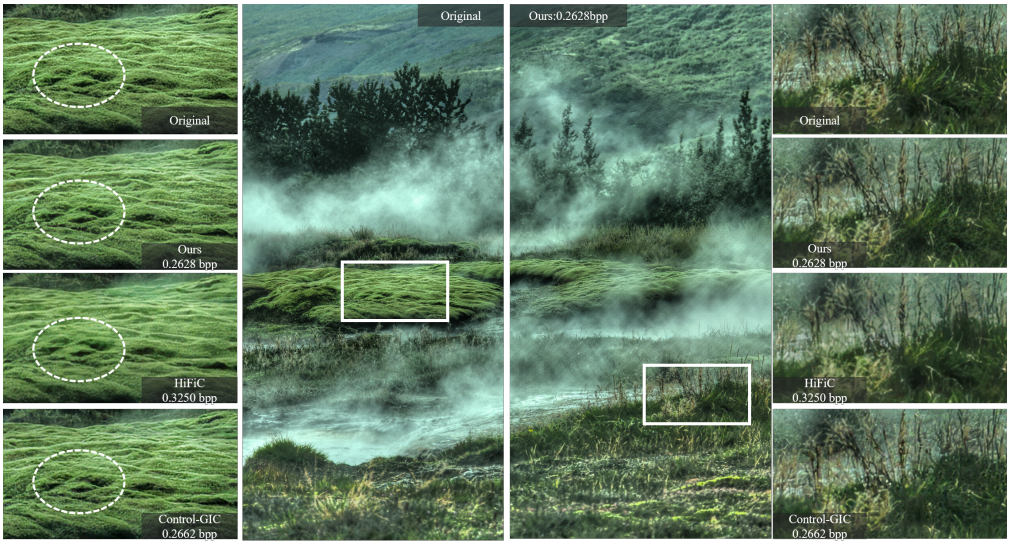

Figure S16: **Qualitative Comparison on CLIC 2020 Dataset.** Comparison of perceptual quality in complex scenes.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification:The abstract and introduction (§1) clearly state the core contributions including the continuous embedding space modeling and the end-to-end RD optimization, which are validated in the experimental section (§4).

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations regarding the isotropic diagonal covariance assumption and its impact on capturing cross-dimensional correlations are explicitly discussed in §5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The mathematical derivation mapping continuous Gaussian densities to discrete probabilities is outlined in §3.2 and fully derived in Appendix D.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The experimental protocol (§4.1) details the baseline architectures, datasets, and metrics used. Comprehensive integration details and training schedules are provided in Appendix C to ensure reproducibility.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [N/A]

   Justification: We will release the source code upon paper publication.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The training setup including datasets, crop sizes, and hyperparameters is specified in §4.3, with extended configuration details provided in Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The experiments are conducted on standard benchmark datasets in the image compression field. It is a common community practice to report the average metrics over the entire dataset, making error bars inapplicable. Furthermore, in this field, there is no random variation caused by hyperparameters, and each encoding and decoding run yields deterministic fixed results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The average encoding and decoding inference latencies are explicitly reported in Table 2, and detailed hardware specifications are documented in the supplementary material.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research focuses on fundamental image compression algorithms and strictly adheres to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

Justification: This work focuses on the foundational algorithmic efficiency of image compression. It does not have direct foreseeable negative societal impacts beyond the general implications of improved generative modeling.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer:[N/A]

Justification: The paper introduces an entropy coding framework for image compression which does not pose a high risk for misuse.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Standard public datasets including ImageNet and Kodak alongside baseline models are appropriately cited in §4.1.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce new datasets or large-scale foundational assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The research does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The research does not involve human subjects and therefore does not require IRB approval.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve Large Language Models as any important or non standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.