CALIBRATING PRETRAINED GENERATIVE MODELS FOR PRACTICAL REVERSE-CHANNEL CODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Pretrained diffusion and flow models can be used for image compression through reverse-channel coding (RCC), but mismatch between the deployed receiver and the source-marginalized ideal receiver can increase communication cost. We decompose ideal expected RCC cost into required source information and receiver-mismatch excess, identifying the part that calibration can reduce. This analysis leads to Calibrated Reverse-Channel Coding (CalRCC), which calibrates prediction, confidence, and path. Prediction balances coding cost and reconstruction quality, confidence adjusts receiver uncertainty, and path reallocates model evaluations along the reverse process. Experiments with controlled DiffC show that CalRCC substantially improves practical compression, with stronger fidelity and perceptual quality without training a new generative backbone. Studies with CoD and Turbo-DDCM further suggest that the principle of prediction calibration can extend across distinct diffusion codec designs. These results show that receiver calibration provides an effective way to adapt pretrained generative models for practical compression.

1 INTRODUCTION

Pretrained diffusion and flow models provide powerful image priors that can be used for compression without training a new generative backbone. Reverse-channel coding (RCC) makes this possible by letting the sender and receiver share a generative model and randomness, while the sender transmits short messages that allow the receiver to reproduce source-conditioned states along a reverse process (Theis et al., 2022; Vonderfecht & Liu, 2025). Practical DiffC implements this approach using finite-candidate Poisson functional representation (PFR) coding (Vonderfecht & Liu, 2025).

However, the sender and receiver have different information. The sender knows the clean source representation and can form the source-conditioned reverse transition, whereas the receiver observes only the current state and uses the pretrained model to estimate the source and form its transition distribution. Ideally, the receiver would average over the unknown source conditioned on the current state; the deployed receiver only approximates this distribution, creating a mismatch that increases coding cost along the reverse path. Moreover, a good clean-source estimate alone is not sufficient: the receiver must also represent its uncertainty and decide where along the reverse process to evaluate the model. We show that the ideal expected communication cost separates into required source information and a receiver-mismatch excess, isolating the part that can be reduced by improving the receiver. The conditional mean is optimal for predicting the clean source representation and for the corresponding Gaussian coding term, whereas the prediction that yields the best final image quality at the receiver can differ. This motivates prediction calibration to balance communication and image quality. In addition, even an exact conditional-mean prediction leaves uncertainty about the source, motivating confidence calibration, while the placement of model evaluations along the reverse process motivates path calibration.

Building on this decomposition, we introduce Calibrated Reverse-Channel Coding (CalRCC), which calibrates three receiver choices: the estimate of the clean source representation, the uncertainty around that estimate, and the model-evaluation path. Prediction calibration uses low-rank adapters to optimize the pretrained clean-source estimator for coding cost and image quality; confidence calibration adjusts the receiver covariance; and path calibration reallocates model evaluations along the communicated reverse trajectory. The pretrained backbone and VAE remain frozen, and confidence

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

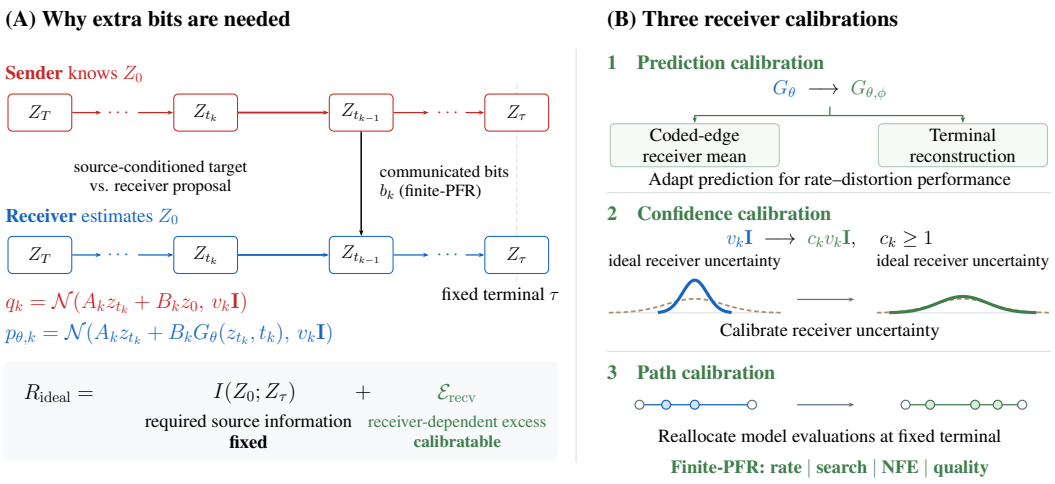

Figure 1: Receiver calibration for practical RCC. The sender knows Z_0, while the receiver estimates it from the current state. Coded messages let both sides reproduce the same selected reverse-transition samples. Ideal communication decomposes into required source information and receiver-dependent excess. CalRCC targets this excess through prediction, confidence, and path calibration: prediction balances coding and final image quality, confidence adjusts receiver uncertainty, and path reallocates model evaluations.

and path require no neural-weight updates. We instantiate CalRCC for fidelity (CalRCC-RD) and perceptual quality (CalRCC-Perceptual). The closed-form confidence calibration transfers to held-out images and also reduces serialized communication in the finite-PFR codec. In controlled DiffC comparisons, CalRCC substantially improves fidelity and perceptual quality, with about 50% bitrate savings at equal PSNR for the fidelity-oriented configuration on DIV2K and Kodak and consistent fidelity gains on the separate MCL-JCI dataset. We further study CoD and Turbo-DDCM, where the idea of prediction calibration also improves codec-specific communication or reconstruction objectives.

Specifically, our contributions are as follows:

- We characterize the receiver-dependent communication cost of RCC and derive a closed-form shared confidence calibration rule, together with its ideal-KL saving and dependence on path resolution.
- We develop CalRCC to calibrate prediction, confidence, and path while keeping the pre-trained generative backbone and VAE frozen; only prediction calibration uses low-rank adapters.
- We demonstrate about 1.6 dB matched-rate PSNR gains over controlled DiffC and about 50% bitrate savings at equal PSNR on DIV2K and Kodak. The perceptual variant retains strong fidelity while substantially improving perceptual quality, and CoD/Turbo-DDCM studies further test the idea of prediction calibration across different codec structures.

2 Preliminaries

2.1 Diffusion Models and Latent Representations

Latent diffusion performs the diffusion process in the representation space of a variational autoencoder (VAE) rather than directly in pixels (Ho et al., 2020; Rombach et al., 2022). We use the frozen encoder E_{VAE} to obtain a deterministic clean source latent Z_0 from an image X, and the frozen decoder D_{VAE} to reconstruct an image after latent-space denoising. We consider the variance-preserving (VP) Gaussian marginals

$$Z_t = \alpha_t Z_0 + \sigma_t \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \mathbf{I}), \qquad \alpha_t^2 + \sigma_t^2 = 1, \tag{1}$$

where $t = 0$ is clean, $\gamma_t = \alpha_t^2/\sigma_t^2$ denotes the signal-to-noise ratio (SNR), and the analytical endpoint $T = 1$ is idealized as pure noise with $\alpha_T = 0$ and $\sigma_T = 1$. The reverse process evolves

noisy latent states toward a clean-latent estimate $\hat{Z}_0$, which is mapped back to image space by the VAE decoder as $\hat{X} = D_{\mathrm{VAE}}(\hat{Z}_0)$.

2.2 Image Compression with Diffusion Models via Reverse-Channel Coding

The reverse process can be used for lossy compression through reverse-channel coding (RCC), where the sender and receiver share randomness and the sender communicates short source-dependent messages (Theis et al., 2022). Starting from the same source-independent noisy latent, the sender uses its knowledge of the clean source latent Z_0 to guide each communicated reverse transition. Communication stops at a terminal latent Z_τ, from which the image is reconstructed without further source bits.

To formalize these communicated reverse transitions, let

$$\mathcal{T} = \{T = t_K > t_{K-1} > \cdots > t_m = \tau > 0\} \tag{2}$$

denote the shared sequence of noise levels at which RCC is applied. The sender and receiver traverse $K - m$ edges $Z_{t_k} \to Z_{t_{k-1}}$ for $k = K, \ldots, m+1$, starting in the ideal setting from the source-independent state $Z_T \sim \mathcal{N}(0, \mathbf{I})$ and stopping at Z_τ. For the VP marginals in (1), the source-conditioned posterior transition on edge k can be written as

$$q_k(z_{t_{k-1}} \mid z_{t_k}, z_0) = \mathcal{N}(A_k z_{t_k} + B_k z_0,\, v_k \mathbf{I}), \tag{3}$$

where A_k, B_k, v_k are the corresponding posterior coefficients; see Appendix B.1 for details. The sender can evaluate this transition since it knows the clean source latent Z_0.

In contrast, the receiver does not know Z_0. DiffC constructs the receiver by replacing Z_0 with a clean-latent estimate $G_\theta(Z_t, t)$ obtained from the pretrained model's native prediction, directly or through the corresponding noise- or velocity-to-clean conversion (Theis et al., 2022; Vonderfecht & Liu, 2025). This gives the matched-covariance receiver

$$p_{\theta,k}(z_{t_{k-1}} \mid z_{t_k}) = \mathcal{N}(A_k z_{t_k} + B_k G_\theta(z_{t_k}, t_k),\, v_k \mathbf{I})\,, \tag{4}$$

which exposes three receiver choices: $G_\theta(z_{t_k}, t_k)$ for prediction, $v_k \mathbf{I}$ for confidence, and $\{t_k\}$ for the evaluation path. The same predictor also provides the terminal estimate $G_\theta(z_\tau, \tau)$, linking prediction to both communication and final image quality.

2.3 Finite-Candidate PFR Coding

We instantiate each RCC transition using a finite-candidate *Poisson functional representation* (PFR), with q_k as the source-conditioned target and $p_{\theta,k}$ as the proposal known to the receiver (Li & El Gamal, 2018; Theis & Ahmed, 2022; Vonderfecht & Liu, 2025). Sender and receiver generate the same proposal candidates from shared randomness; the sender scores them using the target-to-proposal density ratio and transmits the selected candidate's identity rather than its coordinates. We define R_{ideal} as the summed expected edge KL under the fixed ideal Gaussian coupling; the reported rate R_{actual} instead counts the serialized finite-PFR payload and decode-required metadata normalized by image area. Throughout this paper, KL quantities are expressed in nats and converted to bpp by dividing by $\ln 2$ and the image area.

3 Communication Calibration

3.1 What communication cost is avoidable?

Receiver calibration acts on the excess communication induced by the deployed receiver. On each edge, the expected Kullback–Leibler (KL) divergence $\mathbb{E}[D_{\mathrm{KL}}(q_k \| p_k)]$ is the ideal communication quantity for reproducing the source-conditioned transition q_k using receiver law p_k. Its sum R_{ideal} separates required source information from receiver-dependent excess. The receiver observes Z_{t_k} but not Z_0. Marginalizing the unavailable source gives

$$\bar{q}_k(z_{t_{k-1}} \mid z_{t_k}) = \int q_k(z_{t_{k-1}} \mid z_{t_k}, z_0) p(z_0 \mid z_{t_k})\, dz_0. \tag{5}$$

The expected edge KL decomposes as

$$\begin{aligned}&\mathbb{E}\Big[D_{\mathrm{KL}}\big(q_k(\cdot \mid Z_{t_k}, Z_0) \,\|\, p_k(\cdot \mid Z_{t_k})\big)\Big] \\ &\quad = I(Z_0; Z_{t_{k-1}} \mid Z_{t_k}) + \mathbb{E}\Big[D_{\mathrm{KL}}\big(\bar{q}_k(\cdot \mid Z_{t_k}) \,\|\, p_k(\cdot \mid Z_{t_k})\big)\Big].\end{aligned} \tag{6}$$

The first term is source information that must be conveyed on the edge; the second is receiver-mismatch excess. The excess is nonnegative and vanishes for $p_k = \bar{q}_k$, making $\bar{q}_k$ the ideal receiver given Z_{t_k}.

Summing (6) over all communicated edges gives the total ideal communication cost. Recall that communication terminates at the noise level τ. Under the Markov coupling, the source-information terms telescope as $I(Z_0; Z_{t_{k-1}} \mid Z_{t_k}) = I(Z_0; Z_{t_{k-1}}) - I(Z_0; Z_{t_k})$, leaving $I(Z_0; Z_\tau) - I(Z_0; Z_T)$. As Z_T is source-independent, this reduces to $I(Z_0; Z_\tau)$. The receiver-mismatch terms add across edges. Hence, we have

$$R_{\mathrm{ideal}}(\{p_k\}, \mathcal{T}; \tau) = I(Z_0; Z_\tau) + \mathcal{E}_{\mathrm{recv}}(\{p_k\}, \mathcal{T}), \tag{7}$$

where $\mathcal{E}_{\mathrm{recv}} = \sum_{k=m+1}^{K} \mathbb{E}[D_{\mathrm{KL}}(\bar{q}_k \| p_k)]$. With the source law, representation, terminal, and coupling fixed, $I(Z_0; Z_\tau)$ is fixed, whereas $\mathcal{E}_{\mathrm{recv}}$ depends on the deployed receiver and the intermediate path. Appendix A.1 gives the full derivation.

For the matched-covariance receiver in (4), q_k and $p_{\theta,k}$ share covariance $v_k \mathbf{I}$. Using $B_k^2 / v_k = \Delta\gamma_k$ from Appendix B.1 gives

$$D_{\mathrm{KL}}(q_k \| p_{\theta,k}) = \frac{\Delta\gamma_k}{2} \left\| Z_0 - G_\theta(Z_{t_k}, t_k) \right\|_2^2, \qquad \Delta\gamma_k = \gamma_{t_{k-1}} - \gamma_{t_k} > 0, \tag{8}$$

where γ_t denotes the SNR defined in (1). Its expectation contains both required source information and receiver mismatch. Thus, (8) links prediction error and evaluation placement to ideal coding cost, while confidence calibration additionally adjusts receiver covariance.

3.2 Prediction Calibration

Prediction calibration adapts the clean-latent predictor used throughout the codec to balance receiver coding cost against terminal reconstruction quality, while leaving the sender target q_k unchanged. We use LoRA (Hu et al., 2022) to parameterize the adapted predictor, keeping the pretrained backbone and VAE frozen:

$$G_\theta(z_t, t) \longrightarrow G_{\theta,\phi}(z_t, t), \qquad \mu_{p,k}^{\phi}(z_{t_k}) = A_k z_{t_k} + B_k G_{\theta,\phi}(z_{t_k}, t_k). \tag{9}$$

The same $G_{\theta,\phi}$ determines every coded-edge receiver mean and, after communication stops, the terminal estimate $\widehat{Z}_0 = G_{\theta,\phi}(Z_\tau, \tau)$ decoded as $\widehat{X} = D_{\mathrm{VAE}}(\widehat{Z}_0)$. This dual role makes prediction calibration a rate–distortion problem.

For each terminal τ and fixed training path $\mathcal{T}_\tau$, **CalRCC-RD** trains a separate adapter ϕ_τ by minimizing

$$\mathcal{L}_{\mathrm{RD}}(\phi_\tau) = \lambda_{R,\tau} R_{\mathrm{ideal}}(\phi_\tau; \mathcal{T}_\tau) + \lambda_D D_{\mathrm{RGB}}(\phi_\tau; \mathcal{T}_\tau), \tag{10}$$

where $\lambda_{R,\tau}$ is the terminal-specific rate weight selected on development data, λ_D is shared across terminals, and D_{RGB} is mean squared RGB reconstruction error. Here, R_{ideal} is normalized to bits per source pixel. With the source law, representation, terminal, and ideal coupling fixed, only the receiver-mismatch component of R_{ideal} depends on ϕ_τ. The prediction that minimizes the Gaussian coding term can differ from the one that gives the best decoded-image quality, making prediction calibration a rate–distortion problem; Appendix A.4 formalizes this distinction locally.

CalRCC-Perceptual extends the same objective with a perceptual term. For each terminal, the adapter is initialized from the common distortion-oriented predictor and optimized as

$$\mathcal{L}_{\mathrm{perc}}(\phi_\tau) = \lambda_{R,\tau}^{\mathrm{perc}} R_{\mathrm{ideal}}(\phi_\tau; \mathcal{T}_\tau) + \lambda_D D_{\mathrm{RGB}}(\phi_\tau; \mathcal{T}_\tau) + \beta_\tau D_{\mathrm{VGG}}(\phi_\tau; \mathcal{T}_\tau), \tag{11}$$

where D_{VGG} is LPIPS with a frozen VGG backbone (Zhang et al., 2018). The terminal-specific rate weight $\lambda_{R,\tau}^{\mathrm{perc}}$ and perceptual weight β_τ are selected on development data, while λ_D is shared across terminals.

To optimize these objectives, we sample Gaussian edge states directly from source latents. A weighted sampled-edge KL estimates the ideal communication term, while an independently noised terminal state provides the reconstruction loss through the frozen VAE decoder. Finite-PFR coding is used only for configuration selection and evaluation, without differentiating through the finite-PFR candidate-selection procedure. Further training details are given in Appendix C.2.

3.3 Confidence calibration

Confidence calibration adjusts communication cost by changing receiver uncertainty at a fixed predictor and mean. Residual source uncertainty and prediction error motivate inflating the baseline covariance $v_k\mathbf{I}$ with a scalar $c_k \geq 1$ shared across images:

$$p_{\theta,\phi,c,k}(z_{t_{k-1}} \mid z_{t_k}) = \mathcal{N}\Big(\mu^{\phi}_{p,k}(z_{t_k}), c_k v_k \mathbf{I}\Big). \tag{12}$$

For one edge, let $q = \mathcal{N}(\mu_q, v\mathbf{I})$ and $p_c = \mathcal{N}(\mu_p, cv\mathbf{I})$. Also, define $a = \|\mu_q - \mu_p\|_2^2/v$ as the squared mean mismatch normalized by the baseline variance, and let $M = \mathbb{E}[a]$ and d denote the expected normalized mean mismatch and the latent dimension, respectively. With the receiver mean μ_p fixed, increasing c leaves the mean mismatch $\|\mu_q - \mu_p\|_2^2$ unchanged but reduces its impact on KL. Excessive variance inflation, however, also increases KL.

Proposition 1 (KL-optimal variance inflation)**.** *For a fixed receiver mean μ_p, expected KL within the isotropic Gaussian family $p_c = \mathcal{N}(\mu_p, cv\mathbf{I})$ is minimized over $c \geq 1$ by*

$$c^*_{\mathrm{KL}} = 1 + \frac{M}{d}, \tag{13}$$

and its exact saving relative to $c = 1$ is

$$\Delta_{\mathrm{unc}}(M) = \frac{1}{2}\left[M - d\log\left(1 + \frac{M}{d}\right)\right] \geq 0. \tag{14}$$

Thus, the baseline matched-covariance receiver $c = 1$ is optimal only when $M = 0$; larger M favors greater variance inflation. M captures not only prediction error but also unresolved source uncertainty. Therefore, $c^*_{\mathrm{KL}} > 1$ can remain optimal even when the predictor exactly matches the conditional mean. The derivation and source-uncertainty interpretation are provided in Appendix A.2.

For each predictor and evaluation schedule, the scalars $\{c_k\}$ are fitted once on training latents and then fixed as part of the shared codec configuration. During coding, finite-PFR uses the calibrated covariance $c_k v_k \mathbf{I}$ while leaving the sender target q_k and receiver mean unchanged. The shared $\{c_k\}$ therefore require no per-image uncertainty side information. Appendix B.2 gives the corresponding target-to-proposal density-ratio rule for finite-PFR candidate selection.

3.4 Path calibration

Path calibration chooses where the model is evaluated along the reverse process while keeping the terminal fixed. The number and placement of communicated edges determine the noise levels t_k and increments $\Delta\gamma_k$ in (8), thereby changing receiver mismatch and computation while leaving $I(Z_0; Z_\tau)$ unchanged. Corollary 1 in Appendix A.3 shows that, under its assumptions, the additional ideal-KL gain from scalar confidence calibration vanishes as the path becomes finer.

For a fixed predictor, we evaluate paths from a fixed candidate family $\mathcal{F}_\tau$. Confidence is fitted separately for each candidate path, and we select on development images

$$\mathcal{T}^\star_\tau \in \arg\min_{\mathcal{T}\in\mathcal{F}_\tau} J_{\mathrm{actual}}(\mathcal{T}). \tag{15}$$

Here J_{actual} combines measured rate and RGB distortion using fixed reference weights, as specified in Appendix C.2. The candidate paths and selected configurations are listed in Appendix B.3; Algorithm 1 summarizes offline calibration and finite-PFR encoding and decoding.

4 Related Work

RCC with pretrained generative models. DiffC communicates Gaussian-corrupted states through a multi-edge reverse-channel coding framework (Theis et al., 2022). Practical DiffC pro-

vides finite-candidate PFR, chunking, timestep-schedule optimization, and pretrained Stable Diffusion/FLUX codecs, including the FLUX-to-DDPM-style mapping (Vonderfecht & Liu, 2025). Few-step work derives RCC quantities for Rectified Flow, CTM, and MeanFlow through velocity/noise or denoiser reparameterization (Kimishima & Zhou, 2026).

Other generative codecs. A broader taxonomy of diffusion-based generative compression is given in (Yang & Mandt, 2026). MS-ILLM studies distortion–perception trade-offs (Muckley et al., 2023), while DiffEIC and PerCo use different priors, side information, and rate regimes (Li et al., 2025; Careil et al., 2024). Turbo-DDCM provides a flexible zero-shot diffusion codec, while StableCodec targets one-step extreme-rate decoding (Vaisman et al., 2026; Zhang et al., 2025). CoD trains a compression-native diffusion foundation model (Jia et al., 2026). Universal-quantization approaches use different channels and coding methods (Yang et al., 2025; Relic et al., 2025).

RCC algorithms. The same shared-randomness sample-communication problem is also studied as relative-entropy coding (REC); we use RCC throughout for consistency. Poisson functional representation (PFR) provides a general construction for communicating samples with shared randomness (Li & El Gamal, 2018), and finite variants are used for diffusion compression (Theis & Ahmed, 2022; Vonderfecht & Liu, 2025). Alternative RCC algorithms use adaptive search, ordered candidates, greedy rejection, or sample-space partitioning for computational efficiency (Flamich et al., 2022; Theis & Ahmed, 2022; Flamich et al., 2023; He et al., 2024).

5 Evaluation Protocol

Our controlled DiffC comparison keeps the source representation, terminal, reconstruction policy, finite-PFR coding, and evaluation setup fixed, while CalRCC calibrates receiver prediction, confidence, and path. Model-call counts are reported separately. CalRCC-RD and CalRCC-Perceptual each use a separately selected rank-16 prediction adapter for each of seven terminals, while the base FLUX.1-dev and VAE weights remain fixed. Both families and controlled DiffC use the same single-pass terminal reconstruction policy, matching DiffC-MSE rather than Practical DiffC's 50-step DiffC-F configuration (Vonderfecht & Liu, 2025). The shared and changed conditions are summarized in Table 8 of Appendix C, and the selected codec configurations are given in Table 5 of Appendix B.3. Appendix F.1 separately studies iterative post-terminal reconstruction.

Data. Training and model selection use DIV2K training and development splits, with a separate held-out set for mechanism analysis. Final evaluation uses 80 DIV2K-1024 images and all 24 native-resolution Kodak images. We additionally evaluate CalRCC-RD on all 50 pristine MCL-JCI images without further adaptation. No final evaluation set is used for model, path, or confidence selection.

Baselines. The controlled baseline is DiffC. Released VTM-24.0, DiffEIC, StableCodec, MS-ILLM, and LALIC implementations are rerun on the same pixels (Joint Video Experts Team, 2026; Li et al., 2025; Zhang et al., 2025; Muckley et al., 2023; Feng et al., 2025). Tier A has self-contained serialized or operationally equivalent rate; Tier B has direct author-coded rate but no self-contained container and is descriptive only. Published or digitized points never enter exact-input comparisons.

Statistics and metrics. Final matched-rate comparisons use only overlapping measured rate support, without extrapolation. Confidence intervals use 10,000 paired-image bootstrap resamples, rebuilding the matched-rate curves in each resample. All methods use the same PSNR, LPIPS, DISTS, CLIP, and Q-Align implementations (Zhang et al., 2018; Ding et al., 2022; Radford et al., 2021; Wu et al., 2024). LPIPS uses AlexNet unless explicitly labeled VGG. LPIPS and DISTS measure source-relative perceptual similarity (Blau & Michaeli, 2019), while Q-Align evaluates no-reference image quality. The separate COCO FID/KID results appear in Table 22 of Appendix E.5.

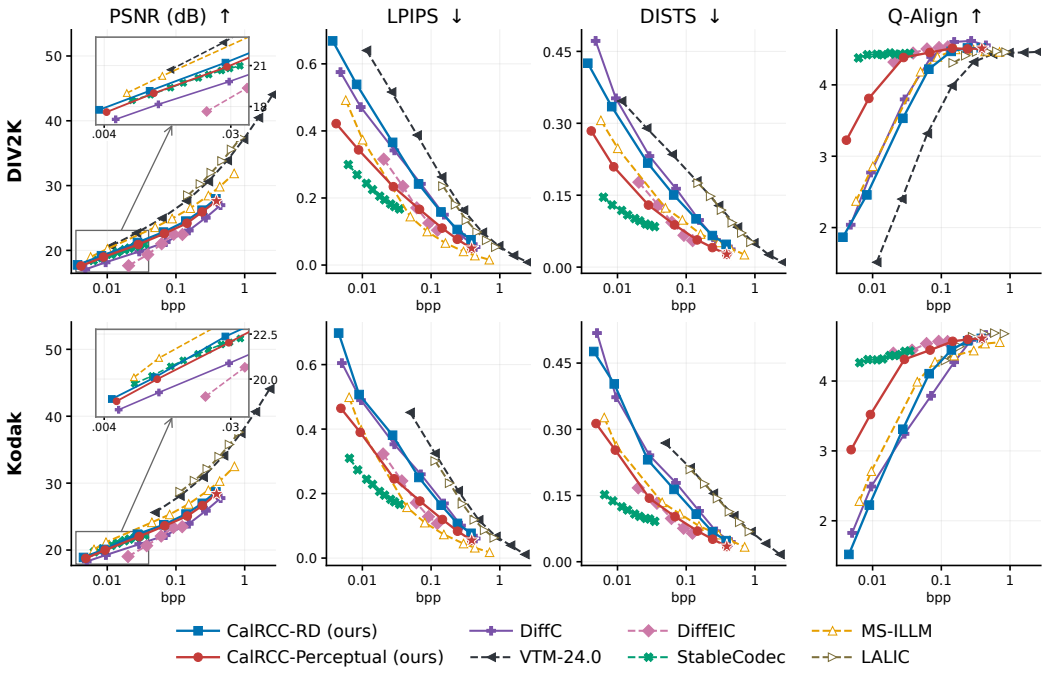

Figure 2: Fidelity and perceptual quality on DIV2K (1024^2, top) and Kodak (native resolution, bottom). Each CalRCC curve contains seven measured points with terminal-specific adapters; insets enlarge the low-rate PSNR region. Red stars mark CalRCC-Perceptual points beyond CalRCC-RD's measured rate support. Open markers indicate descriptive Tier B methods; CLIP results are provided in Fig. 3 of Appendix D.3.

Table 1: BD-rate (%) relative to controlled DiffC. Negative values indicate bitrate savings; bold marks the lower value per dataset and metric.

Dataset	Method	PSNR	LPIPS	DISTS
DIV2K	CalRCC-RD	**−50.13**	+13.21	−20.02
	CalRCC-Perceptual	−41.83	**−58.21**	**−71.14**
Kodak	CalRCC-RD	**−49.34**	+1.43	−7.56
	CalRCC-Perceptual	−41.31	**−53.70**	**−64.84**

6 Results

6.1 Can calibration compete without training a new backbone?

CalRCC-RD consistently improves pixel fidelity over controlled DiffC. Its mean matched-rate PSNR gain is 1.615 dB on DIV2K-1024 (95% CI $[1.550, 1.676]$), 1.545 dB on Kodak ($[1.447, 1.621]$), and 1.575 dB on MCL-JCI ($[1.522, 1.625]$), averaging six supported anchors in each comparison. At all seven measured DIV2K terminals in Fig. 2, it increases PSNR and reduces serialized rate. The fidelity gain persists on the separate 50-image MCL-JCI source set without further adaptation.

CalRCC-Perceptual favors perceptual quality while retaining a strong fidelity gain over DiffC. Relative to CalRCC-RD, it gives up about 0.3 dB of matched-rate PSNR on DIV2K and Kodak while improving LPIPS, DISTS, CLIP, and Q-Align on average, as shown in Fig. 2 and Table 11. At 0.0282 bpp, CalRCC-Perceptual improves PSNR, LPIPS, and DISTS over DiffEIC, while VTM retains higher PSNR and StableCodec lower LPIPS/DISTS, as shown in Table 12 in Appendix D.1. Figs. 9 and 10 in Appendix G illustrate reconstruction differences, including the regular grid pattern in the displayed DiffC images; their RD panels use the shared-adapter RD variant.

Table 2: Component effects on 100 DIV2K development images. Confidence/path rows compare with DiffC; prediction rows compare adapter on/off at fixed path and confidence. Rate and NFE are same-terminal averages, whereas quality is matched in rate within measured support; effects are non-additive. Confidence/path and RD-prediction summaries use seven terminals; perceptual prediction uses three terminals for rate/NFE and two supported terminals for quality.

	Same-terminal changes		Matched-rate quality		
Calibration	Δrate (%)	ΔNFE	ΔPSNR (dB)$\uparrow$	ΔLPIPS$\downarrow$	ΔDISTS$\downarrow$
Confidence	-4.35	$+0.0$	$+0.234$	-0.0084	-0.0059
Path	-6.18	$+13.7$	$+0.252$	-0.0120	-0.0103
Confidence + path	-8.67	$+13.4$	$+0.359$	-0.0141	-0.0102
RD prediction	$+2.25$	$+0.0$	$+0.624$	$+0.0087$	$+0.0114$
Perceptual prediction	$+1.66$	$+0.0$	$+0.610$	-0.1457	-0.1628

Table 3: Matched-rate ΔPSNR (dB)$\uparrow$ from terminal-specific adaptation and configuration selection.

Configuration change	DIV2K	Kodak
Terminal-specific adaptation	$+0.4480$	$+0.4630$
Rate-weight selection	$+0.0203$	-0.0373
Path/confidence refinement	$+0.0097$	$+0.0060$

Table 1 summarizes bitrate savings at equal quality. Relative to controlled DiffC, CalRCC-RD saves about 50% bitrate at equal PSNR on DIV2K and Kodak, while CalRCC-Perceptual achieves substantial savings at equal PSNR, LPIPS, and DISTS. Appendix D.3 gives the full pairwise results and calculation details.

At approximately 0.02824 bpp, single-pass CalRCC-Perceptual retains a 1.28/1.20 dB PSNR advantage over iterative DiffC on DIV2K/Kodak. LPIPS point estimates are close, whereas iterative DiffC achieves lower DISTS and higher Q-Align. Appendix F.1 gives the reconstruction-policy comparison, which holds each codec's bitstream and terminal state fixed and interpolates within measured rate support.

6.2 What does each calibration component contribute?

Across seven matched terminals in Table 2, confidence lowers serialized rate by 4.35% on average without additional model evaluations, while confidence and path together reduce it by 8.67% with more evaluations. Confidence remains useful after prediction adaptation, reducing high-rate serialized rate by 10.7% with NFE unchanged at 27 in Table 17. Proposition 1 determines the confidence factors directly from empirical estimates of M_k; when frozen and evaluated on disjoint COCO images, these factors reduce Gaussian edge KL for pretrained, RD-calibrated, and perceptually calibrated predictors, with the lowest tested KL at the closed-form scale. Path calibration additionally reduces candidate search while using more model evaluations. Appendix E.2 and Appendix E.1 give the corresponding protocols.

Prediction calibration plays a different role. With path and confidence fixed, the RD and perceptual prediction diagnostics increase same-terminal rate by 2.25% and 1.66%, respectively, while improving their targeted quality metrics at matched rates. Table 2 shows that RD prediction improves PSNR while slightly worsening LPIPS and DISTS, whereas perceptual prediction improves all three metrics. At the medium-rate RD setting, prediction spends 10.2% more rate but gains 1.190 dB PSNR and reduces the measured RD objective by 23.5%. Table 17 shows that this prediction-only tradeoff differs from the final terminal-specific codec, which reduces rate while increasing PSNR at all seven DIV2K terminals. Appendix E.1 gives the full controlled comparison.

Switching from the shared-adapter RD setup to the terminal-specific configuration yields a further 0.4480/0.4630 dB matched-rate PSNR gain on DIV2K/Kodak, as shown in Table 3. Training allocation also changes in this comparison, so the gain reflects the overall configuration change rather than adapter sharing alone. Subsequent rate-weight selection and path/confidence refinement produce much smaller, sometimes mixed changes. Appendix E.3 reports the full paired-image comparisons.

6.3 How does the prediction objective change reconstruction?

The prediction objective shapes the fidelity–perception tradeoff. We compare each perceptual adapter with an MSE-trained counterpart matched in initialization, rate weight, and training budget, with confidence fitted separately for each predictor. At the highest-rate comparison, adding the LPIPS-VGG loss improves LPIPS and DISTS at a cost of about 0.34 dB PSNR. These controls isolate the effect of the prediction objective from the separately selected terminal-specific CalRCC-RD family.

Across the measured endpoints, stronger perceptual weighting reduces LPIPS by 23–34% and DISTS by 24–44%, while costing 0.21–0.43 dB PSNR and 0.9–5.2% additional rate. Greater emphasis on pixel fidelity improves PSNR but increases LPIPS and DISTS, while varying the rate weight has smaller effects over the tested range. Appendix E.4 gives the full objective-sensitivity analysis.

6.4 Does prediction adaptation transfer to other codecs?

In CoD (Jia et al., 2026), the coded DiffC edges are unchanged, and prediction adaptation is applied only to the remaining denoising transitions after the decoded terminal state. In Turbo-DDCM (Vaisman et al., 2026), prediction adaptation is applied separately to the coded frontend and the subsequent zero-bit decoder. The frontend can change the transmitted content, while the decoder acts after communication; the comparison in Table 4 isolates decoder adaptation with the adapted frontend held fixed. These codec-specific studies transfer prediction adaptation only; confidence and path calibration are not applied. Fig. 7 in Appendix F.2 summarizes the two implementations.

Table 4: Prediction-adaptation transfer on DIV2K/Kodak. CoD reports post-terminal adaptation at its lowest-rate setting; Turbo reports decoder adaptation with the adapted frontend held fixed. $\Delta = \text{Adapted} - \text{Reference}$.

Codec	ΔPSNR (dB) $\uparrow$	ΔLPIPS $\downarrow$
CoD	$+1.053/+1.308$	$-0.0229/-0.0308$
Turbo	$-0.254/-0.288$	$-0.2512/-0.2593$

CoD improves PSNR and LPIPS at all three tested rates on DIV2K80 and Kodak24. The displayed lowest-rate CoD point uses 0.0112/0.0107 bpp on DIV2K/Kodak, where DISTS increases by 0.0219/0.0105. For Turbo-DDCM, Table 4 compares the adapted and original zero-bit decoders on the same adapted frontend. With the mixed VGG–LPIPS objective, LPIPS and DISTS improve at 0.003273 bpp on both datasets, with a 0.25–0.29 dB PSNR cost. Appendix F.2 gives the full objectives, protocols, and transfer details.

7 Discussion

The full calibration framework could be extended to other generative codecs. Finite-PFR itself also remains uncalibrated: our method adapts the receiver while keeping candidate generation and search fixed, leaving a finite-sampling gap from the ideal transition. Future work could jointly adapt candidate budgets, proposal distributions, and search allocation to reduce this gap and computation. The structured artifacts observed in some DiffC reconstructions motivate testing whether finite-candidate approximation contributes to them and whether sampler calibration can help. The RD rank study further shows that improved reconstruction fidelity does not necessarily improve FID/KID.

8 Conclusion

This paper presents CalRCC, a receiver-calibration framework for adapting pretrained generative models for practical RCC without retraining the generative backbone. By separating required source information from receiver-dependent excess, CalRCC motivates complementary calibration of prediction, confidence, and path to address communication and image quality. The results further suggest that the idea of prediction calibration can extend beyond DiffC to other diffusion codec structures.

REPRODUCIBILITY STATEMENT.

Terminal-specific adapters and codec configurations were selected on development data before final evaluation. Final DIV2K, Kodak, and MCL-JCI images were not used for model, path, or confidence selection. The appendix specifies the training and selection procedures, codec configurations, preprocessing, rate accounting, metric aggregation, and bootstrap protocol.

ETHICS STATEMENT

This work uses public image benchmarks and pretrained generative models for image compression. It does not involve human-subject experiments or private user data. Potential biases and artifacts of the underlying models may also appear in reconstructed images.

AI USE STATEMENT.

Generative AI tools were used to assist with language editing, literature search, code implementation, and refining the presentation of mathematical derivations and figures. All core conceptual ideas, theoretical frameworks, mathematical claims, experimental designs, analyses, and conclusions were developed and independently verified by the authors. The authors take full responsibility for the entire content of the manuscript.

REFERENCES

Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017.

Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018.

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

Marlène Careil, Matthew J. Muckley, Jakob Verbeek, and Stéphane Lathuilière. Towards image compression with perfect realism at ultra-low bitrates. In *International Conference on Learning Representations*, 2024.

Keyan Ding, Kede Ma, Shiqi Wang, and Eero P. Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5):2567–2581, 2022.

Donghui Feng, Zhengxue Cheng, Shen Wang, Ronghua Wu, Hongwei Hu, Guo Lu, and Li Song. Linear attention modeling for learned image compression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Gergely Flamich, Stratis Markou, and José Miguel Hernández-Lobato. Fast relative entropy coding with A* coding. In *International Conference on Machine Learning*, 2022.

Gergely Flamich, Stratis Markou, and José Miguel Hernández-Lobato. Faster relative entropy coding with greedy rejection coding. In *Advances in Neural Information Processing Systems*, 2023.

Jiajun He, Gergely Flamich, and José Miguel Hernández-Lobato. Accelerating relative entropy coding with space partitioning. In *Advances in Neural Information Processing Systems*, 2024.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Zhaoyang Jia, Zihan Zheng, Naifu Xue, Jiahao Li, Bin Li, Zongyu Guo, Xiaoyi Zhang, Houqiang Li, and Yan Lu. CoD: A diffusion foundation model for image compression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Lina Jin, Joe Yuchieh Lin, Sudeng Hu, Haiqiang Wang, Ping Wang, Ioannis Katsavounidis, Anne Aaron, and C.-C. Jay Kuo. Statistical study on perceived JPEG image quality via MCL-JCI dataset construction and analysis. *Electronic Imaging*, 28(13):1–9, 2016.

Joint Video Experts Team. VVCSoftware VTM: VTM-24.0 reference software. `https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tree/VTM-24.0`, 2026. Tag VTM-24.0, commit ca331de.

Fuma Kimishima and Jinjia Zhou. Few-step generative models as lossy compression. *arXiv preprint arXiv:2606.10450*, 2026. URL `https://arxiv.org/abs/2606.10450`.

Cheuk Ting Li and Abbas El Gamal. Strong functional representation lemma and applications to coding theorems. *IEEE Transactions on Information Theory*, 64(11):6967–6978, 2018.

Zhiyuan Li, Yanhui Zhou, Hao Wei, Chenyang Ge, and Jingwen Jiang. Toward extreme image compression with latent feature guidance and diffusion prior. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(1):888–899, 2025.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, 2014.

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

Lucas Relic, Roberto Azevedo, Yang Zhang, Markus Gross, and Christopher Schroers. Bridging the gap between gaussian diffusion models and universal quantization for image compression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

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

Lucas Theis and Noureldin Y. Ahmed. Algorithms for the communication of samples. In *International Conference on Machine Learning*, 2022.

Lucas Theis, Tim Salimans, Matthew D. Hoffman, and Fabian Mentzer. Lossy compression with gaussian diffusion. *arXiv preprint arXiv:2206.08889*, 2022. URL `https://arxiv.org/abs/2206.08889`.

Amit Vaisman, Guy Ohayon, Hila Manor, Michael Elad, and Tomer Michaeli. Turbo-DDCM: Fast and flexible zero-shot diffusion-based image compression. In *International Conference on Learning Representations*, 2026.

Jeremy Vonderfecht and Feng Liu. Lossy compression with pretrained diffusion models. In *International Conference on Learning Representations*, 2025.

Haoning Wu, Zicheng Zhang, Weixia Zhang, Chaofeng Chen, Liang Liao, Chunyi Li, Yixuan Gao, Annan Wang, Erli Zhang, Wenxiu Sun, Qiong Yan, Xiongkuo Min, Guangtao Zhai, and Weisi Lin. Q-Align: Teaching LMMs for visual scoring via discrete text-defined levels. In *International Conference on Machine Learning*, 2024.

Yibo Yang and Stephan Mandt. Advances in diffusion-based generative compression. *IEEE BITS the Information Theory Magazine*, 2026. Early Access.

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

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Tianyu Zhang, Xin Luo, Li Li, and Dong Liu. StableCodec: Taming one-step diffusion for extreme image compression. In *IEEE/CVF International Conference on Computer Vision*, 2025.

APPENDIX

A PROOFS AND THEORETICAL DETAILS

A.1 FIXED-TERMINAL DECOMPOSITION

We first prove the edgewise information/mismatch decomposition and then telescope its source-information terms. All expectations below use the joint source distribution and the fixed DDPM-equivalent Markov communication coupling. By the definition in (5), $\bar{q}_k(\cdot \mid Z_{t_k})$ is the source-marginalized conditional law of $Z_{t_{k-1}}$. Insert it into the edge log-density ratio:

$$\log \frac{q_k(Z_{t_{k-1}} \mid Z_{t_k}, Z_0)}{p_k(Z_{t_{k-1}} \mid Z_{t_k})} = \log \frac{q_k(Z_{t_{k-1}} \mid Z_{t_k}, Z_0)}{\bar{q}_k(Z_{t_{k-1}} \mid Z_{t_k})} + \log \frac{\bar{q}_k(Z_{t_{k-1}} \mid Z_{t_k})}{p_k(Z_{t_{k-1}} \mid Z_{t_k})}. \tag{16}$$

Taking expectation gives (6): the two terms are the source-information increment and receiver-mismatch excess. The same identity holds for any receiver conditional r_k with appropriate support. Its mismatch term is nonnegative and vanishes exactly when $r_k = \bar{q}_k$ almost surely. Thus

$$\bar{q}_k = \arg\min_{r_k} \mathbb{E}\Big[D_{\mathrm{KL}}(q_k(\cdot \mid Z_{t_k}, Z_0) \,\|\, r_k(\cdot \mid Z_{t_k}))\Big], \tag{17}$$

up to almost-sure equality.

The forward degradation order is $Z_0 \to Z_{t_m} = Z_\tau \to \cdots \to Z_{t_K} = Z_T$. Hence $Z_0 \perp Z_{t_k} \mid Z_{t_{k-1}}$, and the mutual-information chain rule gives

$$I(Z_0; Z_{t_{k-1}} \mid Z_{t_k}) = I(Z_0; Z_{t_{k-1}}) - I(Z_0; Z_{t_k}).$$

Summing over $k = m+1, \ldots, K$ gives

$$\sum_{k=m+1}^{K} I(Z_0; Z_{t_{k-1}} \mid Z_{t_k}) = I(Z_0; Z_\tau) - I(Z_0; Z_T) = I(Z_0; Z_\tau), \tag{18}$$

since $\alpha_T = 0$ in the ideal setting makes $Z_T = \epsilon$ source-independent and $I(Z_0; Z_T) = 0$. This proves (7). Nonnegative receiver-excess KL gives $R_{\mathrm{ideal}} \geq I(Z_0; Z_\tau)$, with equality exactly when every communicated-edge excess vanishes, equivalently $p_k = \bar{q}_k$ almost surely on every edge where the expectation is defined. Internal-path invariance of the source-information term requires the same source representation, terminal τ, marginal family, and Markov coupling; it does not apply after changing any of those objects.

A.2 OPTIMAL CONFIDENCE WITH A FIXED RECEIVER MEAN

We hold the receiver mean fixed and optimize a scalar covariance multiplier shared across source images. Expectations in this subsection use the fixed source law and ideal Gaussian coupling of the decomposition above; assume finite second moments of the source and the fixed predictor.

For $q = \mathcal{N}(\mu_q, v\mathbf{I})$ and $p_c = \mathcal{N}(\mu_p, cv\mathbf{I})$, with $c \geq 1$, write $a = \|\mu_q - \mu_p\|_2^2 / v$. The isotropic Gaussian KL is

$$D_{\mathrm{KL}}(q\|p_c) = \frac{1}{2}\left[d\left(\log c - 1 + \frac{1}{c}\right) + \frac{a}{c}\right]. \tag{19}$$

With expected normalized mean mismatch $M = \mathbb{E}[a]$, the objective becomes

$$f(c) = \mathbb{E}[D_{\mathrm{KL}}(q\|p_c)] = \frac{1}{2}\left[d\log c - d + \frac{d+M}{c}\right].$$

Its derivative $f'(c) = [dc - (d+M)]/(2c^2)$ is negative below $c^* = 1 + M/d$ and positive above it. Hence c^* is the unique minimizer on $c \geq 1$, with

$$f(c^*) = \frac{d}{2}\log\left(1 + \frac{M}{d}\right) \leq \frac{M}{2}. \tag{20}$$

Since $f(1) = M/2$, the saving is

$$f(1) - f(c^*) = \frac{1}{2}\left[M - d\log\left(1 + \frac{M}{d}\right)\right] \geq 0,$$

which proves (14). This optimum concerns covariance inflation within the fixed-mean isotropic family.

Source uncertainty and shared confidence. To distinguish source uncertainty from predictor error, let $m_t(z) = \mathbb{E}[Z_0 \mid Z_t = z]$ and $\Sigma_t(z) = \mathrm{Cov}(Z_0 \mid Z_t = z)$. At incoming state z, total covariance in (3) and (5) gives

$$\mathrm{Cov}_{\bar{q}_k}(Z_{t_{k-1}} \mid Z_{t_k} = z) = v_k \mathbf{I} + B_k^2 \Sigma_{t_k}(z).$$

Marginalizing the source adds its conditional variance to the within-source covariance $v_k\mathbf{I}$. Conditional-mean orthogonality removes the prediction-error cross-term. Using $B_k^2/v_k = \Delta\gamma_k$ from (8),

$$M_k = \Delta\gamma_k \Big(\mathbb{E}\big[\mathrm{tr}\, \Sigma_{t_k}(Z_{t_k})\big] + \mathbb{E}\big[\|m_{t_k}(Z_{t_k}) - G_{\theta,\phi}(Z_{t_k}, t_k)\|_2^2\big] \Big).$$

The two contributions are unresolved source uncertainty and conditional-mean approximation error. Even when $G_{\theta,\phi} = m$, positive expected conditional source variance gives $M_k > 0$ and $c^*_{\mathrm{KL},k} > 1$. The shared scalar matches the averaged second moment about the fixed receiver mean, rather than conditioning the variance on each transmitted image; $\bar{q}_k$ itself need not be Gaussian. Required source information remains fixed. These ideal-coupling identities interpret Proposition 1; finite-PFR trajectories and serialized rates remain operational measurements.

A.3 Interaction with path resolution

Corollary 1 (Path–uncertainty interaction)**.** *Fix the source distribution, model, terminal τ, VP coupling, and latent dimension d. Assume a uniform prediction-error bound*

$$\mathbb{E}\big[\|Z_0 - G_\theta(Z_t, t)\|_2^2\big] \le C < \infty \qquad \textit{for all } t \in [\tau, T].$$

Let $\Gamma_\tau = \gamma(\tau) - \gamma(T) < \infty$. For paths whose SNR mesh $\delta = \max_k \Delta\gamma_k$ tends to zero,

$$\begin{aligned} M_k &\le C\Delta\gamma_k, \\ 0 \le c^*_{\mathrm{KL},k} - 1 &= \frac{M_k}{d} \le \frac{C}{d}\Delta\gamma_k \le \frac{C}{d}\delta \to 0, \\ 0 \le \sum_k \Delta_{\mathrm{unc},k} &\le \frac{C^2\Gamma_\tau}{4d}\delta \to 0. \end{aligned} \tag{21}$$

Proof. The matched-covariance identity (8) and (19) at $c = 1$ relate the expected normalized mean mismatch to the SNR increment:

$$\begin{aligned} M_k = \mathbb{E}[a_k] &= \Delta\gamma_k\, \mathbb{E}\big[\|Z_0 - G_\theta(Z_{t_k}, t_k)\|_2^2\big] \\ &\le C\Delta\gamma_k. \end{aligned}$$

Proposition 1 gives $c^*_{\mathrm{KL},k} - 1 = M_k/d$, proving the first two bounds.

To bound the saving on each edge, use $0 \le x - \log(1+x) \le x^2/2$ for $x \ge 0$. The upper bound follows because $h(x) = x^2/2 - x + \log(1+x)$ has $h(0) = 0$ and $h'(x) = x^2/(1+x) \ge 0$. Substituting $x = M_k/d$ into (14) yields

$$0 \le \Delta_{\mathrm{unc},k} \le \frac{M_k^2}{4d} \le \frac{C^2}{4d}(\Delta\gamma_k)^2.$$

Finally, $\Delta\gamma_k \le \delta$ and $\sum_k \Delta\gamma_k = \Gamma_\tau$ give

$$\begin{aligned} 0 \le \sum_k \Delta_{\mathrm{unc},k} &\le \frac{C^2}{4d}\sum_k (\Delta\gamma_k)^2 \\ &\le \frac{C^2\delta}{4d}\sum_k \Delta\gamma_k = \frac{C^2\Gamma_\tau}{4d}\delta \to 0. \end{aligned}$$

□

Thus, under the stated assumptions, finer paths reduce the ideal-KL saving available from scalar confidence inflation to zero. This limit does not imply that total receiver mismatch or finite-PFR error vanishes.

A.4 Prediction and the reconstruction objective

The predictor minimizing Gaussian coding cost need not minimize decoded-image distortion. Fix the source law, representation, terminal, and Gaussian communication path, and retain the matched covariance used in prediction training. Let $m_t(z) = \mathbb{E}[Z_0 \mid Z_t = z]$ and consider $G_\epsilon(z,t) = m_t(z) + \epsilon h_t(z)$ with square-integrable perturbations of the shared predictor. Conditional-mean orthogonality applied to (8) gives, with ideal rate expressed in bpp,

$$R_{\mathrm{ideal}}(G_\epsilon) - R_{\mathrm{ideal}}(m) = \frac{\epsilon^2}{2HW \ln 2} \sum_{k=m+1}^{K} \Delta\gamma_k \, \mathbb{E}\big[\|h_{t_k}(Z_{t_k})\|_2^2\big] . \tag{22}$$

The linear term vanishes because $\mathbb{E}[Z_0 - m_t(Z_t) \mid Z_t] = 0$.

In contrast, consider the terminal training distortion

$$D(\epsilon) = \frac{1}{3HW} \mathbb{E}\big[\|X - D_{\mathrm{VAE}}(G_\epsilon(Z_\tau, \tau))\|_2^2\big] . \tag{23}$$

Assuming sufficient smoothness and integrability for a second-order expansion,

$$D(\epsilon) = D(0) + \epsilon a_h + O(\epsilon^2), \qquad a_h = D'(0). \tag{24}$$

Unlike the Gaussian coding term, decoded-image distortion need not be stationary at the latent conditional mean. If an admissible shared-predictor direction has $a_h < 0$, sufficiently small positive ϵ improves $\lambda_{R,\tau} R_{\mathrm{ideal}} + \lambda_D D$ for positive weights: the reconstruction gain is first order, while the ideal coding penalty is second order.

This local result explains why prediction calibration optimizes a joint rate–distortion objective rather than latent prediction error alone. At an inexact conditional-mean predictor, the rate expansion also contains a linear cross-term with the prediction error. The analysis concerns the fixed ideal Gaussian training law; serialized finite-PFR rates are measured separately.

B Codec Implementation

B.1 FLUX receiver and Gaussian transitions

The deterministic posterior mode of the frozen FLUX.1-dev VAE represents the source. FLUX velocity supplies the Gaussian receiver's clean-latent estimate; both VAE modules remain fixed.

Using the normalized time $T = 1$, rectified flow has $Y_t = (1-t)Z_0 + t\epsilon$ and

$$n(t) = \sqrt{(1-t)^2 + t^2}, \quad Z_t = Y_t/n(t), \quad \alpha_t = (1-t)/n(t), \quad \sigma_t = t/n(t),$$

so $\alpha_t^2 + \sigma_t^2 = 1$ and $\gamma_t = ((1-t)/t)^2$. With true velocity $u_t = \epsilon - Z_0$,

$$G_\theta(Z_t, t) = Y_t - t v_\theta(Y_t, t), \tag{25}$$

and

$$Z_0 - G_\theta = t(v_\theta - u_t), \qquad \|Z_0 - G_\theta\|_2^2 = t^2 \|u_t - v_\theta\|_2^2.$$

Thus (8) becomes schedule-weighted velocity error. This endpoint identity supplies the RCC predictor; the Gaussian Markov communication coupling is separately fixed.

Integer labels in the implementation. Positive scheduler labels ℓ use distinct noise-lookup and model-conditioning coordinates:

$$u_\ell = \frac{\ell - 1}{1000}, \qquad t_{\mathrm{model},\ell} = \frac{\ell}{1000}. \tag{26}$$

Its noise lookup returns the *amplitude* ratio $s_\ell = (1-u_\ell)/u_\ell$, while the VP SNR is $\gamma_\ell = s_\ell^2$. Thus $n_\ell = \sqrt{(1-u_\ell)^2 + u_\ell^2}$,

$$\alpha_\ell = (1-u_\ell)/n_\ell, \qquad \sigma_\ell = u_\ell/n_\ell, \qquad G^{\mathrm{impl}}_{\theta,\phi}(Z,\ell) = n_\ell Z - u_\ell v_{\theta,\phi}(n_\ell Z, t_{\mathrm{model},\ell}).$$

Neural conditioning differs from noise lookup by one label increment. Coding starts from independent Gaussian noise at label 1000, whose lookup gives $u_{1000} = 0.999$; the mathematical pure-noise endpoint $T = 1$ idealizes this boundary. Tables identify terminals by their scheduler labels, with achieved bitrate measured separately.

For a cleaner time $s < t$, let $a_{t|s} = \alpha_t/\alpha_s$ and $\beta_{t|s} = \sigma_t^2 - a_{t|s}^2\sigma_s^2$. The exact VP posterior coefficients used in (3) are

$$A_{s,t} = \frac{a_{t|s}\sigma_s^2}{\sigma_t^2}, \qquad B_{s,t} = \frac{\alpha_s\beta_{t|s}}{\sigma_t^2}, \qquad v_{s,t} = \frac{\sigma_s^2\beta_{t|s}}{\sigma_t^2}.$$

On communicated edge k, $(s,t) = (t_{k-1}, t_k)$ and $(A_k, B_k, v_k) = (A_{s,t}, B_{s,t}, v_{s,t})$.

B.2 Finite-PFR encoding and decoding

The adapter, path, and covariance table are shared codec state and determine the receiver law at both ends. Shared finite-PFR candidates turn each transition into a replayable message.

For one chunk, write $\delta = (\mu_q - \mu_p)/\sqrt{v}$ and generate the same standard-normal candidates $z_i \sim \mathcal{N}(0, \mathbf{I})$ as the unchanged Practical DiffC implementation. Under scalar inflation, candidate z_i represents the proposal sample $x_i = \mu_p + \sqrt{cv}\, z_i$. Its density-ratio score is

$$\log \frac{q(x_i)}{p_c(x_i)} = \frac{1}{2}\left[d \log c - (c-1)\|z_i\|_2^2 + 2\sqrt{c}\, z_i^\top \delta - \|\delta\|_2^2\right]. \tag{27}$$

Here, d denotes the number of latent coordinates in the coded chunk. We change only this score and the shared affine reconstruction of the selected candidate. Candidate generation, deterministic partitioning, chunk sizes, exponential arrivals, candidate selection, shared randomness, and index serialization remain unchanged. Setting $c = 1$ recovers the baseline implementation; for $c > 1$, (27) uses the same candidates and arrivals. The confidence table is fitted on training data and shared between sender and receiver, so only selected candidate indices and decode-required headers are transmitted. Exact sender/receiver agreement does not imply exact target sampling from a finite candidate pool.

After the last communicated transition, the receiver makes one clean-latent prediction and decodes it with the frozen VAE without additional source bits. Actual rate includes the finite-PFR payload and decode-required metadata, normalized by original image area. For the final RD family, path selection in (15) uses the reference-weighted score defined in Appendix C.2; both measured rate and distortion depend on the path-specific fitted confidence.

Algorithm 1 CalRCC calibration and finite-PFR codec

Offline calibration.

Inputs: Terminal τ; training path $\mathcal{T}_\tau^{\mathrm{tr}}$; candidate family $\mathcal{F}_\tau$; frozen $G_\theta, E_{\mathrm{VAE}}, D_{\mathrm{VAE}}$; shared randomness ω.

Train prediction candidates

$$\phi_{\tau,j} \leftarrow \mathrm{Train}_\phi\ \mathcal{L}_j(\phi; \mathcal{T}_\tau^{\mathrm{tr}}), \qquad \mathcal{L}_j \in \{\mathcal{L}_{\mathrm{RD}}, \mathcal{L}_{\mathrm{perc}}\}.$$

for each $(\phi_{\tau,j}, \mathcal{T}), \mathcal{T} \in \mathcal{F}_\tau$ **do**

For each edge k, fit on training latents and states:

$$\widehat{M}_k = \widehat{\mathbb{E}}_{\mathrm{conf}}\left[\frac{\|\mu_{q,k} - \mu_{p,k}^{\phi_{\tau,j}}\|_2^2}{v_k}\right], \qquad c_k = 1 + \frac{\widehat{M}_k}{d}.$$

end for

Select and freeze according to Table 7:

$$(\phi_\tau^\star, \mathcal{T}_\tau^\star, \mathbf{c}_\tau^\star) \leftarrow \mathrm{Select}_{\mathrm{dev}}\{(\phi_{\tau,j}, \mathcal{T}, \mathbf{c})\}.$$

Define the selected predictor and path:

$$G_\tau^\star(\cdot, t) \triangleq G_{\theta,\phi_\tau^\star}(\cdot, t),$$
$$\mathcal{T}_\tau^\star = \{t_K = T > t_{K-1} > \cdots > t_m = \tau\}.$$

Sender: $X \mapsto B$

$Z_0 \leftarrow E_{\mathrm{VAE}}(X)$.
$Z_{t_K} = Z_T \leftarrow \mathrm{Sample}_\omega\, \mathcal{N}(0, \mathbf{I})$.
for $k = K, \ldots, m+1$ **do**
$q_k = \mathcal{N}\big(A_k Z_{t_k} + B_k Z_0,\ v_k \mathbf{I}\big)$.
$p_k^\star = \mathcal{N}\big(A_k Z_{t_k} + B_k G_\tau^\star(Z_{t_k}, t_k),\ c_k^\star v_k \mathbf{I}\big)$.
$(b_k, Z_{t_{k-1}}) \leftarrow \mathrm{PFR\text{-}Enc}(q_k, p_k^\star; \omega_k)$.
end for
$B \leftarrow b_K \| \cdots \| b_{m+1}$.
return B.

Receiver: $B \mapsto \widehat{X}$

Read $b_K, \ldots, b_{m+1}$ from B.
$Z_{t_K} = Z_T \leftarrow \mathrm{Sample}_\omega\, \mathcal{N}(0, \mathbf{I})$.
for $k = K, \ldots, m+1$ **do**
$p_k^\star = \mathcal{N}\big(A_k Z_{t_k} + B_k G_\tau^\star(Z_{t_k}, t_k),\ c_k^\star v_k \mathbf{I}\big)$.
$Z_{t_{k-1}} \leftarrow \mathrm{PFR\text{-}Dec}(p_k^\star, b_k; \omega_k)$.
end for
$\widehat{Z}_0 \leftarrow G_\tau^\star(Z_\tau, \tau)$.
$\widehat{X} \leftarrow D_{\mathrm{VAE}}(\widehat{Z}_0)$.
return $\widehat{X}$.

PFR-Enc and PFR-Dec use the finite-candidate implementation of Appendix B.2 with the calibrated density-ratio score in (27). Here b_k denotes the serialized edge message, including selected finite-PFR indices and decode-required metadata. The selected adapter, path, confidence table, and shared randomness are shared codec state and are not transmitted per image.

B.3 Final codec configurations and model-call counts

CalRCC-RD and CalRCC-Perceptual each use seven terminal-specific rank-16 prediction adapters, with fixed base FLUX and VAE weights. One deployed setting needs one adapter (99.75 MiB of FP32 tensors) and its path/confidence configuration. Storing all seven adapters of one family requires 698.25 MiB, excluding shared backbone weights and diagnostic models. The selected adapter is shared codec state and requires no per-image adapter bits.

Table 5 combines the selected settings and model-call counts. RD uses $\lambda_D = 112.349549$; Perceptual uses $\lambda_D = 112.35$ and $\beta_\tau = 1.3575\rho_\tau$. Displayed coefficients are rounded from stored configurations. The multiplier α_R scales the reference rate weight; dashes denote directly specified λ_R. Both families have the listed active edge counts. For E communicated edges, sender and receiver model calls are E and $E+1$, including terminal prediction, so total NFE is $2E+1$. Equal NFE need not imply equal wall time.

Table 6 separates path resolution from placement; split factors four and eight lie outside this validated family. Selection is described in Appendix C.2, and measured effects in Appendix E.1.

Table 5: Final codec configurations. Panel A gives terminal-specific RD settings and DIV2K operating points; confidence ranges cover active edges. Panel B gives Perceptual coefficients, paths, and confidence ranges, and compares total NFE, which is identical for both CalRCC families. Training and selection are specified in Appendix C.2.

A. CalRCC-RD

Terminal	$\lambda_{R,\tau}$	Path	Confidence range	Edges	Actual bpp	PSNR (dB)
805	128	Later-biased	$[1.0011, 1.0068]$	4	0.003697	17.760
704	$64/3$	Later-biased	$[1.0011, 1.0135]$	6	0.008249	19.148
520	24	Later-biased	$[1.0011, 1.0404]$	10	0.027566	21.213
371	14.816590	Earlier-biased	$[1.0008, 1.0607]$	14	0.065803	22.823
247	4.172530	Split	$[1.0010, 1.1123]$	18	0.139127	24.516
160	0.883484	Split	$[1.0010, 1.1406]$	22	0.239862	26.201
94	0.137511	Split	$[1.0010, 1.2223]$	26	0.380954	27.956

B. CalRCC-Perceptual and shared model-call counts

Terminal	α_R	λ_R	ρ	Path	Confidence range	DiffC NFE	CalRCC NFE
805	–	96	4	Earlier-biased	$[1.0008, 1.0084]$	5	9
704	–	32	4	Earlier-biased	$[1.0008, 1.0133]$	7	13
520	–	24	8	Later-biased	$[1.0011, 1.0414]$	11	21
371	$4/3$	7.40829	4	Earlier-biased	$[1.0008, 1.0630]$	15	29
247	2	3.1294	8	Split	$[1.0010, 1.1226]$	19	37
160	2	1.32523	4	Split	$[1.0010, 1.1374]$	23	45
94	$2/3$	0.137511	8	Split	$[1.0010, 1.1943]$	27	53

Table 6: Shared path family. Edge counts describe the complete configured path, rather than the active prefix to a terminal. Both WARP controls keep the split-path resolution and change only placement.

Path	Family edges	Resolution vs. `COMMON`	Placement vs. `SPLIT2`	Interpretation
`COMMON`	16	Reference	–	Reference model-evaluation path
`SPLIT2`	32	More evaluation nodes	Reference	Resolution control
`WARP0p85_K32`	32	Same as `SPLIT2`	Earlier-biased	Fixed-edge placement control
`WARP1p15_K32`	32	Same as `SPLIT2`	Later-biased	Opposite placement control

C Training and Evaluation Protocol

Development selects the codec; evaluation uses fixed configurations under the following study-specific protocols.

C.1 Data and preprocessing

Training and selection use DIV2K (Agustsson & Timofte, 2017) train/development splits. A disjoint held-out split supports the 25-image post-prediction factorial with the shared-adapter RD variant. Transfer is evaluated on a separate set of all 50 pristine MCL-JCI (Jin et al., 2016) images, with RGB conversion, short-side Lanczos resize to 1024, and a central 1024^2 crop. Source-identity checks exclude overlap with prior project data.

The benchmark uses 80 DIV2K-1024 inputs and 24 native-resolution Kodak images. Metrics exclude Kodak padding; bpp uses original area. Benchmark inputs, codec configurations, and coding randomness are fixed before evaluation. Qualitative examples and crops are selected afterward for illustration, with an identical region across methods; they are not a representative performance sample.

COCO5K (Lin et al., 2014) supports the separate adapter-capacity and distributional study in Appendix E.5, not primary model selection. Turbo/CoD use the separate training populations and codec-specific DIV2K/Kodak preprocessing described in Appendix F.2.

C.2 Training and development selection

Prediction calibration keeps the base FLUX and both VAE modules fixed. Gradients pass through the VAE but not finite-PFR candidate selection. A Gaussian edge state is sampled directly from each source latent, independently of a noised terminal state. The sampled matched-covariance edge KL, weighted by the training-prefix edge count and divided by $HW \ln 2$, estimates summed ideal rate; a decoded terminal prediction supplies reconstruction losses. Confidence is fitted separately on training latents for each predictor and path.

Both final families start from the common distortion-oriented prediction adapter and use rank/LoRA alpha $16/16$ with 1,000 updates per candidate. RD uses learning rate 10^{-4} and $\lambda_D = 112.349549$; its five rate multipliers across seven terminals require 35,000 additional development updates. In (10) and (11), $\lambda_{R,\tau}$ denotes the effective training coefficient after applying any rate multiplier to its reference weight. Perceptual training is independent of final RD selection, uses a newly initialized optimizer and matched source/edge/noise streams, and adds frozen five-stage LPIPS-VGG with $\beta = \rho\beta_1$. The selected settings appear in Table 5.

Final RD development selection minimizes

$$J_{\mathrm{actual}}(\mathcal{T}, c) = \lambda^{\mathrm{ref}}_{R,\tau} R_{\mathrm{actual}}(\mathcal{T}, c) + \lambda_D D_{\mathrm{RGB}}(\mathcal{T}, c).$$

Here R_{actual} is serialized bpp and D_{RGB} is pixel-weighted RGB MSE on $[0, 1]$. The reference rate weight $\lambda^{\mathrm{ref}}_{R,\tau}$ stays fixed across candidates at each terminal, independently of the selected training coefficient $\lambda_{R,\tau}$.

Table 7 records the development rules. Perceptual selection uses the shared-adapter RD variant as its RD reference, not the final terminal-specific family. Except at terminals 805/704, LPIPS and DISTS guide selection but are not the VGG training loss. Final DIV2K/Kodak and MCL-JCI images enter neither selection nor confidence fitting.

The terminal-94 boundary rule was defined after development inspection and fixed before final evaluation. All final Perceptual choices use update 1000. Matched MSE controls use $\rho = 0$ and differ from the separately selected terminal-specific RD family; their same-terminal contrasts include covariance and rate changes.

Table 7: Development selection. α_R multiplies the reference rate weight; ρ scales LPIPS-VGG. MSE controls match initialization, rate penalty, duration, path, and numerical policy, with separately training-fitted confidence.

Family / terminal	Candidate grid	Selection criterion	Reference
RD / all seven	$\alpha_R \in \{1/2, 2/3, 1, 4/3, 2\}$; bounded path family	DevA/DevB select coefficients and path using actual rate plus RGB distortion at each terminal, with the reference rate weight rather than the training multiplier.	Fixed reference rate weight
Perceptual / 805, 704	VGG-based selection	Select using VGG, without LPIPS or DISTS.	Shared-adapter RD
Perceptual / 520	$\lambda_R \in \{6, 12, 24\}$; $\rho \in \{0, 2, 4, 8\}$	Within $\pm 10\%$ of the split-specific actual bpp of the fixed Perceptual reference $(\lambda_R, \rho) = (12, 4)$, apply the perceptual and PSNR conditions below and rank geometric relative LPIPS/DISTS loss. $(24, 8)$ improves the score on both DevA and DevB.	Rate-matched shared-adapter RD; matched MSE control
Perceptual / 371, 247, 160	$\alpha_R \in \{1/2, 2/3, 1, 4/3, 2\}$; $\rho \in \{0, 1/2, 1, 2, 4, 8\}$	Require strict LPIPS/DISTS improvement and PSNR loss at most 0.25/0.30 dB against RD/MSE control, respectively. DevA shortlists up to three eligible models; DevB applies these conditions and ranks increasing-rate combinations by geometric relative perceptual loss.	Rate-matched shared-adapter RD; same-α_R MSE control
Perceptual / 94	Same α_R, ρ grid as above	Outside RD rate support: require PSNR loss ≤ 0.5 dB, rate ratio ≤ 1.05, and strict LPIPS/DISTS improvement. DevA shortlists up to three eligible models; rank geometric relative perceptual loss.	Matched MSE control

C.3 Controlled baselines and rate accounting

The controlled comparison changes receiver prediction, confidence, and path under the shared conditions in Table 8. Reconstruction is single-pass, as in published DiffC-MSE, rather than primary DiffC-F's 50-step DDIM probability flow. Released-model behavior provides context; unavailable exact published subset/crop aggregation precludes direct superiority claims against that protocol.

Table 9 separates operational from descriptive author-coded and mixed-protocol rates. Exact-input methods share the evaluator; Tier A supports formal operational comparisons. Payload/header construction is in Appendix B.2.

Table 8: Shared and changed conditions in the controlled DiffC comparison. Source representation, reconstruction policy, serializer, and evaluation inputs and metrics are fixed.

Item	Controlled DiffC	CalRCC
Backbone	Frozen FLUX.1-dev, identical base weights	Same frozen base plus terminal-specific prediction LoRA
Source latent	Deterministic posterior mode of the frozen FLUX VAE	Identical deterministic source latent
Terminal/path	Seven matched terminals on `COMMON`	Same terminals; terminal-specific paths selected from the same bounded family
Terminal reconstruction	One clean prediction, then decoding with the frozen VAE	Same policy with the adapted predictor
Receiver mean	Estimate from the frozen FLUX predictor	FLUX LoRA-adapted estimate
Prediction objective	No codec-specific training	CalRCC-RD: $\mathcal{L}_{\mathrm{RD}}$; CalRCC-Perceptual: $\mathcal{L}_{\mathrm{perc}}$
Receiver covariance	Matched scalar covariance, $c_k = 1$	Training-fitted shared $c_k \geq 1$; no image-side uncertainty bits
PFR branch	Same candidates, chunking, selection, and replay	Same machinery; unequal-covariance affine proposal and density-ratio score
Serializer/rate	Self-contained payload + decode header	Identical serializer and accounting
Evaluation pixels	Identical DIV2K/Kodak/MCL inputs	Same inputs and metric implementations
Paper role	No-calibration baseline	Receiver-calibrated codec

Table 9: Rate-accounting classes for exact-input comparisons. All methods use the same metric implementations; only Tier A supports formal operational-rate comparisons.

Tier	Methods	Rate interpretation
A	DiffC, CalRCC-RD, CalRCC-Perceptual, VTM, DiffEIC, StableCodec	Self-contained actual serialized or operationally equivalent rate.
B	MS-ILLM, LALIC	Direct author-coded rate on exact pixels, but not a self-contained operational container.
C	Released-author and mixed-resolution curves	Different-protocol context only; excluded from exact-input dominance claims.

C.4 Metrics, aggregation, and statistical comparisons

PSNR measures pixel fidelity; LPIPS (AlexNet) and DISTS measure source-relative perceptual similarity. VGG denotes the training loss or a labeled auxiliary evaluator. CLIP measures cosine similarity between normalized ViT-B/32 source and reconstruction embeddings, without logit scaling; Q-Align estimates no-reference quality. FID/KID compare feature distributions rather than source-paired similarity, as discussed in Appendix E.5.

Aggregation and numerical policy. Final RD and perceptual curves on DIV2K/Kodak, and the MCL-JCI RD comparison, compute PSNR from pixel-weighted pooled MSE. Final perceptual curves pool bits over source pixels and average perceptual scores equally over images. Development, rank, and CoD comparisons instead use mean-image PSNR. Final perceptual and coefficient MSE/PSNR/LPIPS/DISTS use float reconstructions; auxiliary VGG/CLIP/Q-Align use saved RGB8 images. A nonfinite FP16 endpoint prediction is retried in FP32. The separate float/RGB8 boundaries and realization counts for CoD/Turbo are specified in Appendix F.2.

Paired comparisons. Final DIV2K/Kodak RD and perceptual comparisons, MCL-JCI evaluation, and development mechanism comparisons use two coding realizations paired across compared configurations. Realizations are averaged within each image before image-level summaries or resampling. The rank study uses one coding realization; other study-specific conventions are stated beside their results.

Matched-rate comparisons anchor on supported treatment rates and interpolate linearly in actual bpp, even when plots use log-bpp axes, without extrapolation. Final comparisons interpolate the specified

aggregate metric. Supported anchors receive equal weight. Paired image resamples rebuild both curves and the full across-anchor mean, using the same sampled images across anchors and fixed measured-support membership, rather than averaging interval endpoints.

Final intervals, confidence/path comparisons, and coefficient responses use 10,000 bootstrap draws. Reported intervals are pointwise and conditional on the evaluated model parameters and coding realizations, without multiplicity correction. Coefficient endpoint comparisons follow Appendix E.4; rank-study feature conventions and auxiliary-codec protocols remain beside their results.

D Additional Compression Results

We complement the main curves with same-terminal rate changes, matched-rate uncertainty, transfer to MCL-JCI, and bitrate comparisons at equal quality.

D.1 Final RD and perceptual comparisons

CalRCC-RD improves PSNR and lowers serialized rate at every measured DIV2K terminal in Table 10. At matched rates, its clearest advantage over DiffC is fidelity, while perceptual changes are mixed, including lower DIV2K Q-Align. Table 11 reports the aggregate changes and uncertainty for both final families. CalRCC-Perceptual improves all five metrics on average over DiffC and exchanges about 0.3 dB relative to terminal-specific RD for better perceptual quality. These averages do not imply improvement at every operating point. Matched MSE controls isolate the objective separately in Appendix E.4.

Table 10: Exact-input DIV2K-1024 comparison at the same terminals. Deltas are terminal-specific CalRCC-RD minus DiffC; lower bpp, LPIPS, and DISTS are better. These rows are not rate-matched.

Terminal	DiffC bpp	CalRCC-RD bpp	Δbpp	ΔPSNR (dB)	ΔLPIPS	ΔDISTS
805	0.004804	0.003697	-0.001107	$+0.697$	$+0.0931$	-0.0465
704	0.009478	0.008249	-0.001230	$+0.989$	$+0.0678$	-0.0174
520	0.029216	0.027566	-0.001650	$+1.388$	$+0.0240$	-0.0145
371	0.071146	0.065803	-0.005343	$+1.545$	-0.0004	-0.0130
247	0.153839	0.139127	-0.014712	$+1.446$	$+0.0082$	$+0.0025$
160	0.271780	0.239862	-0.031918	$+1.159$	$+0.0175$	$+0.0094$
94	0.445278	0.380954	-0.064324	$+0.981$	$+0.0218$	$+0.0105$

At 0.0282 bpp in Table 12, CalRCC-Perceptual trades 0.337 dB relative to CalRCC-RD for approximately 36% lower LPIPS and 40% lower DISTS. It also achieves higher PSNR and lower LPIPS and DISTS than DiffEIC, while StableCodec retains lower perceptual losses and VTM higher PSNR.

Table 11: Matched-rate changes in final codec performance, averaged equally over six supported anchors per comparison. Brackets are paired-image 95% intervals. Positive PSNR/CLIP/Q-Align and negative LPIPS/DISTS are favorable. Each comparison retains its own supported rate range.

A. CalRCC-RD minus DiffC

Metric	DIV2K	Kodak
PSNR (dB)	+1.6148 $[+1.5503, +1.6762]$	+1.5451 $[+1.4469, +1.6209]$
LPIPS	+0.0070 $[-0.0012, +0.0156]$	-0.0052 $[-0.0144, +0.0044]$
DISTS	-0.0172 $[-0.0215, -0.0129]$	-0.0114 $[-0.0141, -0.0087]$
CLIP	+0.0059 $[+0.0038, +0.0080]$	+0.0026 $[-0.0019, +0.0069]$
Q-Align	-0.1079 $[-0.1424, -0.0731]$	+0.1045 $[+0.0520, +0.1571]$
bpp range	0.0082–0.3810	0.0090–0.3813

B. CalRCC-Perceptual minus each reference

Metric	vs. CalRCC-RD		vs. DiffC	
	DIV2K	Kodak	DIV2K	Kodak
PSNR	-0.3055 $[-0.3227, -0.2898]$	-0.2842 $[-0.3108, -0.2598]$	+1.2639 $[+1.1969, +1.3287]$	+1.2042 $[+1.0962, +1.2899]$
LPIPS	-0.1170 $[-0.1282, -0.1060]$	-0.0997 $[-0.1141, -0.0870]$	-0.0724 $[-0.0793, -0.0657]$	-0.0689 $[-0.0757, -0.0619]$
DISTS	-0.0781 $[-0.0826, -0.0736]$	-0.0835 $[-0.0898, -0.0772]$	-0.0735 $[-0.0783, -0.0689]$	-0.0674 $[-0.0721, -0.0628]$
CLIP	+0.0156 $[+0.0123, +0.0192]$	+0.0278 $[+0.0210, +0.0344]$	+0.0159 $[+0.0138, +0.0181]$	+0.0217 $[+0.0174, +0.0262]$
Q-Align	+0.6191 $[+0.5636, +0.6750]$	+0.6918 $[+0.6188, +0.7792]$	+0.2754 $[+0.2369, +0.3155]$	+0.5373 $[+0.4568, +0.6243]$
bpp range	0.0042–0.2405	0.0049–0.2427	0.0088–0.3891	0.0093–0.3900

Table 12: Exact-input comparison at 0.0282 bpp on DIV2K. CalRCC-Perceptual is measured; other methods are linearly interpolated within measured rate support. Bold/underline mark best/second-best; Tier-B methods appear only in Fig. 2.

Method	PSNR ↑	LPIPS ↓	DISTS ↓	CLIP ↑	Q-Align ↑
VTM-24.0	**22.784**	0.510	0.287	0.903	2.440
DiffC	19.743	0.348	0.237	0.945	3.746
DiffEIC	18.362	0.281	0.157	0.949	4.372
StableCodec	20.738	**0.177**	**0.088**	**0.985**	**4.434**
CalRCC-RD	<u>21.241</u>	0.364	0.216	0.952	3.544
CalRCC-Perceptual	20.904	<u>0.234</u>	<u>0.129</u>	<u>0.968</u>	<u>4.383</u>

D.2 TRANSFER TO MCL-JCI

CalRCC-RD retains its fidelity gain on all 50 pristine MCL-JCI images without further adaptation or selection. Its mean matched-rate gain is 1.5745 dB (95% CI $[1.5221, 1.6245]$), with positive gains at all six supported anchors in Table 13. Terminal 805 is below DiffC's measured support; evaluation follows Appendix C.4.

Table 13: CalRCC-RD on the separate MCL-JCI source set. Rate/PSNR pairs are measured at each terminal; the matched-rate PSNR gain uses linear interpolation within DiffC support. PSNR is computed from pooled MSE. No MCL-JCI image is used for training, confidence fitting, or selection.

Terminal	DiffC bpp	DiffC PSNR	CalRCC-RD bpp	CalRCC-RD PSNR	Matched ΔPSNR	Support
805	0.004789	18.268	0.003643	18.953	–	No
704	0.009328	19.443	0.008028	20.380	+1.274	Yes
520	0.028437	21.252	0.026522	22.593	+1.522	Yes
371	0.068768	22.798	0.063305	24.278	+1.689	Yes
247	0.148335	24.649	0.133831	26.013	+1.701	Yes
160	0.262258	26.615	0.231027	27.708	+1.632	Yes
94	0.431151	28.475	0.368798	29.417	+1.629	Yes

D.3 ADDITIONAL METRIC CURVES AND BD-RATE COMPARISONS

Fig. 3 adds source–reconstruction CLIP to the main performance curves at the same operating points.

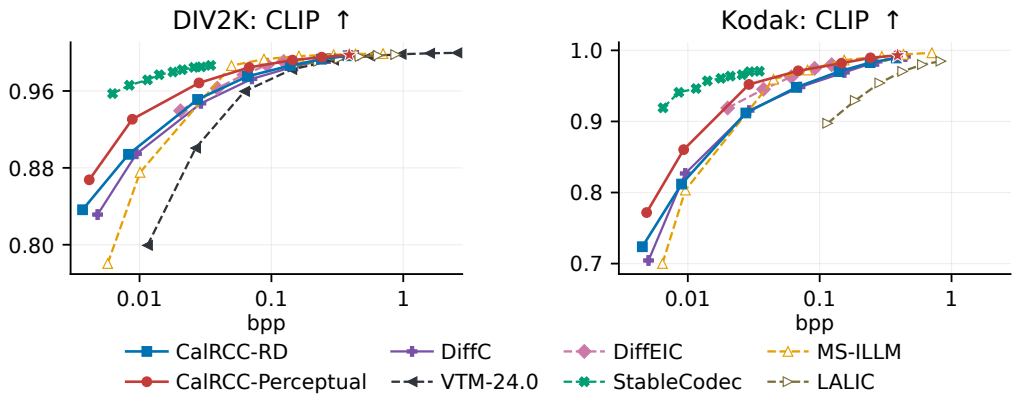

Figure 3: CLIP source–reconstruction similarity for the codecs in Fig. 2. Stars mark CalRCC-Perceptual points outside CalRCC-RD's measured rate range; open markers denote descriptive Tier-B methods.

Tables 14, 15, and 16 summarize bitrate differences at equal PSNR, LPIPS, and DISTS using the dataset-level curves in Fig. 2. For each method pair, we use piecewise cubic Hermite interpolation (PCHIP) of log bitrate as a function of the metric over their common measured quality interval. BD-rate is the exponentiated mean log-rate difference, minus one, expressed as a percentage; negative values favor the row method. LPIPS and DISTS retain their original scales. We neither extrapolate nor remove measured points, and comparison intervals can differ between pairs.

At equal PSNR, CalRCC-RD reduces bitrate by about 50% relative to controlled DiffC on both datasets. CalRCC-Perceptual reduces bitrate relative to DiffC by about 54–58% at equal LPIPS and 65–71% at equal DISTS over the respective common quality intervals.

Table 14: Pairwise BD-rate (%) at equal PSNR on DIV2K and Kodak, computed from Fig. 2. The column method is the reference; negative values indicate bitrate savings for the row method. †Descriptive Tier B comparisons.

DIV2K	DiffC	VTM 24.0	DiffEIC	Stable Codec	MS-ILLM†	LALIC†	CalRCC RD	CalRCC Perceptual
DiffC	0.00	+348.97	*NM*	+88.42	+239.81	—	+100.53	+71.91
VTM-24.0	−77.73	0.00	*NM*	−60.08	−32.88	+18.09	−57.91	−63.15
DiffEIC	*NM*	*NM*	*NM*	*NM*	*NM*	*NM*	*NM*	*NM*
StableCodec	−46.93	+150.48	*NM*	0.00	+88.34	—	+23.91	+3.25
MS-ILLM†	−70.57	+48.98	*NM*	−46.91	0.00	+111.76	−40.64	−48.84
LALIC†	—	−15.32	*NM*	—	−52.78	0.00	—	—
CalRCC-RD	−50.13	+137.58	*NM*	−19.30	+68.47	—	0.00	−13.83
CalRCC-Perceptual	−41.83	+171.37	*NM*	−3.15	+95.46	—	+16.04	0.00

Kodak	DiffC	VTM 24.0	DiffEIC	Stable Codec	MS-ILLM†	LALIC†	CalRCC RD	CalRCC Perceptual
DiffC	0.00	+368.35	−19.37	+122.33	+226.40	—	+97.38	+70.38
VTM-24.0	−78.65	0.00	—	—	−49.82	+13.59	−66.36	−70.01
DiffEIC	+24.02	—	0.00	+195.45	+307.84	—	+179.58	+137.76
StableCodec	−55.02	—	−66.15	0.00	+61.98	—	+2.08	−13.02
MS-ILLM†	−69.36	+99.26	−75.48	−38.26	0.00	+135.22	−37.55	−46.33
LALIC†	—	−11.96	—	—	−57.49	0.00	—	—
CalRCC-RD	−49.34	+197.23	−64.23	−2.04	+60.12	—	0.00	−13.19
CalRCC-Perceptual	−41.31	+233.48	−57.94	+14.97	+86.32	—	+15.20	0.00

—: no common quality interval. NM: nonmonotonic quality-versus-rate curve, for which we do not compute an inverse. The DIV2K DiffEIC PSNR curve has a slight decrease at its highest rate.

Table 15: Pairwise BD-rate (%) at equal LPIPS on DIV2K and Kodak, computed from Fig. 2. The column method is the reference; negative values indicate bitrate savings for the row method. †Descriptive Tier B comparisons.

DIV2K	DiffC	VTM 24.0	DiffEIC	Stable Codec	MS-ILLM†	LALIC†	CalRCC RD	CalRCC Perceptual
DiffC	0.00	−64.58	+97.38	+476.40	+149.85	−47.59	−11.67	+139.32
VTM-24.0	+182.35	0.00	+353.82	+1234.66	+538.77	+16.44	+153.55	+494.10
DiffEIC	−49.34	−77.96	0.00	+194.57	+57.29	−74.07	−47.58	+18.44
StableCodec	−82.65	−92.51	−66.05	0.00	−45.40	−89.85	−81.60	−54.56
MS-ILLM†	−59.98	−84.35	−36.42	+83.16	0.00	−82.38	−63.39	−12.39
LALIC†	+90.79	−14.12	+285.72	+884.89	+467.60	0.00	+91.67	+242.44
CalRCC-RD	+13.21	−60.56	+90.78	+443.59	+173.16	−47.83	0.00	+158.93
CalRCC-Perceptual	−58.21	−83.17	−15.57	+120.08	+14.15	−70.80	−61.38	0.00

Kodak	DiffC	VTM 24.0	DiffEIC	Stable Codec	MS-ILLM†	LALIC†	CalRCC RD	CalRCC Perceptual
DiffC	0.00	−62.16	+126.44	+551.59	+167.89	−49.30	−1.41	+115.99
VTM-24.0	+164.28	0.00	+430.60	+1460.53	+611.87	+14.57	+169.72	+463.65
DiffEIC	−55.84	−81.15	0.00	+188.13	+54.08	−78.20	−50.62	+0.77
StableCodec	−84.65	−93.59	−65.29	0.00	−45.55	−92.56	−82.56	−60.99
MS-ILLM†	−62.67	−85.95	−35.10	+83.64	0.00	−84.96	−62.75	−23.59
LALIC†	+97.23	−12.71	+358.67	+1244.18	+565.02	0.00	+120.61	+304.74
CalRCC-RD	+1.43	−62.92	+102.50	+473.37	+168.49	−54.67	0.00	+118.93
CalRCC-Perceptual	−53.70	−82.26	−0.77	+156.33	+30.88	−75.29	−54.32	0.00

Table 16: Pairwise BD-rate (%) at equal DISTS on DIV2K and Kodak, computed from Fig. 2. The column method is the reference; negative values indicate bitrate savings for the row method. †Descriptive Tier B comparisons.

DIV2K	DiffC	VTM 24.0	DiffEIC	Stable Codec	MS-ILLM†	LALIC†	CalRCC RD	CalRCC Perceptual
DiffC	0.00	−55.76	+168.53	+889.36	+109.56	−63.78	+25.03	+246.51
VTM-24.0	+126.05	0.00	+657.30	+2697.51	+397.51	+2.65	+168.76	+754.46
DiffEIC	−62.76	−86.80	0.00	+267.82	−26.30	−86.47	−57.97	+18.73
StableCodec	−89.89	−96.43	−72.81	0.00	−79.25	−96.33	−88.59	−66.64
MS-ILLM†	−52.28	−79.90	+35.69	+382.04	0.00	−81.30	−45.22	+67.78
LALIC†	+176.08	−2.59	+639.05	+2624.99	+434.88	0.00	+209.00	+762.82
CalRCC-RD	−20.02	−62.79	+137.94	+776.61	+82.55	−67.64	0.00	+203.55
CalRCC-Perceptual	−71.14	−88.30	−15.77	+199.76	−40.40	−88.41	−67.06	0.00

Kodak	DiffC	VTM 24.0	DiffEIC	Stable Codec	MS-ILLM†	LALIC†	CalRCC RD	CalRCC Perceptual
DiffC	0.00	−63.18	+219.55	+977.36	+123.22	−62.63	+8.18	+184.45
VTM-24.0	+171.57	0.00	+793.71	+2840.31	+482.52	+1.76	+228.41	+653.09
DiffEIC	−68.71	−88.81	0.00	+229.05	−33.36	−88.67	−62.00	−10.14
StableCodec	−90.72	−96.60	−69.61	0.00	−78.20	−96.55	−88.67	−70.99
MS-ILLM†	−55.20	−82.83	+50.06	+358.65	0.00	−84.44	−48.60	+28.38
LALIC†	+167.60	−1.73	+782.85	+2798.63	+542.79	0.00	+232.41	+729.69
CalRCC-RD	−7.56	−69.55	+163.15	+782.34	+94.55	−69.92	0.00	+144.30
CalRCC-Perceptual	−64.84	−86.72	+11.28	+244.65	−22.10	−87.95	−59.07	0.00

E Calibration Analysis

We distinguish the communication and computation effects of calibration, then examine terminal-specific adaptation, training objectives, and adapter capacity.

E.1 Calibration effects on rate, quality, and computation

Confidence and path calibration alter communication and computation around a fixed predictor, while prediction calibration balances coding cost and reconstruction quality. Table 2 compares confidence/path calibration with unadapted DiffC and isolates prediction by enabling or disabling the adapter at fixed path and confidence. RD prediction diagnostics use the shared-adapter variant, separately from the final terminal-specific family. The perceptual diagnostics use $(\lambda_R, \rho) = (96, 4), (32, 4), (12, 4)$ at terminals 805/704/520; the terminal-520 diagnostic is not the final $(24, 8)$ model.

With the shared-adapter RD predictor fixed, Table 17 compares the unadapted baseline and a conditional 2×2 confidence/path design on 25 held-out images: prediction alone, prediction plus either calibration, and all three. Confidence and path remain useful after prediction adaptation. These conditional effects are not additive with the different references in Table 2. Here $J_{\text{actual}} = R_{\text{actual}} + \lambda_D D_{\text{RGB}}$, with $\lambda_D = 112.35$ for the low/medium regions and 7.02 for the high region (rounded). Final RD selection uses the reference-weighted score in Appendix C.2.

Confidence can reduce rate without additional model calls, whereas path calibration trades candidate search for more model evaluations. At medium rate, path calibration reduces candidates from 388.7 to 366.8 million relative to prediction alone, and all three calibrations reduce them to 357.5 million; NFE rises from 15 to 29. These counts do not establish a measured runtime speedup. The matched-covariance path choices are earlier-biased, later-biased, and split in the low/medium/high regimes, with 6/14/26 active edges. Confidence fitting can also change the selected placement or resolution.

Comparison protocol. Three disjoint 100-image DIV2K development subsets fit confidence, select paths, and evaluate Table 2 at 1024^2 resolution. Coding realizations are paired and averaged within images as in Appendix C.4; PSNR is mean-image. For image/realization-mean bpp $\bar{R}_j(\tau)$

Table 17: Confidence/path ablation with the shared-adapter RD predictor on 25 held-out images, with DiffC as an unadapted reference. Actual bpp includes payload and headers; candidate counts are in millions.

Region	Method	Actual bpp	PSNR	J_{actual}	Candidates	NFE
Low	DiffC	.009014	19.250	1.607152	41.8	7
	Prediction	.009123	20.055	1.306613	42.4	7
	Prediction + confidence	.009046	20.064	1.305855	41.9	7
	Prediction + path	.008748	20.094	1.297346	38.9	13
	Prediction + confidence + path	.008683	20.099	1.296216	38.6	13
Medium	DiffC	.071192	22.491	.895887	352.2	15
	Prediction	.078424	23.681	.685623	388.7	15
	Prediction + confidence	.075708	23.685	.681818	370.1	15
	Prediction + path	.074917	23.702	.678898	366.8	29
	Prediction + confidence + path	.073637	23.694	.678513	357.5	29
High	DiffC	.453768	28.530	.466935	2300.0	27
	Prediction	.443732	28.719	.456301	2248.5	27
	Prediction + confidence	.396337	28.748	.408813	1936.1	27
	Prediction + path	.398058	28.760	.410504	1990.8	53
	Prediction + confidence + path	.377693	28.770	.390112	1855.5	53

and evaluated terminals $\mathcal{S}$, the reported percentage rate change is

$$100\left[\frac{1}{|\mathcal{S}|}\sum_{\tau\in\mathcal{S}}\frac{\bar{R}_{\mathrm{treat}}(\tau)}{\bar{R}_{\mathrm{ref}}(\tau)}-1\right]\%.$$

This is the arithmetic mean of aggregate-rate ratios, not a pooled-rate ratio or an average of per-image ratios. Quality equally averages treatment-minus-reference differences at supported treatment rates using linear-bpp reference interpolation without extrapolation. NFE averages same-terminal differences over all evaluated terminals, with the model-call convention in Appendix B.3.

Confidence/path-only rows use seven terminals. Prediction rows retain all seven RD and three perceptual terminals for rate and NFE, but perceptual quality comparisons use only 805/704: the adapter-enabled result at 520 exceeds the adapter-disabled rate range, although the adapter-disabled 520 point supports interpolation at 704. Intervals for confidence/path comparisons follow the paired-image protocol in Appendix C.4.

E.2 Held-out validation of the confidence factor

Proposition 1 gives the closed-form covariance multiplier that minimizes expected Gaussian edge KL at a fixed receiver mean. We test whether substituting training estimates of M_k into this rule reduces KL on held-out source images.

We use the pretrained predictor and the final terminal-520 CalRCC-RD and CalRCC-Perceptual predictors, keeping all model parameters fixed. On 100 DIV2K training images, we estimate $M_k = \mathbb{E}[a_k]$ by the empirical average $\widehat{M}_k$ of the normalized mean mismatch, and set $\hat{c}_k = 1 + \widehat{M}_k/d$ directly from (13). These factors are frozen for evaluation on 500 disjoint COCO images. Inputs are resized on the short side and center-cropped to 1024^2. We evaluate the ten edges of the later-biased path ending at label 520, with $d = 262144$. Each image has two noise realizations, and all predictors receive identical states sampled directly from $q(z_{t_k} \mid z_0)$. The initial state retains the finite-SNR lookup at label 1000. No held-out tuning is performed.

For each held-out image, we sum calibrated-minus-unit KL over the ten edges and average the two noise realizations. Table 18 reports the mean of these paired image differences and its 95% percentile interval from 2,000 image-bootstrap resamples, conditional on the frozen training fit. All three intervals lie below zero. These are ideal Gaussian-KL changes, not measured serialized finite-PFR rates. The common-marginal-state fit is separate from the source-conditioned trajectory, noise, and rounding protocol used for the final codec settings in Table 5.

Fig. 4 scales the fitted variance adjustment as $c_k(\lambda) = 1 + \lambda(\hat{c}_k - 1)$. For all three predictors, the lowest held-out KL among the tested scales occurs at $\lambda = 1$, the training-fitted scale. This supports transfer of the closed-form fit to held-out images; the sweep is diagnostic and does not select

Table 18: Held-out confidence validation. Training-fitted c_k is frozen for evaluation on 500 COCO images; negative ΔKL improves on $c_k = 1$. The paired-image 95% intervals follow averaging of the two noise realizations within each image. Gaussian edge KL is an idealized information quantity, not serialized bpp.

Predictor	Training c_k range	ΔKL (nats/image)	95% CI	Relative KL change
Pretrained	$[1.0013, 1.0343]$	-212.4	$[-216.8, -207.5]$	-1.09%
CalRCC-RD	$[1.0011, 1.0403]$	-220.2	$[-225.8, -214.6]$	-1.14%
CalRCC-Perceptual	$[1.0011, 1.0413]$	-264.0	$[-270.0, -257.7]$	-1.28%

factors. The serialized-rate reductions in Tables 2 and 17 provide separate evidence for confidence calibration in the practical codec.

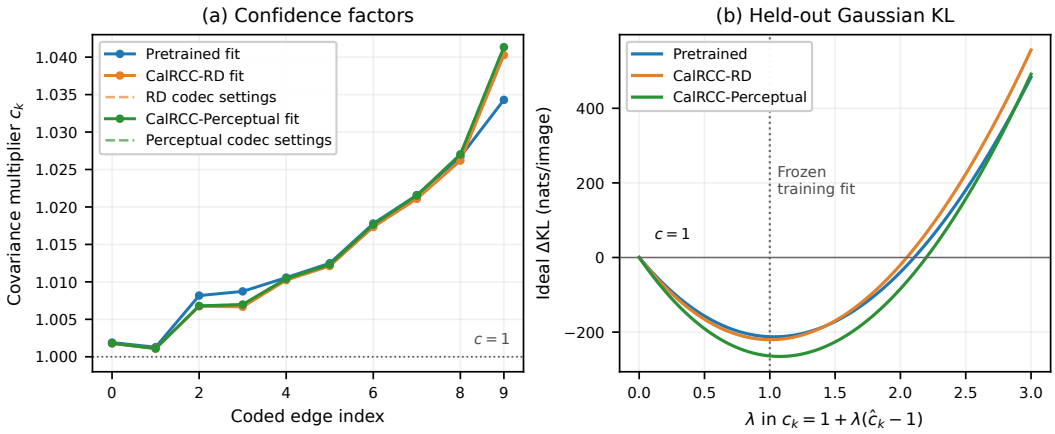

Figure 4: Gaussian-KL response to confidence calibration at terminal 520. Left: training-fitted factors (solid); dashed curves show existing codec settings. Right: held-out KL change along $c_k = 1 + \lambda(\hat{c}_k - 1)$, where $\hat{c}_k$ is the training fit. Unit scaling is $\lambda = 0$ and the frozen training fit is $\lambda = 1$; lower Gaussian KL is favorable.

E.3 Effects of terminal-specific adaptation and configuration selection

Terminal-specific prediction adaptation provides the largest additional fidelity change: about 0.45 dB over the shared-adapter RD variant on DIV2K and Kodak. Table 19 compares separate adapters at reference rate weights with the shared-adapter RD variant, then measures the smaller changes from rate-weight selection and path/confidence refinement. The latter changes are not uniformly positive across datasets.

Table 19: Effects of terminal-specific adaptation and configuration selection on matched-rate PSNR (dB), with 95% paired-image intervals. The first row compares terminal-specific adaptation at reference weights with shared-adapter RD; subsequent rows compare each refinement with the preceding configuration. Each contrast uses its own six supported anchors.

Change	DIV2K	Kodak
Terminal-specific adaptation	+0.4480 $[+0.4242, +0.4699]$	+0.4630 $[+0.4317, +0.4919]$
Rate-weight selection	+0.0203 $[+0.0154, +0.0251]$	-0.0373 $[-0.0673, -0.0123]$
Path/confidence refinement	+0.0097 $[+0.0062, +0.0132]$	+0.0060 $[-0.0045, +0.0174]$

Training allocation also differs between the terminal-specific and shared-adapter variants, so the first contrast does not isolate the causal effect of adapter sharing. The contrasts have different supported rate domains and must not be added to obtain the final gains over DiffC.

E.4 Prediction objectives and coefficient sensitivity

We first isolate the effect of adding perceptual loss to a matched MSE objective. In Table 20, perceptual training improves LPIPS/DISTS while spending additional bits and pixel fidelity. The table includes the final terminal-520 model on both datasets. At the highest-rate boundary, the PSNR changes are -0.3359/-0.3356 dB and rate changes are +1.14/+1.92% on DIV2K/Kodak. These same-terminal comparisons match initialization, rate penalty, training duration, path, and numerical policy, but retain checkpoint-specific training-fitted covariance.

Table 20: CalRCC-Perceptual minus its MSE-trained counterpart at the same terminal and rate penalty. Each has training-fitted covariance; rates differ. Brackets are paired-image 95% intervals.

Dataset	Terminal	Δbpp	ΔPSNR	ΔLPIPS	ΔDISTS
DIV2K	520	+0.000673	-0.3087 $[-0.3311, -0.2884]$	-0.1322 $[-0.1468, -0.1185]$	-0.0877 $[-0.0936, -0.0820]$
DIV2K	371	+0.001322	-0.2491 $[-0.2789, -0.2262]$	-0.0805 $[-0.0908, -0.0706]$	-0.0624 $[-0.0676, -0.0573]$
DIV2K	247	+0.004451	-0.3278 $[-0.3547, -0.3011]$	-0.0486 $[-0.0565, -0.0412]$	-0.0445 $[-0.0494, -0.0398]$
DIV2K	160	+0.001823	-0.2462 $[-0.2666, -0.2222]$	-0.0275 $[-0.0315, -0.0238]$	-0.0239 $[-0.0265, -0.0215]$
DIV2K	94	+0.004391	-0.3359 $[-0.3595, -0.3145]$	-0.0271 $[-0.0314, -0.0231]$	-0.0223 $[-0.0249, -0.0200]$
Kodak	520	+0.001553	-0.3449 $[-0.3888, -0.3090]$	-0.1352 $[-0.1618, -0.1121]$	-0.0870 $[-0.0945, -0.0794]$
Kodak	371	+0.001996	-0.2320 $[-0.2746, -0.1863]$	-0.0739 $[-0.0904, -0.0596]$	-0.0585 $[-0.0660, -0.0513]$
Kodak	247	+0.005403	-0.3163 $[-0.3414, -0.2872]$	-0.0453 $[-0.0560, -0.0362]$	-0.0369 $[-0.0430, -0.0313]$
Kodak	160	+0.002147	-0.3220 $[-0.3494, -0.2920]$	-0.0225 $[-0.0279, -0.0177]$	-0.0152 $[-0.0176, -0.0129]$
Kodak	94	+0.007331	-0.3356 $[-0.3690, -0.3056]$	-0.0244 $[-0.0310, -0.0189]$	-0.0159 $[-0.0180, -0.0138]$

We next vary the relative rate, pixel-fidelity, and perceptual weights. This sensitivity analysis does not change the selected codec configurations. On DIV2K and Kodak, stronger perceptual emphasis lowers LPIPS/DISTS at the cost of PSNR and rate; pixel-fidelity emphasis reverses this preference, while rate penalties produce smaller changes, as shown in Fig. 5 and Table 21. The measured intermediate responses include nonmonotonic changes.

Objective and measured slices. At terminals $\tau \in \{371, 247, 160, 94\}$, rank-16 receivers start from the common distortion-oriented adapter and share the ordered training/noise stream, fixed path, and 1,000 AdamW updates at learning rate 10^{-4}. With the backbone and VAE fixed, we vary

$$L_\tau(\alpha_R, \rho, \gamma_D) = \alpha_R \lambda_{R,0,\tau} R_{\text{ideal}} + \gamma_D \lambda_0 D_{\text{RGB}} + \rho \beta_1 D_{\text{VGG}}. \tag{28}$$

Here $\lambda_0 = 112.349549$, $\beta_1 = 1.357516$, and $\lambda_{R,0,\tau} = (5.556221, 1.564699, 0.662613, 0.206266)$ in terminal order. Starting from $(\alpha_R, \rho, \gamma_D) = (1, 2, 1)$, the measured slices vary $\alpha_R \in \{0.5, 1, 2\}$, $\rho \in \{0, 2, 4, 8\}$, or pixel-fidelity endpoints $\gamma_D \in \{0.5, 2\}$. The distortion multiplier γ_D is distinct from SNR $\gamma(t)$.

Comparison protocol. Each model has confidence fitted on the same first 100 training images; evaluation and paired-image intervals follow Appendix C.4. Percentage changes are ratios of aggregates. At terminals 371/247, the reused reference and direct endpoints have different training

numerical policies, shown by open markers in Fig. 5; they match at 160/94. Endpoint comparisons in Table 21 use the same policy within each terminal and report ranges across terminals. These training differences are separate from the evaluation fallback. Table 21 reports endpoint responses for both datasets; Fig. 5 shows the measured DIV2K slices.

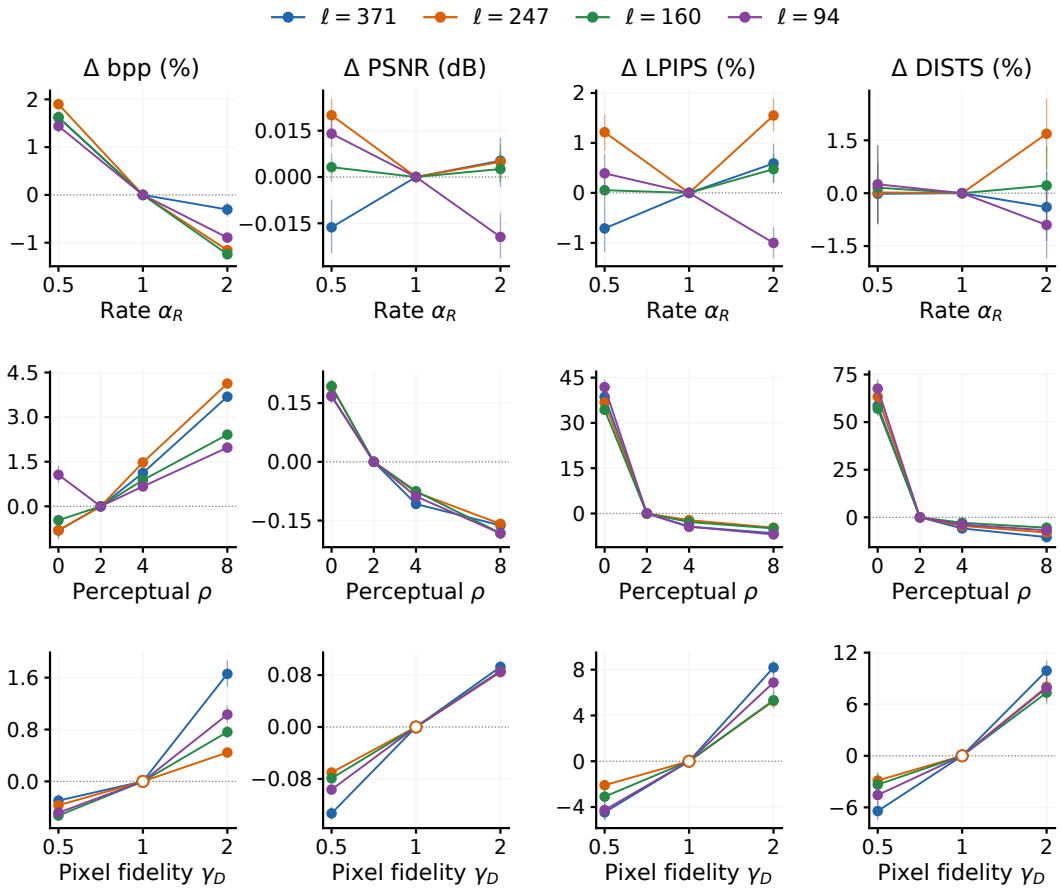

Figure 5: Changes relative to each terminal's reference weights $(\alpha_R, \rho, \gamma_D) = (1, 2, 1)$ on DIV2K. Rows vary the rate, perceptual, and pixel-fidelity weights. Bars show pointwise paired-image 95% intervals; lines connect measured settings. Open reference markers indicate the training-policy differences described in the text.

Table 21: Coefficient endpoint responses: ranges across four terminals, not confidence intervals. Changes are endpoint minus start, with other weights at the reference setting. Rate, LPIPS, and DISTS are percentage changes; PSNR changes are in dB.

Dataset	Change	Δbpp (%)	ΔPSNR (dB)	ΔLPIPS (%)	ΔDISTS (%)
DIV2K	$\alpha_R : 0.5 \to 2$	-3.00 to -1.90	-0.033 to $+0.022$	-1.39 to $+1.31$	-1.15 to $+1.66$
	$\rho : 0 \to 8$	$+0.90$ to $+4.97$	-0.376 to -0.331	-34.39 to -29.24	-44.40 to -39.80
	$\gamma_D : 0.5 \to 2$	$+0.82$ to $+1.96$	$+0.155$ to $+0.226$	$+7.52$ to $+13.25$	$+11.07$ to $+17.45$
Kodak	$\alpha_R : 0.5 \to 2$	-3.41 to -1.55	-0.077 to -0.017	-1.84 to $+2.16$	-1.10 to $+1.82$
	$\rho : 0 \to 8$	$+1.32$ to $+5.22$	-0.425 to -0.211	-31.46 to -22.85	-37.83 to -23.79
	$\gamma_D : 0.5 \to 2$	$+0.61$ to $+2.30$	$+0.143$ to $+0.177$	$+5.45$ to $+12.57$	$+3.59$ to $+13.04$

E.5 Adapter capacity and distributional tradeoffs

Most RGB gains occur at low rank in this matched-budget COCO study: rank 4 captures 97% of rank 16's mean-image-PSNR gain with one quarter of its parameters in Table 22. Rank 16 reduces

RGB MSE by 13.5% at 2.6% higher rate than no adaptation and has the best perceptual scores among trained ranks. Rank 32 adds little benefit: its actual RD-objective change is -0.000223 (paired-image 95% interval $[-0.000496, +0.000042]$). All ranks share 2,000 RD updates, zero-effect initialization, $\alpha/r = 1$, ordered training/noise streams, and no perceptual loss. COCO5K evaluation uses short-side resize and 1024^2 center crop, terminal 520, one coding realization, and fixed later-biased path and covariance. RGB metrics use float reconstructions; FID/KID (Heusel et al., 2017; Bińkowski et al., 2018) use RGB8 PNG inception-v3-compat features of dimension 2,048, with 100 subsets of 1,000 images for KID.

Improved RGB fidelity does not imply lower distributional or latent error: every trained rank worsens FID/KID, latent MSE, and marginal Gaussian error relative to no adaptation. The receiver diagnostics use 500 images with identical independent Gaussian marginal states across predictors at each evaluation time. Gaussian error is $\sum_k \Delta\gamma_k \|Z_0 - G_{\theta,\phi}(Z_{t_k}, t_k)\|^2/(2HW \ln 2)$, in bit-equivalent units per source pixel rather than measured bitstream rate. Different state distributions and metrics leave the cause of this separation unresolved. Image-paired intervals condition on the evaluated parameters and coding realization; KID subset variability is separate from effect uncertainty, and no FID interval is estimated. Supporting nested 1K/2K/5K evaluations check sample-size sensitivity. These conclusions concern this RD-only study.

Table 22: Adapter capacity and diagnostic tradeoffs. Panel A and RGB MSE in Panel B use 5,000 codec reconstructions; latent MSE and Gaussian error use 500 paired Gaussian marginal states. PSNR is mean-image; RGB MSE averages squared pixel errors. No adaptation disables the adapter under these fixed settings, rather than using the separately defined DiffC baseline. MiB counts FP32 adapter tensors without serialization overhead; KID is raw.

A. Reconstruction, rate, and distributional quality

Rank	MiB	bpp	PSNR ↑	LPIPS ↓	DISTS ↓	FID ↓	KID ↓
No adaptation	0.00	0.028641	24.550	0.3021	0.1651	12.111	0.002210
4	24.94	0.029573	25.086	0.2978	0.1644	13.579	0.003000
8	49.88	0.029360	25.098	0.2996	0.1648	13.426	0.002884
16	99.75	0.029395	25.102	0.2977	0.1626	13.163	0.002820
32	199.50	0.029340	25.097	0.2992	0.1635	13.420	0.002884

B. Pixel and receiver errors

Rank	RGB MSE ↓	Latent MSE ↓	Gaussian error ↓
No adaptation	0.00491065	0.125672	0.026970
4	0.00426388	0.130707	0.027722
8	0.00425625	0.130233	0.027539
16	0.00424875	0.130347	0.027575
32	0.00424725	0.130451	0.027527

F Reconstruction Policies and Prediction Adaptation

We first compare single-pass and iterative reconstruction from the same communicated state. We then study prediction adaptation in CoD and Turbo-DDCM, separating communication-stage frontend adaptation from fixed-communication decoder adaptation.

F.1 Single-pass versus iterative reconstruction

We ask how reconstruction changes when the same communicated state is decoded once or iteratively. On DIV2K and Kodak, we replace the terminal clean-latent prediction with deterministic multi-step denoising. The bitstream, decoded terminal state, actual bpp, path, and confidence remain identical within each codec; different codecs use different bitstreams.

Fig. 6 shows a lower-rate example at terminal 704. Single-pass reconstruction smooths fine ceiling detail, whereas iterative decoding produces sharper, more photo-like structure that can depart from

the source. This example is consistent with synthesized detail contributing to perceived sharpness, rather than faithful recovery of the source; it does not establish a general causal mechanism.

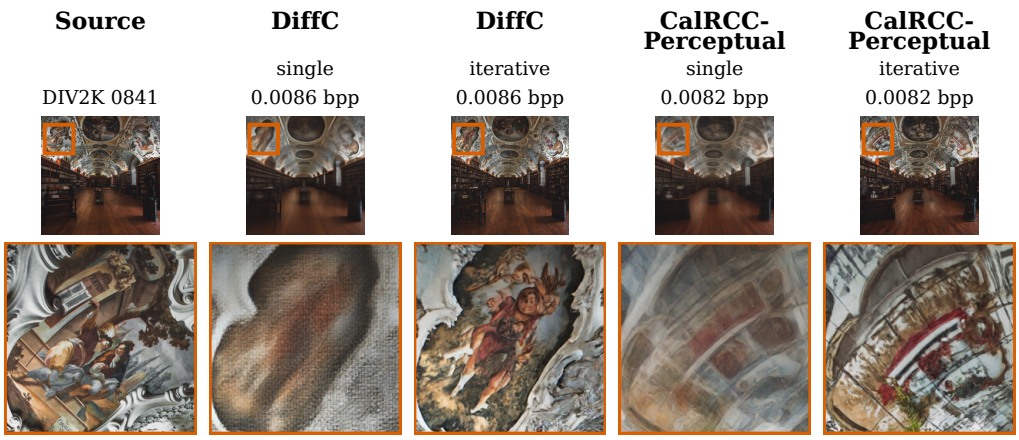

Figure 6: Single-pass and iterative reconstruction of DIV2K 0841. Within each codec, both policies use the same bitstream and decoded terminal state; iterative reconstruction uses the base pretrained predictor. Insets enlarge the same region, and bpp refers to the full coded image.

For DiffC, iterative reconstruction with the base pretrained predictor improves LPIPS and DISTS at all seven terminals on both datasets while reducing PSNR on average. For CalRCC-Perceptual, the same reconstruction costs roughly 1 dB PSNR on average, with little average LPIPS change but lower DISTS and higher Q-Align. Keeping the perceptual adapter active throughout iterative reconstruction instead worsens PSNR and LPIPS on average relative to single-pass reconstruction.

Table 23: Post-terminal reconstruction at approximately 0.02824 actual serialized bpp, using linear interpolation within measured support where needed. Iterative DiffC uses the base pretrained predictor.

Dataset	Codec / reconstruction	PSNR (dB) ↑	LPIPS ↓	DISTS ↓	Q-Align ↑
DIV2K	CalRCC-Perceptual / single	20.90	0.2336	0.1294	4.383
DIV2K	DiffC / iterative	19.62	0.2333	0.1100	4.549
Kodak	CalRCC-Perceptual / single	21.93	0.2520	0.1486	4.276
Kodak	DiffC / iterative	20.73	0.2536	0.1261	4.498

At the representative rate in Table 23, single-pass CalRCC-Perceptual has 1.28/1.20 dB higher PSNR than iterative DiffC on DIV2K/Kodak. LPIPS point estimates are close; paired-image 95% intervals for their difference include zero but do not establish equivalence. Iterative DiffC has lower DISTS and higher Q-Align, with larger average advantages over the supported low-rate points. Single-pass retains higher PSNR across the supported rates.

Single-pass reconstruction uses one post-terminal model evaluation; measured iterative schedules use 7–43. At the representative rate, DiffC is interpolated between actual schedules with 28 and 38 post-terminal evaluations. These counts exclude communicated-edge and sender evaluations and do not establish a measured runtime speedup. Neither policy adds source bits to its underlying codec. Iterative reconstruction thus trades pixel fidelity and additional model evaluations for gains in selected perceptual metrics.

F.2 Prediction adaptation in CoD and Turbo-DDCM

Shared formulation and evaluation protocol

We compare decoders at fixed communication in CoD and Turbo-DDCM; Fig. 7 shows where prediction adaptation enters. Starting from the latent state recovered from the bitstream, the decoder repeatedly predicts the clean latent and uses that prediction to update its current state. The same adapter is reused at every remaining denoising step, after which the fixed image decoder produces the reconstruction. During training, loss on this final image is backpropagated through all remaining steps to update only the adapter. For image x_i and operating point p, the reference bitstream $b_{i,p}^{\mathrm{ref}}$, decoded latent $z_{i,p}^{\mathrm{ref}}$, and conditioning $c_{i,p}^{\mathrm{ref}}$ remain fixed, as do shared randomness and the decoding schedule:

$$\begin{aligned} z^{(0),\phi} &= \mathrm{sg}(z_{i,p}^{\mathrm{ref}}), \qquad z^{(j+1),\phi} = F_{\theta,\phi,j}(z^{(j),\phi}; c_{i,p}^{\mathrm{ref}}), \\ \widehat{x}_{i,p}^{\phi} &= V_{\psi}(z^{(L_p),\phi}), \qquad b_{i,p}^{\phi} = b_{i,p}^{\mathrm{ref}}, \\ R_{i,p}^{\phi} &= \frac{8|b_{i,p}^{\mathrm{ref}}|}{H_i W_i}, \qquad \nabla_{\phi} R_{i,p}^{\phi} = 0. \end{aligned} \tag{29}$$

Here sg denotes stop-gradient, $|b|$ counts serialized bytes, and L_p is the number of remaining denoising steps. Only attention LoRA parameters change, with $W_\phi = W + (\alpha_{\mathrm{L}}/r)B_\phi A_\phi$; the backbone, conditioning modules, and image decoder V_ψ remain fixed. Adapters are shared in advance, without per-image selection or transmission. This decoder stage therefore uses reconstruction losses without a rate term.

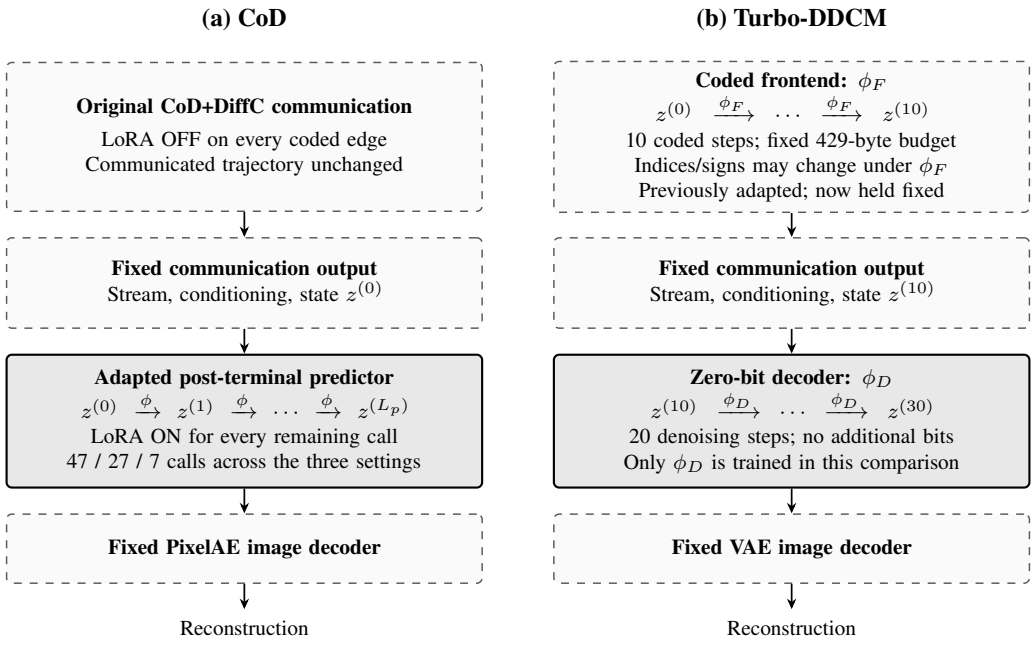

Figure 7: Scope of prediction adaptation in the auxiliary codecs. Dashed boxes are fixed in the decoder comparisons; shaded boxes use LoRA adaptation with a fixed backbone. CoD enables LoRA for every post-terminal denoising call, leaving coded edges unchanged. Turbo first adapts the coded frontend ϕ_F at a fixed 429-byte budget, then freezes it to train the zero-bit decoder ϕ_D on identical streams and start states. Neither transfer includes confidence or path calibration.

Comparison protocol. Both codec studies use DIV2K training subsets separate from the final evaluation on DIV2K80 and Kodak24. CoD resizes inputs to 512^2; Turbo resizes the short side to 1024 and takes a central 1024^2 crop, including Kodak. Rates use processed-image area: CoD

reports mean achieved bpp, while the Turbo decoder comparisons share 429-byte streams. CoD scores clamped float RGB with one coding realization; Turbo scores saved RGB8 and averages three paired realizations within each image. PSNR is mean-image. Paired-image 95% percentile intervals use 10,000 resamples and condition on the evaluated model parameters and realizations, without multiplicity correction.

Predictor interfaces and decoder updates. Both codecs convert their native prediction to a clean latent and apply deterministic denoising, but their time conventions differ. For scheduler label ℓ and flow noise coordinate u_ℓ, define

$$n_\ell = \sqrt{(1-u_\ell)^2 + u_\ell^2}, \qquad a_\ell = \frac{(1-u_\ell)^2}{n_\ell^2}, \qquad y_\ell = n_\ell z_\ell. \tag{30}$$

Thus a_ℓ is signal power, corresponding to α_ℓ^2 in Appendix B.1. Writing $G_\ell^\phi = G_{\theta,\phi}(z_\ell, \ell)$ and suppressing conditioning arguments, the native flow prediction $v_{\theta,\phi}$ gives

$$\begin{aligned} G_\ell^\phi &= y_\ell - u_\ell v_{\theta,\phi}, \qquad \widehat{\epsilon}_\ell^\phi = \frac{z_\ell - \sqrt{a_\ell} G_\ell^\phi}{\sqrt{1-a_\ell}}, \\ F_{\theta,\phi,\ell\to\ell'}(z_\ell) &= \sqrt{a_{\ell'}} G_\ell^\phi + \sqrt{1-a_{\ell'}}\widehat{\epsilon}_\ell^\phi, \qquad \ell' < \ell. \end{aligned} \tag{31}$$

This is the real-valued form of the existing noise-prediction update; each codec retains its deployed operation order and numerical precision.

CoD post-terminal prediction adaptation

CoD runs the original CoD+DiffC communication with LoRA disabled on every coded edge, then reuses exactly the resulting bitstream, conditioning, and receiver-reconstructed terminal latent. The adapter is enabled at the first post-terminal denoising call and remains active for every remaining call. One adapter is shared across all three operating points, acting on the fused query/key/value and output attention projections. The nominal settings 900/500/100 have actual starting labels 897/498/99 and require 47/27/7 adapted denoiser calls, respectively, before the fixed PixelAE inverse reconstructs RGB.

This is post-terminal multi-step decoder prediction calibration. Communication-stage prediction calibration instead changes the receiver on coded DiffC edges, while single-pass terminal prediction uses one final clean-latent estimate. Here the adapter acts throughout latent denoising after communication, with the image decoder unchanged; it is not RGB post-processing. Confidence and path calibration are not transferred.

CoD uses $u_\ell = (\ell - 1)/1000$ and neural time $h_\ell = 1 - \ell/1000$. Its native output $f_{\theta,\phi}$ is combined using classifier-free guidance of strength three, with the sign convention

$$\begin{aligned} f_{\theta,\phi}^{\mathrm{cfg}} &= f_{\theta,\phi}(y_\ell, h_\ell; c_{\mathrm{uncond}}) + 3\big[f_{\theta,\phi}(y_\ell, h_\ell; c) - f_{\theta,\phi}(y_\ell, h_\ell; c_{\mathrm{uncond}})\big], \\ v_{\theta,\phi} &= -f_{\theta,\phi}^{\mathrm{cfg}}. \end{aligned} \tag{32}$$

For clamped RGB in $[0,1]$, let $D_{\mathrm{MSE}} = \|\widehat{x} - x\|_2^2/(3HW)$ and $D_{\mathrm{VGG}} = \mathrm{LPIPS}_{\mathrm{VGG}}(2x-1, 2\widehat{x}-1)$, using the fixed learned LPIPS metric rather than unweighted VGG feature error. Both objectives start from the same previously trained decoder. Its training-image means at each operating point supply fixed positive normalizers $s_{M,p}$, $s_{V,p}$, giving $\widetilde{D}_{\mathrm{MSE}} = D_{\mathrm{MSE}}/s_{M,p}$ and $\widetilde{D}_{\mathrm{VGG}} = D_{\mathrm{VGG}}/s_{V,p}$. The fidelity reference instead uses the matched base decoder with its adapter disabled:

$$g_{i,p}(\phi) = 10\log_{10}\frac{D_{\mathrm{MSE},i,p}(\phi)}{D_{\mathrm{MSE},i,p}^{\mathrm{base}}} = \mathrm{PSNR}_{i,p}^{\mathrm{base}} - \mathrm{PSNR}_{i,p}^{\phi}\,. \tag{33}$$

Suppressing image and operating-point indices, the matched objectives are

$$L_{\mathrm{MSE+VGG}} = \widetilde{D}_{\mathrm{MSE}} + \beta\widetilde{D}_{\mathrm{VGG}}, \qquad L_{\mathrm{constrained}} = \beta\widetilde{D}_{\mathrm{VGG}} + \mathrm{sg}(\eta_p)g. \tag{34}$$

The fixed weight β balances the normalized loss gradients at the common initialization using training data. The two objectives use the same training budget, and the constrained objective has no additional always-active MSE term. Each operating point has its own multiplier, initialized at zero

and updated over nonoverlapping windows $\mathcal{W}_{p,k}$ of training visits:

$$\eta_{p,k+1} = \left[\eta_{p,k} + \frac{\kappa}{|\mathcal{W}_{p,k}|} \sum_{u \in \mathcal{W}_{p,k}} \mathrm{sg}(g_{i_u,p}(\phi_u)) \right]_+, \qquad \eta_{p,0} = 0. \tag{35}$$

Here $\kappa > 0$ is the dual step size and $[\cdot]_+$ projects onto nonnegative values. Violations are measured before the corresponding predictor update; small numerical stabilizers are omitted. The intended constraint $\mathbb{E}_{i|p}[g_{i,p}] \le 0$ limits mean-image PSNR degradation, not each image's distortion or pooled MSE. The signed violation enters the loss; only the multiplier update is projected. Finite training does not guarantee held-out feasibility. The multiplier remains zero in the evaluated CoD training, so its updates use normalized VGG gradients; the results do not establish a benefit from an active fidelity constraint.

At identical communication, the constrained VGG objective improves PSNR and LPIPS over the matched base decoder at all three rates on both datasets in Table 24, but increases DISTS at the lowest rate. The Kodak DISTS interval at setting 500 includes zero. Compared with matched MSE + VGG, it lowers LPIPS and DISTS in every condition at a PSNR cost. At DIV2K setting 900, MSE + VGG obtains 18.19 dB and 0.509 LPIPS, versus 17.11 dB and 0.365 LPIPS for the constrained decoder. Future work could extend this fixed-communication study to the communicated CoD+DiffC trajectory by recalibrating the coded-edge prediction, confidence, and evaluation path.

Turbo coded-frontend prediction adaptation

Turbo-DDCM separates communication-stage frontend prediction adaptation ϕ_F from zero-bit decoder prediction adaptation ϕ_D. The former acts during ten coded steps; the latter during the twenty subsequent denoising steps. Learning ϕ_F can change the selected codebook indices and signs, and hence the transmitted content, while the current coding configuration fixes the serialized length at 429 bytes. The adapted-frontend reference in Fig. 8, denoted by the superscript RF below, uses the previously calibrated ϕ_F on coded steps and the original predictor thereafter.

Turbo uses $u_\ell = \ell/1000$ and the native FLUX velocity at neural time $\ell/1000$, with fixed empty-text conditioning and guidance embedding value one. This embedding differs from CoD's two-branch guidance, and CoD's shifted lookup and sign reversal do not apply.

At each coded step, the sender compares the clean source latent with the current prediction and selects signed vectors from a shared codebook to correct the mismatch. Their indices and signs are transmitted, allowing the receiver to reproduce the correction using the same codebook and predictor. The frontend adapter is trained to reduce the error remaining after this correction, measured against the source-conditioned mean in (36). On step j, the proposal and source-conditional means use the coefficient convention of Appendix B.1, evaluated with Turbo's noise coordinates:

$$\mu^p_{\phi_F,j} = A_j z^{(j)} + B_j G_{\theta,\phi_F}(z^{(j)}, \ell_j), \qquad \mu^q_j = A_j z^{(j)} + B_j z_0. \tag{36}$$

Here z_0 is the clean source latent. For shared codebook C_j and residual $r_j = z_0 - G_{\theta,\phi_F}(z^{(j)}, \ell_j)$, the deployed selector retains the M largest absolute inner products and their signs:

$$\begin{aligned} I_j &= \mathrm{TopM}(|C_j^\top r_j|), \quad h_{j,k} = \mathrm{sign}([C_j^\top r_j]_k), \\ w_j &= \sum_{k \in I_j} h_{j,k} C_{j,k}, \qquad \xi_j = w_j / \mathrm{std}(w_j). \end{aligned} \tag{37}$$

The normalization is the deployed standard deviation across latent coordinates. With posterior variance v_j, the coded update and frontend objective are

$$\begin{aligned} z^{(j+1)} &= \mu^p_{\phi_F,j} + \sqrt{v_j}\xi_j, \qquad r^{\mathrm{post}}_j = z^{(j+1)} - \mu^q_j, \\ L_F &= \frac{1}{K_c} \sum_{j=0}^{K_c-1} \frac{\|r^{\mathrm{post}}_j\|_2^2}{d\, s^{\mathrm{post}}_j}, \qquad K_c = 10. \end{aligned} \tag{38}$$

The positive scales s^{post}_j are fixed from training data, and d is latent dimension. This objective fits post-selection latent residuals; it contains neither an actual-rate loss nor a final-image perceptual loss. The original zero-bit decoder is unchanged during this frontend stage.

Hard-selection gradients. Hard indices and signs are recomputed on every forward pass, then held fixed for the backward pass. The correction $\sqrt{v_j}\xi_j$ and target μ^q_j are also detached. To retain

exact deployed values while supplying gradients through the predictor, write ϵ^{dep} for the deployed noise prediction, ϵ^{cont} for its smooth conversion, and $\mu^{p,\mathrm{smooth}}(z,\epsilon)$ for the Gaussian mean computed from that conversion. Suppressing the step index, the local surrogate is

$$\begin{aligned}\widetilde{\epsilon} &= \mathrm{sg}(\epsilon^{\mathrm{dep}}) + \epsilon^{\mathrm{cont}} - \mathrm{sg}(\epsilon^{\mathrm{cont}}),\\ \widetilde{\mu}^{p} &= \mathrm{sg}(\mu^{p,\mathrm{dep}}) + \mu^{p,\mathrm{smooth}}(z,\widetilde{\epsilon}) - \mu^{p,\mathrm{smooth}}(z,\mathrm{sg}(\epsilon^{\mathrm{dep}})),\\ \widetilde{r}^{\mathrm{post}} &= \widetilde{\mu}^{p} + \mathrm{sg}(\sqrt{v}\xi) - \mathrm{sg}(\mu^{q}).\end{aligned} \tag{39}$$

Both smooth mean terms share the same incoming-state graph. The scalar objective is anchored as $\mathrm{sg}(L_F^{\mathrm{exact}}) + (L_F^{\mathrm{sur}} - \mathrm{sg}(L_F^{\mathrm{sur}}))$, where L_F^{sur} uses $\widetilde{r}^{\mathrm{post}}$. Thus the forward pass retains the hard selection, while training uses a surrogate derivative with its winners fixed. This prior frontend optimization is separate from the final fixed-stream decoder training, whose gradient graph contains no codebook selection.

Turbo zero-bit decoder prediction adaptation

The fixed-stream decoder comparison freezes that frontend, its transmitted stream, conditioning, and receiver-reconstructed start state, and trains only ϕ_D:

$$z^{(10),\phi_D} = \mathrm{sg}(z^{(10),\mathrm{RF}}), \qquad z^{(j+1),\phi_D} = F_{\theta,\phi_D,j}(z^{(j),\phi_D}), \quad j = 10,\dots,29. \tag{40}$$

These two stages transfer prediction adaptation only; neither confidence nor path calibration is used. The state after coding is at label 231; the deterministic continuation ends at label 0 and transmits no further indices. The decoder adapter starts with zero effect on the pretrained predictor. It replaces the frontend adapter at the boundary; the two adapters are not added together.

The frontend is trained on the remaining coded-step error, whereas the decoder is trained on the final reconstructed image. The same decoder adapter is reused across all twenty remaining denoising steps, and the final-image loss is propagated through the complete trajectory. Positive normalizers are training-image means from the adapted-frontend reference. VGG-only training uses $P_{\mathrm{VGG}} = \widetilde{D}_{\mathrm{VGG}}$; mixed training uses

$$P_{\mathrm{mix}} = \tfrac{1}{2}\left(\frac{D_{\mathrm{VGG}}}{s_V} + \frac{D_{\mathrm{LPIPS}}}{s_{\mathrm{LPIPS}}}\right). \tag{41}$$

For either objective P, the matched comparisons use the same image order and training budget:

$$L_{\mathrm{free}} = P, \qquad L_{\mathrm{constrained}} = P + \mathrm{sg}(\eta)g, \qquad g = 10\log_{10}\frac{D_{\mathrm{MSE}}}{D_{\mathrm{MSE}}^{\mathrm{RF}}}. \tag{42}$$

A single nonnegative multiplier follows (35) at this operating point, with zero allowed mean PSNR degradation. Mixed training directly optimizes LPIPS; DISTS is not optimized.

Image-decoder gradients. Turbo preserves the deployed FP16 image-decoder output but uses a frozen FP32 copy of the same stored weights to propagate gradients:

$$\widetilde{x} = \mathrm{sg}(V_\psi^{16}(z)) + \big(V_\psi^{32}(z) - \mathrm{sg}(V_\psi^{32}(z))\big). \tag{43}$$

Both paths include the same latent unpacking, inverse scaling, shift, and clamping. This supplies a surrogate input Jacobian, rather than an exact derivative of the quantized forward computation. The full denoising trajectory remains connected for backpropagation; the fixed coding frontend remains outside that graph.

Mixed perceptual training lowers LPIPS and DISTS relative to the adapted-frontend reference at a PSNR cost in Fig. 8. Against unconstrained mixed training, the fidelity constraint raises mean PSNR by 0.143 dB on DIV2K but lowers it by 0.075 dB on Kodak. VGG-only training improves its feature loss but worsens LPIPS and DISTS. Against Original Turbo-DDCM, the constrained mixed system improves PSNR and LPIPS but increases DISTS on both datasets; differing streams prevent isolating the decoder effect.

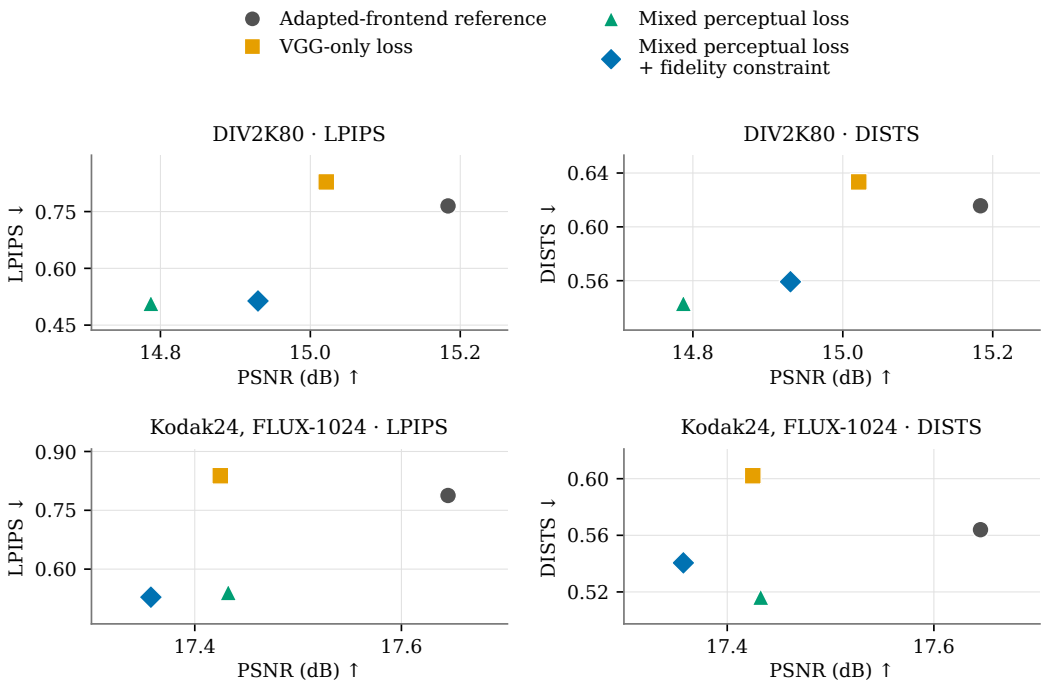

Figure 8: Turbo-DDCM decoder objectives on DIV2K (top) and Kodak (bottom), both using 1024×1024 inputs. All configurations use the same fixed, previously adapted coding frontend and share the same 429-byte bitstreams, conditioning, and decoder start states. Rightward and downward changes are favorable.

Table 24: Adapted-minus-reference effects at fixed communication, extending Table 4 across all three CoD operating points. Brackets are paired-image 95% intervals under the protocol above. Turbo uses the previously adapted coding frontend with the original zero-bit decoder as its adapted-frontend reference.

Dataset	Setting	bpp	ΔPSNR (dB) ↑	ΔLPIPS ↓	ΔDISTS ↓
CoD: fidelity-constrained LPIPS-VGG − matched base decoder					
DIV2K80	900	0.011229	$+1.053$ $[+0.959, +1.154]$	-0.0229 $[-0.0271, -0.0188]$	$+0.0219$ $[+0.0174, +0.0265]$
DIV2K80	500	0.067338	$+0.112$ $[+0.058, +0.153]$	-0.0117 $[-0.0134, -0.0100]$	-0.0063 $[-0.0088, -0.0038]$
DIV2K80	100	0.780811	$+0.192$ $[+0.163, +0.227]$	-0.0019 $[-0.0024, -0.0015]$	-0.0071 $[-0.0081, -0.0061]$
Kodak24	900	0.010735	$+1.308$ $[+1.164, +1.443]$	-0.0308 $[-0.0385, -0.0238]$	$+0.0105$ $[+0.0044, +0.0167]$
Kodak24	500	0.054536	$+0.254$ $[+0.205, +0.304]$	-0.0178 $[-0.0214, -0.0142]$	-0.0006 $[-0.0048, +0.0035]$
Kodak24	100	0.648240	$+0.122$ $[+0.076, +0.170]$	-0.0034 $[-0.0048, -0.0021]$	-0.0029 $[-0.0044, -0.0013]$
Turbo: constrained mixed perceptual loss − adapted-frontend reference					
DIV2K80	30	0.003273	-0.254 $[-0.288, -0.219]$	-0.2512 $[-0.2709, -0.2305]$	-0.0565 $[-0.0611, -0.0517]$
Kodak24	30	0.003273	-0.288 $[-0.354, -0.227]$	-0.2593 $[-0.2869, -0.2316]$	-0.0234 $[-0.0296, -0.0173]$

G Additional Qualitative Comparisons

Fig. 9 shows architectural detail and fine texture at two low-rate settings. The RD panels use the shared-adapter diagnostic variant; the full-image bpp labels describe the displayed reconstructions.

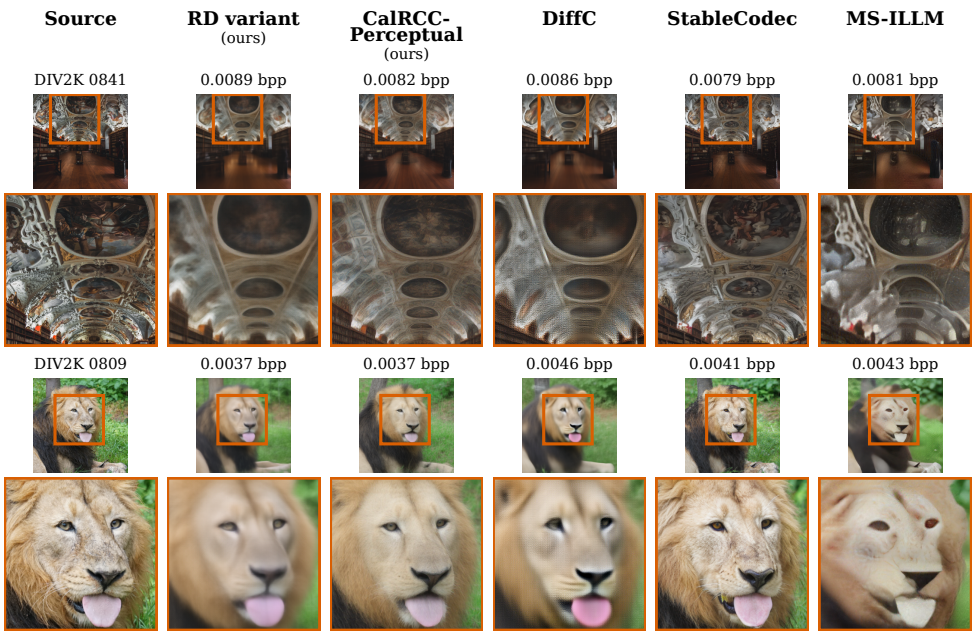

Figure 9: Illustrative DIV2K reconstructions of architectural detail (top) and fine texture (bottom) at two representative low-rate settings. The RD variant column uses the shared-adapter RD predictor.

Fig. 10 illustrates cat facial detail, stone arches, and statue detail. The enlarged arch region shows a regular grid pattern in the displayed DiffC reconstruction. The bottom row uses the final selected CalRCC-Perceptual model; the RD column remains the shared-adapter diagnostic variant. These selected examples illustrate local differences rather than representative performance.

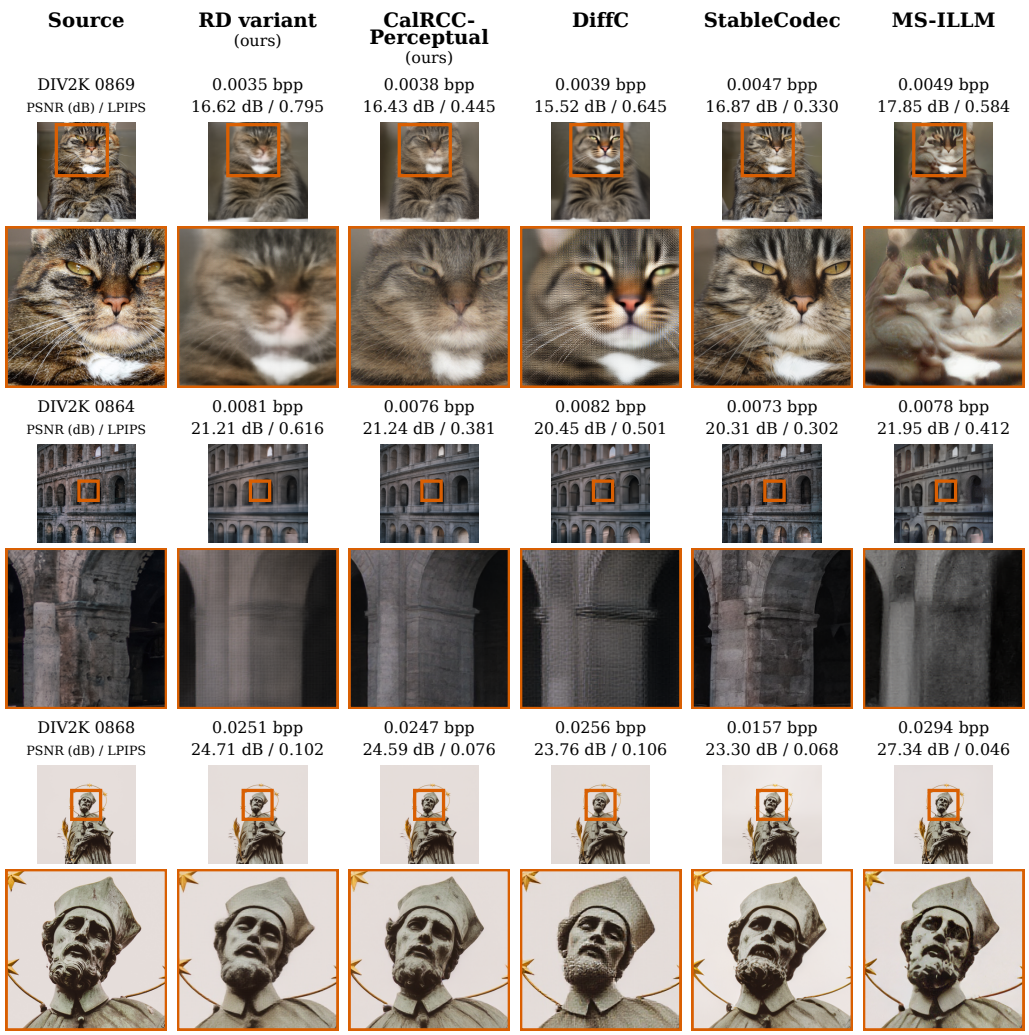

Figure 10: Illustrative DIV2K reconstructions at three low-rate settings, with the same region enlarged across methods. Labels report full-image bpp, PSNR (dB), and LPIPS, not crop metrics; rates are not exactly matched. The RD variant uses the shared-adapter predictor, and MS-ILLM is descriptive context.

Fig. 11 shows DIV2K 0894 (skier) and 0839 (mushroom). CoD uses its post-terminal multi-step decoder comparison at setting 900. The illustrative Turbo coding settings differ from the quantitative fixed-stream study in Appendix F.2, while reusing its trained frontend and decoder without retraining. CoD directly resizes to 512^2; Turbo uses a short-side resize to 1024 and central 1024^2 crop, so their processed source images differ. CoD scores float RGB and Turbo saved RGB8. The examples retain imperfect fine detail, including Turbo blur and grid artifacts.

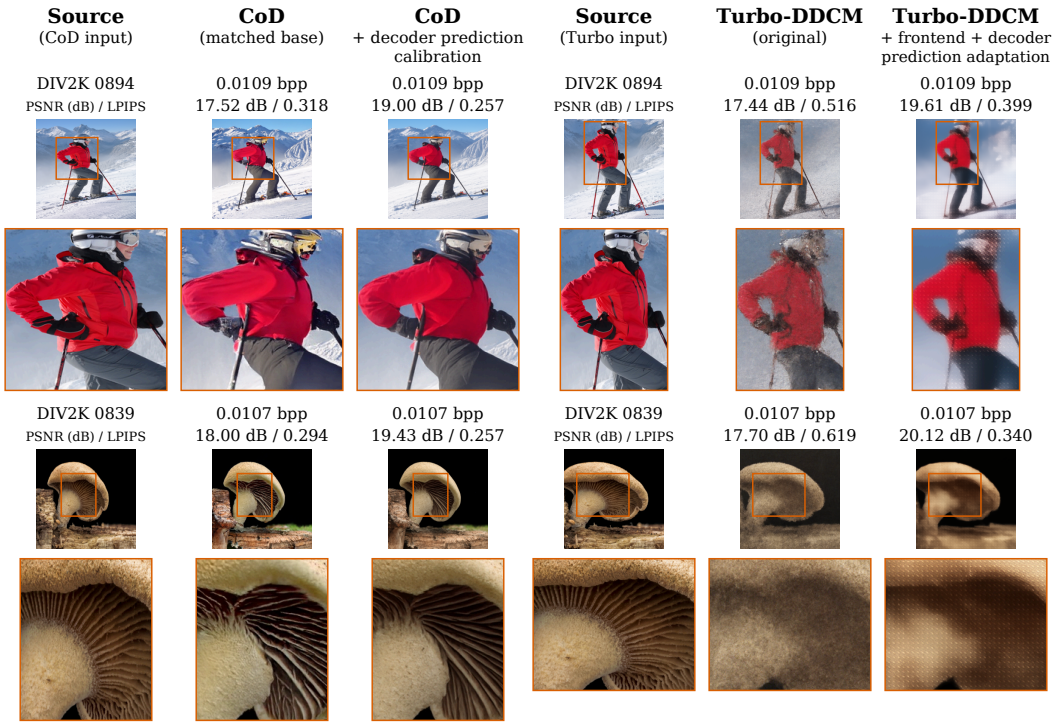

Figure 11: Illustrative CoD and Turbo-DDCM reconstructions of a skier (top) and a mushroom (bottom). CoD compares the base and adapted post-terminal multi-step decoders with the same bitstream and decoded terminal state; Turbo compares original and prediction-adapted frontend–decoder pipelines with different streams. Neither transfer includes confidence or path calibration. Separate source panels reflect different preprocessing. Labels report full-image PSNR (dB), LPIPS, and bpp per processed pixel; Turbo rate excludes shared headers. Boxes identify corresponding enlarged regions.