UNIFIED IMAGE GENERATION AND COMPRESSION WITH HIERARCHICAL PROGRESSIVE CONTEXT MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Learned image compression methods typically transform images into the latent domain and model the probability distributions of latents for efficient entropy coding. Image generation models, particularly latent-space generative frameworks, rely on learned probability distributions for latent sampling. More accurate latent distribution modeling can therefore benefit both tasks. However, existing studies, including current generative image compression methods, have rarely explored this synergistic property. To address this research gap, we propose Uni-HPCM, a unified compression–generation framework that uses shared conditional latent distributions for both entropy coding and sampling. Built on the advanced Hierarchical Progressive Context Modeling (HPCM) model, Uni-HPCM is deliberately trained to jointly optimize the rate–distortion cost and perceptual quality. After training, the single unified model supports three versatile paradigms: reconstructive compression via full main-latent coding, pure image generation via full latent sampling, and generative compression across a spectrum of coding–sampling combinations. Experiments demonstrate competitive compression performance, favorable generation quality, and promising results in generative image compression. In addition, a larger model enhances both rate–distortion performance and generation quality, highlighting their potential synergy.

1 INTRODUCTION

Learned image compression and latent-space image generation both rely on learned probability distributions, but use them for different purposes. In learned image compression, an encoder maps an image into latent representations, and an entropy model estimates their probability distributions for efficient entropy coding (Ballé et al., 2018; Minnen et al., 2018; Cheng et al., 2020). Recent methods improve rate–distortion performance through stronger entropy models (He et al., 2022; Liu et al., 2023; Jiang & Wang, 2023; Li et al., 2024a; Lu et al., 2025; Li et al., 2025a). In latent-space image generation, a model samples latent representations from learned distributions and decodes them into images (Esser et al., 2021; Chang et al., 2022; Rombach et al., 2022). More accurate latent distribution modeling can therefore benefit both coding efficiency and generation quality, motivating the use of shared distributions for entropy coding and sampling.

However, existing models typically learn latent distributions for either compression or generation in isolation. Generative image compression brings the two tasks closer by using generative models to reconstruct perceptually realistic images from compressed representations (Muckley et al., 2023; Careil et al., 2024; Li et al., 2025c; Ke et al., 2025; Guo et al., 2025; Zhang et al., 2025b; Chen et al., 2026c). This scheme, however, does not by itself establish a shared latent distribution for entropy coding and sampling. Related work also connects compression and generation through product-quantized masked image modeling (El-Nouby et al., 2023), shared token priors for coding and completion (Xue et al., 2024), and diffusion codebooks (Ohayon et al., 2025). However, the potential of shared latent distributions remains insufficiently explored, particularly for jointly optimizing entropy coding and continuous latent sampling within a single model while maintaining competitive rate–distortion performance.

In this paper, we propose Uni-HPCM, a unified framework that uses shared latent distributions for both entropy coding and image generation. Built on recent advanced Hierarchical Progressive Context Modeling (HPCM) (Li et al., 2025a), Uni-HPCM uses a shared HPCM context model to predict

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

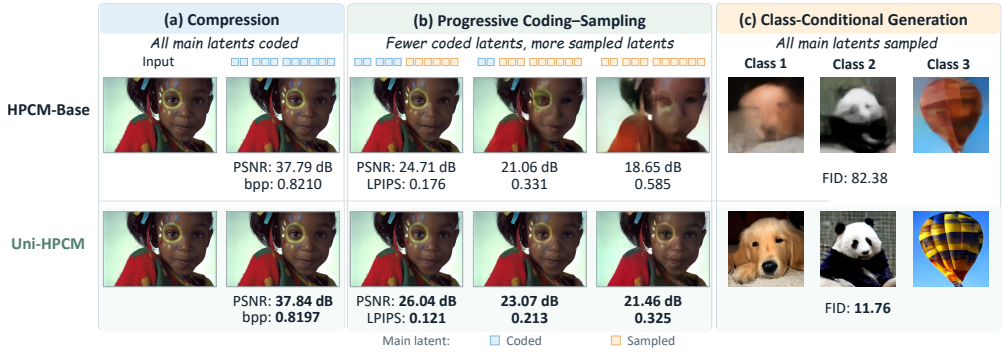


Figure 1: Three capabilities of Uni-HPCM: (a) image compression; (b) generative compression with $k = 6, 9, 11$ sampled subsets; and (c) class-conditional generation on ImageNet-256. Rows compare HPCM-Base and Uni-HPCM. Blue and orange squares denote coded and sampled subsets. PSNR, bpp, and LPIPS are Kodak averages at $\lambda = 0.0483$. FID is evaluated on ImageNet-256.

conditional distributions for coding and sampling, and a shared synthesis transform to reconstruct images from the resulting latents. As illustrated in Fig. 1, its progressive structure allows the proportion of coded and sampled main-latent subsets to be adjusted within a single model. Coding all main-latent subsets yields standard image compression, while coding only a portion and sampling the remainder enables generative compression. Sampling all main-latent subsets with suitable conditioning information further enables conditional image generation.

Building this unified framework requires training the model to handle both coded and sampled latents. Training solely for rate–distortion performance leaves the context model and synthesis transform poorly adapted to sampled latents. To address this issue, we introduce Hybrid Coding–Sampling Training (HCST), which jointly optimizes rate–distortion performance and perceptual reconstruction quality. For each training image, we construct one reconstruction from fully coded latents and another from a combination of coded and sampled latents. The fully coded reconstruction is optimized with the rate–distortion objective, while the mixed reconstruction is optimized for perceptual quality. This joint training adapts the model to latent sampling while retaining rate–distortion optimization for compression.

At inference time, Uni-HPCM supports different combinations of coding and sampling through Progressive Coding–Sampling (PCS). By adjusting the proportion of coded and sampled main-latent subsets, PCS enables flexible generative compression within Uni-HPCM. For class-conditional image generation, we introduce Class-Conditional Latent Sampling (CCLS), which uses hyper-latents extracted from images produced by a pretrained class-conditional model to guide the sampling of all main-latent subsets. Experiments demonstrate that Uni-HPCM maintains competitive rate–distortion performance and achieves favorable class-conditional generation quality on ImageNet. In generative compression, varying the proportion of coded and sampled main-latent subsets enables flexible bitrate control while yielding favorable perceptual quality. Moreover, increasing model capacity further improves both rate–distortion performance and generation quality, highlighting their potential synergy. Our contributions can be summarized as follows:

- We propose Uni-HPCM, a unified framework that uses shared conditional latent distributions for entropy coding and progressive sampling, supporting image compression, conditional image generation, and generative compression within a single model.
- We introduce Hybrid Coding–Sampling Training (HCST), which jointly optimizes rate–distortion performance and perceptual reconstruction quality, adapting the model to both coded and sampled latents.
- We develop Progressive Coding–Sampling (PCS) to enable flexible generative compression by adjusting the proportion of coded and sampled main-latent subsets, and Class-Conditional Latent Sampling (CCLS) to enable class-conditional image generation through full main-latent sampling.

2 Related Work

Learned Image Compression. Learned image compression jointly optimizes transforms and entropy models under a rate–distortion objective. Transform architectures have advanced through

attention, Transformer–CNN hybrids, frequency-aware Transformers, and content-aware Mamba models (Chen et al., 2021; Liu et al., 2023; Li et al., 2024a; Chen et al., 2026b). Entropy models exploit hyperpriors and autoregressive context (Ballé et al., 2018; Minnen et al., 2018), with Gaussian mixture or generalized Gaussian likelihoods (Cheng et al., 2020; Zhang et al., 2025a). Context modeling has evolved through channel-wise, checkerboard, and grouped space–channel dependencies (Minnen & Singh, 2020; He et al., 2021; 2022), Transformer-based and multi-reference models (Qian et al., 2022; Jiang & Wang, 2023), and adaptive and dictionary-based entropy models (Kim et al., 2024; Lu et al., 2025). Recent work also studies the rate–distortion–latency trade-off in Transformer architectures (Chen et al., 2025) and cross-granularity online optimization (Kuang et al., 2025). HPCM (Li et al., 2025a) combines hierarchical coding with progressive context fusion across spatial scales and coding steps. These methods primarily learn latent distributions for coding efficiency. Uni-HPCM extends HPCM's conditional distributions to both entropy coding and latent sampling.

Image Generation in Latent Space. Latent-space generative models synthesize images by decoding sampled representations. VQ-VAE and VQ-VAE-2 learn discrete representations with autoregressive priors (van den Oord et al., 2017; Razavi et al., 2019), while VQGAN combines perceptual tokenization with Transformer priors (Esser et al., 2021). RQ-Transformer and MaskGIT explore residual quantization and masked token prediction (Lee et al., 2022; Chang et al., 2022), and LDM performs diffusion in continuous latent space (Rombach et al., 2022). Subsequent work advances discrete-token generation and continuous-latent sampling (Sun et al., 2024; Yu et al., 2025; Li et al., 2024b; Ren et al., 2025). We investigate how conditional distributions learned within an image codec can support generation while retaining their entropy coding role.

Generative Compression and Unified Modeling. Generative compression considers perceptual quality alongside bitrate and distortion (Blau & Michaeli, 2019). MS-ILLM improves statistical fidelity through implicit local likelihood models (Muckley et al., 2023), while GLC and DLF explore generative latent representations at ultra-low bitrates (Jia et al., 2024; Xue et al., 2025). Diffusion-based codecs leverage generative priors for perceptual reconstruction (Careil et al., 2024; Li et al., 2025c; Ke et al., 2025; Vonderfecht & Liu, 2025), with recent work exploring one-step decoding (Guo et al., 2025; Zhang et al., 2025b) and video diffusion conditioned on compact image anchors (Chen et al., 2026c).

Connecting compression and generation, PQ-MIM combines product quantization with masked conditional entropy modeling to support both tasks without additional training (El-Nouby et al., 2023). UIGC uses a shared prior for entropy coding and maximum-probability completion of omitted discrete tokens (Xue et al., 2024). DDCM uses finite diffusion-noise codebooks for compressed image generation and image-dependent noise selection for lossy compression (Ohayon et al., 2025). Uni-HPCM shares conditional latent distributions between coding and progressive stochastic sampling in continuous latent space, with hybrid training optimizing rate–distortion performance for fully coded reconstructions and perceptual quality for partially sampled ones. Adjusting the coding–sampling balance enables image compression, conditional image generation, and generative compression within a single model.

3 METHOD

3.1 OVERVIEW OF UNI-HPCM

Figure 2 summarizes the Uni-HPCM architecture. Uni-HPCM integrates entropy coding and progressive latent sampling within a learned image compression model. As shown in Fig. 2, it uses analysis transform g_a, synthesis transform g_s, hyper-analysis transform h_a, hyper-synthesis transform h_s, and progressive context model of HPCM.

Given an input image x, the analysis transform produces a main latent $y = g_a(x)$ and a hyper-latent $z = h_a(y)$. After quantization $\hat{y} = Q(y)$ and $\hat{z} = Q(z)$, g_s reconstructs $\hat{x} = g_s(\hat{y})$, while $h_s(\hat{z})$ provides the side information that initializes progressive context modeling. Following HPCM (Li et al., 2025a), we partition the main latent into T ordered masked subsets $\hat{y}_1, \ldots, \hat{y}_T$ across three spatial scales. The coding schedule proceeds from low to high spatial resolution, progressively modeling long-range and then short-range dependencies. The three stages contain 2, 3, and 6 subsets, respectively, giving $T = 11$. The joint distribution factorizes as

$$p_\theta(\hat{y}, \hat{z}) = p_\theta(\hat{z}) \prod_{t=1}^{T} p_\theta(\hat{y}_t \mid \hat{y}_{<t}, h_s(\hat{z})) \,, \tag{1}$$

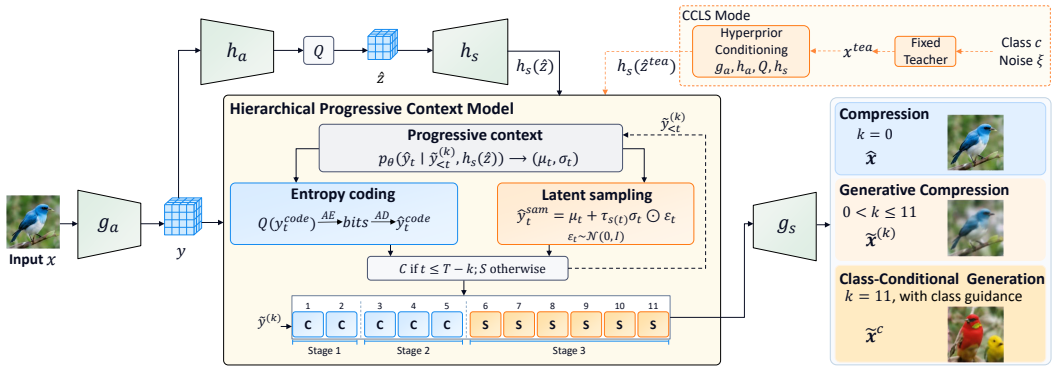


Figure 2: Uni-HPCM framework. The shared hierarchical progressive context model predicts conditional distributions for both entropy coding and latent sampling. The first $T - k$ main-latent subsets are coded and the remaining k are sampled, supporting image compression, generative compression, and class-conditional generation within a single model.

where $\hat{y}_{<t}$ contains the subsets available before coding step t. At each step, the context model predicts a mean and scale (μ_t, σ_t) using the hyperprior information and the preceding subsets. For entropy coding, these parameters characterize the distribution of the quantized latent $\hat{y}_t^{\text{code}}$.

To generate latent subsets, Uni-HPCM uses the predicted distribution parameters for sampling. For step t in stage $s(t)$, the sampled subset is

$$\hat{y}_t^{\text{sam}} = \mu_t + \tau_{s(t)}\sigma_t \odot \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, I), \tag{2}$$

where $\tau_{s(t)}$ is the sampling temperature for stage $s(t)$. Each sampled subset uses the mask defined for the corresponding coding step. Following HPCM's coding order (Li et al., 2025a; 2023), the context model predicts each subsequent distribution using the hyperprior information and previously coded or sampled subsets.

For k sampled subsets, the combined latent representation is

$$\tilde{y}_t^{(k)} = \begin{cases} \hat{y}_t^{\text{code}}, & t \leq T - k, \\ \hat{y}_t^{\text{sam}}, & t > T - k. \end{cases} \tag{3}$$

Uni-HPCM retains the first $T - k$ coded subsets and samples the remaining k subsets following the hierarchical coding schedule. By coding the first $T - k$ subsets before sampling, Uni-HPCM aims to preserve structural information from the input image while synthesizing image details conditioned on the encoded representation.

The synthesis transform reconstructs the image as $\tilde{x}^{(k)} = g_s(\tilde{y}^{(k)})$. This representation is used in HCST and in both sampling modes.

3.2 Shared Distribution Modeling

Shared distribution modeling connects the likelihood objective for compression with distribution matching for latent sampling. Consider a scalar latent Y under a fixed context, with predicted Gaussian density $f_\theta(u) = \mathcal{N}(u; \mu, \sigma^2)$. Let ν denote the density of $\mathcal{U}(-\frac{1}{2}, \frac{1}{2})$. Adding independent noise $\eta \sim \nu$ gives the training latent $V = Y + \eta$, whose density is $\bar{q}$. The compression branch uses the likelihood $\bar{p}_\theta = f_\theta * \nu$ and the rate objective $\mathcal{R}_{\text{train}} = -\mathbb{E}_{\bar{q}}[\log_2 \bar{p}_\theta(V)]$ (Ballé et al., 2018), where $*$ denotes convolution.

The base sampler draws $U \sim f_\theta$. To compare source and model samples under the same training relaxation, let $W = U + \eta'$ with independent $\eta' \sim \nu$. Marginalizing U gives the density of W:

$$p_W(v) = \int f_\theta(u)\nu(v - u)\,\mathrm{d}u = (f_\theta * \nu)(v) = \bar{p}_\theta(v). \tag{4}$$

Substituting this equality into the divergence between source and model samples yields

$$\begin{aligned} D_{\text{KL}}(\bar{q} \| p_W) &= \int \bar{q}(v) \log_2 \frac{\bar{q}(v)}{\bar{p}_\theta(v)}\,\mathrm{d}v \\ &= \mathcal{R}_{\text{train}} - h(\bar{q}), \end{aligned} \tag{5}$$

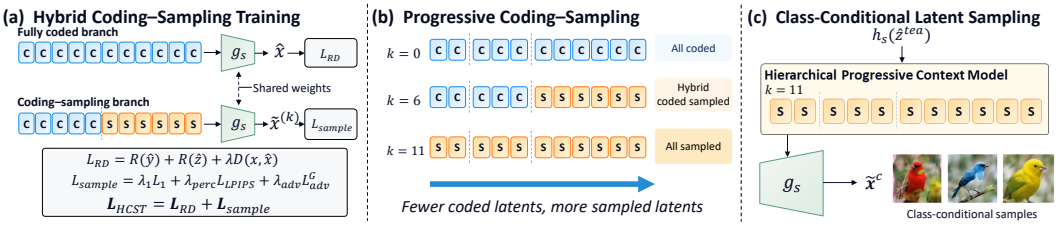


Figure 3: Training and sampling modes of Uni-HPCM. (a) HCST jointly trains fully coded and coding–sampling branches. (b) PCS varies the number of sampled subsets k for progressive coding–sampling. (c) CCLS samples all main-latent subsets using class-conditional hyper-latent guidance.

where h is differential entropy. For fixed $\bar{q}$ and context, reducing the rate objective therefore reduces the mismatch between source and model samples under the training relaxation. This provides a distribution-fitting basis for reusing the Gaussian parameters learned for entropy coding in latent sampling. Appendix A provides further analysis.

HCST extends supervision to the images produced along the actual sampling trajectories, using stage-wise temperatures and previously sampled subsets as context. Each sampled subset enters the synthesis transform and guides subsequent context predictions, so HCST trains both components along mixed coding–sampling trajectories while retaining the fully coded rate–distortion objective.

3.3 Hybrid Coding–Sampling Training

As shown in Fig. 3(a), for each training image x, we construct a fully coded reconstruction $\hat{x}$ and a mixed coding–sampling reconstruction $\tilde{x}^{(k)}$. We retain HPCM's multi-scale latent partition and coarse-to-fine coding order, which progressively models long-range and then short-range dependencies (Li et al., 2025a). We randomly select k from $\{1, \ldots, T\}$, where $T = 11$, encode the first $T - k$ subsets, and sequentially sample the remaining subsets using the available context. Retaining the earlier coded subsets aims to preserve structural information from the input image while allowing subsequent sampling to synthesize details and textures. Varying k covers different coding–sampling combinations, including full main-latent sampling conditioned on the hyper-latent. We use lower sampling temperatures at coarse spatial scales to limit variation in the early subsets that guide subsequent predictions, and higher temperatures at finer scales to allow greater variation in synthesized details. The same stage-wise temperatures are used during training and inference. The fully coded reconstruction is supervised by the rate–distortion objective

$$\mathcal{L}_{\mathrm{RD}} = R(\hat{y}) + R(\hat{z}) + \lambda D(x, \hat{x}), \tag{6}$$

where $R(\hat{y})$ and $R(\hat{z})$ are the estimated rates of the main and hyper-latents, respectively. Training operates directly on latent representations, with rates estimated without running the entropy coder. The mixed reconstruction is supervised using the input image as the target, giving the joint objective

$$\mathcal{L}_{\mathrm{HCST}} = \mathcal{L}_{\mathrm{RD}} + \lambda_1 \|\tilde{x}^{(k)} - x\|_1 + \lambda_{\mathrm{perc}} \mathcal{L}_{\mathrm{LPIPS}}(x, \tilde{x}^{(k)}) + \lambda_{\mathrm{adv}} \mathcal{L}^G_{\mathrm{adv}}. \tag{7}$$

The mixed-reconstruction losses adapt the context model and synthesis transform to sampled latents, while the fully coded branch retains rate–distortion supervision. The discriminator is optimized separately with a hinge loss. We optimize HCST in two stages: first updating the hyperprior and progressive entropy model with g_a and g_s fixed, then jointly fine-tuning the full codec.

3.4 Progressive Coding–Sampling

At inference time, Progressive Coding–Sampling (PCS) controls the balance between transmitted information and sampled content within a single trained model, as illustrated in Fig. 3(b). Given an input image x, we encode its hyper-latent $\hat{z}$ and the first $T - k$ main-latent subsets, where $k \in \{0, \ldots, T\}$. The decoder recovers these subsets and sequentially samples the remaining k subsets, conditioned on the hyperprior information and previously decoded or sampled subsets. The shared synthesis transform then produces the reconstruction $\tilde{x}^{(k)} = g_s(\tilde{y}^{(k)})$.

PCS follows HPCM's coarse-to-fine coding order, which progressively models long-range and then short-range dependencies. Retaining the earlier coded subsets provides context from the input image to guide subsequent sampling. This design aims to preserve global structure through the coded representation while allowing sampling to synthesize local details and textures. Increasing the number of coded subsets extends this guidance to finer spatial scales and leaves fewer subsets to be sampled.

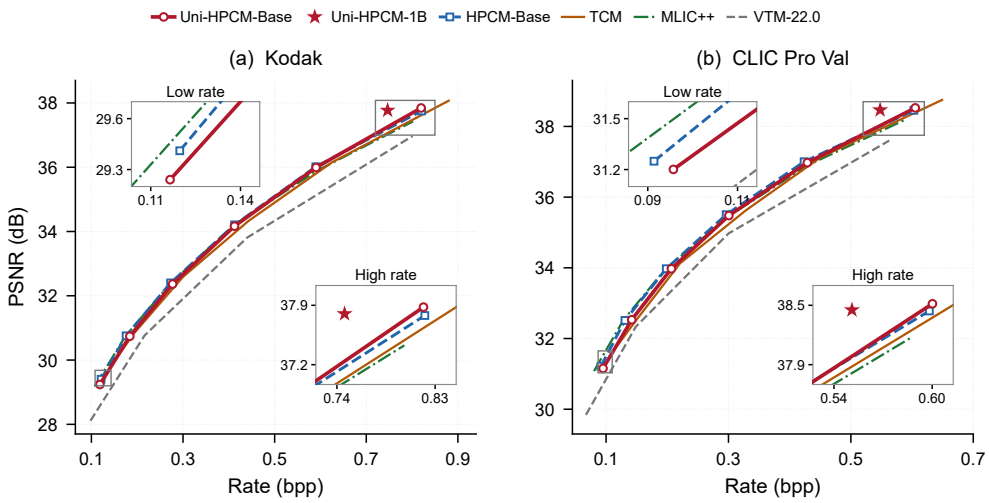


Figure 4: Rate–distortion performance on (a) Kodak and (b) CLIC Pro Val.

For $0 < k < T$, coding and sampling are combined for generative compression. At $k = T$, all main-latent subsets are sampled, conditioned only on the hyper-latent retained from the input image. By varying k, a single trained model supports generative image compression across a broad bitrate range, with flexible control over reconstruction fidelity and perceptual quality.

3.5 Class-Conditional Latent Sampling

Class-Conditional Latent Sampling (CCLS) extends full main-latent sampling to class-conditional image generation, as illustrated in Fig. 3(c). We use a pretrained class-conditional generator to provide conditioning information through the hyper-latent. Given a class label c and noise ξ, the generator produces an image

$$x^{\text{tea}} = G_\phi(c, \xi), \tag{8}$$

from which Uni-HPCM extracts the quantized hyper-latent

$$\hat{z}^{\text{tea}} = Q\big(h_a\big(g_a(x^{\text{tea}})\big)\big)\,. \tag{9}$$

The hyper-synthesis transform converts $\hat{z}^{\text{tea}}$ into conditioning information for the shared context model. Conditioned on this information, Uni-HPCM sequentially samples all T main-latent subsets following the same coarse-to-fine order as PCS. At each step, the context model predicts (μ_t, σ_t) from $h_s(\hat{z}^{\text{tea}})$ and the previously sampled subsets. Sampling and image synthesis are given by

$$\tilde{y}_t^c = \mu_t + \tau_{s(t)}\sigma_t \odot \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, I), \qquad \tilde{x}^c = g_s(\tilde{y}^c), \tag{10}$$

where $t = 1, \ldots, T$. CCLS thus enables class-conditional image generation using the same model that supports compression and generative compression.

4 Experiments

4.1 Experimental Setup

Datasets. Compression is evaluated on Kodak (Eastman Kodak), CLIC Pro Val (Challenge on Learned Image Compression (CLIC), 2020), and Tecnick, with Tecnick results reported in Appendix B. Generative compression is evaluated on Kodak and CLIC Pro Val, with CLIC Pro Val results reported in Appendix C. CCLS is evaluated on ImageNet-1K at 256×256, with 50 samples per category and 50,000 samples in total.

Training. Uni-HPCM-Base and Uni-HPCM-1B are fine-tuned with HCST from MSE-optimized HPCM-Base (Li et al., 2025a) and HPCM-1B (Li et al., 2025b) checkpoints, respectively. We use Flickr2W (Liu et al., 2020), random 256×256 crops, batch size 8, and Adam with $(\beta_1, \beta_2) = (0.9, 0.999)$. We update only the entropy model for 400K steps, then the full codec for 300K steps. Generator learning rates start at 10^{-4} and 5×10^{-5}, respectively, with stepwise decay to 10^{-6} per stage. Each update samples k uniformly from $\{1, \ldots, 11\}$. The fully coded branch uses MSE. Mixed-reconstruction loss weights are $(\lambda_1, \lambda_{\text{perc}}, \lambda_{\text{adv}}) = (1.0, 0.1, 0.005)$.

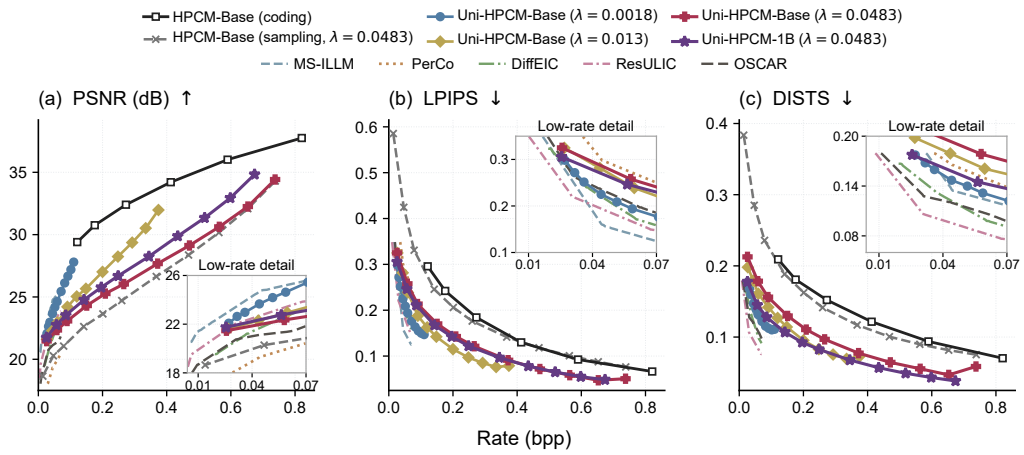


Figure 5: Generative compression performance on Kodak: (a) PSNR, (b) LPIPS, and (c) DISTS versus bitrate. Each PCS curve varies the number of sampled subsets k.

Evaluation settings. Uni-HPCM-Base uses six rate points, $\lambda \in \{0.0018, 0.0035, 0.0067, 0.013, 0.025, 0.0483\}$, while Uni-HPCM-1B uses $\lambda = 0.0483$. For CCLS, the reported Uni-HPCM-Base and Uni-HPCM-1B results use the checkpoint at $\lambda = 0.0483$. Appendix D.3 evaluates this rate-point choice for CCLS. PCS varies $k = 1, \ldots, 11$, with five samples per input for $k > 0$. PCS and CCLS use stage-wise temperatures $(\tau_1, \tau_2, \tau_3) = (0.2, 0.4, 0.6)$, supported by the ablations in Appendices D.1 and D.2. We compare with HPCM-Base coding and sampling, and published results for MS-ILLM (Muckley et al., 2023), PerCo (Careil et al., 2024), DiffEIC (Li et al., 2025c), ResULIC (Ke et al., 2025), and OSCAR (Guo et al., 2025). For CCLS, HPCM-Base, Uni-HPCM-Base, and Uni-HPCM-1B use the same conditioning images generated by DiT-XL/2 (Peebles & Xie, 2023) with a 25-step DPM solver and classifier-free guidance scale 4.0. Each Uni-HPCM model uses the same checkpoint for compression, PCS, and CCLS. All Uni-HPCM encoding and decoding times reported in the main text and appendix are measured on an NVIDIA 3090 GPU, including network inference and entropy encoding/decoding.

Metrics. Compression is measured by bitrate in bits per pixel (bpp), computed from actual bitstreams, and PSNR. For PCS, latent bitrate includes the hyper-latent and coded main-latent subsets. PSNR, LPIPS, and DISTS measure reconstruction fidelity and perceptual similarity. Metrics are averaged first over the five samples per input and then over the dataset. For CCLS, we report FID-50K using the ADM ImageNet-256 evaluator.

4.2 Evaluation of Three Unified Paradigms

We evaluate the three paradigms supported by a single Uni-HPCM model: reconstructive compression via full main-latent coding, generative compression via coding–sampling combinations, and class-conditional generation via full main-latent sampling, using the same checkpoint for all three.

4.2.1 Reconstructive Compression

We evaluate compression performance on Kodak and CLIC Pro Val. As shown in Fig. 4, Uni-HPCM-Base achieves comparable rate–distortion performance to HPCM-Base across six rate points, with improvements at high bitrates and slight degradation at lower bitrates. Uni-HPCM-1B further improves performance at $\lambda = 0.0483$. More results are provided in Appendix B.

4.2.2 Generative Compression

We evaluate generative compression performance through PCS. As shown in Fig. 5, Uni-HPCM-Base achieves lower LPIPS and DISTS than both HPCM-Base coding and direct sampling at comparable bitrates. Uni-HPCM-1B further improves perceptual quality, particularly in DISTS compared with Uni-HPCM-Base at $\lambda = 0.0483$.

Compared with MS-ILLM, PerCo, DiffEIC, ResULIC, and OSCAR, Uni-HPCM achieves competitive generative compression performance while covering a wider bitrate range. The low-rate Uni-HPCM-Base model outperforms PerCo in both LPIPS and DISTS over their common bitrate range. Notably, these results are obtained by fine-tuning a strong MSE-optimized compression

Table 1: Class-conditional generation on ImageNet-256. † marks codec-only parameters and sampling steps. Baseline configurations and teacher-conditioning settings are detailed in Appendix F.

Method	Params.	Sampling steps	FID↓
VQ-VAE-2 (Razavi et al., 2019)	–	5120	31.11
VQGAN (Esser et al., 2021)	227M	256	18.65
RQ-Transformer (Lee et al., 2022)	1.4B	64×4	8.71
LDM-8 (Rombach et al., 2022)	395M	200	15.51
LDM-4 (Rombach et al., 2022)	400M	250	10.56
MaskGIT (Chang et al., 2022)	227M	8	6.18
DiT-XL/2 (Peebles & Xie, 2023)	675M	250	2.27
RAR-XXL (Yu et al., 2025)	1.5B	256	1.48
xAR-H (Ren et al., 2025)	1.1B	50	1.24
HPCM-Base (w/o HCST)	68.5M†	11†	82.38
Uni-HPCM-Base	68.5M†	11†	11.76
Uni-HPCM-1B	1.0B†	11†	9.68

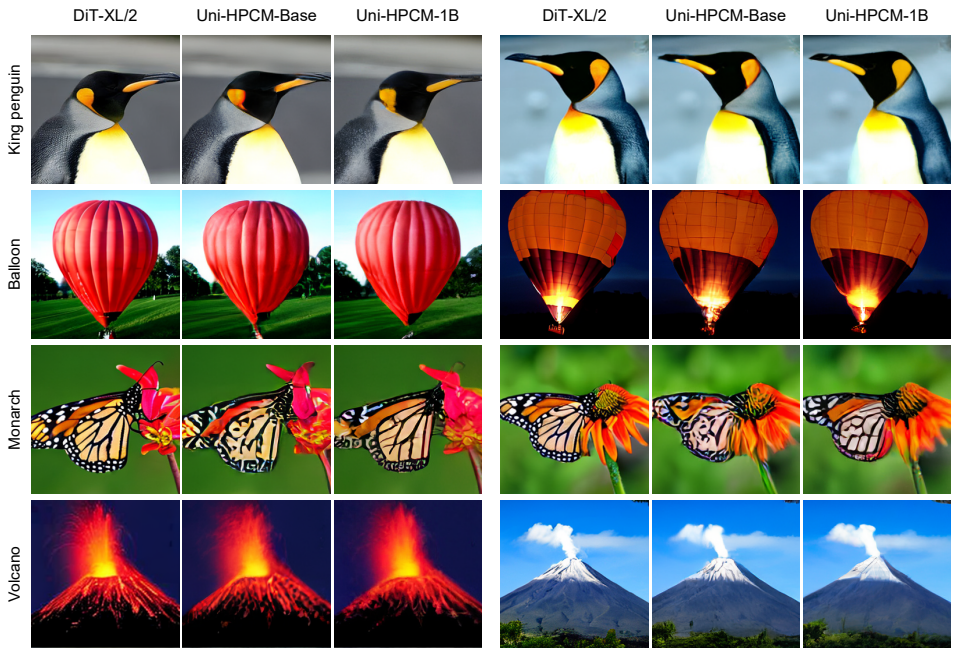


Figure 6: Class-conditional generation results. Each triplet shows a DiT-XL/2 conditioning image and samples from Uni-HPCM-Base and Uni-HPCM-1B.

model, without relying on a pretrained generative model. Additional PCS results on six rate points on Kodak and comparisons on CLIC Pro Val are provided in Appendix C.

4.2.3 Class-Conditional Image Generation

We evaluate generation performance through CCLS. As shown in Table 1, HCST reduces the FID of HPCM-Base from 82.38 to 11.76, and Uni-HPCM-1B further improves FID to 9.68. Compared with latent-space generators, Uni-HPCM-Base achieves lower FID than VQ-VAE-2 (Razavi et al., 2019), VQGAN (Esser et al., 2021), and LDM-8 (Rombach et al., 2022). The Uni-HPCM-Base codec has 68.5M parameters and samples the main latents in 11 progressive steps. Uni-HPCM-1B also outperforms LDM-4. Recent diffusion and autoregressive models, including DiT-XL/2 (Peebles & Xie, 2023), RAR-XXL (Yu et al., 2025), and xAR-H (Ren et al., 2025), achieve lower FID. Appendix E further compares DiT conditioning images, HPCM-Base sampling, and Uni-HPCM-Base mean completion and stochastic sampling in terms of generation quality, distribution coverage, and class consistency, indicating the benefits of HCST and stochastic sampling. Our results demonstrate that a model trained for compression can also achieve competitive conditional generation through its learned latent distributions while retaining strong compression performance. Figure 6 shows that

Table 2: HCST ablation on Kodak ($\lambda = 0.0483$). k denotes the number of sampled main-latent subsets. Bold marks the best PSNR, LPIPS, and DISTS values.

	$k=0$	$k=6$			$k=11$		
Training	PSNR↑	PSNR↑	LPIPS↓	DISTS↓	PSNR↑	LPIPS↓	DISTS↓
No HCST	37.79	24.71	0.176	0.141	18.65	0.585	0.384
RD only	37.81	24.69	0.175	0.140	18.65	0.583	0.381
Entropy only	37.76	25.63	0.142	0.118	20.63	0.387	0.243
Transform only	Diverged (NaN loss)						
Direct joint	Diverged (NaN loss)						
Two-stage HCST (Ours)	**37.84**	**26.04**	**0.121**	**0.097**	**21.46**	**0.325**	**0.213**

Uni-HPCM models preserve recognizable class content while generating varied details. Additional samples are provided in Appendix G.

4.3 Ablation Study

We compare HCST training strategies on HPCM-Base at $\lambda = 0.0483$, under the same settings and total training budget. RD-only fine-tuning follows the same two-stage schedule using only $\mathcal{L}_{\mathrm{RD}}$. Entropy-only training updates the hyperprior and progressive entropy model with g_a and g_s fixed. Transform-only training updates g_a and g_s alone. Direct joint training updates the full codec throughout. Two-stage HCST first updates the entropy-model components, then the full codec. Table 2 shows that, at the same rate point, two-stage HCST improves PSNR, LPIPS, and DISTS over other settings at $k = 6$ and 11, while maintaining comparable compression performance at $k = 0$. RD-only fine-tuning yields nearly unchanged sampling performance, supporting the benefit of HCST beyond additional RD training. Transform-only and direct joint training diverge, supporting entropy-model adaptation before joint fine-tuning. Additional ablations are provided in Appendix D.

4.4 Complexity

Table 3 reports encoding and decoding costs. Uni-HPCM-Base with PCS, using $\lambda = 0.0483$ and $k = 6$, requires less decoding computation than the compared generative codecs. By reusing HPCM's progressive context model and synthesis transform, PCS maintains comparable computational complexity to HPCM. Baseline runtimes are reported under their original evaluation settings. For standard compression, Table 4 in Appendix B reports forward-pass computation and coding times.

Table 3: Complexity of generative compression.

	kMACs/pixel		Time (s)	
Method	Enc.	Dec.	Enc.	Dec.
MS-ILLM	147	1013	0.064	0.070
PerCo	2666	$>$ 10,000	0.080	2.551
DiffEIC	2240	70,610	0.210	4.661
ResULIC	2080	8790	0.096	0.428
OSCAR	2110	6070	0.065	0.163
Uni-HPCM-Base (PCS)	748	742	0.048	0.044

5 Discussion

Uni-HPCM demonstrates the potential for compression and generation to benefit from shared distribution modeling. Increasing model capacity improves both rate–distortion performance and generation quality, suggesting that the two tasks can benefit from advances in a shared model. These findings motivate further exploration of joint improvements in compression and generation.

Our study is limited to the HPCM model family and a limited range of model sizes. Exploring other architectures and larger models is an important direction for assessing the broader potential of this synergy. Future work can build on this framework to investigate how compression and generation can better support each other under joint modeling and optimization.

6 Conclusion

We presented Uni-HPCM, a unified framework that connects entropy coding and latent sampling through shared conditional distributions. Hybrid Coding–Sampling Training enables a single model to support image compression, conditional image generation, and generative compression. Adjusting the balance between coded and sampled latent subsets provides flexible control over bitrate and reconstruction quality. Experiments demonstrate competitive rate–distortion performance, effective class-conditional generation, and favorable perceptual quality in generative compression. Beyond these capabilities, this work demonstrates the possibility of compression–generation synergy and provides a foundation for exploring their joint development.

AI USE STATEMENT

Generative AI tools were used to assist with literature discovery, experimental table organization, language polishing, sentence-level rewriting, comparison of alternative prose versions, and LaTeX template organization. The authors reviewed all AI-assisted edits, checked the mathematical definitions, experimental settings, numerical values, and citation keys against the source manuscript, and take responsibility for the final content.

REFERENCES

Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rkcQFMZRb.

Yochai Blau and Tomer Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 675–685. PMLR, 2019. URL https://proceedings.mlr.press/v97/blau19a.html.

Marlène Careil, Matthew J. Muckley, Jakob Verbeek, and Stéphane Lathuilière. Towards image compression with perfect realism at ultra-low bitrates. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ktdETU9JBg.

Challenge on Learned Image Compression (CLIC). Clic professional validation set, 2020. Dataset used in the Challenge on Learned Image Compression.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. MaskGIT: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11315–11325, 2022.

Tong Chen, Haojie Liu, Zhan Ma, Qiu Shen, Xun Cao, and Yao Wang. End-to-end learnt image compression via non-local attention optimization and improved context modeling. *IEEE Transactions on Image Processing*, 30:3179–3191, 2021. doi: 10.1109/TIP.2021.3058615.

Yunuo Chen, Qian Li, Bing He, Donghui Feng, Ronghua Wu, Qi Wang, Li Song, Guo Lu, and Wenjun Zhang. S2CFormer: Revisiting the RD-latency trade-off in transformer-based learned image compression. *arXiv preprint arXiv:2502.00700*, 2025. URL https://arxiv.org/pdf/2502.00700v3.

Yunuo Chen, Bing He, Zezheng Lyu, Hongwei Hu, Qunshan Gu, Yuan Tian, and Guo Lu. Adaptive learned image compression with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12150–12161, 2026a.

Yunuo Chen, Zezheng Lyu, Bing He, Hongwei Hu, Qi Wang, Yuan Tian, Li Song, Wenjun Zhang, and Guo Lu. Content-aware mamba for learned image compression. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=WwDNiisZQm.

Yunuo Chen, Chuqin Zhou, Jiangchuan Li, Xiaoyue Ling, Bing He, Jincheng Dai, Li Song, and Guo Lu. Next-frame decoding for ultra-low-bitrate image compression with video diffusion priors. In *European Conference on Computer Vision*, 2026c. URL https://arxiv.org/abs/2603.15129.

Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro 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*, pp. 7939–7948, 2020.

Eastman Kodak. Kodak lossless true color image suite. http://r0k.us/graphics/kodak/.

Alaaeldin El-Nouby, Matthew J. Muckley, Karen Ullrich, Ivan Laptev, Jakob Verbeek, and Hervé Jégou. Image compression with product quantized masked image modeling. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=Z2L5d9ay4B.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. URL https://arxiv.org/abs/2012.09841.

Jinpei Guo, Yifei Ji, Zheng Chen, Kai Liu, Min Liu, Wang Rao, Wenbo Li, Yong Guo, and Yulun Zhang. OSCAR: One-step diffusion codec across multiple bit-rates. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/7b372097a499ce99a4b1e41feb79493a-Abstract-Conference.html.

Dailan He, Yaoyan Zheng, Baocheng Sun, Yan Wang, and Hongwei Qin. Checkerboard context model for efficient learned image compression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14771–14780, 2021.

Dailan He, Ziming Yang, Weikun Peng, Rui Ma, Hongwei Qin, and Yan Wang. ELIC: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5718–5727, 2022.

Zhaoyang Jia, Jiahao Li, Bin Li, Houqiang Li, and Yan Lu. Generative latent coding for ultra-low bitrate image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26088–26098, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Jia_Generative_Latent_Coding_for_Ultra-Low_Bitrate_Image_Compression_CVPR_2024_paper.html.

Wei Jiang and Ronggang Wang. MLIC++: Linear complexity multi-reference entropy modeling for learned image compression. In *ICML 2023 Workshop Neural Compression: From Information Theory to Applications*, 2023. URL https://openreview.net/forum?id=hxIpcSoz2t.

Anle Ke, Xu Zhang, Tong Chen, Ming Lu, Chao Zhou, Jiawen Gu, and Zhan Ma. Ultra lowrate image compression with semantic residual coding and compression-aware diffusion. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 29626–29650. PMLR, 2025. URL https://proceedings.mlr.press/v267/ke25c.html.

Jun-Hyuk Kim, Seungeon Kim, Won-Hee Lee, and Dokwan Oh. Diversify, contextualize, and adapt: Efficient entropy modeling for neural image codec. In *Advances in Neural Information Processing Systems*, volume 37, pp. 45956–45974, 2024.

Haowei Kuang, Wenhan Yang, Zongming Guo, and Jiaying Liu. Cross-granularity online optimization with masked compensated information for learned image compression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16514–16523, 2025.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. URL https://arxiv.org/abs/2203.01941.

Han Li, Shaohui Li, Wenrui Dai, Chenglin Li, Junni Zou, and Hongkai Xiong. Frequency-aware transformer for learned image compression. In *International Conference on Learning Representations (ICLR)*, 2024a.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In *Advances in Neural Information Processing Systems*, volume 37, 2024b. doi: 10.52202/079017-1797.

Yuqi Li, Haotian Zhang, and Dong Liu. Flexible coding order for learned image compression. In *IEEE International Conference on Visual Communications and Image Processing (VCIP)*, pp. 1–5, 2023.

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

Yuqi Li, Haotian Zhang, Li Li, Dong Liu, and Feng Wu. Scaling learned image compression models up to 1 billion. *arXiv preprint arXiv:2508.09075*, 2025b. URL https://arxiv.org/abs/2508.09075.

Zhiyuan Li, Yanhui Zhou, Hao Wei, Chenyang Ge, and Jingwen Jiang. Toward extreme image compression with latent feature guidance and diffusion prior. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(1):888–899, 2025c. doi: 10.1109/TCSVT.2024.3455576. URL https://github.com/huai-chang/DiffEIC.

Jiaheng Liu, Guo Lu, Zhihao Hu, and Dong Xu. A unified end-to-end framework for efficient deep image compression. *arXiv preprint arXiv:2002.03370*, 2020. URL https://arxiv.org/abs/2002.03370.

Jinming Liu, Heming Sun, and Jiro Katto. Learned image compression with mixed Transformer-CNN architectures. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14388–14397, 2023. doi: 10.1109/CVPR52729.2023.01383.

Jingbo Lu, Leheng Zhang, Xingyu Zhou, Mu Li, Wen Li, and Shuhang Gu. Learned image compression with dictionary-based entropy model. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 12850–12859, June 2025.

David Minnen and Saurabh Singh. Channel-wise autoregressive entropy models for learned image compression. In *IEEE International Conference on Image Processing (ICIP)*, pp. 3339–3343, 2020. doi: 10.1109/ICIP40778.2020.9190935.

David Minnen, Johannes Ballé, and George D. Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/53edebc543333dfbf7c5933af792c9c4-Abstract.html.

Matthew J. Muckley, Alaaeldin El-Nouby, Karen Ullrich, Hervé Jégou, and Jakob Verbeek. Improving statistical fidelity for neural image compression with implicit local likelihood models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 25426–25443. PMLR, 2023. URL https://proceedings.mlr.press/v202/muckley23a.html.

Guy Ohayon, Hila Manor, Tomer Michaeli, and Michael Elad. Compressed image generation with denoising diffusion codebook models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 47044–47089. PMLR, 2025. URL https://proceedings.mlr.press/v267/ohayon25a.html.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Yichen Qian, Ming Lin, Xiuyu Sun, Zhiyu Tan, and Rong Jin. Entroformer: A Transformer-based entropy model for learned image compression. In *International Conference on Learning Representations (ICLR)*, 2022.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://arxiv.org/abs/1906.00446.

Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Beyond next-token: Next-X prediction for autoregressive visual generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. URL https://arxiv.org/abs/2502.20388.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. URL https://arxiv.org/abs/2112.10752.

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1711.00937.

Jeremy Vonderfecht and Feng Liu. Lossy compression with pretrained diffusion models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=raUnLe0Z04.

Naifu Xue, Qi Mao, Zijian Wang, Yuan Zhang, and Siwei Ma. Unifying generation and compression: Ultra-low bitrate image coding via multi-stage transformer. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2024.

Naifu Xue, Zhaoyang Jia, Jiahao Li, Bin Li, Yuan Zhang, and Yan Lu. DLF: Extreme image compression with dual-generative latent fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19227–19236, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Xue_DLF_Extreme_Image_Compression_with_Dual-generative_Latent_Fusion_ICCV_2025_paper.html.

Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Randomized autoregressive visual generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. URL https://arxiv.org/abs/2411.00776.

Haotian Zhang, Li Li, and Dong Liu. Generalized Gaussian model for learned image compression. *IEEE Transactions on Image Processing*, 34:1950–1965, 2025a. doi: 10.1109/TIP.2025.3550013.

Tianyu Zhang, Xin Luo, Li Li, and Dong Liu. StableCodec: Taming one-step diffusion for extreme image compression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17379–17389, 2025b. doi: 10.1109/ICCV51701.2025.01614. URL https://openaccess.thecvf.com/content/ICCV2025/html/Zhang_StableCodec_Taming_One-Step_Diffusion_for_Extreme_Image_Compression_ICCV_2025_paper.html.

A Analysis of Shared Distribution Modeling

This appendix derives the connection between likelihood-based rate estimation and latent sampling through the shared Gaussian density. The analysis uses the continuous training objective with additive uniform noise.

A.1 Conditional Densities for Rate Estimation

Fix a context and let Y denote a source latent. Write ν for the density of $\mathcal{U}(-\frac{1}{2}, \frac{1}{2})$. Adding independent noise $\eta \sim \nu$ gives $V = Y + \eta$, whose density is denoted by $\bar{q}$. The context model predicts $f_\theta(u) = \mathcal{N}(u; \mu, \sigma^2)$ with $\sigma > 0$. Under the same noise relaxation, its likelihood is

$$\bar{p}_\theta(v) = (f_\theta * \nu)(v) = \int f_\theta(u)\nu(v-u)\,\mathrm{d}u. \tag{11}$$

If $U \sim f_\theta$ and $\eta' \sim \nu$ are independent, then $W = U + \eta'$ has density $p_W = \bar{p}_\theta$, establishing Eq. (4). Thus, $\bar{q}$ and $\bar{p}_\theta$ describe source and model samples under the same noise relaxation. The additional noise on model samples defines this comparison; the generation network receives the sampled latent values themselves.

We assume finite differential entropies and divergences, with positive model densities wherever the source density is positive. All logarithms in entropies and divergences are base two. The identities below hold for each fixed context and can be averaged over a specified distribution of contexts.

A.2 Shared Distribution-Fitting Objective

The training rate estimate is the cross entropy between $\bar{q}$ and $\bar{p}_\theta$. Subtracting the source differential entropy gives

$$\begin{aligned}\mathcal{R}_{\text{train}} - h(\bar{q}) &= -\int \bar{q}(v)\log_2 \bar{p}_\theta(v)\,\mathrm{d}v + \int \bar{q}(v)\log_2 \bar{q}(v)\,\mathrm{d}v \\ &= \int \bar{q}(v)\log_2 \frac{\bar{q}(v)}{\bar{p}_\theta(v)}\,\mathrm{d}v = D_{\mathrm{KL}}(\bar{q}\|\bar{p}_\theta).\end{aligned} \tag{12}$$

Together with Eq. (4), this proves Eq. (5). For fixed $\bar{q}$, reducing the rate objective reduces the mismatch between the noisy source latents and the correspondingly smoothed model samples. The Gaussian density f_θ supplies both this training likelihood and the base sampling distribution. HCST further supervises the images produced along the actual sampling trajectories.

A.3 Temperature-Adjusted Sampling

For temperature $\tau > 0$, define the sampling density and its smoothed counterpart as

$$r_{\theta,\tau}(u) = \mathcal{N}(u; \mu, \tau^2\sigma^2), \qquad \bar{s}_{\theta,\tau} = r_{\theta,\tau} * \nu. \tag{13}$$

Inserting $\bar{p}_\theta$ into the likelihood ratio gives

$$\begin{aligned}D_{\mathrm{KL}}(\bar{q}\|\bar{s}_{\theta,\tau}) &= \int \bar{q}(v)\log_2 \frac{\bar{q}(v)}{\bar{s}_{\theta,\tau}(v)}\,\mathrm{d}v \\ &= D_{\mathrm{KL}}(\bar{q}\|\bar{p}_\theta) + \mathbb{E}_{\bar{q}}\left[\log_2 \frac{\bar{p}_\theta(V)}{\bar{s}_{\theta,\tau}(V)}\right].\end{aligned} \tag{14}$$

At $\tau = 1$, $r_{\theta,1} = f_\theta$ and $\bar{s}_{\theta,1} = \bar{p}_\theta$, so the correction term vanishes. For other temperatures, the correction depends on the source and the predicted distribution. Uni-HPCM uses the same stage-wise temperatures during training and inference, with HCST supervising the images produced along the resulting sampling trajectories.

B Additional Compression Results and Compression Complexity

Table 4 compares fully coded compression performance and complexity with representative learned image compression methods. Using VTM-22.0 as the anchor, Uni-HPCM-Base achieves PSNR BD-rate reductions of 13.45%, 11.54%, and 14.92% on Kodak, CLIC Pro Val, and Tecnick, respectively. It outperforms TCM on all three datasets while requiring fewer parameters and lower computation.

Table 4: Compression performance and complexity comparison. VTM-22.0 serves as the anchor for PSNR BD-rate calculation. The best available compression performance is in **bold**.

Model	Enc. T (ms)	Dec. T (ms)	kMACs /pixel	Params (M)	BD-rate Kodak	BD-rate CLIC Pro Val	BD-rate Tecnick
TCM (CVPR'23) (Liu et al., 2023)	103	122	1823.58	76.57	-10.70%	-8.32%	-11.84%
MLIC++ (NCW ICML'23) (Jiang & Wang, 2023)	90	91	1282.81	116.72	-15.15%	-14.05%	-17.90%
DCAE (CVPR'25) (Lu et al., 2025)	83	82	940.40	119.4	-15.36%	-15.40%	-20.35%
HPCM-Base (ICCV'25) (Li et al., 2025a)	79	85	918.57	68.5	-15.31%	-14.23%	-18.16%
HPCM-Large (ICCV'25) (Li et al., 2025a)	93	94	1261.29	89.71	**-19.19%**	**-18.37%**	-22.20%
GLIC (CVPR'26) (Chen et al., 2026a)	110	102	1285.55	69.37	-18.38%	-18.07%	**-22.70%**
Uni-HPCM-Base (Ours)	81	84	918.57	68.5	-13.45%	-11.54%	-14.92%

Table 5: Fully coded compression on Kodak. Bold marks the lower rate–distortion objective at each λ.

λ	Model	bpp↓	PSNR↑	$\mathcal{L}_{\mathrm{RD}}\downarrow$
0.0483	HPCM-Base	0.8210	37.79	1.3647
0.0483	Uni-HPCM-Base	0.8197	37.84	**1.3564**
0.013	HPCM-Base	0.4125	34.20	**0.7559**
0.013	Uni-HPCM-Base	0.4123	34.17	0.7610
0.0018	HPCM-Base	0.1211	29.40	**0.2715**
0.0018	Uni-HPCM-Base	0.1185	29.24	0.2772

Uni-HPCM-Base has 68.5M parameters and requires 918.57 forward kMACs/pixel, which is the same as HPCM-Base. These results show that Uni-HPCM retains competitive compression performance while extending the same codec to generative compression and class-conditional generation, with no increase in parameter count or computation for fully coded compression.

Table 5 compares compression performance on Kodak at three rate points. At $\lambda = 0.0483$, Uni-HPCM-Base improves PSNR from 37.79 to 37.84 dB while reducing bitrate from 0.8210 to 0.8197 bpp. At $\lambda = 0.013$ and 0.0018, PSNR decreases by 0.03 and 0.16 dB, respectively, with slightly lower bitrates. These results show that Uni-HPCM-Base improves compression performance at the high-rate point, with slight degradation at the medium- and low-rate points, retaining comparable compression performance overall.

C Additional Generative Compression Results

C.1 PCS across Six Rate–Distortion Weights

Figure 7 presents PCS results on Kodak for all six rate–distortion weights of Uni-HPCM-Base. Each curve uses a fixed λ and varies the number of sampled subsets k. The six models cover different bitrate ranges, while PCS adjusts the bitrate within each model. As shown in the low-rate insets, models trained with smaller λ generally achieve lower LPIPS and DISTS at comparable bitrates. These results show that the training weight and coding–sampling balance provide complementary control over bitrate and perceptual quality.

C.2 Generative Compression on CLIC Pro Val

Figure 8 compares generative compression performance on CLIC Pro Val. Uni-HPCM-Base achieves lower LPIPS and DISTS than HPCM-Base coding and direct sampling at comparable bitrates, consistent with the Kodak results. Uni-HPCM-1B further improves DISTS over Uni-HPCM-Base at $\lambda = 0.0483$ in the higher-bitrate region. Compared with existing generative compression methods, Uni-HPCM provides competitive perceptual quality over a broad bitrate range. The low-rate Uni-HPCM-Base model achieves lower LPIPS and DISTS than PerCo over their common bitrate range.

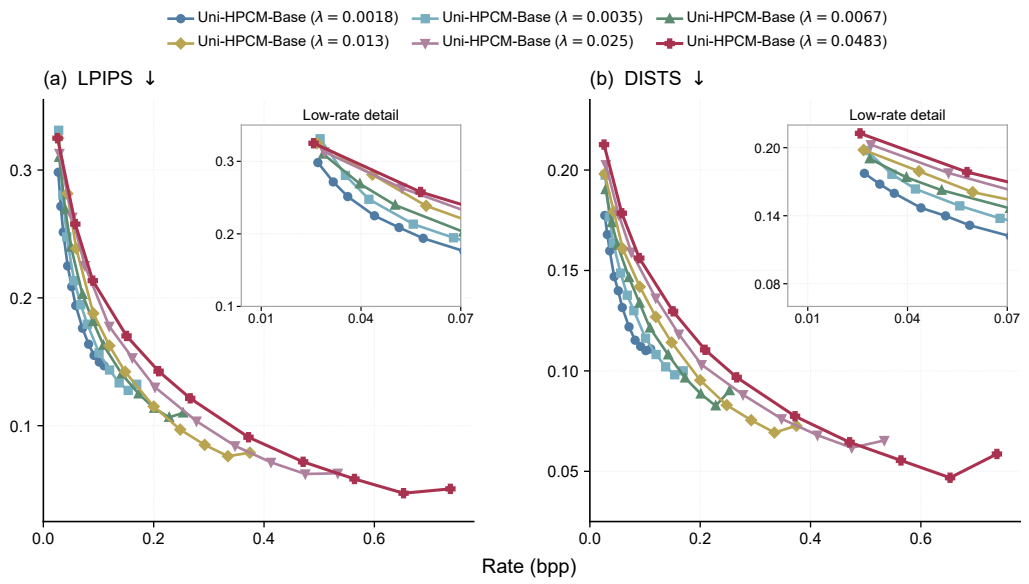


Figure 7: PCS performance across six rate–distortion weights on Kodak: (a) LPIPS and (b) DISTS versus bitrate. Each Uni-HPCM-Base curve uses a fixed λ and varies the number of sampled subsets k.

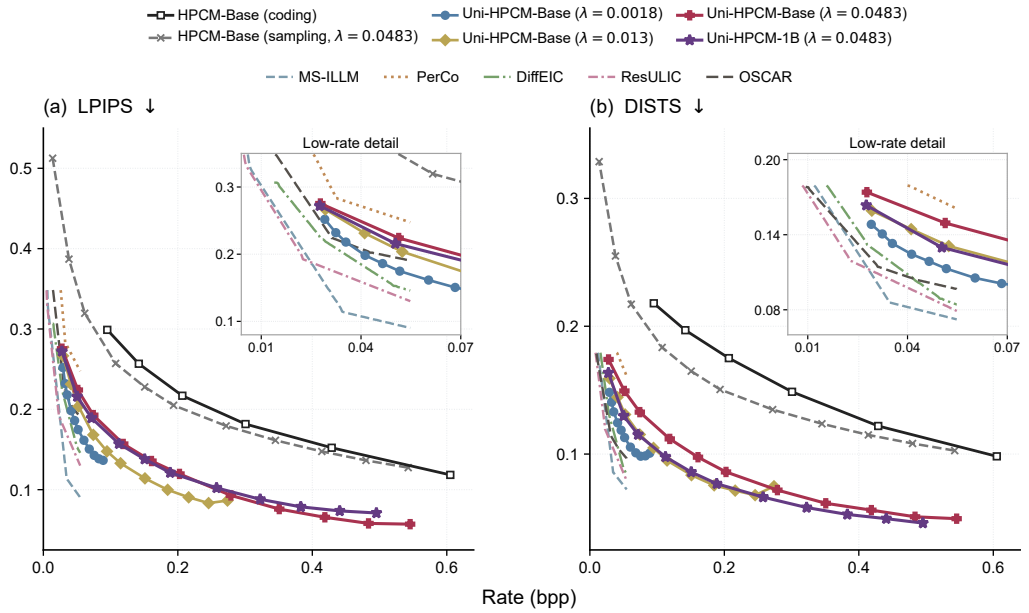


Figure 8: Generative compression performance on CLIC Pro Val: (a) LPIPS and (b) DISTS versus bitrate.

D Additional Ablation Study

D.1 Temperature Configuration in PCS

We compare four stage-wise temperature settings on Kodak using Uni-HPCM-Base at $\lambda = 0.0483$. Each model uses the same temperatures during training and inference, with k sampled from $\{1, \ldots, 11\}$ during training. Table 6 reports BD-rate relative to $(0.2, 0.4, 0.6)$. As shown in Ta-

Table 6: PCS temperature ablation on Kodak. BD-rate is relative to $(0.2, 0.4, 0.6)$.

(τ_1, τ_2, τ_3)	BD-rate (%)↓		
	PSNR	LPIPS	DISTS
$(0.2, 0.4, 0.6)$ (default)	**0.00**	**0.00**	**0.00**
$(0.1, 0.3, 0.5)$	$+0.61$	$+3.26$	$+3.83$
$(0.3, 0.5, 0.7)$	$+2.48$	$+5.88$	$+5.07$
$(0.6, 0.4, 0.2)$	$+0.84$	$+0.46$	$+2.02$

Table 7: CCLS temperature ablation on ImageNet-256. Bold marks the lowest FID.

(τ_1, τ_2, τ_3)	FID↓
$(0.2, 0.4, 0.6)$ (default)	**11.76**
$(0.1, 0.3, 0.5)$	12.57
$(0.3, 0.5, 0.7)$	12.17
$(0.6, 0.4, 0.2)$	12.53

ble 6, $(0.2, 0.4, 0.6)$ achieves the best BD-rate for all three metrics. Lowering, raising, or reversing the temperatures increases BD-rate. These results support the default temperature setting for PCS.

D.2 Temperature Configuration in CCLS

We compare the same four temperature settings for CCLS using Uni-HPCM-Base at $\lambda = 0.0483$. Each model uses the same temperature during training and inference. As shown in Table 7, $(0.2, 0.4, 0.6)$ achieves the lowest FID of 11.76, compared with 12.17–12.57 for the other settings. This agrees with the PCS results and supports the default temperatures for both sampling modes.

D.3 Rate-Point Selection in CCLS

In this paper, we use models trained with $\lambda = 0.0483$ for CCLS by default. We further evaluate Uni-HPCM-Base at two additional rate points on ImageNet-256. As shown in Table 8, $\lambda = 0.0483$ achieves the lowest FID of 11.76, compared with 21.01 at $\lambda = 0.013$ and 25.57 at $\lambda = 0.0018$, supporting our default choice. This advantage may because the high-rate model retains richer information, providing more detailed guidance for generation.

E Additional Evaluation and Results on Class-Conditional Generation

We further evaluate CCLS in terms of generation quality, distribution coverage, and class consistency. All codec settings use the same $50{,}000$ conditioning images generated by DiT-XL/2 with a 25-step DPM solver and classifier-free guidance scale 4.0 (675M-parameter DiT-XL/2 teacher with 25 solver steps). For Uni-HPCM-Base, we use the checkpoint at $\lambda = 0.0483$ and compare stochastic sampling with deterministic mean completion. Mean completion sets all inference temperatures to zero and sequentially predicts all 11 main-latent subsets, using each predicted mean as context for subsequent subsets. Stochastic sampling uses the default temperatures $(0.2, 0.4, 0.6)$.

We compute FID, Precision, and Recall using the ADM ImageNet-256 evaluator with Inception-v3 features and the real ImageNet reference set. Precision measures sample fidelity, while Recall measures coverage of the real image distribution across the generated set. Class consistency is measured by target-class Top-1 accuracy using a fixed torchvision ResNet-50 with `IMAGENET1K_V1` weights.

Table 9 shows that HCST improves all four metrics over direct HPCM-Base sampling. With the same Uni-HPCM-Base checkpoint, stochastic sampling also outperforms mean completion on all four metrics. These results show that HCST enables the compression model to generate images through full main-latent sampling while substantially improving fidelity, coverage, and class consistency. Compared with the DiT conditioning images, Uni-HPCM-Base achieves lower FID and

Table 8: CCLS performance across rate–distortion weights on ImageNet-256 using Uni-HPCM-Base.

λ	FID↓
0.0483 (default)	11.76
0.013	21.01
0.0018	25.57

Table 9: CCLS comparisons on ImageNet-256 using the same DiT conditioning images. Mean completion and stochastic sampling use the same Uni-HPCM-Base checkpoint. Acc. denotes target-class Top-1 accuracy. Bold indicates the best value in each column.

Setting	FID↓	Precision↑	Recall↑	Acc.↑
DiT conditioning images	16.19	**0.8955**	0.3451	**95.77%**
HPCM-Base sampling (No HCST)	82.38	0.3950	0.3342	21.11%
Uni-HPCM-Base mean completion	18.24	0.5321	0.3359	58.79%
Uni-HPCM-Base stochastic sampling	**11.76**	0.8925	**0.4525**	92.72%

higher Recall, but slightly lower Precision and class accuracy. This reflects a trade-off between distribution coverage, sample fidelity, and semantic preservation. These results show that codec sampling expands distribution coverage under teacher-provided semantic conditioning.

F Class-Conditional Generation Evaluation Settings

In Table 1, VQ-VAE-2 and the 227M-parameter VQGAN results are from Chang et al. (2022). RQ-Transformer uses the public 1.4B checkpoint without rejection sampling. Its $8 \times 8 \times 4$ code map has 64 spatial positions with four sequential depth predictions per position. Both LDM variants use unguided sampling.

HPCM parameter counts refer to the codec, and the 11 steps refer to progressive main-latent sampling. CCLS also uses a fixed DiT-XL/2 teacher with 675M parameters and 25 solver steps to generate conditioning images. Parameter counts and sampling steps for the other methods follow their reported configurations.

G Additional Class-Conditional Generation Results

Figure 9 presents additional CCLS samples across nine ImageNet classes. Both Uni-HPCM-Base and Uni-HPCM-1B preserve recognizable class content and the overall structure of the conditioning images, while producing variations in local details such as plumage, flower centers, and mushroom textures. These examples further illustrate class-conditional generation using the same models that support compression and generative compression.

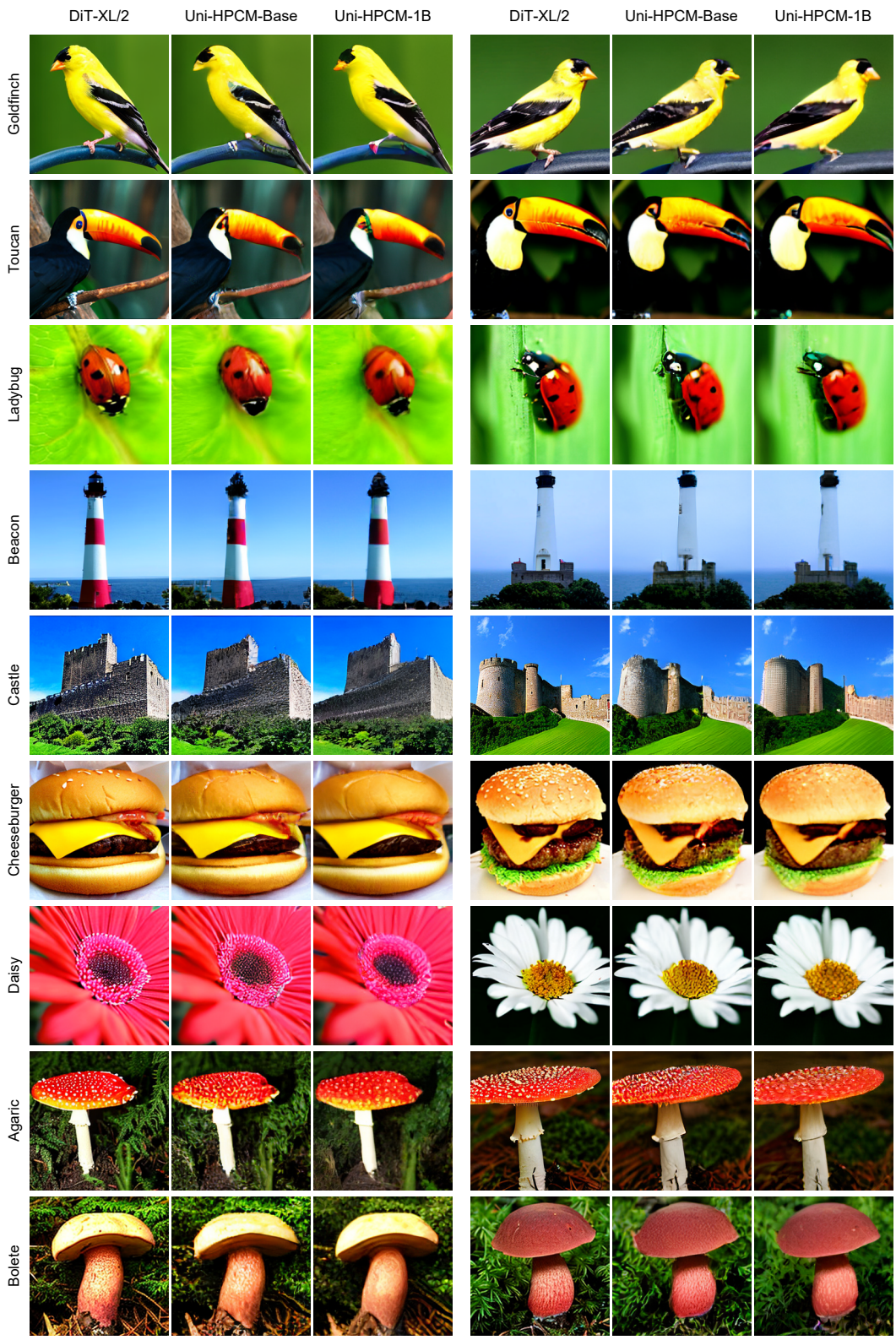


Figure 9: Additional class-conditional generation results on ImageNet-256.