NeurIPS 2026

HyperVQ

Enabling Hyperprior Entropy Modeling for VQ-Based Generative Image Compression

Yi Niu Tianyi Xu Mingming Ma Xinkun Wang
Xidian University

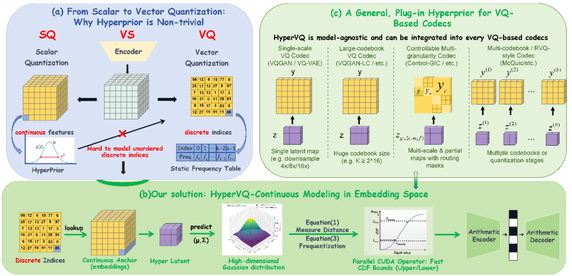
西安电子科技大学 XIDIAN UNIVERSITY

NEURAL INFORMATION PROCESSING SYSTEMS

WeChat: Tianyi Xu

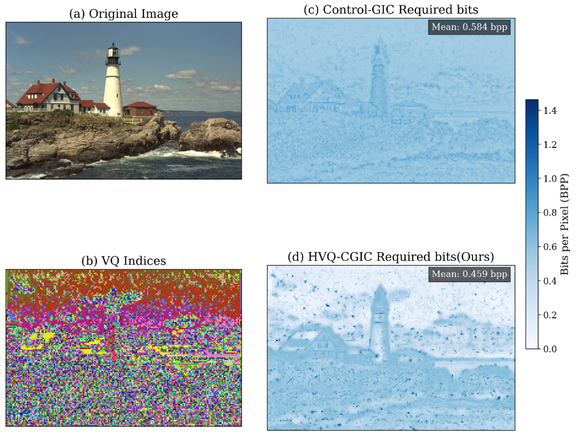
01 Motivation

Hyperpriors exploit spatial redundancy in scalar-quantized latents, but extending them to unordered VQ indices is non-trivial. Static frequency tables miss content-dependent redundancy; rate optimization also requires differentiable index probabilities.



HyperVQ shifts probability modeling into the continuous embedding space. Codebook entries act as anchors: a predicted Gaussian is converted into categorical index probabilities, enabling spatially adaptive entropy coding and joint rate–distortion optimization.

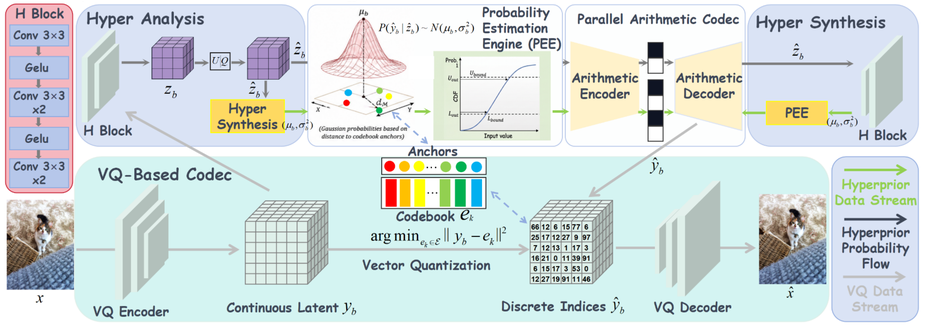
Spatial bit usage

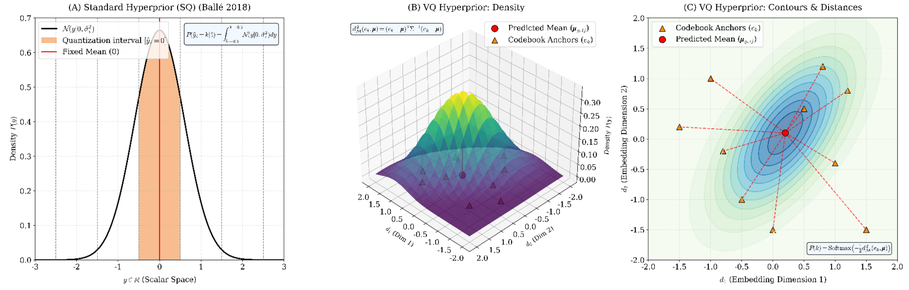


Spatial bit-cost maps on Kodim21 at 4× downsampling. The displayed mean decreases from 0.584 to 0.459 bpp in this example.

02 Method

The base VQ codec produces indices. HyperVQ predicts embedding-space distributions; the Probability Estimation Engine converts them into frequencies for arithmetic coding.





Insight: Codebook geometry turns a continuous hyperprior into content-adaptive probabilities for unordered indices, making the index-rate objective differentiable.

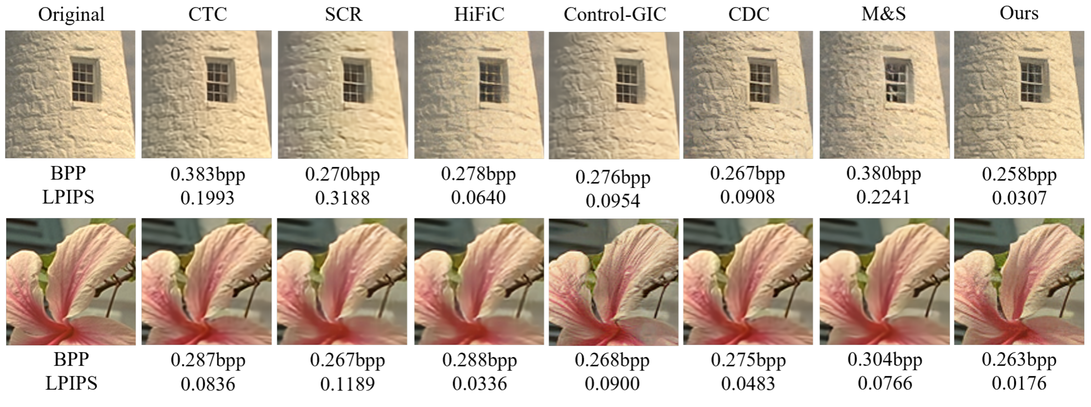
SQ: scalar-bin integration
Zero-mean scale prior

$$P(\hat{y}_i = k \mid \hat{z}) = \int_{k-\frac{1}{2}}^{k+\frac{1}{2}} \mathcal{N}(u; 0, \hat{\sigma}_i^2)\, du$$

HyperVQ: anchor normalization
Isotropic model

$$P(k \mid \hat{z}) = \frac{\exp\left[-\|\mathbf{e}_k - \boldsymbol{\mu}\|^2/(2\sigma^2)\right]}{\sum_{\ell=1}^{K} \exp\left[-\|\mathbf{e}_\ell - \boldsymbol{\mu}\|^2/(2\sigma^2)\right]}$$
$$(\boldsymbol{\mu}, \sigma) = H_s(\hat{z}); \quad \mathbf{e}_k \text{ is codeword } k.$$

03 Visualization



04 Experiments

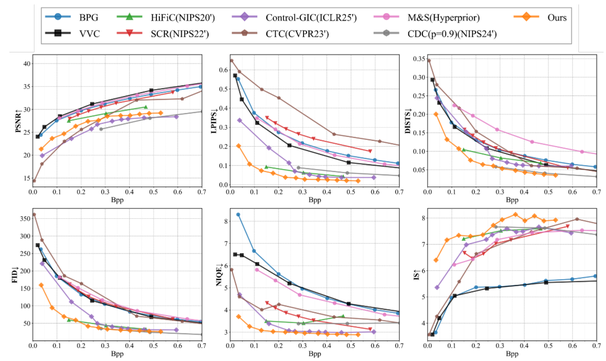
Generality across VQ codecs

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 (+).

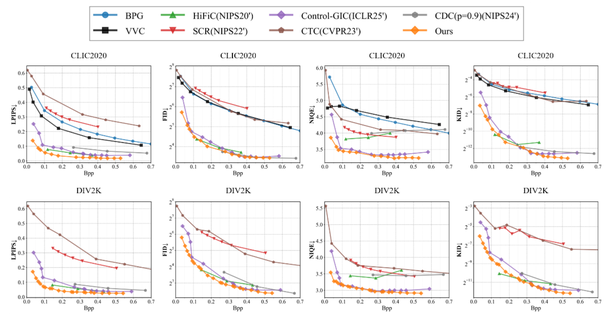
Dataset	Method	Codebook	Scale	Theoretical bpp	Huffman (Gain)	HyperVQ (Gain)
ImageNet-1K Val	VQGAN-LC (NeurIPS'24)	16384	16×	0.0547	0.0541 (1.1%)	0.0458 (16.3%)
		100K	16×	0.0641	0.0642 (-0.2%)	0.0564 (12.0%)
		100K	8×	0.2595	0.2516 (3.0%)	0.2072 (20.2%)
	VQGAN (CVPR'21)	1024	4×	0.6250	0.5844 (6.5%)	0.4703 (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 × 2048	+32×	0.1230	0.1216 (1.1%)	0.0945 (23.2%)
		+2 × 512	+64×	0.1277	0.1263 (1.1%)	0.0983 (23.0%)
				Average Gain	**2.54%**	**18.5%**

Across 15 configurations, HyperVQ saves 18.5% bitrate on average relative to theoretical fixed-length coding. Huffman saves 2.54% on the same basis.

Rate–distortion on Kodak



CLIC2020 and DIV2K



Perceptual rate–distortion performance of HVQ-CGIC.