PerQ: Inverse Generative Modeling for Neural Image Compression via Quantization Error Compensation

Tianhao Peng[1], **Ho Man Kwan**[1], **Fan Zhang**[1], **Shan Liu**[2], **Dave Bull**[1]
[1]Visual Information Lab, University of Bristol, UK
[2]Tencent Media Lab, Palo Alto, USA
{tianhao.peng, hm.kwan, fan.zhang, dave.bull}@bristol.ac.uk
shanl@global.tencent.com

Abstract

Neural image codecs face a fundamental trade-off between pixel fidelity and perceptual quality. Distortion-optimized codecs preserve pixel-wise fidelity but produce blurred reconstructions at low bitrates, whereas generative codecs leverage learned priors to improve perceptual quality but tend to exhibit saturating PSNR at higher bitrates and require separate models for different bitrate targets. We propose **PerQ**, which reframes perceptual reconstruction as inverse generative modeling with bounded support. Specifically, we define the quantization cell as the set of latents that yield the same transmitted symbols and derive a transport bound showing that reconstructions from this cell remain within a rate-determined radius of the codec output. This motivates performing generative reconstruction within the codec-induced quantization cell, rather than sampling from an unconstrained image prior. PerQ implements this principle by attaching a rate-conditioned flow-matching compensator to a frozen, multi-rate distortion-optimized backbone, enabling perceptual enhancement while preserving the backbone's coding performance. Experiments on Kodak and CLIC2020 show that PerQ achieves perceptual performance comparable to generative codecs such as MS-ILLM in LPIPS and FID, while maintaining competitive PSNR relative to distortion-optimized codecs such as ELIC. A single model checkpoint supports a wide bitrate range with fast decoding, and the compensator provides consistent gains across three different backbones. The implementation is available at `https://jasminepp.github.io/PerQ/`.

1 Introduction

Image compression has long been a fundamental research topic in signal processing and a core component of modern visual data infrastructure [11]. Conventional image codecs such as JPEG [55], BPG [5], and the intra modes of AV1 [19] and VVC [10], have been widely adopted over the past decades. More recently, end-to-end learned image compression [3, 4, 44] have achieved competitive or superior rate-distortion performance compared with conventional codecs based on pixel-wise metrics, such as PSNR [57]. However, optimizing pixel-wise distortion does not necessarily yield perceptually pleasing reconstructions, especially at low bitrates, where severe bitrate constraints limit the preservation of fine details and high-frequency image content, often resulting in visually blurred reconstructions. This limitation has motivated growing interest in perceptual image compression, which prioritizes perceptual fidelity over pixel-wise similarity and has led to the development of perceptual and generative image codecs.

Recent generative codecs have demonstrated promising perceptual quality at extremely low bitrates [27, 62]. However, extending this perceptual advantage to practical deployment remains challenging. (i) *Pixel-wise fidelity*: downstream applications often require reconstructions to remain faithful to the source pixels, but most generative codecs, whose reconstructions are dominated by a generative prior,

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

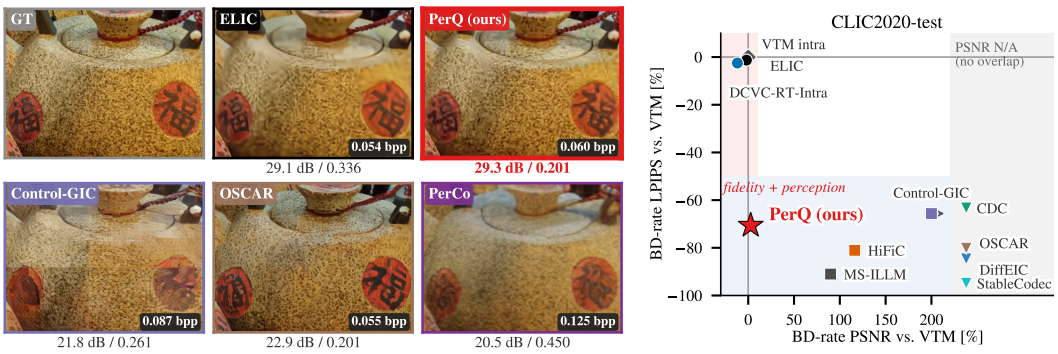

Figure 1: **Balance between Perceptual quality and pixel fidelity.** (Left) Visual comparison at comparable bitrates, with PSNR (dB) / LPIPS. (Right) BD-rate of PSNR against BD-rate of LPIPS on CLIC2020-test, both relative to VTM intra, lower is better).

tend to plateau on PSNR even as the bitrate becomes high, since additional bits do not bring the prior closer to the source pixels [43, 46]. (ii) *Bitrate coverage*: practical deployment would benefit from a single model that supports a broad range of bitrates, avoiding the need to store and maintain separate models for different rates. However, most existing perceptual codecs require separate models for different target rates [43, 63]. Although a few approaches support multiple bitrates with a single model [12, 18], their operating ranges are typically concentrated in a relatively narrow low-bitrate window. The results of existing works shown in Table 1 reveal that none of them delivers high perceptual quality, competitive pixel fidelity, and wide bitrate coverage with a single trained model.

We ask, *Can these competing objectives be reconciled within a single codec without sacrificing the advantages of distortion-optimized compression?* We propose **PerQ**, an image codec that reframes perceptual reconstruction as *inverse generative modeling with bounded support*, leveraging the quantization structure of an existing codec to constrain perceptual refinement. We characterize the set of latent representations that yield the same transmitted symbols as a *quantization cell* and derive a *transport bound* relating the quantization step to a rate-determined radius in pixel space. Unlike prior approaches that augment distortion-optimized backbones with perceptual losses [21, 31, 43] and use learned generative priors to explore the image manifold, PerQ employs a flow-matching-based compensator as a decoder-side plug-in to improve perceptual quality while keeping the refined latent within this bounded region. Integrated into the decoder, the compensator leaves the bitstream unchanged and supports the backbone's existing multi-rate operating range with a single model, while retaining fast decoding. Importantly, the derived transport bound guarantees that the refined reconstruction remains within a rate-determined pixel-space radius of the backbone's output, even with perceptual optimization. This design brings together the perceptual advantages of generative codecs, the pixel fidelity of distortion-optimized codecs, and broad bitrate coverage within a single model (Figure 1). The main contributions of this work are:

- **(C1) Inverse generative modeling with bounded support.** We formulate perceptual reconstruction as inverse generative modeling over a rate-determined bounded support, characterized by a transport bound (§3.1) that relates the quantization step to a pixel-space radius around the output of a multi-rate distortion-optimized codec.
- **(C2) A rate-conditioned flow-matching compensator design** is designed (§3.3) for perceptual reconstruction within the bounded support. A single checkpoint supports multiple backbone rates, while the decoder-side design preserves the bitstream and generalizes across different backbones.
- **(C3) PerQ: A perceptual-fidelity-balanced image codec** integrates a bounded-support compensator with a frozen distortion-optimized backbone, bridging the gap between generative perceptual codecs and distortion-oriented codecs by combining strong perceptual quality with competitive pixel fidelity in a single, multi-rate model.

Extensive experiments on Kodak and CLIC2020-test (Table 1) show that PerQ retains most of the perceptual gains of strong generative codecs such as MS-ILLM while achieving competitive pixel fidelity against VVC (VTM intra), while enabling efficient four-step decoding at 86 ms/MP. Moreover, a single checkpoint covers nearly two orders of magnitude in bitrate, and the same compensator delivers consistent gains across two additional backbones. These results demonstrate that PerQ balances perceptual quality, pixel fidelity, and bitrate flexibility within a single, efficient codec.

Table 1: **Positioning of image codecs across compression paradigms.** Each row reports single-model multi-rate operation, decoding complexity and BD-rate against VTM intra on the Kodak [16] and CLIC2020-test [14] datasets.

Image codecs	Single-model multi-rate	Dec. time (ms/Mp) ↓	Dec. kMACs/px ↓	**Kodak**: BD-rate vs. VVC (%) ↓				**CLIC2020-test**: BD-rate vs. VVC (%) ↓			
				PSNR	LPIPS	DISTS	FID	PSNR	LPIPS	DISTS	FID
Conventional codec											
VTM intra [10]	–	–	–	0	0	0	0	0	0	0	0
VAE-based codecs											
ELIC [20]		206	332	−2.0	−2.1	+2.4	+7.3	−1.9	−1.3	+9.4	−13.8
EVC [56]	✓	57	291	+2.0	−9.0	−5.4	−4.0	+3.7	−8.7	−2.3	−6.2
DCVC-RT-Intra [25]	✓	46	362	−11.2	−8.7	−2.3	−1.8	−11.8	−2.5	+8.8	−30.1
GAN-based perceptual codecs											
MS-ILLM [46]		150	675	+85.9	−86.4	−82.9	−86.3	+89.9	−91.0	−90.7	−97.4
HiFiC [43]		150	546	+128.6	−69.8	−59.0	−64.3	+116.3	−81.1	−76.9	−83.0
Control-GIC [34]	✓	416	6299	+372.4	−74.0	−68.5	−69.7	+1045.8	−65.6	−73.8	−85.2
Diffusion-based perceptual codecs											
PerCo [12]		6822 (20-step)	30930	N/A	−64.1	−84.4	−92.9	N/A	N/A	N/A	−94.3
DiffEIC [36]		11628 (50-step)	89899	N/A	−81.0	−88.8	−91.3	N/A	−84.6	−87.4	−96.5
CDC [59]		4854 (50-step)	39440	+318.7	−48.6	−71.7	−68.8	N/A	−63.3	−74.7	−86.8
StableCodec [63]		494 (1-step)	6138	N/A	−92.7	−96.2	−96.8	N/A	−94.9	−97.1	−99.0
OSCAR [18]	✓	456 (1-step)	8262	N/A	−79.0	−91.8	−92.3	N/A	−80.2	−88.7	−96.0
PerQ (Ours)	✓	**86 (4-step)**	**702**	**−1.6**	**−57.4**	**−63.2**	**−67.9**	**+2.7**	**−70.6**	**−77.4**	**−91.9**

* All BD-rates anchored to VVC intra (VTM 22.0, AI mode, YCbCr 4:4:4 BT.709, 10-bit) [10], sweeping six QP values (17, 22, 27, 32, 37, 42); N/A: the quality ranges do not overlap. † For neural codecs that ship multiple checkpoints, we report decoding time and complexity on the mid-rate checkpoint.

2 Related Work

Learned image compression originates from the joint optimization of an analysis/synthesis transform and a learned entropy model [51, 3], with subsequent progress driven primarily by more effective entropy models and stronger transforms. Hyperpriors capture spatially-varying scale [4]; discretized Gaussian mixture likelihoods refine the conditional entropy model [13]; channel-wise autoregressive factorizations sharpen it further [44]; checkerboard contexts parallelize spatial autoregression [22]; and recent designs combine grouped channel attention [20] with multi-reference context [29, 28] or transformer-CNN hybrid transforms [38]. The best performing neural image codecs match or exceed VVC all intra mode on both PSNR and MS-SSIM, with real-time variants closing the latency gap [25], and they remain visually faithful across moderate to high bitrates. The picture changes at low rates, where the bit budget is limited to preserve high-frequency content, and reconstructions become visibly blurred. The results in [8, 9] show that the divergence between source and reconstruction marginals widens at low rates and cannot be closed with further distortion-oriented training.

Generative image codecs are designed to primarily target lower bitrates by conditioning reconstructions on a generative prior trained to match natural-image statistics, trading pixel-wise fidelity for perceptual realism. GAN-based codecs integrate adversarial training into the rate-distortion objective [52, 2, 43, 46, 21], with subsequent work exposing the distortion-perception trade-off during decoding [1, 31] and obtaining variable-rate operation from a single codec [34]. Alternatively, diffusion-based codecs employ learned denoising at the decoder, with early work casting compression as reverse channel coding of noise samples through a Gaussian diffusion process [50]. Follow-up studies generalize this to pretrained foundation priors [54] and perform multi-rate operations on top of a frozen prior [27]. A separate line trains a conditional diffusion model as the codec decoder. CDC [59] replaces the VAE decoder with a conditional diffusion model, PerCo [12] couples a spatial latent with a global textual code for ultra-low-rate decoding, and other designs guide a frozen text-to-image or semantic prior [33, 36, 48, 24] with compressed content. These designs typically incur multi-step sampling costs during decoding. More recent work distills the iterative process into a single step [26, 63, 18, 58, 62]. Since reconstruction is conditioned on the generative prior rather than driven by the bitstream alone, pixel-space fidelity saturates as bpp increases, and PSNR improves sub-linearly with rate. Moreover, most designs commit to a single rate point at training time, requiring a separate model per operating point on the rate-distortion curve.

3 Method

As mentioned in Section 1, unconstrained generative compression typically conditions reconstructions on a learned image prior, often incurring a penalty in pixel-wise fidelity as a result. This design is acceptable in applications that can tolerate hallucinated details and deviations from source fidelity, as

opposed to distortion-oriented codecs. In this paper, we aim to close the perceptual gap of distortion-optimized image codecs while preserving reconstructions that remain faithful to the source. To this end, we constrain perceptual reconstruction within the quantization-error space induced by the compression backbone. In this section, we first derive the structural properties of this constraint in §3.1 and §3.2. We then propose a flow-matching based compensator in §3.3, which is integrated with a compression codec backbone (§3.3.2). We describe the training methodology in §3.4.

3.1 The Transport Bound

Let $x \in \mathbb{R}^{H\times W\times 3}$ denote the source image, f_ϕ^{enc} the encoder, f_θ^{dec} the decoder, and $y = f_\phi^{\text{enc}}(x) \in \mathbb{R}^{h\times w\times C}$ the latent. In a conventional setting, quantization is performed using a positive scale factor q, which maps the latent into the symbol space (i.e., the integer domain) for rounding. The complete quantization and decoding process is given by:

$$\text{Quantization:} \quad \hat{y}^{\text{sym}} = \lfloor y^{\text{sym}} \rceil = \lfloor \frac{y}{q} \rceil, \tag{1}$$

$$\text{Decoding:} \quad \hat{x} = f_\theta^{\text{dec}}(\hat{y}), \quad \text{where} \quad \hat{y} = q \cdot \hat{y}^{\text{sym}}. \tag{2}$$

in which $q \in \mathbb{R}_+^{h,w,C}$ is the quantization step size, which may vary over spatial locations and channels in modern neural compression models [56, 25]. Following the common practice [4], the non-differentiable rounding operation is emulated during training with additive uniform noise [3] or a straight-through estimator (STE) [51, 6]. In this paper, we utilize the latter in the reconstruction path, together with a distortion loss.

Under round-to-nearest quantization, the latent error in the symbol space is bounded by $\|y^{\text{sym}} - \hat{y}^{\text{sym}}\|_\infty \le \frac{1}{2}$ [17]. Rescaling by q gives the following element-wise bound in the original latent space:

$$|y_{i,j,c} - \hat{y}_{i,j,c}| \le \tfrac{q_{i,j,c}}{2}, \quad \text{for } i, j, c, \tag{3}$$

where i, j, c are the spatial coordinates and channel index of the latent.

To translate the latent-space quantization bound into a bound on reconstruction deviation, we exploit the local Lipschitz continuity of the decoder f_θ^{dec} over a compact operating range [53]. By (3), the source latent y lies in the ***quantization cell*** of $\hat{y}$, defined as the set of latents within $q/2$ of $\hat{y}$ element-wise. For each latent channel c, the Lipschitz constant of the decoder with respect to that channel,

$$L_c \;:=\; \sup_{u\in\mathcal{Y}} \left\| \tfrac{\partial f_\theta^{\text{dec}}(u)}{\partial u_{:,:,c}} \right\|_{\text{op}}, \tag{4}$$

measures the sensitivity of the full output to a perturbation confined to channel c of the input, where the partial derivative is taken over the spatial slice of channel c at the points where it exists, and $\mathcal{Y}$ is an operating range that contains every quantization cell of the backbone, so that the segment between $\hat{y}$ and any latent of its cell lies in $\mathcal{Y}$. By (3), each per-channel difference $\|y_{:,:,c} - \hat{y}_{:,:,c}\|_2$ is at most $\frac{q_c\sqrt{hw}}{2}$, where $q_c := \max_{i,j} q_{i,j,c}$ is the largest step of channel c. Composing with (4) and summing over channels yields the *transport bound*:

$$\|f_\theta^{\text{dec}}(y) - f_\theta^{\text{dec}}(\hat{y})\|_2 \;\le\; \sum_{c=1}^{C} L_c \,\|y_{:,:,c} - \hat{y}_{:,:,c}\|_2 \;\le\; \tfrac{\sqrt{hw}}{2} \sum_{c=1}^{C} L_c\, q_c \;=:\; R(q). \tag{5}$$

The transport bound establishes that every reconstruction decoded from the quantization cell, including the source latent y, lies within a radius $R(q)$ of the backbone reconstruction. Importantly, this radius is linear in the quantization step, with the decoder contributing only the fixed constants L_c. Although the bound is loose due to the supremum-based sensitivity estimates and the nonuniform distribution of actual quantization errors (Figure 3), its key value lies in characterizing a rate-dependent bounded region for reconstruction rather than providing a tight numerical estimate.

3.2 Bounded-support Modeling

The transport bound turns the quantization cell from an error characterization into a feasible region for perceptual reconstruction. In the latent space, (3) defines the quantization cell $\mathcal{C}(\hat{y}, q) = \{\, y' : \|y' - \hat{y}\|_\infty \le q/2 \,\}$ around the latent $\hat{y}$: the source latent y lies in this cell, and every latent in

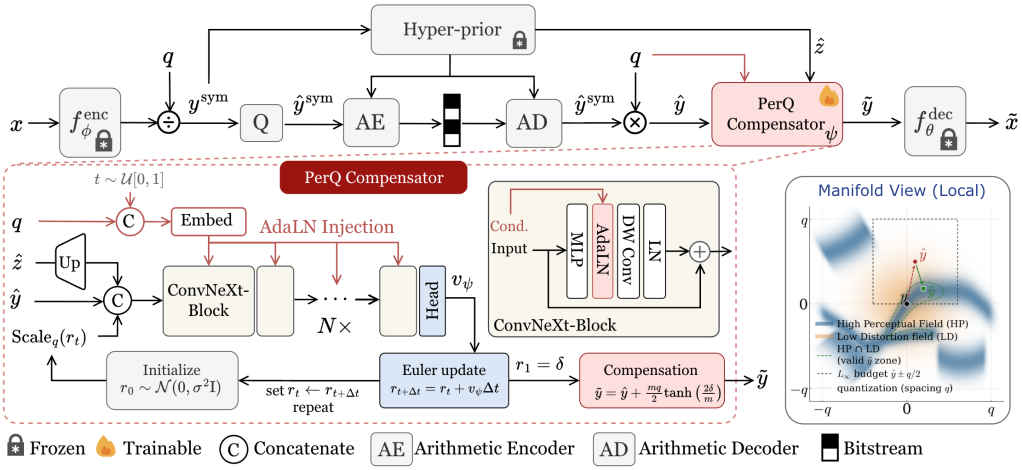

Figure 2: **PerQ overview. Top:** the compression pipeline. **Bottom-left:** architecture of the flow-matching based compensator. **Bottom-right:** illustrative 2D slice of latent space. The cell is centered at the quantized latent $\hat{y}$ and contains the source latent y by the quantization bound (3). The compensator generates $\tilde{y}$ within the cell, targeting the high-perceptual-quality field (blue) while remaining inside the low-distortion field (orange).

it rounds to the same transmitted symbols. In the pixel space, (5) confines every reconstruction decoded from the cell to the closed ball $\mathcal{B}(\hat{x}, q) := \{\, u : \|u - \hat{x}\|_2 \le R(q) \,\}$ around the backbone reconstruction $\hat{x} = f_\theta^{\mathrm{dec}}(\hat{y})$. This ball constrains decoder outputs relative to $\hat{x}$, rather than bounding the reconstruction error relative to the source image x, which also depends on the distortion introduced by the encoder-decoder transform. This transform-induced distortion remains fixed once the backbone is frozen (§3.3.2).

The rate-distortion-perception trade-off [8, 9] implies that a distortion-optimized backbone may retain a perceptual distribution gap $d_p(p_X, p_{\hat{X}}) > 0$. Rather than recovering the source latent itself, we formulate perceptual reconstruction as inverse generative modeling over the quantization cell, treating it as a feasible design space in which to seek perceptually enhanced reconstructions. Accordingly, we introduce a compensator ψ that models the conditional distribution:

$$\tilde{y} \sim p_\psi(\tilde{y} \mid \hat{y}, \hat{z}, q), \qquad \operatorname{supp} p_\psi \subseteq \mathcal{C}(\hat{y}, q), \tag{6}$$

in which $\hat{z}$ is the auxiliary latent produced by the backbone's hyper-prior and the support is enforced by clamping at the decoder side (§3.3.1). Every sample $\tilde{y}$ quantizes to the same transmitted symbols, and its decoded image lies within $\mathcal{B}(\hat{x}, q)$. Unlike prior work [60], which models quantization error as a latent-recovery problem by adding learned noise to $\hat{y}$ to approximate the source latent y, PerQ treats the quantization cell as a design space. Any $\tilde{y}$ within the cell is a valid reconstruction candidate, and the goal is to find one whose decoded image better closes the perceptual gap, rather than to recover y exactly. Moreover, as shown in Appendix A, matching the conditional distribution of the source latent would reproduce the distribution of the unquantized reconstruction.

3.3 Overall framework

Figure 2 provides an overview of the proposed PerQ framework, in which a compensator ψ is proposed to be located between the entropy decoder and the image decoder within the compression backbone. Given the entropy-decoded latent $\hat{y}$ and the auxiliary latent $\hat{z}$ produced by the backbone's hyper-prior, the compensator generates a refined latent $\tilde{y}$ which satisfies the cell constraint outlined in (6), which is then passed to the image decoder f_θ^{dec} to produce the reconstructed image $\tilde{x}$.

3.3.1 Compensator Architecture

Flow-matching parameterization. We instantiate the compensator as a flow-matching network [37, 39]. Specifically, we parameterize a conditional velocity field $v_\psi(r_t, t \mid \hat{y}, \hat{z}, q)$ and obtain the symbol-space residual δ as the final state of an ODE initialized at Gaussian noise:

$$r_0 \sim \mathcal{N}(0, \sigma^2 I), \qquad \frac{dr_t}{dt} = v_\psi(r_t, t \mid \hat{y}, \hat{z}, q), \qquad \delta = r_1. \tag{7}$$

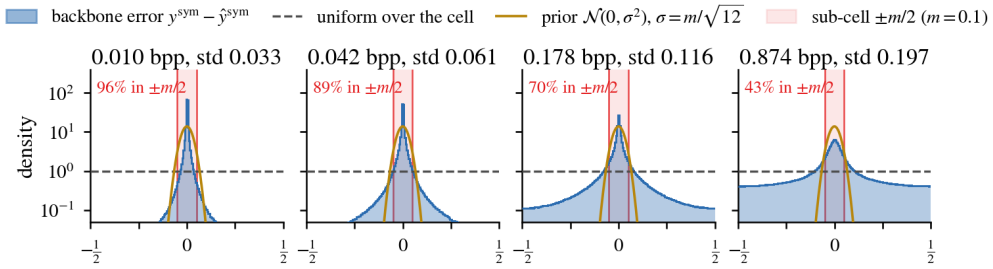

Figure 3: **Symbol-space distribution of the backbone quantization error** $y^{\text{sym}} - \hat{y}^{\text{sym}}$ on Kodak at four rate points (log density). The sub-cell $\pm m/2$ (red, $m = 0.1$) contains most of the error at low rates, and the prior $\mathcal{N}(0, \sigma^2)$ with $\sigma = m/\sqrt{12}$ (gold) is set to the scale of this sub-cell.

We solve this ODE using K Euler steps of size $\Delta t = 1/K$:

$$r_{t+\Delta t} = r_t + v_\psi(r_t, t \mid \hat{y}, \hat{z}, q)\,\Delta t, \qquad t = 0, \tfrac{1}{K}, \ldots, \tfrac{K-1}{K}. \tag{8}$$

The compensator is trained with the rectified-flow term [39]. We define the training target as the symbol-space quantization error $\delta_{\text{target}} = y^{\text{sym}} - \hat{y}^{\text{sym}} = (y - \hat{y})/q$, i.e., the displacement from the quantized latent to its pre-rounding source. For each sample, we draw $t \sim \mathcal{U}[0, 1]$ and perform linear interpolation with the noise prior and the target, $r_t = (1 - t)r_0 + t\delta_{\text{target}}$. The velocity field is regressed against the constant velocity along this interpolant:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{r_0, t, \delta_{\text{target}}} \left\| v_\psi(r_t, t \mid \hat{y}, \hat{z}, q) - (\delta_{\text{target}} - r_0) \right\|_2^2. \tag{9}$$

We set the noise scale to match the range allowed by the final residual clamp. By default, we set $\sigma = \frac{m}{\sqrt{12}}$, the standard deviation of a uniform distribution over the symbol-space sub-cell $[-m/2, m/2]$ that the bounded compensator can reach, where m is the bound scale (see next). The prior $r_0 \sim \mathcal{N}(0, \sigma^2 I)$ then matches the scale of the compensator's reachable region.

Residual clamping. The integrated residual δ is soft-bounded element-wise via a $\tanh$ nonlinearity, and the final compensated latent is

$$\tilde{y} = \hat{y} + \tfrac{mq}{2} \tanh\left(\tfrac{2\delta}{m}\right), \tag{10}$$

where $m \in (0, 1]$ is the bound scale that restricts the compensation to $\pm m/2$ in symbol space, a sub-cell of width mq centered at $\hat{y}$. Since the transport bound (5) is linear in the half-width of the cell, a sub-cell of width mq gives a radius $mR(q)$: the decoded reconstruction stays within $mR(q)$ of the backbone reconstruction, so the bound scale is an explicit control of the fidelity cost of perceptual compensation, which the bound-scale sweep of Table 3 confirms.

Empirically (Figure 3), the latent quantization residual occupies a range that varies substantially across rate points, with the bulk of its mass shrinking at low rates. This is because of the variable rate compression model design: at lower rate points, the quantization step q is larger, compacting the latent $y^{\text{sym}} = \frac{y}{q}$ into a small region of symbol space, where the residual inherits this small magnitude. Generating compensations over the full cell would therefore introduce excessive distortion, as a $\pm\frac{q}{2}$ perturbation is large relative to the signal. We use $m = 0.1$ by default. While this restricts the compensation magnitude at high rates, distortion-optimized backbones already achieve high quality at high bitrates, so a small-magnitude refinement is sufficient to close the perceptual gap.

Velocity field architecture. In our implementation, v_ψ is a ConvNeXt network [40] with N blocks, whose configuration is given in Appendix B. To keep the velocity field at one operating scale across rate points, we multiply the flow state r_t by q before the network, and divide the output by q to return to symbol space, where the bound of (10) is applied. We observe that this scaling improves performance over operating directly in symbol space. The hyper-prior latent $\hat{z}$ is upsampled by $4\times$ to the spatial resolution of $\hat{y}$ through two stages of 3×3 convolution and $2\times$ pixel shuffle with a SiLU activation in between. Following DiT [47], the timestep t is embedded by a sinusoidal embedding and an MLP, and enters each block through a zero-initialized linear projection to the shift, scale, and gate of an adaLN-zero modulation.

3.3.2 Compression backbone

We focus on modeling the quantization-error space and adopt an existing neural image compression model as the codec backbone with minor modifications. Specifically, we use the intra-frame model of DCVC-RT [25], which delivers strong rate–distortion performance at low complexity, supports fast encoding and decoding, and provides variable-bitrate coding by selecting a quantization parameter for the target rate point.

One implementation detail of [25] is that the quantization step size is applied at two stages on both the encoder and decoder sides. The encoder f_ϕ^{enc} is decomposed into two sub-networks f_ϕ^{enc1} and f_ϕ^{enc2}: the rate-specific parameter q^{enc} is applied between them, and the per-pixel parameter $q_{\text{local}}^{\text{enc}}$ (predicted by the hyper-prior) is applied at the bottleneck:

$$y^{\text{sym}} = f_\phi^{\text{enc2}}\big((f_\phi^{\text{enc1}}(x) \cdot q^{\text{enc}}\big) \cdot q_{\text{local}}^{\text{enc}}. \tag{11}$$

The decoder mirrors this structure during inverse quantization. While this distributed scaling provides additional capacity for rate adaptation, it conflicts with the analysis in §3.1: when the rate-specific scaling is absorbed into the encoder/decoder networks, f_ϕ becomes rate-adaptive, so different rate points produce different bottleneck representations of the same image, and the encoder–decoder pair itself becomes an information bottleneck. In such a case, a decoder that knows it operates at a low rate can learn to discard detail, e.g., producing blurrier outputs, which empirically lowers the MSE at low rate points. The quantization error remains bounded in symbol space, but the unrounded decode itself is far from the source at low rates, so the cell no longer contains a high-fidelity latent and in-cell compensation has little headroom.

We therefore modify [25] by consolidating all quantization scaling at the bottleneck[1]:

$$y^{\text{sym}} = f_\phi^{\text{enc2}}\big(f_\phi^{\text{enc1}}(x)\big) \cdot q^{\text{enc}} \cdot q_{\text{local}}^{\text{enc}}, \tag{12}$$

and symmetrically on the decoder side:

$$\hat{y} = \hat{y}^{\text{sym}} \cdot q^{\text{dec}} \cdot q_{\text{local}}^{\text{dec}}, \qquad \hat{x} = f_\theta^{\text{dec}}(\hat{y}) = f_\theta^{\text{dec2}}\big(f_\theta^{\text{dec1}}(\hat{y})\big), \tag{13}$$

where the decoder-side scales are paired with the encoder-side ones, $q^{\text{dec}} = 1/q^{\text{enc}}$ and $q_{\text{local}}^{\text{dec}} = 1/q_{\text{local}}^{\text{enc}}$, and the rate-specific scale q^{enc} increases monotonically with the rate point i by parameterizing it as an accumulation of positive increments:

$$q_i^{\text{enc}} = 0.025 + (40 - 0.025)\,\text{sigmoid}\Big(q_0^{\text{raw}} + \sum_{j=1}^{i} \text{softplus}\big(q_j^{\text{raw}}\big)\Big), \tag{14}$$

with learnable parameters q_j^{raw} and the rescaled sigmoid keeps q_i^{enc} within $(0.025, 40)$ to span a wide range of quantization levels. With this modification, the encoder and decoder are rate-agnostic, the information loss is concentrated at a single rounding step, and the element-wise bound of (3) holds. We also verify this empirically: without rounding, $f_\theta^{\text{dec}}(f_\phi^{\text{enc}}(x))$ reaches 40.4 dB / 41.1 dB PSNR on Kodak / CLIC2020-test at every evaluated rate point, whereas the unrounded decode of [25] drops to 26.9 dB / 30.1 dB at its lowest rate point. This indicates that its encoder and decoder are rate-dependent and form an information bottleneck at low rates, which would limit what a compensator in the latent space can recover. We utilize $q = q^{\text{dec}} \cdot q_{\text{local}}^{\text{dec}}$ when performing the computation with the PerQ compensator.

3.4 Training

Stage 1: Backbone training. We first train the variable-rate compression backbone with the rate-distortion objective $R + \lambda D$, following the typical image compression training pipeline [4]. We then freeze the encoder, decoder, and entropy models, preserving the quantization structure of (3) exactly.

Stage 2: Compensator training. We train the compensator with a combined objective comprising the rectified-flow loss (9), distortion and perceptual terms, and an adversarial term:

$$\mathcal{L} = \lambda\big(\mathcal{L}_{\text{MSE}} + \mathcal{L}_{\text{E-LPIPS}}\big) + \lambda_{\text{FM}}\mathcal{L}_{\text{FM}} + \lambda_{\text{GAN}}\mathcal{L}_{\text{GAN}}, \tag{15}$$

where $\mathcal{L}_{\text{E-LPIPS}}$ is the E-LPIPS perceptual distance [61, 30], $\mathcal{L}_{FM}$ is the flow matching loss [39], and $\mathcal{L}_{\text{GAN}}$ is a generator loss against a patch discriminator conditioned on the rate point (cf. [43, 60]). Since the backbone is frozen, no rate term appears in the Stage 2 objective.

[1] For brevity, we omit the quantization offset (mean subtraction) used in the actual implementation.

4 Experimental Configurations

Implementation Details.

We use the intra-frame model of DCVC-RT [25] as our default compression backbone, supporting 64 rate points, with the architectural modifications described in Section 3.3.2. We trained it from scratch on the MLIC training dataset [29, 28], with a batch size of 64 for 1M steps in the first stage (backbone training) and a batch size of 16 for 200K steps in the second stage (compensator training). More training details and configurations are provided in the Appendix B.

Evaluation. We evaluate PerQ on Kodak [16] (24 images at 768×512), CLIC2020-test [14] (428 images at $\sim$2K resolution). We measure pixel fidelity using PSNR in RGB space, and perceptual quality using LPIPS [61], DISTS [15], and FID [23] computed on 256×256 patches following [46].

Benchmark Methods. We compare against four baseline categories. The conventional reference is VVC (VTM-22.0 intra) [10]. For distortion-optimal learned codecs we include ELIC [20] (PSNR anchor), EVC [56] and DCVC-RT intra [25]. For GAN-based generative codecs we include MS-ILLM [46] (perceptual anchor), HiFiC [43], and Control-GIC [34]. For diffusion-based generative codecs, we include PerCo [12], CDC [59], DiffEIC [36], StableCodec [63], and OSCAR [18]. All neural baselines are re-run from publicly available checkpoints. Reconstructions are generated on a single NVIDIA-4090 GPU, except for DiffEIC [36], whose 50-step decoding is prohibitively slow on the 4090 and is therefore generated on NVIDIA-H200. Decoding latency and complexity reported in Table 1 are measured on a single NVIDIA-4090 GPU for fair comparison.

5 Results and Discussion

Quantitative comparison. Table 1 and Figure 4 position PerQ across the deployment Pareto frontier on the Kodak and CLIC2020-test datasets. With a single checkpoint, PerQ performs close to ELIC on PSNR (-1.6% vs. -2.0% BD-rate against VVC on Kodak, $+2.7\%$ vs. -1.9% on CLIC2020-test) while achieving large perceptual gains (-57.4% in LPIPS, -63.2% in DISTS and -67.9% in FID on Kodak, and -70.6%, -77.4% and -91.9% on CLIC2020-test). As shown in Figure 1 (right), PerQ is the only codec that combines such perceptual gains with VVC-level PSNR, whereas existing perceptual codecs offer significantly inferior PSNR performance ($+89.9\%$ BD-rate against VVC for MS-ILLM and $+116.3\%$ for HiFiC on CLIC2020-test). It can also be observed that per-rate trained generative codecs achieve higher perceptual BD-rate gains (e.g., MS-ILLM, StableCodec), but they require separate checkpoints per rate point or operate over narrower rate ranges.

Qualitative comparison. Figure 1 (left) and Figure 5 compare reconstructions at comparable bitrates. At low rates, ELIC produces visibly blurred textures, while generative codecs recover sharpness but with a substantial PSNR drop (more than 5 dB below ELIC). PerQ recovers much of this sharpness with PSNR results close to ELIC. In Figure 1, for example, it matches the LPIPS of OSCAR with a 6.5 dB higher PSNR. At high rates, generative codecs introduce unfaithful texture details, while PerQ delivers sharpness with similarly high fidelity to ELIC (28.22 vs. 28.40 dB) using the fewest bits. This pattern confirms the quantitative results in Table 1 and verifies the effectiveness of the bounded compensator. More visual examples are available in Appendix D.

Ablation Study. We ablate the design choices of PerQ on the CLIC2020-test dataset [14]. We study ***the compensator design***, ***the different backbone***, and ***the bound size*** that sets the width of the sub-cell the compensator can reach. Specifically, Table 2 ablates four components of the default PerQ design. All variants share the frozen backbone and therefore decode identical bitstreams. It can be observed that replacing E-LPIPS with LPIPS (**v1.1**) slightly decreases performance, and reducing the E-LPIPS weight to 0.25 (**v1.3**) increases the FID BD-rate by 7.20%, while a weight of 0.5 (**v1.2**) performs similarly to the default. Reducing flow-matching from four integration steps to one (**v2.1**) degrades perception ($+7.96\%$ LPIPS, $+5.63\%$ DISTS, $+14.75\%$ FID), and lowering λ_{FM} (**v3.1**, **v3.2** and **v3.3**) degrades all metrics, by up to $+28.31\%$ in FID. Finally, the $0.5\times$ compensator (18.1M parameters, **v4.1**) performs close to the default (at most $+3.38\%$, in FID), while the $0.25\times$ one (8.8M, **v4.2**) loses perceptual quality ($+8.85\%$ LPIPS, $+15.64\%$ FID).

The bound scale m in (10) sets the width of the sub-cell and thus the trade-off between perceptual quality and pixel fidelity (Table 3). Larger m increases the PSNR cost, while the perceptual savings saturate around $m = 0.2$. Removing the bound brings no further gain. This reflects the geometric

Table 2: **PerQ design ablation on CLIC2020-test.** BD-rate against the default setting (DCVC-RT intra backbone, 31.0M parameters (the compensator), 4 steps, $\lambda_{\text{FM}} = 10^4$ and trained with the patch-GAN and E-LPIPS perceptual loss (weight 1.0)).

Configuration	PSNR ↑	MS-SSIM ↑	LPIPS ↓	DISTS ↓	FID ↓
Loss setting					
v1.1 LPIPS instead of E-LPIPS	+1.92%	+1.16%	+2.85%	+1.39%	+5.13%
v1.2 E-LPIPS weight 0.5	+0.24%	+0.97%	+1.38%	-2.57%	-1.60%
v1.3 E-LPIPS weight 0.25	+0.61%	+2.02%	+3.49%	+1.15%	+7.20%
Flow matching steps					
v2.1 1 step	-1.06%	-1.83%	+7.96%	+5.63%	+14.75%
Flow matching loss weight					
v3.1 $\lambda_{\text{FM}} = 10$	+6.88%	+5.58%	+17.28%	+12.07%	+27.51%
v3.2 $\lambda_{\text{FM}} = 10^2$	+6.51%	+5.18%	+16.29%	+12.23%	+28.31%
v3.3 $\lambda_{\text{FM}} = 10^3$	+4.14%	+2.51%	+10.59%	+7.49%	+22.69%
Compensator size					
v4.1 PerQ-0.5× (18.1M)	-0.03%	-0.22%	-0.59%	-0.05%	+3.38%
v4.2 PerQ-0.25× (8.8M)	-0.27%	-1.10%	+8.85%	+11.05%	+15.64%

Table 3: **Bound scale ablation on CLIC2020-test.** BD-rate against the unmodified backbone for the bound scale m in (10). PSNR_{min}/PSNR_{max} report the average PSNR at the lowest/highest rate point.

Bound scale m	PSNR ↑	MS-SSIM ↑	LPIPS ↓	DISTS ↓	FID ↓	PSNR_{min}	PSNR_{max}
0.05	+13.77%	+16.20%	-60.56%	-76.61%	-91.87%	24.88	39.16
0.10 **(default)**	+18.24%	+20.60%	-67.84%	-79.68%	-91.51%	24.76	38.81
0.20	+23.23%	+24.40%	-70.79%	-82.53%	-92.39%	24.41	38.47
unbounded	+28.58%	+25.38%	-64.45%	-81.73%	-92.22%	24.47	38.11

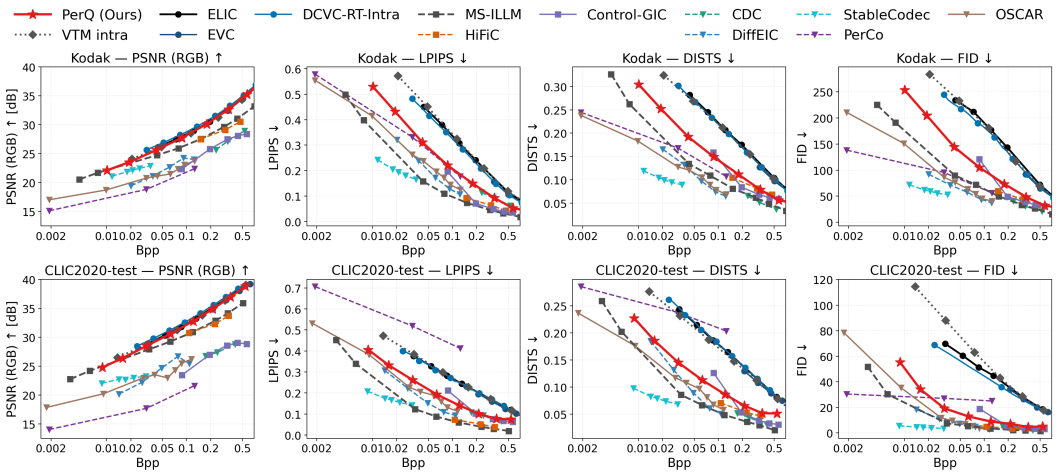

Figure 4: **Rate–distortion curves on Kodak (top) and CLIC2020-test (bottom).** Panels report PSNR (RGB), LPIPS (AlexNet), DISTS, and FID across bitrates. PerQ (red star) traces a single curve from one trained checkpoint by varying the rate parameter at the decoder. Solid markers denote multi-rate or distortion-optimal baselines; dashed markers denote per-rate trained generative codecs.

role of the transport bound, and the default $m = 0.1$ balances perceptual gains with fidelity while still reaching a peak PSNR of 38.81 dB.

Table 4 further attaches the same compensator to two other frozen backbones, ELIC [20] and MLIC++ [28], both retrained for variable-rate coding. The compensator brings similar gains on all three backbones, reducing the LPIPS BD-rate by 42–45 points on Kodak and by 61–64 points on CLIC2020-test at a similar PSNR cost. These consistent gains suggest that PerQ generalizes across backbone architectures.

We note that these results are obtained with a frozen backbone and the compensator as an add-on module and more aggressive modification of the compression model is expected to widen the distortion-perception trade-off further. We discuss the limitation in the Appendix E.

Table 4: **PerQ compensator on three frozen backbones.** The same compensator, bound and training recipe are attached to three variable-rate backbones (retrained), each covering 64 rate points with one checkpoint. BD-rate (%) against VVC intra.

	Kodak					CLIC2020-test				
	bpp	PSNR	LPIPS	DISTS	FID	bpp	PSNR	LPIPS	DISTS	FID
DCVC-RT backbone [25] (default)										
Backbone only	0.010–0.87	−10.6	−12.7	−5.7	−10.4	0.009–0.54	−11.1	−7.3	+3.5	−42.2
+ PerQ compensator	0.010–0.87	**−1.6**	**−57.4**	**−63.2**	**−67.9**	0.009–0.54	**+2.7**	**−70.6**	**−77.4**	**−91.9**
ELIC backbone [20]										
Backbone only	0.061–0.91	−4.9	−13.8	−5.2	−5.5	0.045–0.56	−5.5	−8.5	+4.9	−28.4
+ PerQ compensator	0.061–0.91	**+4.0**	**−57.4**	**−64.4**	**−64.2**	0.045–0.56	**+9.0**	**−72.5**	**−77.8**	**−88.5**
MLIC++ backbone [28]										
Backbone only	0.055–0.85	−9.1	−20.5	−10.5	−13.0	0.041–0.52	−11.0	−15.8	−4.0	−35.4
+ PerQ compensator	0.055–0.85	**+0.0**	**−63.0**	**−68.4**	**−68.0**	0.041–0.52	**+1.2**	**−76.9**	**−79.2**	**−89.4**

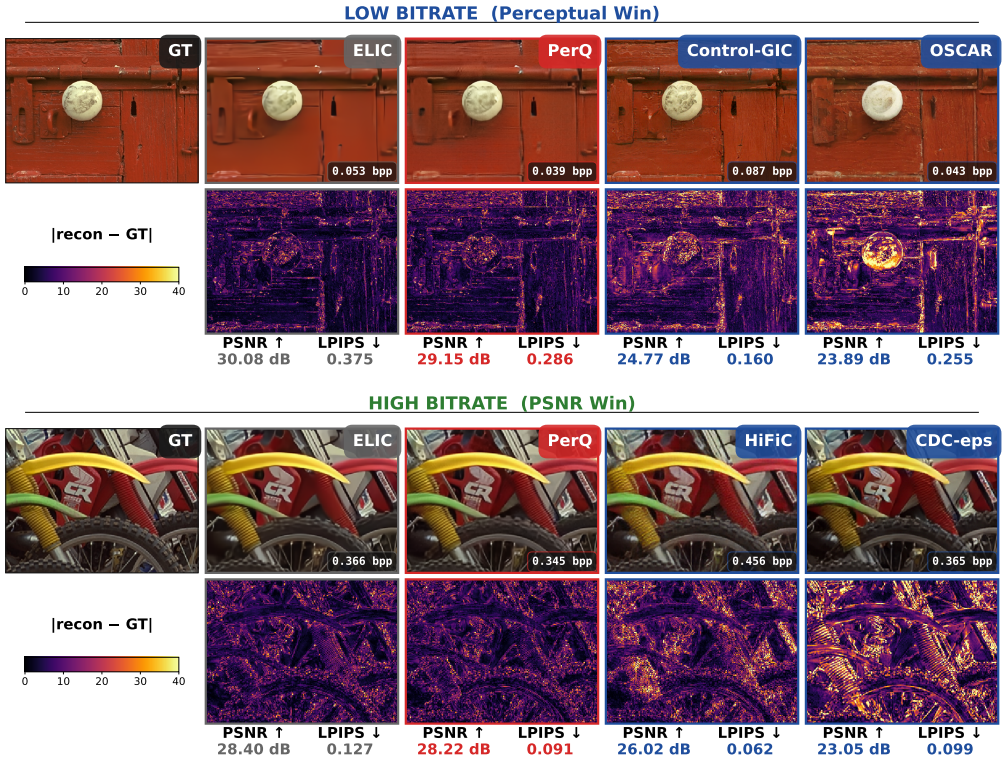

Figure 5: **Qualitative comparison.** Source crop (left) compared with reconstructions from a distortion-optimized VAE-based codec (grey), PerQ (red, ours), and two generative codecs, GAN-based and diffusion-based (blue), at similar bitrates with the error maps.

6 Conclusion

We presented PerQ, a learning-based image codec that addresses the trade-off between pixel-wise fidelity and perceptual quality by formulating perceptual reconstruction as inverse generative modeling with bounded support. Specifically, we characterized the quantization cell of a multi-rate distortion-optimized codec as the set of latents that yield the same transmitted symbols, and derived a transport bound that constrains the deviation of reconstructions within this cell from the backbone's output. Building on this formulation, we developed a rate-conditioned flow-matching compensator that performs perceptual reconstruction within the codec-induced bounded support, preserving the backbone's multi-rate capability. Experiments on Kodak and CLIC2020-test demonstrate that PerQ achieves strong perceptual quality with competitive pixel fidelity, while supporting a wide bitrate range through a single model and efficient decoding. To the best of our knowledge, PerQ is the first generative image codec to combine these properties in a single framework.

Acknowledgements

This work was supported by UK EPSRC, the UKRI MyWorld Strength in Places Programme, Tencent (US) and the University of Bristol. Part of the computational work was also supported by the facilities provided by the Advanced Computing Research Centre at the University of Bristol. The authors acknowledge the use of resources provided by the Isambard-AI National AI Research Resource (AIRR) [42]. Isambard-AI is operated by the University of Bristol and is funded by the UK Government's Department for Science, Innovation and Technology (DSIT) via UK Research and Innovation; and the Science and Technology Facilities Council [ST/AIRR/I-A-I/1023].

References

[1] E. Agustsson, D. Minnen, G. Toderici, and F. Mentzer. Multi-realism image compression with a conditional generator. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22324–22333, 2023.

[2] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. Van Gool. Generative adversarial networks for extreme learned image compression. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 221–231, 2019.

[3] J. Ballé, V. Laparra, and E. P. Simoncelli. End-to-end optimized image compression. In *International Conference on Learning Representations (ICLR)*, 2017.

[4] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations (ICLR)*, 2018.

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

[6] Y. Bengio, N. Léonard, and A. C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[7] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations (ICLR)*, 2018.

[8] Y. Blau and T. Michaeli. The perception-distortion tradeoff. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6228–6237, 2018.

[9] Y. Blau and T. Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff. In *International Conference on Machine Learning (ICML)*, pages 675–685, 2019.

[10] 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 (TCSVT)*, 31(10):3736–3764, 2021.

[11] D. Bull and F. Zhang. *Intelligent Image and Video Compression: Communicating Pictures*. Academic Press, 2021.

[12] M. Careil, M. J. Muckley, J. Verbeek, and S. Lathuilière. Towards image compression with perfect realism at ultra-low bitrates. In *International Conference on Learning Representations (ICLR)*, 2024.

[13] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto. Learned image compression with discretized Gaussian mixture likelihoods and attention modules. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7936–7945, 2020.

[14] CLIC 2020: Challenge on learned image compression. `https://compression.cc`, 2020.

[15] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44(5):2567–2581, 2022.

[16] Eastman Kodak Company. Kodak lossless true color image suite (PhotoCD PCD0992). `https://r0k.us/graphics/kodak/`.

[17] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Springer, 1992.

[18] J. Guo, Y. Ji, Z. Chen, K. Liu, M. Liu, W. Rao, W. Li, Y. Guo, and Y. Zhang. OSCAR: One-step diffusion codec across multiple bit-rates. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 85267–85286, 2025.

[19] J. Han, B. Li, D. Mukherjee, C.-H. Chiang, A. Grange, C. Chen, H. Su, S. Parker, S. Deng, U. Joshi, Y. Chen, Y. Wang, P. Wilkins, Y. Xu, and J. Bankoski. A technical overview of AV1. *Proceedings of the IEEE*, 109(9):1435–1462, 2021.

[20] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5708–5717, 2022.

[21] D. He, Z. Yang, H. Yu, T. Xu, J. Luo, Y. Chen, C. Gao, X. Shi, H. Qin, and Y. Wang. PO-ELIC: Perception-oriented efficient learned image coding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1763–1768, 2022.

[22] D. He, Y. Zheng, B. Sun, Y. Wang, and H. Qin. Checkerboard context model for efficient learned image compression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14771–14780, 2021.

[23] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6626–6637, 2017.

[24] E. Hoogeboom, E. Agustsson, F. Mentzer, L. Versari, G. Toderici, and L. Theis. High-fidelity image compression with score-based generative models. *arXiv preprint arXiv:2305.18231*, 2023.

[25] Z. Jia, B. Li, J. Li, W. Xie, L. Qi, H. Li, and Y. Lu. Towards practical real-time neural video compression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12543–12552, 2025.

[26] Z. Jia, J. Li, B. Li, H. Li, and Y. Lu. Generative latent coding for ultra-low bitrate image compression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26088–26098, 2024.

[27] Z. Jia, Z. Zheng, N. Xue, J. Li, B. Li, Z. Guo, X. Zhang, H. Li, and Y. Lu. CoD: A diffusion foundation model for image compression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 38420–38429, 2026.

[28] W. Jiang, J. Yang, Y. Zhai, F. Gao, and R. Wang. MLIC++: Linear complexity multi-reference entropy modeling for learned image compression. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 21(5):142:1–142:25, 2025.

[29] W. Jiang, J. Yang, Y. Zhai, P. Ning, F. Gao, and R. Wang. MLIC: Multi-reference entropy model for learned image compression. In *ACM International Conference on Multimedia (ACM MM)*, pages 7618–7627, 2023.

[30] M. Kettunen, E. Härkönen, and J. Lehtinen. E-LPIPS: Robust perceptual image similarity via random transformation ensembles. *arXiv preprint arXiv:1906.03973*, 2019.

[31] N. Körber, E. Kromer, A. Siebert, S. Hauke, D. Mueller-Gritschneder, and B. W. Schuller. EGIC: Enhanced low-bit-rate generative image compression guided by semantic segmentation. In *European Conference on Computer Vision (ECCV)*, pages 202–220, 2024.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1106–1114, 2012.

[33] E. Lei, Y. B. Uslu, H. Hassani, and S. Saeedi Bidokhti. Text + Sketch: Image compression at ultra low rates. *arXiv preprint arXiv:2307.01944*, 2023.

[34] A. Li, F. Li, Y. Liu, R. Cong, Y. Zhao, and H. Bai. Once-for-all: Controllable generative image compression with dynamic granularity adaptation. In *International Conference on Learning Representations (ICLR)*, 2025.

[35] J. Li, B. Li, and Y. Lu. Neural video compression with feature modulation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26099–26108, 2024.

[36] Z. Li, Y. Zhou, H. Wei, C. Ge, and J. Jiang. Toward extreme image compression with latent feature guidance and diffusion prior. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 35(1):888–899, 2025.

[37] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

[38] J. Liu, H. Sun, and J. Katto. Learned image compression with mixed transformer-CNN architectures. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14388–14397, 2023.

[39] X. Liu, C. Gong, and Q. Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023.

[40] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. A ConvNet for the 2020s. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11966–11976, 2022.

[41] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[42] S. McIntosh-Smith, S. R. Alam, and C. J. Woods. Isambard-AI: a leadership-class supercomputer optimised specifically for artificial intelligence. In *Cray User Group Conference (CUG)*, pages 44–54, 2024.

[43] F. Mentzer, G. Toderici, M. Tschannen, and E. Agustsson. High-fidelity generative image compression. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 11913–11924, 2020.

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

[45] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[46] 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 (ICML)*, pages 25426–25443, 2023.

[47] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4172–4182, 2023.

[48] L. Relic, R. Azevedo, M. Gross, and C. Schroers. Lossy image compression with foundation diffusion models. In *European Conference on Computer Vision (ECCV)*, pages 303–319, 2024.

[49] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[50] L. Theis, T. Salimans, M. D. Hoffman, and F. Mentzer. Lossy compression with Gaussian diffusion. *arXiv preprint arXiv:2206.08889*, 2022.

[51] L. Theis, W. Shi, A. Cunningham, and F. Huszár. Lossy image compression with compressive autoencoders. In *International Conference on Learning Representations (ICLR)*, 2017.

[52] M. Tschannen, E. Agustsson, and M. Lucic. Deep generative models for distribution-preserving lossy compression. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5933–5944, 2018.

[53] A. Virmaux and K. Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3839–3848, 2018.

[54] J. Vonderfecht and F. Liu. Lossy compression with pretrained diffusion models. In *International Conference on Learning Representations (ICLR)*, 2025.

[55] G. K. Wallace. The JPEG still picture compression standard. *Communications of the ACM (CACM)*, 34(4):30–44, 1991.

[56] G.-H. Wang, J. Li, B. Li, and Y. Lu. EVC: Towards real-time neural image compression with mask decay. In *International Conference on Learning Representations (ICLR)*, 2023.

[57] Z. Wang and A. C. Bovik. Mean squared error: Love it or leave it? A new look at signal fidelity measures. *IEEE Signal Processing Magazine (SPM)*, 26(1):98–117, 2009.

[58] N. Xue, Z. Jia, J. Li, B. Li, Y. Zhang, and Y. Lu. One-step diffusion-based image compression with semantic distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 37108–37144, 2025.

[59] R. Yang and S. Mandt. Lossy image compression with conditional diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 64971–64995, 2023.

[60] F. Yi, Z. Gong, and H. Zeng. Rethinking latent variable in learned image compression. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 8531–8540, 2026.

[61] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018.

[62] T. Zhang, D. Liu, and C. W. Chen. Ultra-low bitrate perceptual image compression with shallow encoder. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12118–12128, 2026.

[63] T. Zhang, X. Luo, L. Li, and D. Liu. StableCodec: Taming one-step diffusion for extreme image compression. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17379–17389, 2025.

Appendix

A From the quantization cell to the compensated distribution

This section formalizes the connection between the quantization cell and the distribution induced by the compensator. Throughout, we write $g := f_\theta^{\mathrm{dec}}$ for simplicity.

Let X denote the random variable of source images with distribution p_X, let $Y = f_\phi^{\mathrm{enc}}(X)$ be its latent, and let $(\hat{Y}, \hat{Z}, Q)$ be the entropy-decoded latent, the hyper-prior latent, and the rate state obtained by quantizing Y. The backbone reconstruction is $\hat{X} = g(\hat{Y})$ and the unquantized reconstruction is $X^* := g(Y)$. Although $\hat{Y}$, $\hat{Z}$, and Q are deterministic functions of X given the frozen backbone, we treat them as random variables with the joint distribution induced by p_X. The joint distribution of the source latent and the decoder inputs factors as

$$p_{Y,\hat{Y},\hat{Z},Q}(y, \hat{y}, \hat{z}, q) = p_{Y|\hat{Y},\hat{Z},Q}(y \mid \hat{y}, \hat{z}, q) p_{\hat{Y},\hat{Z},Q}(\hat{y}, \hat{z}, q), \tag{16}$$

where $p_{\hat{Y},\hat{Z},Q}$ is fixed by the frozen backbone and $p_{Y|\hat{Y},\hat{Z},Q}$ is the posterior distribution of the source latent given the decoder inputs. This conditional distribution characterizes the source latents compatible with the decoder inputs, whose support we establish next.

The posterior has support inside the quantization cell. The quantization bound (3) states that $|Y - \hat{Y}| \le Q/2$ holds element-wise for every encoded image. Thus, $Y \in \mathcal{C}(\hat{Y}, Q)$ holds deterministically under $p_{Y,\hat{Y},\hat{Z},Q}$. Conditioning on $(\hat{Y}, \hat{Z}, Q) = (\hat{y}, \hat{z}, q)$ gives

$$\mathrm{supp}\big(p_{Y|\hat{Y},\hat{Z},Q}(\cdot \mid \hat{y}, \hat{z}, q)\big) \subseteq \mathcal{C}(\hat{y}, q). \tag{17}$$

Conversely, every $y' \in \mathcal{C}(\hat{y}, q)$ quantizes to the same transmitted symbols, up to ties on the cell boundary, given the hyper-prior latent $\hat{z}$ that determines the step size q. The cell therefore characterizes the latents consistent with the transmitted symbols. This makes it a natural support constraint for the compensator: it contains the source latent while restricting generation to symbol-consistent alternatives.

The compensator induces a marginal. The compensator defines a conditional distribution $p_\psi(\tilde{y} \mid \hat{y}, \hat{z}, q)$ supported on $\mathcal{C}(\hat{y}, q)$ for every $(\hat{y}, \hat{z}, q)$, as specified in (6). Combined with the fixed marginal $p_{\hat{Y},\hat{Z},Q}$, this conditional distribution induces a marginal distribution on the compensated latent $\tilde{Y}$:

$$p_{\tilde{Y}}(\tilde{y}) = \int p_\psi(\tilde{y} \mid \hat{y}, \hat{z}, q) p_{\hat{Y},\hat{Z},Q}(\hat{y}, \hat{z}, q)\, d\hat{y}\, d\hat{z}\, dq. \tag{18}$$

Through the fixed decoder, it further induces a marginal $p_{\tilde{X}}$ on the compensated reconstruction $\tilde{X} = g(\tilde{Y})$. Closing the perceptual gap of [9] ideally requires $p_{\tilde{X}} = p_X$. We next characterize the distribution obtained by matching the source-latent posterior, providing a reference for the compensator's achievable perceptual improvement under the frozen backbone.

Conditional matching reproduces the unquantized reconstruction. Since $p_{\hat{Y},\hat{Z},Q}$ is fixed, if

$$p_\psi(\tilde{y} \mid \hat{y}, \hat{z}, q) = p_{Y|\hat{Y},\hat{Z},Q}(\tilde{y} \mid \hat{y}, \hat{z}, q) \quad \text{for } p_{\hat{Y},\hat{Z},Q}\text{-almost every } (\hat{y}, \hat{z}, q), \tag{19}$$

then substituting (19) into (18) yields $p_{\tilde{Y}} = p_Y$ by (16). Since g is fixed, it follows that $p_{\tilde{X}} = p_{X^*}$. Thus, matching the latent posterior reproduces the distribution of the unquantized reconstruction $X^* = g(Y)$, rather than necessarily recovering the source-image distribution p_X. The remaining divergence $d_p(p_X, p_{X^*})$ arises from the backbone transform itself and remains fixed once the backbone is frozen. The compensator instead targets the rate-dependent component of the perceptual gap, which generally increases as the rate decreases. Several existing generative codecs also pursue conditional distribution matching, typically conditioning on quantized latents [59, 12] or learned semantic codes [36, 12]. PerQ conditions on $(\hat{y}, \hat{z}, q)$ and constrains the conditional distribution to the quantization cell, yielding the bounded support in (17).

The decoded samples stay inside $\mathcal{B}(\hat{x}, q)$. Beyond this distributional result, the support constraint also provides a geometric guarantee on the compensated reconstructions through the transport bound. For every $\tilde{y} \in \mathcal{C}(\hat{y}, q)$, (5) gives

$$\|g(\tilde{y}) - g(\hat{y})\|_2 \le R(q).$$

Hence, every sample from p_ψ decodes into $\mathcal{B}(\hat{x}, q)$, where $\hat{x} = g(\hat{y})$. With the clamp in (10), the decoded sample lies within the smaller ball of radius $mR(q)$. In particular, $X^* = g(Y) \in \mathcal{B}(\hat{X}, Q)$. This bound limits how far the compensator can move a reconstruction from the backbone output, allowing perceptual refinement within a rate-dependent bounded region.

B Implementation details

B.1 Training configurations

Stage 1: backbone (distortion only). The compression backbone (DCVC-RT intra-frame model [25]) is trained from scratch on the MLIC training set [29, 28] with the rate-distortion objective $L = R + \lambda \cdot L_{\text{MSE}}$ for 1M steps at batch size 64. The Lagrange multiplier λ is sampled from [4, 4096] for 64 rate points, following [35].

We use the AdamW optimizer [41] with $(\beta_1, \beta_2) = (0.9, 0.999)$ and weight decay 10^{-2}. The peak learning rate is 2×10^{-4}, reached by a linear warmup from 10^{-6} over the first 100K steps, with a multi-step schedule that halves the learning rate at 50%, 75%, 87.5%, and 93.75% of training. Gradients are clipped to a global ℓ_2 norm of 1.0. Over the first 10% of training, the weight of the rate term and λ are ramped up linearly, from 0 to 1 and from 4 to the sampled value, respectively. Training images are randomly cropped to 256×256.

Stage 2: compensator (perceptual optimization). In stage 2 we freeze the encoder, decoder, hyper-prior, and entropy models, and add the compensator and discriminator for joint training. We train for 200K steps at batch size 16 with $\lambda \in [4, 4096]$ and the Stage 2 objective of Section 3.4. The weight of the latent flow-matching term is $\lambda_{\text{FM}} = 10^4$, the weight of the adversarial term is $\lambda_{\text{GAN}} = 0.1$, and the E-LPIPS term has the same weight as the MSE term.

$L_{\text{E-LPIPS}}$ uses the VGG backbone [30, 61, 49] during training; we report the LPIPS variant with AlexNet [32] features for evaluation. Optimizer and schedule follow stage 1, except the peak learning rate is 1×10^{-4}, the warmup is 10K steps without the rate and λ warmup, and we keep the weight decay to 10^{-2} on the compensator's and discriminator's parameters. The GAN term is disabled for the first 20K iterations and then linearly warmed up over the next 20K iterations to its full weight.

Compensator architecture. The velocity field v_ψ is a stack of 10 ConvNeXt blocks [40] with hidden width of 512, an MLP expansion ratio of 4 and a kernel size of 5. It first computes the sum of a 1×1 convolution of the scaled flow state $q\, r_t$ and a 1×1 convolution of the channel-wise concatenation of $\hat{y}$, the upsampled $\hat{z}$, and q. Timestep conditioning is injected through a sinusoidal positional embedding followed by a linear layer with SiLU, and each block maps it through a zero-initialized linear layer to the shift, scale, and gate of an adaLN-zero modulation [47]. The output head is a LayerNorm followed by a zero-initialized 1×1 convolution, which makes the compensator output only the bounded prior noise at the start of training, i.e., it acts as the identity up to this noise. We use 4 Euler steps for flow matching at both training and inference, with bound scale m=0.1 (Eq. 10) and prior noise standard deviation $\sigma = 0.029 \approx m/\sqrt{12}$ (Section 3.3.1).

The default compensator has 31.0M parameters. The compensator-size ablation (**v4.1** and **v4.2** in Table 2) uses two smaller compensators that change only the hidden width and keep the depth, kernel size, bound and training recipe: PerQ-$0.5\times$ with a hidden width of 384 (18.1M parameters) and PerQ-$0.25\times$ with a hidden width of 256 (8.8M parameters).

Discriminator. The adversarial term uses a multi-scale rate-point-conditioned patch discriminator [43] with 3 spatial scales (input downsampled by $1\times, 2\times, 4\times$ via average pooling) and a separate per-rate-point parameterization (per-rate kernels at each conv layer). Each scale uses 3 strided convolutional layers with base width 48, doubled at each layer (capped at 512), followed by a 4×4 stride-1 conv and a final 4×4 patch logit head. All convolutions except the final logit head use spectral normalization [45] and LeakyReLU(0.2) activations. The discriminator is trained jointly with the compensator, where both are updated at every step by the same AdamW optimizer, with hinge losses for the discriminator and the generator.

Backbones of the generality study. The three backbones of Table 4 are all trained from scratch on the MLIC training set with the Stage 1 objective, 64 rate points and the consolidated bottleneck

quantization of Section 3.3.2. DCVC-RT is the default backbone described above. ELIC [20] keeps its original architecture (320 latent channels in five uneven groups) and follows the DCVC-RT recipe, except that λ is sampled from $[32, 4096]$. MLIC++ [28] keeps its original architecture (320 latent channels in 10 slices) with $\lambda \in [32, 4096]$. It is first trained for 1M steps on 256×256 crops at batch size 64 with a peak learning rate of 10^{-4}, and then fine-tuned for 400K steps on 512×512 crops at batch size 32, with a peak learning rate of 10^{-4}, 5K warmup steps, no rate or λ warmup, and the learning rate halved at 37.5%, 75%, 87.5%, and 93.75% of the fine-tuning. Since $\lambda \geq 32$, ELIC and MLIC++ do not reach the lowest rates of DCVC-RT (see the bpp ranges in Table 4). On each backbone, the compensator and all Stage 2 settings are those of the default model, except that λ follows the range of the backbone and MLIC++ is trained on 512×512 crops.

C Additional quantitative results

Tables 5 and 6 extend Table 1 with the bitrate range of every codec and with the BD-rate in KID [7], computed on the same 256×256 patches as FID. Since KID uses an unbiased estimator, it is less affected than FID by the small number of patches on Kodak (192), although its estimate can fall slightly below zero when the reconstructions are statistically close to the source. Figure 6 shows the corresponding rate-KID curves next to the rate-PSNR curves. The KID results follow those of FID: on Kodak, PerQ (-84.1%) is within 2.5 points of HiFiC (-86.6%) and Control-GIC (-85.7%), and on CLIC2020-test it surpasses Control-GIC (-95.0% vs. -90.8%), while keeping its PSNR BD-rate within 3% of VVC.

Table 5: **BD-rate against VVC on Kodak (%, lower is better) with bitrate ranges and KID.** Same anchor, fitting method and N/A rule as Table 1. The bpp range covers the rate points evaluated for each codec.

Codec	Single-model multi-rate	bpp range	PSNR	LPIPS	DISTS	FID	KID
Conventional codec							
VTM intra [10]		0.113–2.316	0	0	0	0	0
VAE-based codecs							
ELIC [20]		0.044–0.865	−2.0	−2.1	+2.4	+7.3	+3.8
EVC [56]	✓	0.329–0.983	+2.0	−9.0	−5.4	−4.0	−7.5
DCVC-RT-Intra [25]	✓	0.031–0.913	−11.2	−8.7	−2.3	−1.8	−2.8
GAN-based perceptual codecs							
MS-ILLM [46]		0.0045–0.708	+85.9	−86.4	−82.9	−86.3	−95.9
HiFiC [43]		0.149–0.468	+128.6	−69.8	−59.0	−64.3	−86.6
Control-GIC[§] [34]	✓	0.087–0.568	+372.4	−74.0	−68.5	−69.7	−85.7
Diffusion-based perceptual codecs							
PerCo [12]		0.0019–0.125	N/A	−64.1	−84.4	−92.9	−99.0
DiffEIC [36]		0.020–0.123	N/A	−81.0	−88.8	−91.3	−97.6
CDC [59]		0.232–0.530	+318.7	−48.6	−71.7	−68.8	−85.9
StableCodec [63]		0.012–0.035	N/A	−92.7	−96.2	−96.8	−99.0
OSCAR [18]	✓	0.0019–0.125	N/A	−79.0	−91.8	−92.3	−97.8
PerQ (Ours)	✓	0.010–0.875	**−1.6**	**−57.4**	**−63.2**	**−67.9**	**−84.1**

D More visual results

Figures 7, 8, and 9 compares the visual quality of the backbone models (DCVC-RT [25] and MLIC++ [28]) with PerQ. In all examples, PerQ significantly improve the visual quality, like synthesizing the texture which are missing in the original backbone outputs, and improving the the sharpness of the images.

Additional visual comparison results for comparing with baseline methods are provided in Figures 10, 11. 10 provides more examples for the comparison with the frozen backbone and other generative-based codecs. In all of them, the compensator reduces LPIPS by 53–68% (e.g., $0.279 \rightarrow 0.088$ for the lamp) at a cost of only 0.6–0.9 dB in PSNR. The generative codecs at similar bitrates fall

Table 6: **BD-rate against VVC on CLIC2020-test (%, lower is better) with bitrate ranges and KID.** Same protocol as Table 5.

Codec	Single-model multi-rate	bpp range	PSNR	LPIPS	DISTS	FID	KID
Conventional codec							
VTM intra [10]		0.076–1.717	0	0	0	0	0
VAE-based codecs							
ELIC [20]		0.032–0.538	−1.9	−1.3	+9.4	−13.8	−31.6
EVC [56]	✓	0.211–0.626	+3.7	−8.7	−2.3	−6.2	−11.9
DCVC-RT-Intra [25]	✓	0.024–0.562	−11.8	−2.5	+8.8	−30.1	−39.7
GAN-based perceptual codecs							
MS-ILLM [46]		0.0035–0.506	+89.9	−91.0	−90.7	−97.4	−98.7
HiFiC [43]		0.107–0.336	+116.3	−81.1	−76.9	−83.0	N/A
Control-GIC[§] [34]	✓	0.087–0.567	+1045.8	−65.6	−73.8	−85.2	−90.8
Diffusion-based perceptual codecs							
PerCo [12]		0.0019–0.125	N/A	N/A	N/A	−94.3	−94.5
DiffEIC [36]		0.014–0.108	N/A	−84.6	−87.4	−96.5	−97.6
CDC [59]		0.172–0.405	N/A	−63.3	−74.7	−86.8	N/A
StableCodec [63]		0.0086–0.031	N/A	−94.9	−97.1	−99.0	N/A
OSCAR [18]	✓	0.0018–0.115	N/A	−80.2	−88.7	−96.0	−96.8
PerQ (Ours)	✓	0.0088–0.539	**+2.7**	**−70.6**	**−77.4**	**−91.9**	**−95.0**

3.2–12.0 dB below PerQ in PSNR, and their error maps show large deviations from the source in textured regions. Figure 11 follows two images across rates. As the rate grows, the error of PerQ fades steadily and its PSNR rises from 26.8 to 38.5 dB and from 27.0 to 36.5 dB, whereas Control-GIC and OSCAR stay below 25 dB.

E Limitations and Societal Impacts

Limitations. Alongside the positive characteristics, limitations still remain with PerQ. First, it does not reach the highest perceptual quality attained by fully generative codecs, although it remains very competitive on distortion-based metrics and has achieved a better trade-off between fidelity and realism. Moreover, our design relies on a frozen distortion-optimized backbone and jointly fine-tuning the backbone may further improve performance. A unified theoretical framework that jointly optimizes the rate-distortion-perception trade-off should be investigated as future work.

Societal impacts. This paper proposes an advanced general-purpose learned image codec that can potentially reduce image storage requirements while maintaining both high fidelity and perceptual quality. This capability may also help reduce the carbon footprint of a wide range of related applications.

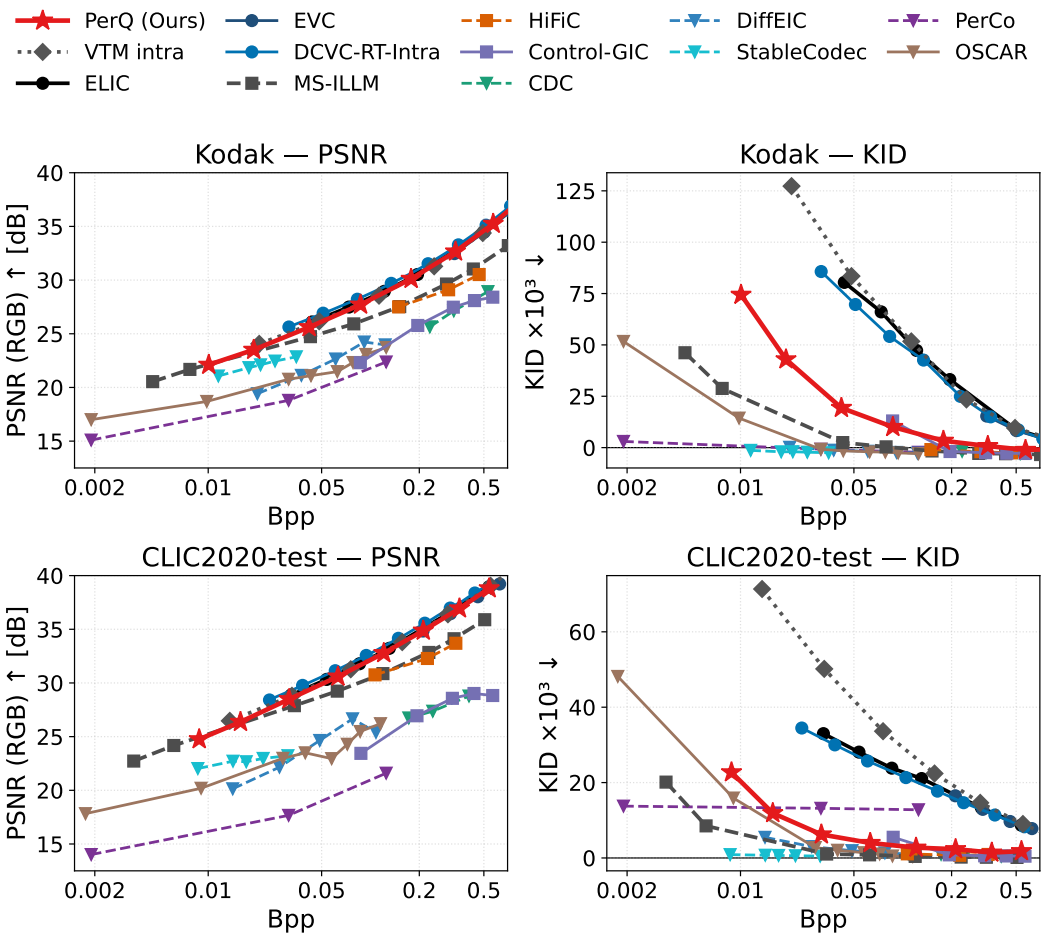

Figure 6: **Rate-PSNR and rate-KID curves on Kodak (top) and CLIC2020-test (bottom).** KID ($\times 10^3$, lower is better) is computed on the same 256×256 patches as FID.

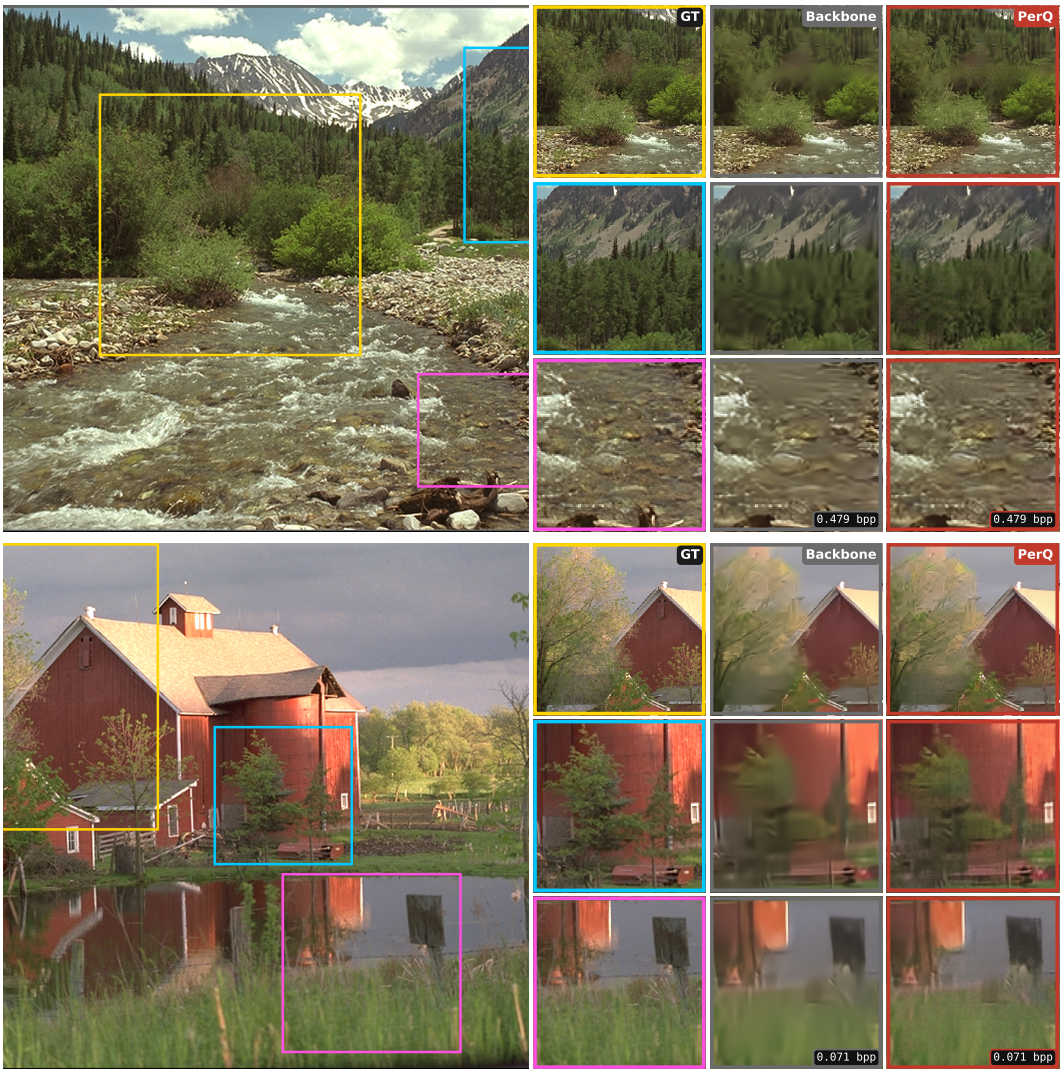

Figure 7: **Visual comparison between the backbone and PerQ on Kodak**, with the DCVC-RT (top) and MLIC++ (bottom) backbones. Additional examples on CLIC2020-test are shown in Figure 8 (DCVC-RT) and Figure 9 (MLIC++).

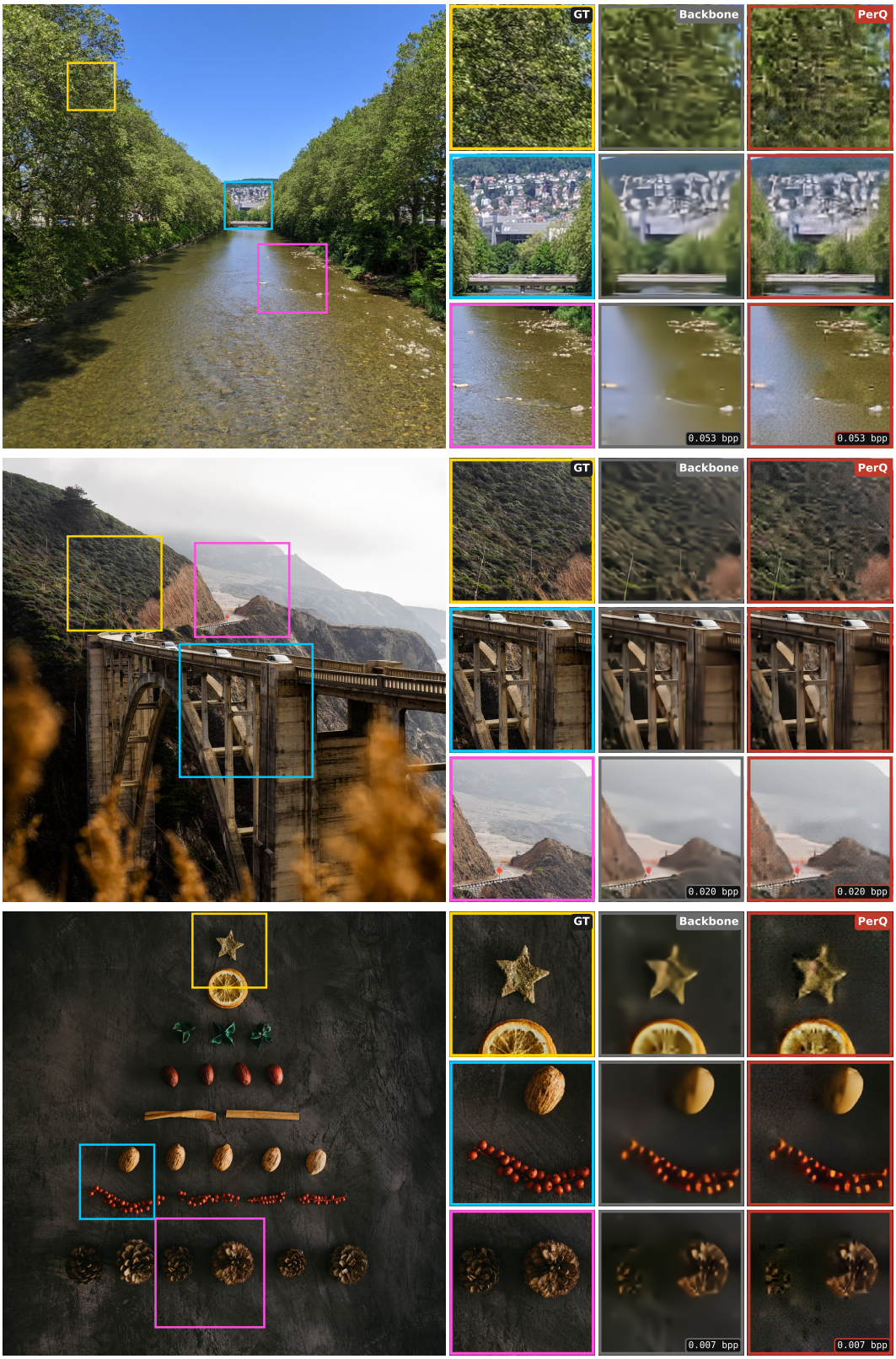

Figure 8: **Visual comparison between the backbone and PerQ on CLIC2020-test** with the DCVC-RT backbone.

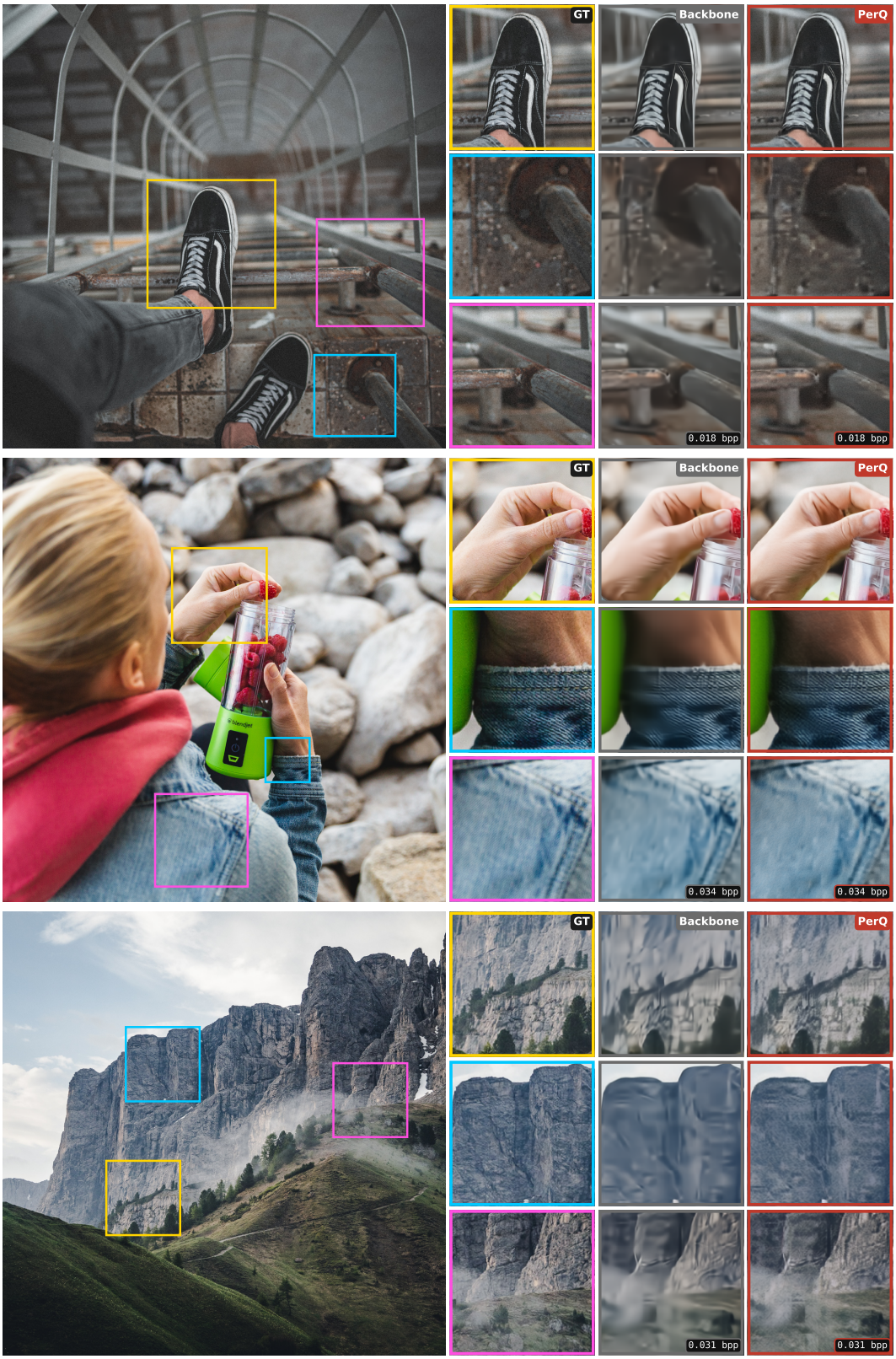

Figure 9: **Visual comparison between the backbone and PerQ on CLIC2020-test** with the MLIC++ backbone of the generality study (Table 4).

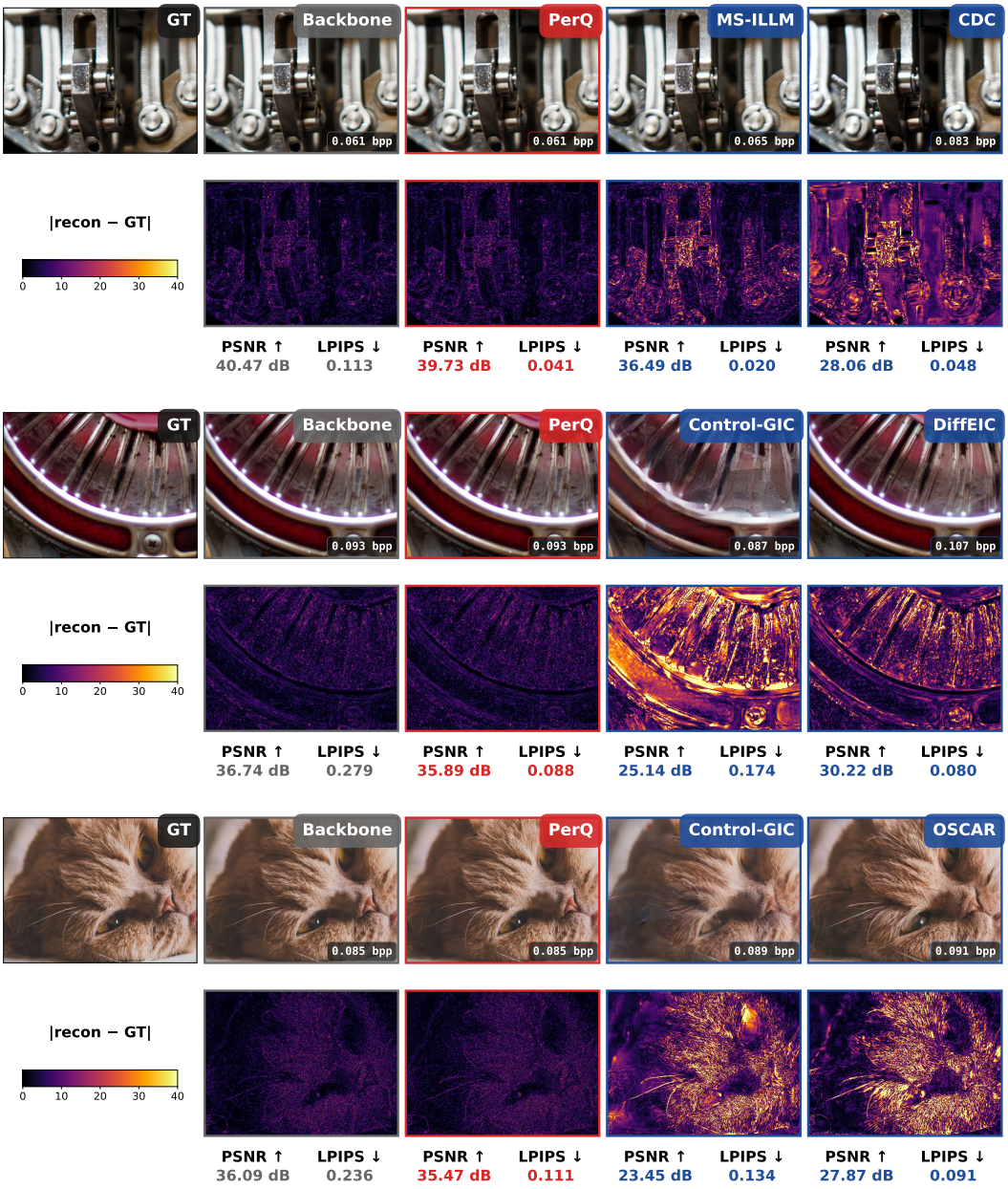

Figure 10: **Additional visual comparison between PerQ and baseline methods on CLIC2020-test [14].** Source crop (left) compared with reconstructions from our frozen backbone (grey), PerQ (red, ours) at the same rate point, and two generative codecs, GAN-based and diffusion-based (blue), at similar bitrates. Below each reconstruction, the error map shows its absolute difference to the source on a shared color scale, and the numbers give the PSNR and LPIPS of the whole image.

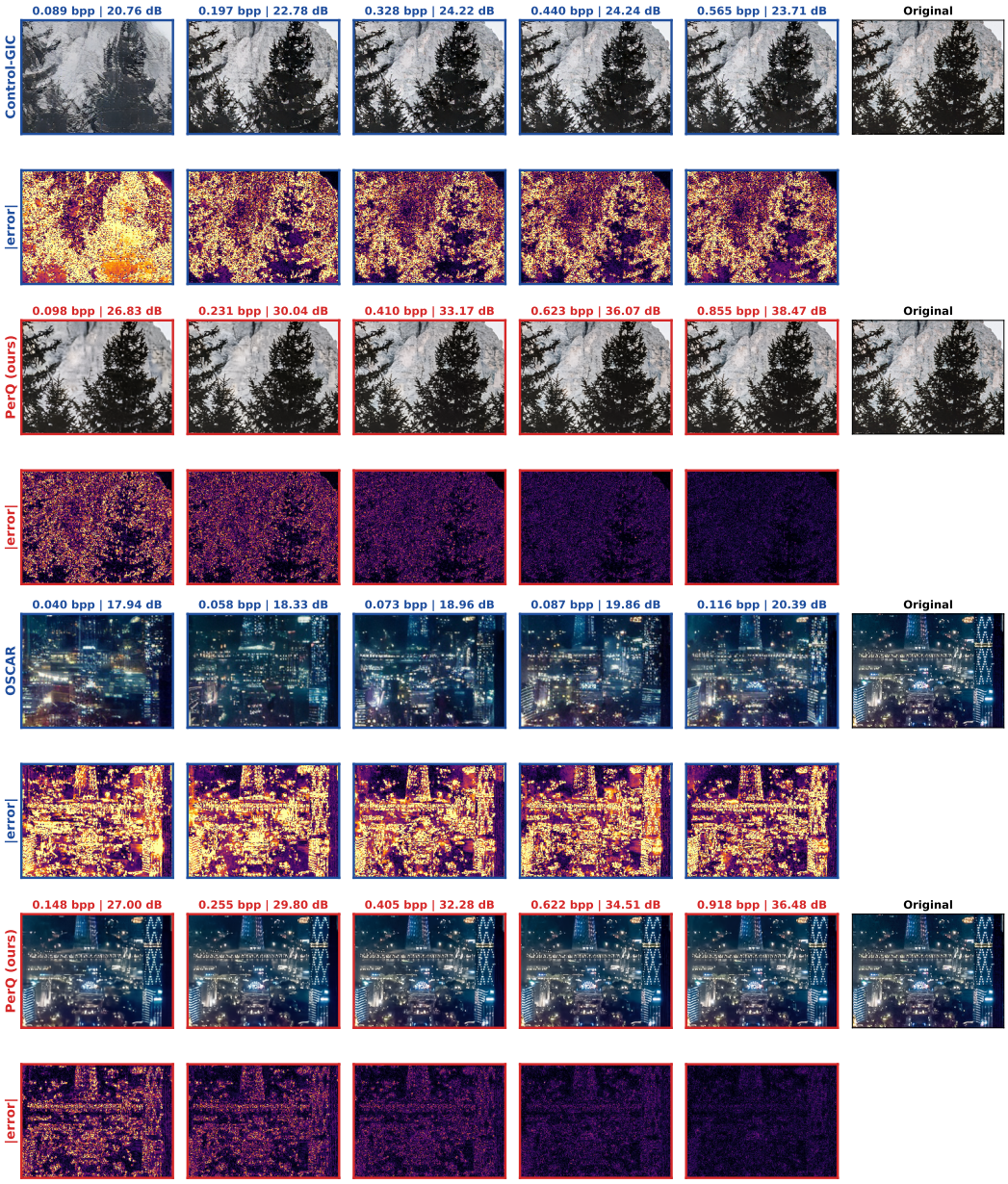

Figure 11: **Reconstructions across rates on CLIC2020-test [14].** Generative codecs (blue) and PerQ (red) at their own rate points, with the absolute error map below each crop. The PSNR of PerQ keeps rising with the rate, while Control-GIC and OSCAR stay below 25 dB.

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 claims in the abstract and introduction accurately reflect the paper's contributions and scope: a transport bound on the perception-optimal reconstruction and the PerQ codec instantiating it , with empirical results.

   Guidelines:

   - 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: We have mentioned the limitation of this work in the paper.

   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: We clearly stated/referenced all assumptions and proofs in the paper and *Appendix*.

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: We did our best to disclose all the information required to reproduce the main experimental results of the paper.

   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: [Yes]

Justification: We will release the code and the synthetic datasets to the public.

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: We clearly specify all the training and test details in the paper.

   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: [N/A]

   Justification: This does not apply to the results in this work.

   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: We did provide the information related to the hardware resources used for generating the results.

   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 reported in this paper does conform with the NeurIPS Code of Ethics, in every respect.

   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: [Yes]

    Justification: We have discuss the societal impact in the Appendix.

    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: We did not identify such risks with this work.

    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: We have mentioned and cited the code/data/models used in this paper, and respected the license and terms while using them.

    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: [Yes]

    Justification: The PerQ implementation, configurations and trained weights are released at `https://github.com/jasminepp/PerQ` under the MIT licence.

    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: We did not perform experiments involving 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: We did not perform experiments involving human subjects.

    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 LLMs as any important, original, or non-standard components.

    Guidelines: The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.

    - 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.