Noise-Regularized Training for Learned Image Compression

Renjie Zou **Zhiwei Huang*** **Dong Jiang***
Malanshan Audio & Video Laboratory
{zourenjie,huangzhiwei,jiangdong}@mlslabs.com.cn

Abstract

Learned image compression (LIC) is trained almost exclusively on clean images: the only stochastic perturbation in the loop is an additive uniform proxy injected in the *latent* domain to keep the entropy model differentiable. Outside compression, by contrast, a long line of work—from Tikhonov regularization and contractive autoencoders through denoising score matching and modern diffusion models—has established that Gaussian noise at the *input* of a network with a clean target acts as a structured probe of the data manifold rather than as data augmentation. Building on this view, we introduce *noise-regularized training* for LIC: at every training step we add Gaussian noise to the encoder input while keeping the distortion target, the latent uniform proxy, the hyperprior, and the entropy coding pipeline unchanged. The recipe is architecture-agnostic and adds essentially no training or inference overhead. Across backbones and operating regimes it yields consistent BD-Rate gains: -2.60% on a reproduced ELIC and $-0.85\%/-1.05\%/-2.38\%$ on the Base/Medium/Large scales of a variable-rate DMCI codec, with DMCI-Large reaching -22.02% against VTM-22.0. A second-order expansion of the noisy RD loss accounts for these gains: because the noise enters *upstream* of the analysis transform, it converts the clean RD objective into a *structured* $O(\sigma^2)$ regularizer comprising an information-weighted encoder-Jacobian penalty (rate side) and a Frobenius contractive penalty on the end-to-end reconstruction map (distortion side). A finite-difference Lipschitz analysis shows corresponding sensitivity reductions of $\sim 12\%$ for the encoder and $\sim 8\%$ for the end-to-end codec, while an exact-JVP test further shows preferential contraction of high-information directions. Code and trained models are available at `https://github.com/Googolxx/NoiseRegu-LIC`.

1 Introduction

Learned image compression (LIC) [2–9] now rivals and surpasses VVC [1] on standard benchmarks [37] by jointly optimizing an analysis transform g_a, a synthesis transform g_s, and a differentiable entropy model under the rate–distortion (RD) objective $\mathcal{L} = R + \lambda D$. Effort at the field level has concentrated on two axes: more expressive nonlinear transforms [5, 7, 8, 10] and more accurate entropy models [4, 6, 13, 9, 14, 16]. The training signal itself, however, has remained largely fixed: the codec is fed clean images and supervised against the same clean target, and the only stochastic perturbation in the loop is an additive *uniform* proxy injected in the *latent* domain to keep the entropy model differentiable [2].

A different role for noise. Outside compression, a parallel literature has long taught a complementary lesson. Bishop [20] showed that Gaussian input noise is equivalent to Tikhonov regularization,

*Corresponding authors: Zhiwei Huang and Dong Jiang.

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

and Rifai et al. [21] used the same Jacobian-Frobenius penalty to derive contractive autoencoders. More importantly, Vincent [18] and Alain & Bengio [19] proved that a denoising autoencoder trained with Gaussian input corruption and a clean target implicitly learns the Stein score $\nabla \log p_\sigma$ of the noise-smoothed data distribution; modern score-based generative models [22, 24, 25] turn this into a generative procedure by walking samples *onto* the data manifold via Langevin dynamics across a noise-scale ladder. The shared takeaway is that noise injected at the *input* of a network with a clean target is not augmentation but a structured probe of the geometry of the data distribution.

From latent noise to input noise. The latent-domain proxy used in standard LIC is, by contrast, an *operator approximation*: it mimics the statistics of rounding but never asks the encoder to be insensitive to plausible variations in the input image. Inspired by the score matching view above, we study the simplest pixel-domain counterpart of that recipe: at every training step we replace the encoder input by $\tilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\eta}$ with $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 I)$, while leaving the distortion target, the latent uniform proxy, the hyperprior, and the entropy coding pipeline all untouched. There is no architectural change and essentially no overhead at either end of the pipeline: training adds a single additive perturbation in data preprocessing, and at test time the codec runs unmodified.

Empirical effectiveness. This single change yields consistent gains across architectures and bitrates, achieving a -2.60% BD-Rate on a reproduced ELIC [41]. It also improves every scale of the variable-rate DMCI (the intra-image codec of DCVC-RT [42]), with the strongest variant reaching -22.02% against VTM-22.0 on Kodak [37], Tecnick [39], and CLIC [38] (Table 1). These results motivate the central question of the paper: what regularization does pixel-domain input noise impose on the RD objective, and why is it consistently beneficial?

Theoretical picture. A second-order expansion of the noisy RD loss (Section 3) shows that, because the noise enters *upstream* of g_a, it adds two structured $O(\sigma^2)$ corrections: a Frobenius contractive penalty on the end-to-end Jacobian (LIC analogue of the contractive autoencoder [21]) and an *information-weighted* encoder-Jacobian penalty rescaled by the entropy model's curvature, so directions expensive to encode are penalized most. Empirically, the encoder's local Lipschitz drops by $\sim 12\%$ and the end-to-end codec's by $\sim 8\%$, while an exact-JVP test further confirms stronger contraction of high-I_ϕ directions (Fig. 1, Tables 5 and 4).

Contributions.

- **A pixel-domain noise-regularization recipe for LIC.** Motivated by the score matching view of input noise as a manifold probe, we *add* a zero-mean Gaussian perturbation at the encoder input while leaving the standard latent uniform proxy and everything else in the pipeline unchanged, with a clean distortion target. The change is architecture-agnostic and adds essentially no training or inference overhead.
- **A regularization-based explanation.** A local second-order expansion of the noisy RD loss shows that the resulting training signal is a *structured* regularizer: an information-weighted encoder Jacobian penalty (rate side) plus a Frobenius contractive penalty on the end-to-end reconstruction map (distortion side), with a closed-form LIC-specific residual–Hessian correction that is forced to survive by the rate constraint.
- **State-of-the-art RD performance and direct geometric evidence.** The recipe improves a reproduced ELIC by -2.60% BD-Rate on average and every DMCI scale by $-0.85\%/-1.05\%/-2.38\%$ (Base/Medium/Large); DMCI-Large reaches -22.02% against VTM-22.0 without any architectural modification. A finite-difference Lipschitz analysis shows both predicted contractions ($\sim 12\%$ encoder, $\sim 8\%$ end-to-end codec), and an exact-JVP test shows preferential contraction of high-I_ϕ directions.

2 Related Work

Learned image compression. LIC has progressed along two main axes. *Nonlinear transforms* have evolved from GDN-based convolutions [2, 3] through attention [5], window-based Transformers [7, 15], hybrid CNN–Transformer backbones [8], large-receptive-field and adaptive transforms [10], and linear-attention [11]. *Entropy models* have similarly advanced from scale hyperpriors [3] through autoregressive [4], checkerboard [6], channel-wise [13], global–local attention [14],

multi-reference [9], progressive context [12], and Transformer-based [16] context models. All of these improvements optimize a *clean-input* RD loss; the training-signal modification we propose is orthogonal and combines with all of them. A few works examine training dynamics for the quantization step itself, e.g. straight-through estimators applied to LIC [31] or variable-rate conditioning [33, 34], but *input-domain* Gaussian noise has been largely ignored.

Input noise as a manifold probe. A separate line of work has long studied input Gaussian noise as a tool to shape the geometry of a learned representation. Bishop [20] first showed that it is equivalent to Tikhonov regularization. Vincent et al. [17] introduced the denoising autoencoder, and Rifai et al. [21] made the connection to Jacobian-Frobenius penalization explicit. The decisive theoretical step came from Vincent [18] and Alain & Bengio [19]: a noisy-input, clean-target denoising autoencoder trained at corruption level σ implicitly learns the Stein score $\nabla \log p_\sigma$ of the noise-smoothed data distribution. Score-based generative models [22–25] later turned this observation into a generative procedure, walking samples onto the data manifold via Langevin dynamics along a noise-scale ladder. The shared takeaway, that input noise paired with a clean target acts as a structured probe of the data manifold, is the conceptual basis of our method; to our knowledge it has not previously been imported into the compression setting.

Latent vs. input noise in LIC. Standard LIC injects an additive uniform proxy $\tilde{\mathbf{y}} = \mathbf{y} + \mathbf{u}$, $\mathbf{u} \sim \mathcal{U}(-\frac{1}{2}, \frac{1}{2})$, in the *latent* domain purely as a continuous relaxation of quantization [2]; it does not constrain the encoder's sensitivity to input variations. The input-domain perturbation we study acts upstream of g_a and, as shown in Section 3, induces a structured penalty on the encoder's Jacobian under the RD loss. The two mechanisms operate at different stages and are complementary, and the input-side perturbation is conceptually related to randomized smoothing [35]. A small concurrent line of work trains LIC with noisy inputs for the explicit task of joint denoising and compression [36]; the goal there is to *deploy* the codec on corrupted images, whereas we use noise purely as a training-time regularizer and evaluate strictly on clean inputs.

3 Why Pixel-Domain Noise Helps: A Local RD Expansion

We expand the noisy RD loss to second order in the noise scale σ and show that pixel-domain input noise turns the clean RD objective into a *structured* regularizer with two interpretable components: an information-weighted encoder Jacobian penalty (rate side) and a Frobenius contractive penalty on the end-to-end reconstruction map (distortion side). All algebra is collected in Appendix A.3–A.5.

3.1 Setup

Let g_a, g_s be the analysis and synthesis transforms. At inference, $\mathbf{y} = g_a(\mathbf{x})$, $\hat{\mathbf{y}} = Q(\mathbf{y})$, $\hat{\mathbf{x}} = g_s(\hat{\mathbf{y}})$. During training we keep the standard additive-uniform proxy $\mathbf{u} \sim \mathcal{U}(-\frac{1}{2}, \frac{1}{2})$ for the latent and inject Gaussian noise only at the encoder input,

$$\tilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\eta}, \qquad \boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 I), \qquad \bar{\mathbf{y}} = g_a(\tilde{\mathbf{x}}) + \mathbf{u}. \tag{1}$$

The hyperprior variable is suppressed in the notation for clarity; the additional z-pathway and context-model terms are made explicit in Appendix A.4. The noise-regularized training objective is then

$$\mathcal{L}_{\text{noise}} \;=\; \mathbb{E}_{\mathbf{x},\boldsymbol{\eta},\mathbf{u}}\big[\, R(\tilde{\mathbf{x}}) + \lambda\, D(\tilde{\mathbf{x}}, \mathbf{x}) \,\big], \quad D = \|g_s(\bar{\mathbf{y}}) - \mathbf{x}\|_2^2, \quad R = -\log p_\phi(\bar{\mathbf{y}}). \tag{2}$$

With a slight abuse of notation, $R(\tilde{\mathbf{x}})$ should be read as the composite end-to-end rate $R(\tilde{\mathbf{x}}) := -\log p_\phi\big(g_a(\tilde{\mathbf{x}}) + \mathbf{u}\big)$, i.e. the cost of coding the noisy image's latent under the entropy model. The key structural observation is that, because $\boldsymbol{\eta}$ is injected *upstream* of g_a, it propagates through both the distortion path $g_s \circ g_a$ and the rate path $-\log p_\phi(g_a(\cdot) + \mathbf{u})$. Compared with classical denoising-autoencoder analyses [20, 21], where input noise perturbs only a single reconstruction loss, the noisy LIC objective therefore receives *two* second-order corrections from $\boldsymbol{\eta}$, with different physical content. The latent proxy $\mathbf{u}$ is held fixed throughout the $\boldsymbol{\eta}$-expansion (it is independent of $\boldsymbol{\eta}$ and enters only through $\bar{\mathbf{y}}_0 = g_a(\mathbf{x}) + \mathbf{u}$; see Appendix A.4), so it generates no genuine cross terms. Standard mild smoothness assumptions on g_a, g_s, p_ϕ suffice to make the second-order Taylor remainder $O(\sigma^4)$ under Gaussian averaging; the precise statement, and a brief discussion of how modern LIC backbones satisfy it in practice, is deferred to Appendix A.1.

3.2 Distortion branch: end-to-end contraction with an LIC-specific burden

Define the training-time reconstruction map for a fixed quantization proxy $\mathbf{u}$, $f(\mathbf{x}) := g_s\big(g_a(\mathbf{x}) + \mathbf{u}\big)$, and let $D_0(\mathbf{x}) = \|f(\mathbf{x}) - \mathbf{x}\|_2^2$ be the clean-input distortion. A second-order Taylor expansion in $\boldsymbol{\eta}$ followed by Gaussian averaging gives (Appendix A.3)

$$\mathbb{E}_{\boldsymbol{\eta}}\big[D(\tilde{\mathbf{x}}, \mathbf{x})\big] \;=\; D_0(\mathbf{x}) \;+\; \sigma^2\, \|J_f(\mathbf{x})\|_F^2 \;+\; \sigma^2\, \big\langle f(\mathbf{x}) - \mathbf{x},\, \Delta f(\mathbf{x}) \big\rangle \;+\; O(\sigma^4), \tag{3}$$

where Δf is the component-wise Laplacian of f. The first correction $\sigma^2 \|J_f\|_F^2$ is the LIC analogue of the contractive autoencoder penalty [21] and upper-bounds the squared spectral norm $\|J_f\|_2^2$, serving as the natural surrogate of the codec's local Lipschitz constant; we measure the contractions of J_{g_a} and J_f directly in Section 5.3. The second correction is a residual–Hessian interaction *specific to lossy compression*: a classical DAE satisfies $f^\star \approx \mathbf{x}$ at the optimum and the term collapses to $O(\sigma^4)$, but the rate constraint in LIC keeps $f - \mathbf{x} \neq 0$ at any finite bitrate, so the term survives at $O(\sigma^2)$, is sign-indefinite, and is most pronounced at low rate.

3.3 Rate branch: an information-weighted encoder Jacobian

Let $\bar{\mathbf{y}}_0 := g_a(\mathbf{x}) + \mathbf{u}$ denote the clean-input training-time latent, and let $s_\phi := \nabla \log p_\phi$ and $I_\phi := -\nabla^2 \log p_\phi$ be the score and observed information of the entropy model. Expanding the rate $R(\tilde{\mathbf{x}}) = -\log p_\phi(\bar{\mathbf{y}})$ to second order around $\bar{\mathbf{y}}_0$ in the latent perturbation $\delta\mathbf{y} = J_{g_a}(\mathbf{x})\boldsymbol{\eta} + \frac{1}{2} H_{g_a}(\mathbf{x})[\boldsymbol{\eta}, \boldsymbol{\eta}] + O(\|\boldsymbol{\eta}\|^3)$ and averaging over $\boldsymbol{\eta}$ yields (Appendix A.4)

$$\mathbb{E}_{\boldsymbol{\eta}}\big[R(\tilde{\mathbf{x}})\big] \;=\; R_0(\mathbf{x}) \;+\; \tfrac{\sigma^2}{2}\, \mathrm{tr}\big(J_{g_a}(\mathbf{x})^\top\, I_\phi(\bar{\mathbf{y}}_0)\, J_{g_a}(\mathbf{x})\big) \;-\; \tfrac{\sigma^2}{2}\, s_\phi(\bar{\mathbf{y}}_0)^\top\, \Delta g_a(\mathbf{x}) \;+\; O(\sigma^4), \tag{4}$$

with $R_0(\mathbf{x}) := -\log p_\phi(\bar{\mathbf{y}}_0)$ denoting the clean-input rate and Δg_a denoting the component-wise Laplacian of g_a. The Mahalanobis form $\mathrm{tr}(J_{g_a}^\top I_\phi J_{g_a})$ is an *information-weighted* encoder contraction whose strength tracks the rate landscape automatically: for standard LIC entropy models, I_ϕ is positive semi-definite, so this term acts as a genuine penalty; where p_ϕ is sharp, I_ϕ is large and the encoder is strongly penalized for being sensitive to input variation *in those very directions*, while in flatter regions the effect correspondingly weakens. This is the LIC-native counterpart of the contractive penalty of [21], but with no extra hyperparameter beyond σ. The second correction $-\,s_\phi(\bar{\mathbf{y}}_0)^\top \Delta g_a$ is a rate-side score–Laplacian term, locally sign-indefinite; at a well-calibrated entropy model it oscillates around zero and is dominated by the sign-definite $\mathrm{tr}(J_{g_a}^\top I_\phi J_{g_a})$, so we focus the interpretation on the latter while keeping it explicit in Eq. (5).

3.4 Combined regularizer

Substituting Eqs. (3)–(4) into Eq. (2) yields the local form of the noisy RD loss as a clean-input RD term plus four explicit $O(\sigma^2)$ corrections,

$$\mathcal{L}_{\mathrm{noise}}(\mathbf{x}) \;=\; \mathcal{L}_0(\mathbf{x}) \;+\; \tfrac{\sigma^2}{2}\Big(\Omega_R^{(1)} + \Omega_R^{(2)}\Big) \;+\; \lambda\sigma^2\Big(\Omega_D^{(1)} + \Omega_D^{(2)}\Big) \;+\; O(\sigma^4), \tag{5}$$

$$\begin{aligned} \Omega_R^{(1)} &= \mathrm{tr}\big(J_{g_a}^\top I_\phi(\bar{\mathbf{y}}_0) J_{g_a}\big), & \Omega_R^{(2)} &= -\,s_\phi(\bar{\mathbf{y}}_0)^\top \Delta g_a(\mathbf{x}), \\ \Omega_D^{(1)} &= \|J_f(\mathbf{x})\|_F^2, & \Omega_D^{(2)} &= \big\langle f(\mathbf{x}) - \mathbf{x},\, \Delta f(\mathbf{x}) \big\rangle. \end{aligned} \tag{6}$$

The headline regularizers are the two sign-definite penalties $\Omega_R^{(1)}$ and $\Omega_D^{(1)}$, which actively shape the learned codec and are measured directly in Section 5.3; the sign-indefinite $\Omega_R^{(2)}, \Omega_D^{(2)}$ are kept explicit for completeness, with $\Omega_D^{(2)}$ acting as the LIC-specific residual–Hessian burden that the rate constraint cannot make vanish at finite bitrate.

Spectral and manifold interpretation. $\Omega_R^{(1)}$ rescales each input direction by the entropy-model curvature along its image, $\langle J_{g_a}\mathbf{v}_i,\, I_\phi J_{g_a}\mathbf{v}_i \rangle$, where $\mathbf{v}_i$ are the right singular vectors of J_{g_a}. For the mean-scale Gaussian conditionals used in standard LIC entropy models, I_ϕ is diagonal with entries equal to the inverse predicted variances, which the hyperprior adapts spatially: directions the model commits to encoding precisely carry large I_ϕ and are penalized most. The result is an *anisotropic* encoder that retains on-manifold variation and suppresses off-manifold sensitivity, formalizing the manifold-shaping intuition behind the recipe; we test this interpretation in Section 5.3.

Connections to information bottleneck and score matching. Two complementary perspectives further support this picture. An information-bottleneck [30] argument shows that minimizing $\mathcal{L}_{\text{noise}}$ pushes the encoder toward conditional independence of $\mathbf{y}$ from $\boldsymbol{\eta}$ given $\mathbf{x}$, mechanistically consistent with $\Omega_R^{(1)}$ shrinking the very directions through which $\boldsymbol{\eta}$ leaks into the latent. A score matching argument shows that, absent the rate term, the optimum reduces to a classical DAE whose residual aligns at $O(\sigma^2)$ with the Stein score of the noise-smoothed data distribution [18, 19]; the rate constraint perturbs this stationary point at $O(\sigma^2)$ through the LIC-specific scalar penalty $\Omega_D^{(2)}$. The full derivations are deferred to Appendix A.6.

4 Method

Standard pipeline. Given $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$, the encoder produces $\mathbf{y} = g_a(\mathbf{x})$, which is rounded to $\hat{\mathbf{y}} = \lfloor \mathbf{y} \rceil$ at inference and replaced by $\tilde{\mathbf{y}} = \mathbf{y} + \mathcal{U}(-\frac{1}{2}, \frac{1}{2})$ during training [2]. A hyperprior $\mathbf{z} = h_a(\mathbf{y})$ parameterizes the entropy model $p_{\hat{\mathbf{y}}|\hat{\mathbf{z}}}$, and the decoder returns $\hat{\mathbf{x}} = g_s(\hat{\mathbf{y}})$. The standard objective is $\mathcal{L}_{\text{std}} = R + \lambda\, d(\mathbf{x}, \hat{\mathbf{x}})$ with $R = \mathbb{E}[-\log_2 p_{\hat{\mathbf{y}}|\hat{\mathbf{z}}} - \log_2 p_{\hat{\mathbf{z}}}]$.

Noise-regularized objective. Following Section 3, we replace the clean encoder input by a clamped Gaussian-perturbed version while keeping the distortion target clean:

$$\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 I), \quad \tilde{\mathbf{x}} = \text{clamp}(\mathbf{x} + \boldsymbol{\eta},\, 0,\, 1), \quad \mathcal{L}_{\text{noise}} = R(\tilde{\mathbf{x}}) + \lambda\, d\big(\mathbf{x},\, g_s(\bar{\mathbf{y}})\big), \tag{7}$$

with $\bar{\mathbf{y}} = g_a(\tilde{\mathbf{x}}) + \mathbf{u}$ as in Section 3.1. The noisy image is fed into *both* the analysis transform and the hyperprior pathway. The local expansions are understood on the unclamped region. Clamping keeps training inputs in range, and its boundary effect diminishes as σ is annealed; a no-clamp ablation gives similar RD performance (Appendix A.2).

Noise schedule. Annealing σ from a large σ_0 down to 0 separates the geometric-shaping role of $(\Omega_R^{(1)}, \Omega_D^{(1)})$ from the persistent low-rate cost of $\Omega_D^{(2)}$: a large σ early in training biases the encoder–decoder along the data manifold, and $\sigma \to 0$ later sheds all four $O(\sigma^2)$ corrections at once, returning optimization to the clean RD loss (Appendix A.7). Concretely, with T noise-active epochs and initial scale σ_0, we use the exponential schedule

$$\sigma(t) \;=\; \sigma_0 \, \exp\!\big(-k\, t/(T-1)\big), \quad k = \ln 100 \approx 4.6, \tag{8}$$

so $\sigma(T-1) \approx 0.01\, \sigma_0$ and $\sigma(t) = 0$ for $t \geq T$. Within each epoch, individual minibatches draw a noise level $\sigma_{\text{batch}} = r \cdot \sigma(t)$ with $r \sim \mathcal{U}(0, 1)$ (a fresh scalar per minibatch, distinct from the latent proxy $\mathbf{u}$), so the model sees a continuum of noise scales at every point of the schedule rather than a single instantaneous value.

Algorithm. Algorithm 1 summarizes the recipe; the only change from standard LIC training is to perturb the encoder input and keep the reconstruction target clean. A minimal PyTorch sketch with a drop-in `lic_noise` helper is given in Appendix A.8.

5 Experiments

5.1 Experimental Setup

Backbones. We evaluate the recipe on two representative codecs that span both operating regimes used in the field. For the *single-rate* setting we use ELIC [41]; for the *variable-rate* setting we use the intra image codec of DCVC-RT [42], referred to as DMCI in the literature. We reproduce DMCI at three model scales, denoted *DMCI-Base*, *DMCI-Medium*, and *DMCI-Large*, obtained by jointly scaling three channel widths: the analysis/synthesis transform ch_g, the latent $\mathbf{y}$ dimension ch_y, and the hyperprior $\mathbf{z}$ dimension ch_z. Concretely, $(\text{ch}_g, \text{ch}_y, \text{ch}_z) = (368, 256, 128)$ for Base, $(512, 368, 192)$ for Medium, and $(736, 512, 256)$ for Large; all other architectural blocks remain identical across scales.

Training. All models are trained on MLIC-Train-100K [45] with Adam [32]. For *single-rate* ELIC, we use an initial learning rate of 1×10^{-4}, decayed by a factor of 10 at $\sim 85\%$ and $\sim 99\%$ of the

Algorithm 1 Noise-Regularized Training for LIC

Require: Codec (g_a, g_s, h_a, h_s), dataset $\mathcal{X}$, initial noise σ_0, noise epochs T, total epochs T_{total}, RD trade-off λ (a constant for single-rate codecs; drawn from a logarithmic grid $\lambda \in [16, 4096]$ over 64 QPs per minibatch for variable-rate codecs)

for epoch $t = 0, \ldots, T_{\text{total}} - 1$ **do**
  Compute $\sigma(t)$ via Eq. (8); set $\sigma(t) = 0$ if $t \geq T$
  for each minibatch $\mathbf{x} \in \mathcal{X}$ **do**
    Sample $r \sim \mathcal{U}(0, 1)$; $\sigma_{\text{batch}} = r \cdot \sigma(t)$ ▷ schedule multiplier (distinct from latent $\mathbf{u}$)
    if $\sigma_{\text{batch}} > 0$ **then**
      $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2_{\text{batch}} I)$; $\tilde{\mathbf{x}} \leftarrow \text{clamp}(\mathbf{x} + \boldsymbol{\eta}, 0, 1)$
    else
      $\tilde{\mathbf{x}} \leftarrow \mathbf{x}$
    end if
    $\bar{\mathbf{y}} \leftarrow g_a(\tilde{\mathbf{x}}) + \mathbf{u}$; $\hat{\mathbf{x}} \leftarrow g_s(\bar{\mathbf{y}})$; $\mathcal{L} \leftarrow R(\tilde{\mathbf{x}}) + \lambda\, d(\mathbf{x}, \hat{\mathbf{x}})$ ▷ clean target
    Update θ with Adam on $\nabla_\theta \mathcal{L}$
  end for
end for

total $\sim$2M iterations. The first 90% uses batch size 32 and patch size 256; the remaining 10% uses batch size 16 and patch size 512. For *variable-rate* DMCI, we start from a fully trained single-rate counterpart and fine-tune in two stages: 1.4M iterations with the noise-regularized objective (7) at learning rate 1×10^{-4}, followed by 0.1M iterations at patch size 512 and learning rate 1×10^{-5}. Each stage applies a single factor-of-ten decay late in training. Throughout DMCI training, the RD trade-off λ is sampled per minibatch from a logarithmic grid of 64 QPs covering $[16, 4096]$, and the distortion target remains the clean image.

Datasets and metrics. We evaluate on Kodak [37] (24 images, 512×768), CLIC Professional Validation [38] (41 images), and Tecnick [39] (100 images, 1200×1200). All results are reported as BD-Rate [40] savings (negative means bitrate savings at matched reconstruction quality). All baseline and noise-regularized numbers are reproduced under the same training pipeline unless otherwise noted.

5.2 RD Performance

Within-backbone gain (Table 1). On the single-rate ELIC backbone, noise-regularized training reduces BD-Rate by 2.10% (Kodak), 3.28% (Tecnick), and 2.42% (CLIC), with a 2.60% average gain over the reproduced clean-trained baseline. On the variable-rate DMCI backbone, the same recipe yields consistent gains across all three scales: -0.85%, -1.05%, and -2.38% for Base, Medium, and Large, respectively. The largest gain appears on DMCI-Large, suggesting that the regularizer becomes more beneficial as model capacity increases. The official open-source release of DMCI-Base is reported here only for reference; it is $+4.66\%$ worse than our reproduced baseline under the same evaluation setup. Appendix A.10 additionally reports a preliminary measurement on HPCM-Large [12], where the noise-regularized variant lowers bitrate by 3.7–4.6% across Kodak, CLIC, and Tecnick at a slight PSNR degradation; a full BD-Rate sweep on this backbone is deferred to follow-up work.

Across eight DMCI-Base runs with $\sigma_0 \in [0.4, 0.8]$, every run improves over the shared clean baseline on all three datasets, with within-σ_0 standard deviation at most 0.04 percentage points.

Sensitivity to the initial noise scale (Table 2). The recipe exposes a single hyperparameter, the initial noise scale σ_0 in Eq. (8). Sweeping σ_0 on the DMCI-Base backbone (Kodak) shows that very small values (e.g. 0.125) collapse to the no-noise solution at $+0.01\%$, while the BD-Rate saturates around $\sigma_0 \in [0.5,\ 0.75]$ at roughly -0.9%, with $\sigma_0 = 0.5$ giving the best result. The narrow spread across the entire sweep (~ 0.9 pp from worst to best) indicates that sufficiently large initial scales yield similar benefits; we therefore recommend $\sigma_0 \in [0.4, 0.8]$.

Comparison against VTM-22.0 and prior codecs (Table 3). Table 3 situates our trained models alongside VTM-22.0 and a broader set of recent LIC systems—TCM [8], MLIC++ [43], LALIC [11],

Table 1: BD-Rate on Kodak, Tecnick, and CLIC Pro Valid. Each sub-group is anchored to its own baseline (0.00%). All models reproduced under a unified training pipeline unless otherwise noted. Negative: bitrate savings at matched PSNR.

Model	Variant	Kodak	Tecnick	CLIC	Avg.
ELIC	baseline	0.00%	0.00%	0.00%	0.00%
	w/ Noise Reg. (Ours)	**−2.10%**	**−3.28%**	**−2.42%**	**−2.60%**
DMCI	Base (baseline)	0.00%	0.00%	0.00%	0.00%
	w/ Noise Reg. (Ours)	**−0.90%**	**−0.85%**	**−0.80%**	**−0.85%**
	official release†	+2.91%	+5.60%	+5.45%	+4.66%
	Medium (baseline)	0.00%	0.00%	0.00%	0.00%
	w/ Noise Reg. (Ours)	**−0.98%**	**−0.93%**	**−1.24%**	**−1.05%**
	Large (baseline)	0.00%	0.00%	0.00%	0.00%
	w/ Noise Reg. (Ours)	**−2.32%**	**−2.32%**	**−2.50%**	**−2.38%**

† Official open-source release of DMCI-Base; not reproduced under our unified training pipeline.

Table 2: Sensitivity of noise-regularized DMCI-Base to the initial noise scale σ_0 on Kodak. BD-Rate anchored to our reproduced DMCI-Base (clean training).

Initial noise scale σ_0	0.125	0.25	0.50	0.75
BD-Rate on Kodak	+0.01%	−0.16%	**−0.90%**	−0.86%

DCAE [44], and HPCM [12]—reported as absolute BD-Rate against VTM-22.0. Noise-regularized ELIC improves the reproduced ELIC baseline from -6.13% to -8.71% average BD-Rate, a further 2.58 pp gain over VTM-22.0 even on this relatively lightweight baseline. The three DMCI scales reach -18.11%, -20.13%, and -22.02% with the proposed training; in particular, DMCI-Medium w/ Noise Reg. already surpasses HPCM-Large (-19.86%) while using 953 vs. 1261 kMACs/pixel. While this absolute comparison serves as a benchmark snapshot, the isolated marginal gain from noise-regularized training is reported in the within-backbone column of Table 1 (RD curves in Appendix A.11).

5.3 Jacobian Contraction and Information Weighting

The local objective (5) predicts both contraction and its directional structure: the rate-side penalty $\Omega_R^{(1)} = \mathrm{tr}(J_{g_a}^\top I_\phi J_{g_a})$ contracts the *encoder* Jacobian preferentially along high-I_ϕ coordinates, while the distortion-side penalty $\Omega_D^{(1)} = \|J_f\|_F^2$ contracts the *end-to-end* reconstruction map $f = g_s \circ g_a$. We first test both contractions on the variable-rate DMCI-Base backbone with the same randomized finite-difference protocol applied to two probe functions $\Phi \in \{g_a,\ f\}$. For $n = 64$ random unit-norm directions $\mathbf{d}_i$ in pixel space and perturbation magnitude δ, we define a finite-direction local sensitivity estimate

$$\hat{L}_\Phi(\mathbf{x}, \mathrm{QP}) \;=\; \max_i \; \frac{\big\|\Phi(\mathbf{x} + \delta \mathbf{d}_i; \mathbf{q}_{\mathrm{enc}}) - \Phi(\mathbf{x}; \mathbf{q}_{\mathrm{enc}})\big\|_{\mathrm{rms}}}{\delta \, \|\mathbf{d}_i\|_{\mathrm{rms}}}, \qquad \Phi \in \{g_a,\ f\}, \tag{9}$$

where $\mathbf{q}_{\mathrm{enc}}$ is the QP-conditioning vector of the variable-rate codec and $\|\cdot\|_{\mathrm{rms}} = \|\cdot\|_F / \sqrt{\mathrm{numel}}$ is the RMS norm. We use f in its differentiable, quantization-free form $f(\mathbf{x}) = g_s\big(g_a(\mathbf{x})\big)$ ($\mathbf{u} = 0$), matching the theoretical map of Section 3.2. RMS normalization removes the output-dimensionality factor and facilitates comparison across QPs. We sweep $\delta \in \{0.005, 0.01, 0.02, 0.05\}$ and treat as robust only reductions that persist across the entire grid.

Encoder contraction (verifying $\Omega_R^{(1)}$). Figure 1 (a,b) reports $\hat{L}_{g_a}$ at the focal probe scale $\delta = 0.01$. The noise-regularized encoder is *consistently less sensitive* at every operating point, with an average 12.0% reduction across QPs (per-QP range 9.2%–14.7%). For concreteness, $\hat{L}_{g_a}$ contracts from $18.12 \to 16.10$ at QP 0 and from $38.25 \to 32.66$ at QP 63. Both models follow the expected monotone increase of $\hat{L}_{g_a}$ with QP, since higher rates demand finer spatial discrimination; noise regularization reduces this sensitivity across all operating points.

Table 3: Main RD comparison against VTM-22.0 in BD-Rate on Kodak, Tecnick, and CLIC. Negative values indicate bitrate savings at matched PSNR. Our noise-regularized variants are highlighted in bold; the best overall result is additionally bolded in the metric columns.

Method	kMACs/pixel	Param. (M)	Type	Kodak	Tecnick	CLIC	Avg.
TCM	1824	77	Single-Rate	-10.46%	-10.91%	-9.23%	-10.20%
MLIC++	1283	117	Single-Rate	-13.61%	-17.18%	-14.50%	-15.10%
LALIC	667	66	Single-Rate	-14.01%	-17.71%	-15.61%	-15.78%
DCAE	940	119	Single-Rate	-15.72%	-20.16%	-17.13%	-17.67%
HPCM-Base	919	69	Single-Rate	-14.67%	-17.37%	-15.26%	-15.77%
HPCM-Large	1261	90	Single-Rate	-18.70%	-21.44%	-19.43%	-19.86%
ELIC (reproduced)	578	32	Single-Rate	-5.87%	-8.11%	-4.42%	-6.13%
ELIC w/ Noise Reg.	578	32	Single-Rate	$\mathbf{-7.98\%}$	$\mathbf{-11.35\%}$	$\mathbf{-6.79\%}$	$\mathbf{-8.71\%}$
DMCI-Base (official)	486	46	Variable-Rate	-12.67%	-15.38%	-12.63%	-13.56%
DMCI-Base	486	46	Variable-Rate	-15.23%	-19.87%	-16.99%	-17.36%
DMCI-Base w/ Noise Reg.	486	46	Variable-Rate	$\mathbf{-16.04\%}$	$\mathbf{-20.59\%}$	$\mathbf{-17.70\%}$	$\mathbf{-18.11\%}$
DMCI-Medium	953	91	Variable-Rate	-17.37%	-21.85%	-18.45%	-19.23%
DMCI-Medium w/ Noise Reg.	953	91	Variable-Rate	$\mathbf{-18.24\%}$	$\mathbf{-22.63\%}$	$\mathbf{-19.53\%}$	$\mathbf{-20.13\%}$
DMCI-Large	1924	181	Variable-Rate	-18.50%	-22.67%	-18.93%	-20.03%
DMCI-Large w/ Noise Reg.	1924	181	Variable-Rate	$\mathbf{-20.47\%}$	$\mathbf{-24.55\%}$	$\mathbf{-21.06\%}$	$\mathbf{-22.02\%}$

Table 4: Encoder information-weighting test on DMCI-Large with exact JVPs and a shared noise-model I_ϕ. Weighted contraction exceeds unweighted contraction at every QP, yielding a positive gap of 7.70–17.72 percentage points.

QP	0	9	18	27	36	45	54	63
Weighted $1 - R_w$	39.15%	41.41%	46.56%	50.99%	46.08%	40.52%	38.05%	38.03%
Unweighted $1 - R_u$	29.75%	27.77%	31.13%	33.27%	32.28%	30.52%	30.13%	30.33%
Gap (pp)	$+9.41$	$+13.64$	$+15.43$	$+17.72$	$+13.80$	$+10.00$	$+7.91$	$+7.70$

Which encoder directions contract? The 12% reduction is an overall average: it does not tell us whether all directions shrink equally or some shrink more. To examine this directional structure at a larger model scale, we next evaluate exact JVPs on DMCI-Large and compare the same encoder response with and without entropy-model weighting. For each direction, let $v = J_{g_a} d$; $u = \|v\|^2$ measures its plain response energy, while $p = v^\top I_\phi v$ weights that response by coding curvature. We sum these energies over 24 Kodak images and the same $n = 64$ directions per image and QP, using a shared noise-model I_ϕ, and define

$$R_u = \frac{\sum_{\mathbf{x},d} u_{\text{noise}}}{\sum_{\mathbf{x},d} u_{\text{base}}}, \qquad R_w = \frac{\sum_{\mathbf{x},d} p_{\text{noise}}}{\sum_{\mathbf{x},d} p_{\text{base}}}. \tag{10}$$

A positive gap $R_u - R_w$ means that high-I_ϕ directions contract more strongly. Table 4 shows this pattern at every QP.

End-to-end codec contraction (verifying $\Omega_D^{(1)}$). We next return to overall sensitivity, applying the same protocol to the complete encoder–decoder map $\Phi = f$ to probe the distortion-side penalty $\|J_f\|_F^2$ directly (Figure 1 (c,d)). Noise regularization again contracts the codec at every operating point, with an 8.4% average reduction (per-QP range 7.4–10.0%). Two qualitative differences with the encoder result are worth noting. First, $\hat{L}_f$ is both *smaller* in absolute scale (9.08 at QP 0 vs. 18.12 for $\hat{L}_{g_a}$) and *less QP-dependent* ($9.08 \to 7.51$ from QP 0 to QP 63), consistent with the synthesis transform g_s partially undoing the encoder's amplification. Second, the relative reduction is *stronger at low rate* (QP 0: -9.6%) and weakens slowly at high rate (QP 63: -7.9%), the opposite of the encoder trend.

Summary. Both contractions persist across an order-of-magnitude sweep of δ (Table 5; full per-QP, per-δ numbers in Appendix A.9). The JVP result adds a simple directional picture: the encoder does not shrink all directions equally, but contracts high-I_ϕ directions more strongly. Together, these measurements capture both the magnitude and directional structure of the local RD regularizer derived in Section 3.

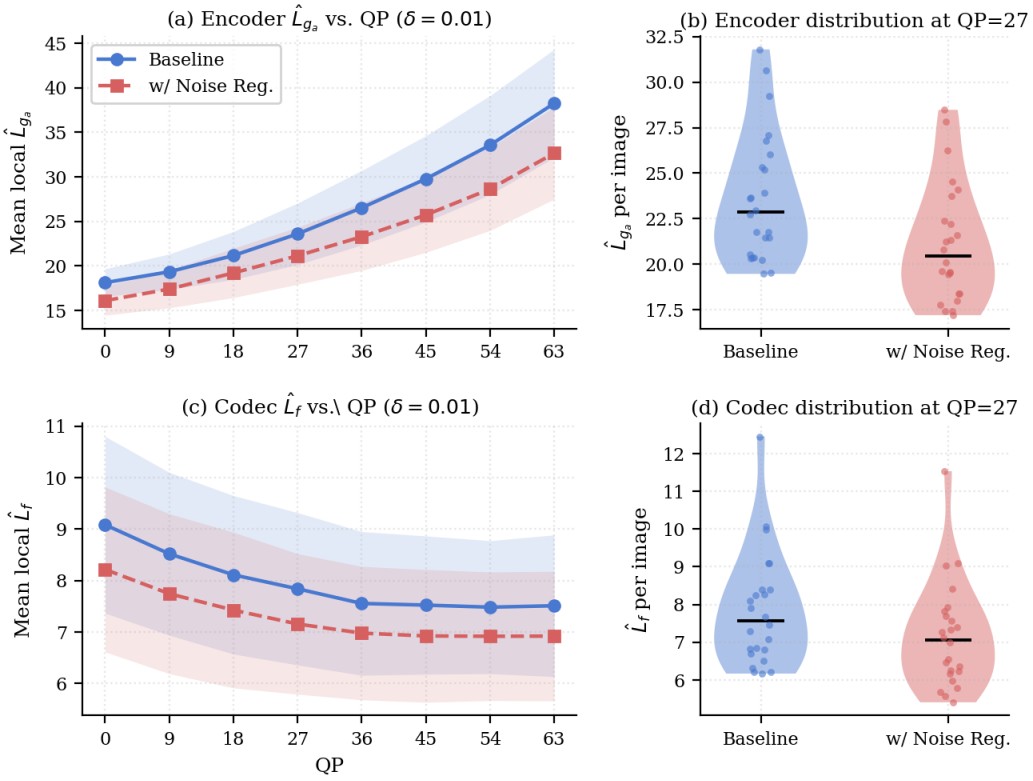


Figure 1: **Local Lipschitz analysis on Kodak (DMCI-Base, $\delta = 0.01$).** Top row, encoder $\hat{L}_{g_a}$ (tests $\Omega_R^{(1)}$); bottom row, end-to-end codec $\hat{L}_f$ on the quantization-free map $f = g_s \circ g_a$ (tests $\Omega_D^{(1)} = \|J_f\|_F^2$). Left panels (a,c): mean $\hat{L}$ vs. QP for the baseline and the noise-regularized model. Right panels (b,d): per-image distribution at QP $=$ 27, estimated with $n = 64$ random directions. Noise regularization contracts both Jacobians at every operating point (12.0% encoder, 8.4% codec on average; see Table 5).

Table 5: **Summary of relative reduction in local Lipschitz $\hat{L}$ on Kodak DMCI-Base (baseline $\rightarrow$ noise-regularized).** Encoder column tests $\Omega_R^{(1)}$; end-to-end column tests $\Omega_D^{(1)}$. Each cell averages Eq. (9) over 24 Kodak images, 8 QPs, and $n = 64$ random directions. Full per-QP per-δ tables are in Appendix A.9.

Probe scale δ	0.005	0.01	0.02	0.05	QP range
Encoder $\hat{L}_{g_a}$ (tests $\Omega_R^{(1)}$)	11.69%	12.00%	12.23%	12.24%	8.5%–14.8%
End-to-end $\hat{L}_f$ (tests $\Omega_D^{(1)}$)	8.21%	8.36%	8.45%	8.52%	7.4%–10.0%

6 Discussion and Conclusion

A single conceptual step—adding a pixel-domain Gaussian perturbation to the encoder input on top of the standard latent quantization proxy—yields a structured RD regularizer with a closed-form local description: an information-weighted encoder-Jacobian penalty (rate side) and a Frobenius contractive penalty on the end-to-end reconstruction map (distortion side), alongside a sign-indefinite residual–Hessian term that the rate constraint forces to survive at finite bitrate. The two perturbations are complementary, operating at different stages: the latent proxy provides differentiability around quantization, while pixel-domain noise penalizes encoder sensitivity to plausible input variation. The Jacobian analysis (Section 5.3) supports both predicted contractions and preferential contraction of high-I_ϕ directions, while BD-Rate gains are uniform across backbones, scales, and bitrates.

Table 6: Perceptual DMCI-Large: BD-Rate of noise-regularized training against the corresponding clean model. Negative values indicate bitrate savings.

Dataset	PSNR	MS-SSIM	LPIPS	DISTS
Kodak	-1.63%	-1.99%	-2.13%	-2.84%
Tecnick	-0.63%	-0.77%	-1.12%	-1.94%
CLIC	-1.01%	-1.49%	-2.83%	-4.85%

Table 7: Perceptual DMCI-Large: BD-Rate against VTM-22.0. Negative values indicate bitrate savings relative to VTM.

Dataset	Model	PSNR	MS-SSIM	LPIPS	DISTS
Kodak	Clean	-14.20%	-18.29%	-90.41%	-90.67%
Kodak	NoiseRegu	$\mathbf{-15.75\%}$	$\mathbf{-20.04\%}$	$\mathbf{-90.66\%}$	$\mathbf{-91.01\%}$
Tecnick	Clean	-17.10%	-14.62%	-87.34%	-88.29%
Tecnick	NoiseRegu	$\mathbf{-17.65\%}$	$\mathbf{-15.32\%}$	$\mathbf{-87.43\%}$	$\mathbf{-88.43\%}$
CLIC	Clean	-8.59%	-6.26%	-91.49%	-92.92%
CLIC	NoiseRegu	$\mathbf{-9.69\%}$	$\mathbf{-7.87\%}$	$\mathbf{-91.65\%}$	$\mathbf{-93.07\%}$

Connection to diffusion-style training. Pixel-domain noise with a clean reconstruction target gives LIC a denoising-autoencoder interpretation: the codec still predicts the clean image and is trained with an x-space loss, loosely analogous to x_0-prediction in diffusion models. The optimal denoising residual further connects this reconstruction objective to the score of the noise-smoothed data distribution [18, 19]. This connection alone does not implement $\boldsymbol{\epsilon}$- or v-prediction, which require reparameterizing the prediction target and typically introduce noise-level-dependent loss weighting [23–26]. Exploring such reparameterizations, including an x-prediction variant with a v-space loss, is a natural direction for future work.

Beyond PSNR: gains on perceptual and structural metrics. We also fine-tune DMCI-Large with a mixed MSE, MS-SSIM [27], LPIPS [28], and DISTS [29] objective. Tables 6 and 7 report consistent gains over both the corresponding clean model and VTM-22.0 across all three datasets and four metrics.

Limitations. The theoretical analysis is a second-order expansion in σ derived for MSE distortion, while the perceptual results are empirical extensions beyond this setting. We use a single exponential annealing schedule, whose interaction with rate conditioning remains to be studied.

Acknowledgements and Disclosures

Funding in direct support of this work was provided by Malanshan Audio & Video Laboratory through institutional research and computing resources.

References

[1] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, "Overview of the Versatile Video Coding (VVC) standard and its applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3736–3764, 2021.

[2] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *Proc. ICLR*, 2017.

[3] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *Proc. ICLR*, 2018.

[4] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Proc. NeurIPS*, 2018.

[5] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized Gaussian mixture likelihoods and attention modules," in *Proc. CVPR*, 2020.

[6] D. He, Y. Zheng, B. Sun, Y. Wang, and H. Qin, "Checkerboard context model for efficient learned image compression," in *Proc. CVPR*, 2021.

[7] R. Zou, C. Song, and Z. Zhang, "The devil is in the details: Window-based attention for image compression," in *Proc. CVPR*, 2022.

[8] J. Liu, H. Sun, and J. Katto, "Learned image compression with mixed transformer-CNN architectures," in *Proc. CVPR*, 2023.

[9] W. Jiang, J. Yang, Y. Zhai, P. Ning, F. Gao, and R. Wang, "MLIC: Multi-reference entropy model for learned image compression," in *Proc. ACM MM*, 2023.

[10] W. Jiang, P. Ning, J. Yang, Y. Zhai, F. Gao, and R. Wang, "LLIC: Large receptive field transform coding with adaptive weights for learned image compression," *IEEE Trans. Multimedia*, vol. 26, pp. 10937–10951, 2024.

[11] D. Feng, Z. Cheng, S. Wang, R. Wu, H. Hu, G. Lu, and L. Song, "Linear Attention Modeling for Learned Image Compression," in *Proc. CVPR*, 2025.

[12] Y. Li, H. Zhang, L. Li, and D. Liu, "Learned Image Compression with Hierarchical Progressive Context Modeling," in *Proc. ICCV*, 2025.

[13] D. Minnen and S. Singh, "Channel-wise autoregressive entropy models for learned image compression," in *Proc. ICIP*, 2020.

[14] J.-H. Kim, B. Heo, and J.-S. Lee, "Joint global and local hierarchical priors for learned image compression," in *Proc. CVPR*, 2022.

[15] Y. Zhu, Y. Yang, and T. Cohen, "Transformer-based transform coding," in *Proc. ICLR*, 2022.

[16] Y. Qian, M. Lin, X. Sun, Z. Tan, and R. Jin, "Entroformer: A transformer-based entropy model for learned image compression," in *Proc. ICLR*, 2022.

[17] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. ICML*, 2008.

[18] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural Computation*, vol. 23, no. 7, pp. 1661–1674, 2011.

[19] G. Alain and Y. Bengio, "What regularized auto-encoders learn from the data-generating distribution," *Journal of Machine Learning Research*, vol. 15, no. 110, pp. 3743–3773, 2014.

[20] C. M. Bishop, "Training with noise is equivalent to Tikhonov regularization," *Neural Computation*, vol. 7, no. 1, pp. 108–116, 1995.

[21] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proc. ICML*, 2011.

[22] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *Proc. NeurIPS*, 2019.

[23] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. NeurIPS*, 2020.

[24] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *Proc. ICLR*, 2021.

[25] T. Karras, M. Aittala, T. Aila, and S. Laine, "Elucidating the design space of diffusion-based generative models," in *Proc. NeurIPS*, 2022.

[26] T. Salimans and J. Ho, "Progressive distillation for fast sampling of diffusion models," in *Proc. ICLR*, 2022.

[27] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, 2003.

[28] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. CVPR*, 2018.

[29] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Image quality assessment: Unifying structure and texture similarity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2567–2581, 2022.

[30] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *arXiv:physics/0004057*, 2000.

[31] L. Theis, W. Shi, A. Cunningham, and F. Huszár, "Lossy image compression with compressive autoencoders," in *Proc. ICLR*, 2017.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *arXiv preprint arXiv:1412.6980*, 2014.

[33] Y. Choi, M. El-Khamy, and J. Lee, "Variable rate deep image compression with a conditional autoencoder," in *Proc. ICCV*, 2019.

[34] Z. Cui, J. Wang, S. Gao, T. Guo, Y. Feng, and B. Bai, "Asymmetric gained deep image compression with continuous rate adaptation," in *Proc. CVPR*, 2021.

[35] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *Proc. ICML*, 2019.

[36] B. Brummer and C. De Vleeschouwer, "On the importance of denoising when learning to compress images," in *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023.

[37] E. Kodak, "Kodak lossless true color image suite (PhotoCD PCD0992)," `http://r0k.us/graphics/kodak/`, 1993.

[38] G. Toderici, W. Shi, R. Timofte, L. Theis, J. Ballé, E. Agustsson, N. Johnston, and F. Mentzer, "Workshop and Challenge on Learned Image Compression (CLIC2020)," `https://archive.compression.cc/`, 2020.

[39] N. Asuni and A. Giachetti, "TESTIMAGES: A large-scale archive for testing visual devices and basic image processing algorithms," in *Proc. STAG*, 2014.

[40] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," *ITU-T SG16/Q6, Doc. VCEG-M33*, 2001.

[41] D. He, Z. Yang, W. Peng, R. Ma, H. Qin, and Y. Wang, "ELIC: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding," in *Proc. CVPR*, 2022.

[42] Z. Jia, B. Li, J. Li, W. Xie, L. Qi, H. Li, and Y. Lu, "Towards practical real-time neural video compression," in *Proc. CVPR*, 2025.

[43] W. Jiang, J. Yang, Y. Zhai, F. Gao, and R. Wang, "MLIC++: Linear complexity multi-reference entropy modeling for learned image compression," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 21, no. 5, 2025.

[44] J. Lu, L. Zhang, X. Zhou, M. Li, W. Li, and S. Gu, "Learned Image Compression with Dictionary-based Entropy Model," in *Proc. CVPR*, 2025.

[45] W. Jiang, "MLIC-Train-100K," Hugging Face dataset repository, 2024. Available: `https://huggingface.co/datasets/Whiteboat/MLIC-Train-100K`

A Technical Appendices and Supplementary Material

This appendix collects the longer derivations that would otherwise interrupt the main narrative of Section 3. Its goal is to make every algebraic step that produces the local objective (5) explicitly verifiable, with no missing intermediate manipulations. The notation is kept identical to the main text and is summarized in Table 8 for quick reference. Throughout, all derivatives are evaluated at the clean input $\mathbf{x}$ unless stated otherwise.

A.1 Smoothness Assumptions

For the Gaussian-averaged Taylor expansions of Sections A.3–A.4 to admit an $O(\sigma^4)$ remainder, we assume that g_a, g_s, and the entropy log-density $-\log p_\phi$ are C^4 on the relevant supports of $\tilde{\mathbf{x}}$ and $\bar{\mathbf{y}}$. The residual is then bounded by an $O(\|\boldsymbol{\eta}\|^4)$ Lagrange remainder, whose Gaussian expectation is $O(\sigma^4)$. Third-order odd moments vanish, so the next nonzero correction after the $O(\sigma^2)$ terms is collected in this remainder. (A weaker C^3 assumption would yield an expected $O(\sigma^3)$ remainder with the same $O(\sigma^2)$ structure.) The standard analysis and synthesis transforms of LIC, built from convolutions with smooth activations (e.g. WSiLU), satisfy this in practice.

A.2 Clamping

The expansions below use the unclamped perturbation $\tilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\eta}$, for which the zero-mean, isotropic Gaussian moment identities apply. Clamping can introduce boundary corrections, but its effect diminishes as σ is annealed to zero. Empirically, a no-clamp DMCI-Base ablation at $\sigma_0 = 0.75$ changes the average BD-Rate across Kodak, Tecnick, and CLIC only from -0.77% with clamping to -0.65% without clamping, a difference of 0.13 percentage points. Thus, the noise-regularization gain is retained without clamping.

A.3 Distortion-Side Expansion

This subsection derives the $O(\sigma^2)$ correction to the distortion branch announced in Eq. (3) of the main text. We work in five elementary steps.

Setup. For a fixed quantization proxy $\mathbf{u}$, define the training-time end-to-end reconstruction map and the per-sample distortion as

$$f(\mathbf{x}) \;:=\; g_s\big(g_a(\mathbf{x}) + \mathbf{u}\big), \qquad D(\tilde{\mathbf{x}}, \mathbf{x}) \;:=\; \|f(\tilde{\mathbf{x}}) - \mathbf{x}\|_2^2. \tag{11}$$

Because the distortion target is the clean image $\mathbf{x}$, the noise $\boldsymbol{\eta}$ enters D *only* through the argument of f; this is the structural reason why the distortion branch produces both a Jacobian penalty and a residual–Hessian term.

Step 1: Taylor expansion of each output coordinate. For any output coordinate f_k of the map f, a second-order Taylor expansion of f_k around the clean input $\mathbf{x}$ gives

$$f_k(\tilde{\mathbf{x}}) \;=\; f_k(\mathbf{x}) \;+\; \nabla f_k(\mathbf{x})^\top \boldsymbol{\eta} \;+\; \tfrac{1}{2}\,\boldsymbol{\eta}^\top \nabla^2 f_k(\mathbf{x})\,\boldsymbol{\eta} \;+\; O(\|\boldsymbol{\eta}\|^3), \tag{12}$$

where ∇f_k and $\nabla^2 f_k$ denote the gradient and Hessian of the scalar function $f_k : \mathbb{R}^d \to \mathbb{R}$. From now on, derivatives are evaluated at $\mathbf{x}$ unless stated otherwise.

Step 2: Squaring and grouping by order. Substituting Eq. (12) into the squared error, expanding the bracket, and collecting terms by their order in $\boldsymbol{\eta}$, we obtain

$$\begin{aligned} \big\|f(\tilde{\mathbf{x}}) - \mathbf{x}\big\|_2^2 \;=\; & \sum_k \big(f_k(\mathbf{x}) - x_k\big)^2 \\ & +\; 2\sum_k \big(f_k(\mathbf{x}) - x_k\big)\,\nabla f_k^\top \boldsymbol{\eta} \\ & +\; \sum_k \big(f_k(\mathbf{x}) - x_k\big)\,\boldsymbol{\eta}^\top \nabla^2 f_k\,\boldsymbol{\eta} \\ & +\; \sum_k \big(\nabla f_k^\top \boldsymbol{\eta}\big)^2 \;+\; O\big(\|\boldsymbol{\eta}\|^3\big). \end{aligned} \tag{13}$$

Table 8: Notation used throughout this appendix. Cross-references to the main text equations are given in the right column where applicable.

Symbol	Meaning
Inputs and noise	
$\mathbf{x}$	clean input image (the distortion target)
$\boldsymbol{\eta}$	encoder-side Gaussian noise, $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 I)$
σ	noise scale (the small parameter of the local expansion)
$\tilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\eta}$	noisy encoder input
$\mathbf{u}$	uniform proxy for latent quantization, $\mathbf{u} \sim \mathcal{U}(-\frac{1}{2}, \frac{1}{2})$
Codec, Jacobians, and Hessians	
$g_a,\ g_s$	analysis (encoder) and synthesis (decoder) transforms
$f(\mathbf{x}) = g_s\big(g_a(\mathbf{x}) + \mathbf{u}\big)$	training-time end-to-end reconstruction map (fixed $\mathbf{u}$)
$J_{g_a}(\mathbf{x}),\ J_f(\mathbf{x})$	Jacobian of g_a and f at $\mathbf{x}$
$\nabla^2 f_k(\mathbf{x})$	Hessian of the scalar coordinate $f_k : \mathbb{R}^d \to \mathbb{R}$ at $\mathbf{x}$
$H_{g_a}(\mathbf{x})$	component-wise Hessian tensor of g_a, see Eq. (19)
$\Delta f(\mathbf{x}),\ \Delta g_a(\mathbf{x})$	component-wise Laplacian of f and g_a at $\mathbf{x}$
Latent and entropy model	
$\bar{\mathbf{y}}_0 = g_a(\mathbf{x}) + \mathbf{u}$	training-time latent at the clean input
$\bar{\mathbf{y}} = g_a(\tilde{\mathbf{x}}) + \mathbf{u}$	training-time latent at the noisy input
$\delta\mathbf{y} = \bar{\mathbf{y}} - \bar{\mathbf{y}}_0$	latent perturbation induced by $\boldsymbol{\eta}$
p_ϕ	entropy model on the latent $\mathbf{y}$
$s_\phi(\mathbf{y}) = \nabla \log p_\phi(\mathbf{y})$	score of the entropy model
$I_\phi(\mathbf{y}) = -\nabla^2 \log p_\phi(\mathbf{y})$	observed information matrix of p_ϕ
Loss decomposition	
$D(\tilde{\mathbf{x}}, \mathbf{x}) = \lVert f(\tilde{\mathbf{x}}) - \mathbf{x} \rVert_2^2$	per-sample distortion (clean target)
$R(\tilde{\mathbf{x}}) = -\log p_\phi(\bar{\mathbf{y}})$	per-sample rate
$\mathcal{L}_0,\ \mathcal{L}_{\text{noise}}$	clean-input and noisy-input RD losses, Eq. (2)
$\Omega_R^{(1)}$	information-weighted Jacobian penalty $\operatorname{tr}(J_{g_a}^\top I_\phi J_{g_a})$
$\Omega_R^{(2)}$	rate-side score–Laplacian term $-s_\phi^\top \Delta g_a$
$\Omega_D^{(1)}$	end-to-end Frobenius penalty $\lVert J_f \rVert_F^2$
$\Omega_D^{(2)}$	residual–Laplacian term $\langle f - \mathbf{x},\ \Delta f \rangle$
λ	RD trade-off coefficient

The four lines on the right-hand side are, in order, the clean-input distortion $D_0(\mathbf{x}) = \lVert f(\mathbf{x}) - \mathbf{x} \rVert_2^2$, a linear-in-$\boldsymbol{\eta}$ term, a residual×Hessian quadratic, and a squared-Jacobian quadratic.

Step 3: Gaussian moment identities. For $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 I)$, the first two moments are $\mathbb{E}[\boldsymbol{\eta}] = \mathbf{0}$ and $\mathbb{E}[\boldsymbol{\eta}\boldsymbol{\eta}^\top] = \sigma^2 I$. These imply, for any deterministic vector $\mathbf{a} \in \mathbb{R}^d$ and any deterministic symmetric matrix $A \in \mathbb{R}^{d \times d}$, the elementary contractions

$$\mathbb{E}\big[\mathbf{a}^\top \boldsymbol{\eta}\big] \;=\; 0, \qquad \mathbb{E}\big[(\mathbf{a}^\top \boldsymbol{\eta})^2\big] \;=\; \sigma^2 \lVert \mathbf{a} \rVert_2^2, \qquad \mathbb{E}\big[\boldsymbol{\eta}^\top A\, \boldsymbol{\eta}\big] \;=\; \sigma^2 \operatorname{tr}(A). \tag{14}$$

These three identities are the workhorses of every step that follows in both Section A.3 and Section A.4.

Step 4: Term-by-term expectation. Apply Eq. (14) to each line of Eq. (13):

- the linear-in-$\boldsymbol{\eta}$ term has zero expectation;
- the residual×Hessian term contracts via $\mathbb{E}\big[\boldsymbol{\eta}^\top \nabla^2 f_k\, \boldsymbol{\eta}\big] = \sigma^2 \operatorname{tr}\big(\nabla^2 f_k\big) = \sigma^2\, \Delta f_k(\mathbf{x})$;
- the squared-Jacobian term contracts via $\mathbb{E}\big[(\nabla f_k^\top \boldsymbol{\eta})^2\big] = \sigma^2 \lVert \nabla f_k \rVert_2^2$.

Summing over the output index k, the total $O(\sigma^2)$ contribution to the expected distortion becomes

$$\sigma^2 \sum_k \big(f_k(\mathbf{x}) - x_k\big)\, \Delta f_k(\mathbf{x}) \;+\; \sigma^2 \sum_k \big\lVert \nabla f_k(\mathbf{x}) \big\rVert_2^2. \tag{15}$$

Step 5: Compact form. The two sums above are, respectively, the residual–Laplacian inner product and the squared Frobenius norm of the Jacobian:

$$\sum_k \big(f_k(\mathbf{x}) - x_k\big)\,\Delta f_k(\mathbf{x}) \;=\; \big\langle f(\mathbf{x}) - \mathbf{x},\, \Delta f(\mathbf{x})\big\rangle, \qquad \sum_k \big\|\nabla f_k(\mathbf{x})\big\|_2^2 \;=\; \|J_f(\mathbf{x})\|_F^2. \tag{16}$$

Combining the clean-input contribution with the two $O(\sigma^2)$ corrections yields the distortion-side expansion in the form quoted as Eq. (3) of the main text:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\eta}}\big[D(\tilde{\mathbf{x}},\mathbf{x})\big] \;=\; & \underbrace{\|f(\mathbf{x}) - \mathbf{x}\|_2^2}_{D_0(\mathbf{x})} \\ & + \sigma^2\, \underbrace{\|J_f(\mathbf{x})\|_F^2}_{\Omega_D^{(1)}} \\ & + \sigma^2\, \underbrace{\big\langle f(\mathbf{x}) - \mathbf{x},\, \Delta f(\mathbf{x})\big\rangle}_{\Omega_D^{(2)}} + O(\sigma^4).\end{aligned} \tag{17}$$

The first $O(\sigma^2)$ correction $\sigma^2\,\Omega_D^{(1)}$ is the end-to-end contractive Jacobian penalty discussed in Section 3.2. The second, $\sigma^2\,\Omega_D^{(2)}$, is the LIC-specific residual–Hessian term: it survives at $O(\sigma^2)$ precisely because the rate constraint keeps $f(\mathbf{x}) \neq \mathbf{x}$ at finite bitrate, whereas a classical denoising autoencoder satisfies $f^\star(\mathbf{x}) \approx \mathbf{x}$ at the optimum and the same term is suppressed to $O(\sigma^4)$.

Equivalent Laplacian form. Differentiating the squared error $\sum_k (f_k(\mathbf{x}) - x_k)^2$ twice with respect to $\mathbf{x}$ and contracting indices yields the pointwise identity

$$\tfrac{1}{2}\,\Delta\big\|f(\mathbf{x}) - \mathbf{x}\big\|_2^2 \;=\; \big\|J_f(\mathbf{x}) - I\big\|_F^2 \;+\; \big\langle f(\mathbf{x}) - \mathbf{x},\, \Delta f(\mathbf{x})\big\rangle, \tag{18}$$

which can be substituted into Eq. (17) to obtain an equivalent decomposition of the $O(\sigma^2)$ distortion correction in terms of $\|J_f\|_F^2$, $\|J_f - I\|_F^2$, and $\Delta\|f - \mathbf{x}\|_2^2$. The two formulations agree to leading order in σ.

A.4 Rate-Side Expansion

This subsection derives the $O(\sigma^2)$ correction to the rate branch announced as Eq. (4), in three logical movements: expand g_a in $\boldsymbol{\eta}$ to obtain the latent perturbation $\delta\mathbf{y}$; read off its first two moments; and propagate them through a second-order expansion of $-\log p_\phi$. All derivatives are evaluated at the clean point ($\mathbf{x}$ for g_a and its derivatives, $\bar{\mathbf{y}}_0$ for s_ϕ and I_ϕ) unless stated otherwise.

Step 1: Latent perturbation from a noisy input. Taylor-expanding g_a around $\mathbf{x}$ to second order in $\boldsymbol{\eta}$ and subtracting $\bar{\mathbf{y}}_0 = g_a(\mathbf{x}) + \mathbf{u}$ from both sides of $\bar{\mathbf{y}} = g_a(\tilde{\mathbf{x}}) + \mathbf{u}$, the proxy $\mathbf{u}$ cancels and we obtain

$$\delta\mathbf{y} \;=\; J_{g_a}\,\boldsymbol{\eta} \;+\; \tfrac{1}{2}\,H_{g_a}[\boldsymbol{\eta},\boldsymbol{\eta}] \;+\; O\big(\|\boldsymbol{\eta}\|^3\big), \qquad \big(H_{g_a}[\boldsymbol{\eta},\boldsymbol{\eta}]\big)_k := \boldsymbol{\eta}^\top \nabla^2 (g_a)_k\, \boldsymbol{\eta}, \tag{19}$$

where the component-wise contraction is indexed by the output coordinate k.

Step 2: First two moments of $\delta\mathbf{y}$. Applying the Gaussian identities (14) component-wise to Eq. (19): the linear term has zero mean; the Hessian term contributes to the mean at $O(\sigma^2)$ and to the covariance only at $O(\sigma^4)$. We read off

$$\mathbb{E}[\delta\mathbf{y}] \;=\; \tfrac{\sigma^2}{2}\,\Delta g_a, \qquad \mathbb{E}[\delta\mathbf{y}\,\delta\mathbf{y}^\top] \;=\; \sigma^2\, J_{g_a}\, J_{g_a}^\top + O(\sigma^4). \tag{20}$$

Step 3: Local expansion of the rate and its expectation. A second-order expansion of $R(\tilde{\mathbf{x}}) = -\log p_\phi(\bar{\mathbf{y}})$ around $\bar{\mathbf{y}}_0$ in $\delta\mathbf{y}$ gives, with $\nabla(-\log p_\phi) = -s_\phi$ and $\nabla^2(-\log p_\phi) = I_\phi$,

$$R(\tilde{\mathbf{x}}) \;=\; R_0(\mathbf{x}) \;-\; s_\phi^\top \delta\mathbf{y} \;+\; \tfrac{1}{2}\,\delta\mathbf{y}^\top I_\phi\,\delta\mathbf{y} \;+\; O\big(\|\delta\mathbf{y}\|^3\big), \tag{21}$$

where $R_0(\mathbf{x}) := -\log p_\phi(\bar{\mathbf{y}}_0)$ is the clean-input rate. Taking the expectation over $\boldsymbol{\eta}$ and substituting Eq. (20) (using $\mathbb{E}[\delta\mathbf{y}^\top I_\phi\,\delta\mathbf{y}] = \mathrm{tr}(I_\phi\,\mathbb{E}[\delta\mathbf{y}\delta\mathbf{y}^\top])$ and the trace's cyclicity) yields the rate-side expansion in the form quoted as Eq. (4) of the main text,

$$\mathbb{E}_{\boldsymbol{\eta}}\big[R(\tilde{\mathbf{x}})\big] \;=\; R_0(\mathbf{x}) \;+\; \tfrac{\sigma^2}{2}\,\Omega_R^{(1)}(\mathbf{x}) \;+\; \tfrac{\sigma^2}{2}\,\Omega_R^{(2)}(\mathbf{x}) \;+\; O(\sigma^4), \tag{22}$$

with the two rate-side $O(\sigma^2)$ corrections

$$\Omega_R^{(1)}(\mathbf{x}) := \mathrm{tr}\big(J_{g_a}^\top I_\phi J_{g_a}\big), \qquad \Omega_R^{(2)}(\mathbf{x}) := -\, s_\phi^\top \Delta g_a. \tag{23}$$

Cross terms with the latent proxy $\mathbf{u}$ vanish at this order: $\mathbf{u}$ is independent of $\boldsymbol{\eta}$ and enters only through $\bar{\mathbf{y}}_0$, contributing no $\boldsymbol{\eta}$-dependent term up to $O(\sigma^4)$. The interpretation of the two corrections is taken up jointly with the distortion-side terms in Appendix A.5.

Hyperprior and context models. For a hyperprior, the perturbation also propagates through $\bar{\mathbf{z}} = h_a(\bar{\mathbf{y}})$, adding a curvature term

$$\Omega_z = \mathrm{tr}\big(J_{h_a \circ g_a}^\top I_z J_{h_a \circ g_a}\big)\,,$$

together with cross-Hessian and drift terms. The curvature term is positive semidefinite for Gaussian-type hyperpriors; the remaining terms need not have a definite sign. Context models further introduce off-diagonal information entries, so the diagonal inverse-variance reading of I_ϕ applies only to elementwise mean-scale Gaussian conditionals.

A.5 Combined Local Objective

We now assemble the distortion-side expansion (17) and the rate-side expansion (22) into the local form of the noisy RD loss quoted as Eq. (5). Multiplying Eq. (17) by λ, adding it to Eq. (22), and grouping each $O(\sigma^2)$ correction with its branch yields

$$\begin{aligned} \mathcal{L}_{\mathrm{noise}}(\mathbf{x}) \;&= \mathcal{L}_0(\mathbf{x}) \;+\; \tfrac{\sigma^2}{2}\,\big(\Omega_R^{(1)}(\mathbf{x}) + \Omega_R^{(2)}(\mathbf{x})\big) \\ &\quad + \lambda\,\sigma^2\,\big(\Omega_D^{(1)}(\mathbf{x}) + \Omega_D^{(2)}(\mathbf{x})\big) \;+\; O(\sigma^4), \end{aligned} \tag{24}$$

with $\mathcal{L}_0(\mathbf{x}) := R_0(\mathbf{x}) + \lambda\, D_0(\mathbf{x})$ the clean-input RD loss, the rate-side corrections $\Omega_R^{(1)}, \Omega_R^{(2)}$ given by Eq. (23), and the distortion-side corrections $\Omega_D^{(1)}, \Omega_D^{(2)}$ given by Eq. (17). Eq. (24) is the local objective of the main text (Eq. (5)) in expanded form. The four corrections have distinct status: $\Omega_R^{(1)}$ is the sign-definite, information-aware encoder contraction; $\Omega_R^{(2)}$ is the rate-side score–Laplacian term, locally nonzero and sign-indefinite per sample, with a marginal expectation that is not strictly zero in general but exhibits zero-mean oscillatory behavior at a well-calibrated entropy model; $\Omega_D^{(1)} = \|J_f\|_F^2$ is the sign-definite, end-to-end codec contraction, measured directly in Section 5.3; $\Omega_D^{(2)}$ is the LIC-specific, sign-indefinite residual–Laplacian burden. The dynamic role of σ in trading off the useful pair $\{\Omega_R^{(1)}, \Omega_D^{(1)}\}$ against the LIC-specific cost $\Omega_D^{(2)}$ is taken up in Appendix A.7.

A.6 Connections to Information Bottleneck and Score Matching

This subsection expands the two perspectives sketched at the end of Section 3.4. The full derivations are not strictly needed to use the local objective (5), but they make explicit the connections to two well-known principles.

Information-bottleneck connection. With $\tilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\eta}$ and a clean distortion target $\mathbf{x}$, the rate term upper-bounds $I(\mathbf{y}; \tilde{\mathbf{x}})$, and the squared-error distortion is, up to a data entropy that does not depend on the encoder, an upper bound on $H(\mathbf{x} \mid \mathbf{y})$, so its negation *lower*-bounds $I(\mathbf{y}; \mathbf{x})$. The data-processing inequality gives $I(\mathbf{y}; \mathbf{x}) \le I(\mathbf{y}; \tilde{\mathbf{x}})$, with equality iff $\mathbf{y}$ is conditionally independent of $\boldsymbol{\eta}$ given $\mathbf{x}$. Minimizing $\mathcal{L}_{\mathrm{noise}}$ therefore pushes the encoder toward the conditional-independence optimum: the latent retains image content and discards $\boldsymbol{\eta}$. This is mechanistically consistent with $\Omega_R^{(1)}$, which shrinks the encoder Jacobian precisely along the expensive directions through which $\boldsymbol{\eta}$ would otherwise leak into the latent.

Score-matching connection. If we drop the rate term and focus only on the noisy-input, clean-target distortion, classical DAE theory [18, 19] gives the optimum

$$f^\star(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}} \;=\; \sigma^2\, \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}), \qquad q_\sigma = p_{\mathrm{data}} * \mathcal{N}(0, \sigma^2 I), \tag{25}$$

i.e. the optimal residual aligns at $O(\sigma^2)$ with the Stein score of the noise-smoothed data distribution, biasing the codec toward high-density image regions. In LIC the rate constraint perturbs this optimal vector field at $O(\sigma^2)$, and the resulting shift in the stationary point of the loss is governed by the inclusion of the LIC-specific scalar penalty $\Omega_D^{(2)}$ in Eq. (5).

Listing 1: Minimal PyTorch sketch of the training loop. The callable `lic_noise` applies the schedule in Eq. 8 and injects the Gaussian perturbation of Algorithm 1.

```
from lic_noise import lic_noise

noise = lic_noise(noise_max=sigma_0, noise_epoch=T, total_epoch=T_total)

for epoch in range(T_total):
    for x in dataloader:
        x_tilde, sigma_batch = noise(x, epoch)
        x_hat, rate = codec(x_tilde)
        loss = rate + lam * mse(x_hat, x)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
```

A.7 Why Annealing Matters

The local objective (5) shows that every $O(\sigma^2)$ correction carries an explicit factor of σ^2; so does the gradient field,

$$\nabla_\theta \mathcal{L}(\theta; \sigma(t)) \;\approx\; \nabla_\theta \mathcal{L}_0(\theta) \;+\; \sigma(t)^2\, \nabla_\theta \Omega(\theta), \tag{26}$$

with Ω the combined regularizer of Eq. (5). Because σ multiplies all four corrections jointly, scheduling it modulates the regularization *as a whole*; what changes between training phases is therefore not the relative weighting between $\Omega_R^{(1)}, \Omega_D^{(1)}$ and $\Omega_D^{(2)}$, but how much the entire Ω is allowed to deflect the gradient field away from the clean RD loss $\nabla_\theta \mathcal{L}_0$.

Why anneal at all (and not just train at fixed σ). A fixed-σ run pays a permanent $O(\sigma^2)$ correction from all four terms at convergence; among them, the sign-indefinite residual–Hessian cost $\Omega_D^{(2)}$ is the dominant remaining penalty at low rate, since the rate constraint forces $f - \mathbf{x} \neq 0$ regardless of how well-trained the network is. Annealing instead separates the regularizer's two jobs in time: (i) *geometric shaping*, front-loaded. Early in training the network is still searching for a manifold parameterization, and a large σ activates the helpful pair $(\Omega_R^{(1)}, \Omega_D^{(1)})$ as a strong prior that contracts both encoder and end-to-end Jacobians along expensive directions. (ii) *Rate–distortion fine-tuning*, back-loaded. Late in training the manifold structure has been mostly captured, and any residual $O(\sigma^2)$ correction acts as a constraint rather than a guide; sending $\sigma(t) \to 0$ removes *all four* corrections together, so the helpful pair is shed alongside $\Omega_D^{(2)}$, but by this point the helpful pair has already done its shaping work, while the harmful term would otherwise persist throughout training. The optimizer then completes convergence on the clean RD loss, where it can fine-tune the rate–distortion trade-off precisely.

Coupling with entropy-model maturation. The information-weighted Jacobian penalty $\Omega_R^{(1)}$ is only as informative as the entropy model that supplies I_ϕ. Early in training I_ϕ tracks an immature density estimator and the penalty is correspondingly noisy; late in training I_ϕ is accurate but $\sigma^2(t)$ is small. The two effects pull in opposite temporal directions, so a sensible schedule keeps σ moderate during the window in which the entropy model matures, rather than front-loading all of the noise. In practice $\sigma_0 \in [0.1, 0.5]$ on the $[0, 1]$ pixel scale, an annealing horizon overlapping the entropy model's calibration period, and a terminal $\sigma \approx 0$ jointly satisfy these requirements; this is the design behind Eq. (8), and it is empirically corroborated by the σ_0 sweep in Table 2, where any sufficiently large initial scale ($\sigma_0 \gtrsim 0.5$) saturates the BD-Rate gain on DMCI-Base.

A.8 Drop-in Training Interface

Listing 1 gives a minimal PyTorch sketch of the training loop in Algorithm 1. The variables `x`, `x_tilde`, `x_hat`, and `sigma_batch` correspond to $\mathbf{x}$, $\tilde{\mathbf{x}}$, $\hat{\mathbf{x}}$, and σ_{batch}, respectively; the schedule in Eq. (8) is wrapped by `lic_noise`.

A.9 Full Lipschitz Sweep Tables

For completeness we report the full per-QP, per-δ relative reductions underlying the summary in Table 5 of the main text. Table 9 covers the encoder local Lipschitz constant $\hat{L}_{g_a}$ ($\Phi = g_a$ in Eq. (9)), the empirical signature of the rate-side penalty $\Omega_R^{(1)}$. Table 10 covers the end-to-end codec Lipschitz constant $\hat{L}_f$ ($\Phi = f$), the empirical signature of the distortion-side penalty $\Omega_D^{(1)} = \|J_f\|_F^2$.

Table 9: **Full sweep: relative reduction (%) in encoder local Lipschitz $\hat{L}_{g_a}$ on Kodak (DMCI-Base, baseline $\rightarrow$ noise-regularized).** Each cell averages Eq. (9) ($\Phi = g_a$) over 24 Kodak images and $n = 64$ random directions. Row averages stay within a 0.6-pp band as δ varies over an order of magnitude.

δ	QP = 0	9	18	27	36	45	54	63	avg.
0.005	10.28	9.33	8.54	10.17	11.99	13.71	14.82	14.70	11.69
0.01	11.14	10.06	9.22	10.42	12.25	13.60	14.68	14.61	12.00
0.02	11.91	10.76	9.71	10.59	12.73	13.32	14.42	14.39	12.23
0.05	12.90	11.49	9.85	10.27	12.42	12.75	14.03	14.22	12.24

Table 10: **Full sweep: relative reduction (%) in end-to-end codec local Lipschitz $\hat{L}_f$ on Kodak (DMCI-Base, baseline $\rightarrow$ noise-regularized).** Each cell averages Eq. (9) applied to $\Phi = f$ over 24 Kodak images and $n = 64$ random directions. Row averages stay within a 0.31-pp band; the per-QP trend is opposite to that of the encoder, with the strongest reduction at the lowest rate.

δ	QP = 0	9	18	27	36	45	54	63	avg.
0.005	9.33	8.84	8.36	8.75	7.51	7.82	7.44	7.62	8.21
0.01	9.56	9.10	8.46	8.72	7.64	7.97	7.56	7.85	8.36
0.02	9.81	9.30	8.47	8.79	7.83	8.02	7.47	7.86	8.45
0.05	9.99	9.57	8.45	8.56	7.98	8.20	7.38	8.05	8.52

A.10 Additional Backbone: HPCM-Large

For completeness, we also apply the noise-regularized training to HPCM-Large [12] as a preliminary out-of-sample check beyond the headline ELIC/DMCI evaluations. On this auxiliary backbone we evaluate at a single calibrated operating point per dataset; a multi-rate sweep that would yield a proper BD-Rate is left to follow-up work. Table 11 reports paired bpp/PSNR numbers for the reproduced clean-trained baseline and the noise-regularized variant on Kodak, CLIC, and Tecnick. At each anchor, the noise-regularized model lowers bitrate by 3.7–4.6% with only a small PSNR drop (-0.07 to -0.12 dB), a trend in the same direction as the rate–distortion improvements observed on ELIC and DMCI in Table 1. While a single operating point per dataset does not allow us to compute a BD-Rate, these measurements are consistent with a favorable transfer of the recipe to this third architecture; the full multi-rate verification is deferred to follow-up work.

Table 11: HPCM-Large at a single calibrated operating point on Kodak, CLIC, and Tecnick. We list bpp and PSNR for the reproduced clean-trained baseline and the noise-regularized variant; Δbpp is the relative bitrate change and ΔPSNR the absolute quality change of the noise-regularized variant relative to the corresponding baseline. We do not aggregate these single-point pairs into a BD-Rate.

Dataset	Variant	bpp	PSNR (dB)	Δbpp	ΔPSNR (dB)
Kodak	HPCM-Large (baseline)	0.8313	38.022	—	—
	HPCM-Large w/ Noise Reg.	0.7965	37.906	-4.19%	-0.116
CLIC	HPCM-Large (baseline)	0.6151	38.691	—	—
	HPCM-Large w/ Noise Reg.	0.5867	38.598	-4.61%	-0.093
Tecnick	HPCM-Large (baseline)	0.5613	38.667	—	—
	HPCM-Large w/ Noise Reg.	0.5403	38.598	-3.74%	-0.070

A.11 Per-Dataset RD Curves

As a visual complement to the BD-Rate numbers in Tables 1 and 3, Figures 2–4 plot the full PSNR-vs.-bpp curves on Kodak, CLIC Professional Validation, and Tecnick.

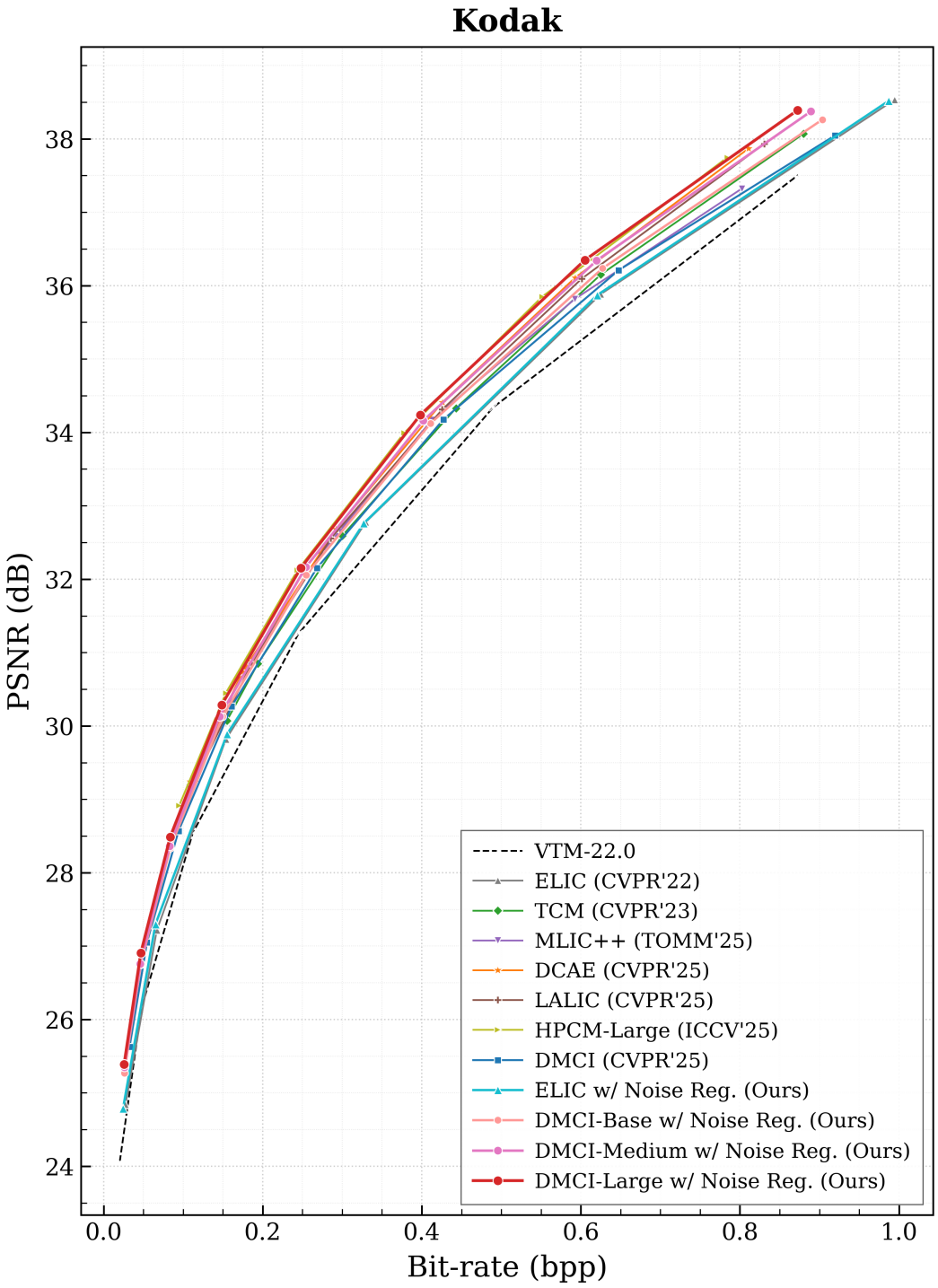


Figure 2: RD curves on **Kodak**. PSNR vs. bitrate (bpp). Curves closer to the upper-left corner are better.

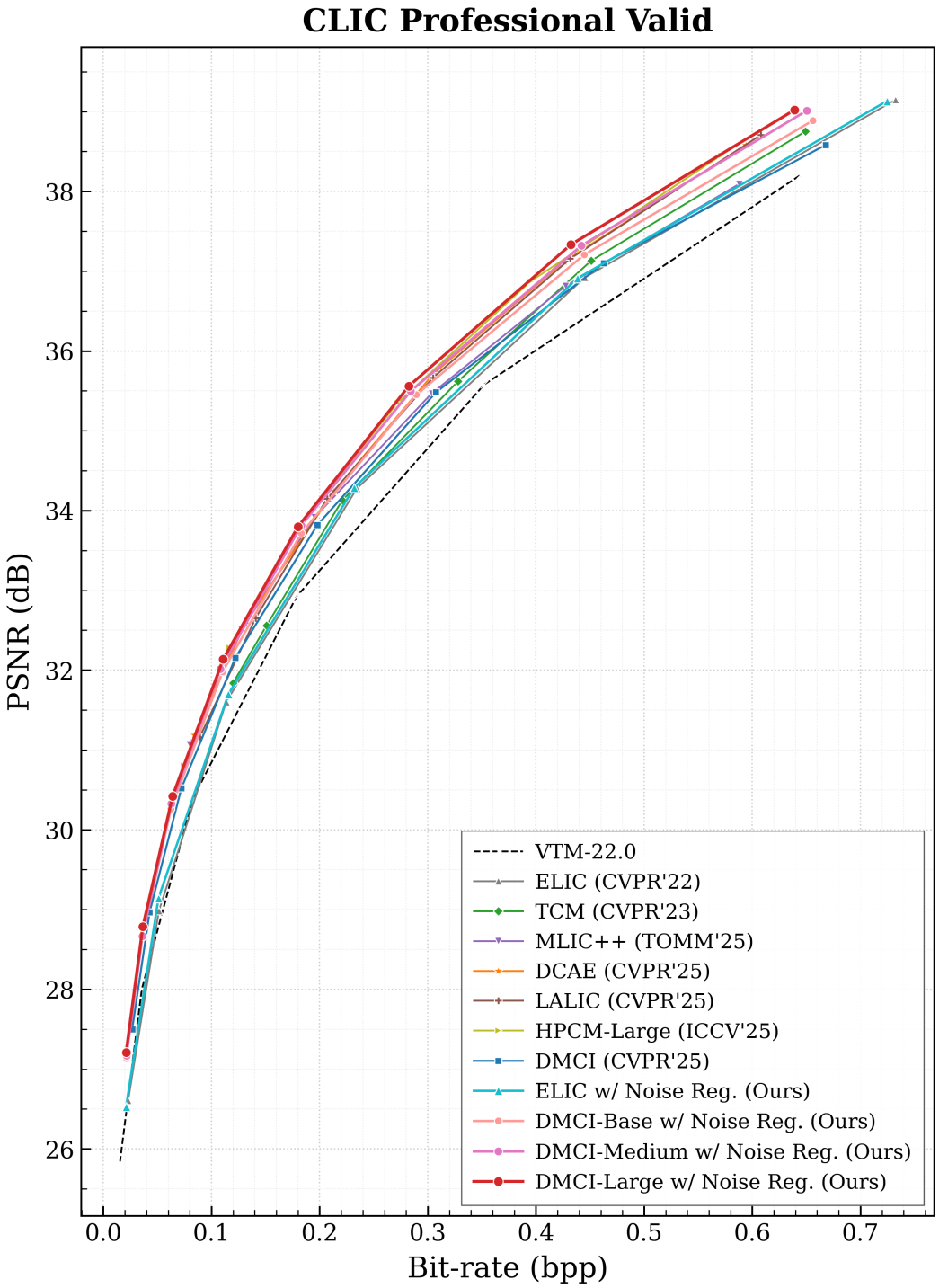


Figure 3: RD curves on **CLIC Professional Valid**.

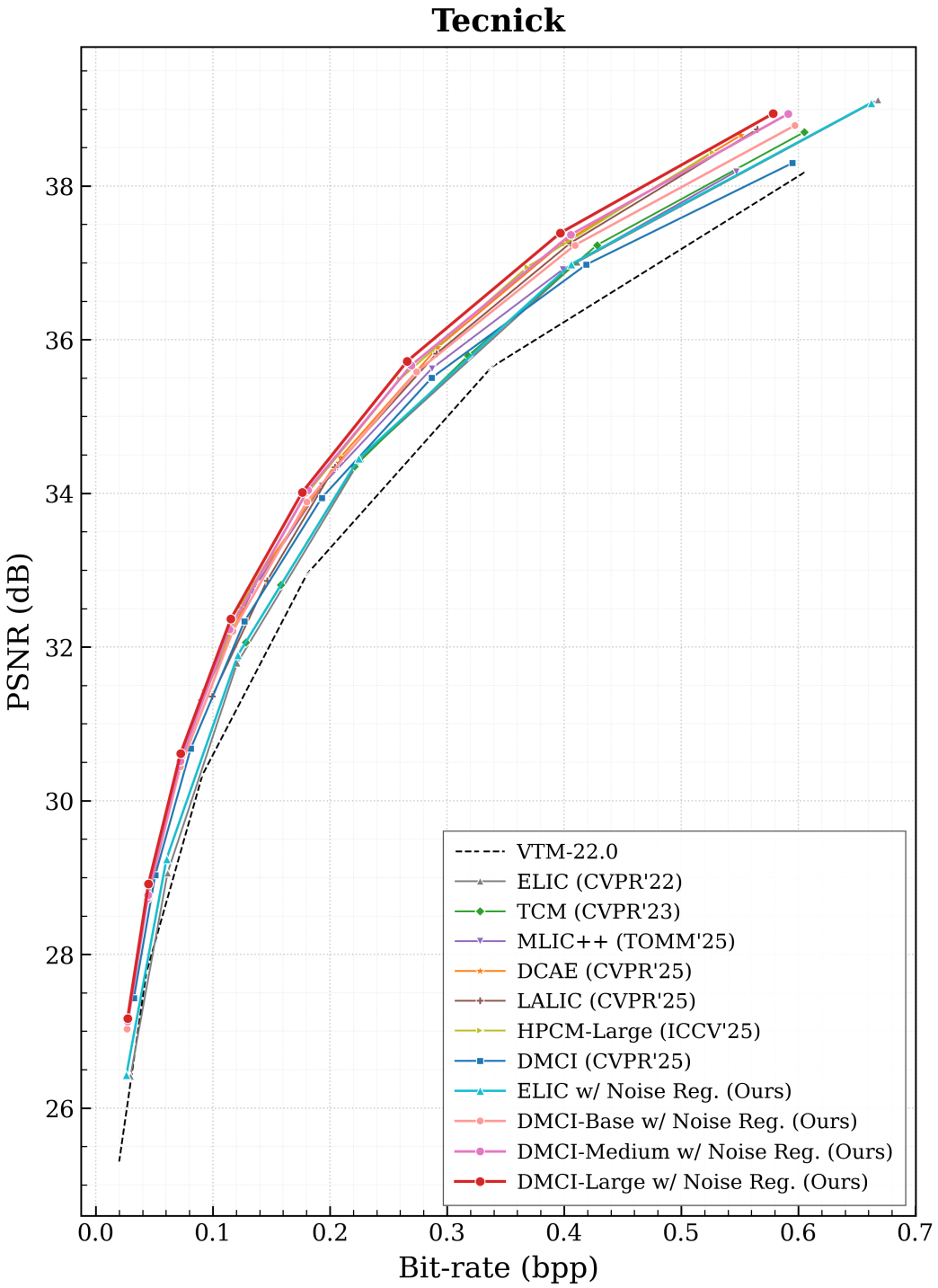


Figure 4: RD curves on **Tecnick**.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and Introduction (Section 1) state three concrete claims—(i) a pixel-domain noise-regularization recipe for LIC, (ii) a second-order RD expansion that yields a structured regularizer (Eq. 5), and (iii) state-of-the-art BD-Rate performance together with a direct Lipschitz verification on both the encoder and the end-to-end codec—which are exactly the contents of Section 3 (theory), Section 5 (BD-Rate Tables 1–3), and Section 5.3 (Lipschitz analysis), respectively. Numerical headlines in the abstract (-2.60% ELIC, -22.02% DMCI-Large vs. VTM-22.0, $\sim 12\%$ encoder and $\sim 8\%$ end-to-end codec Lipschitz reduction) match Tables 1–3 and Table 5.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section 6 contains a dedicated "Limitations" paragraph: the theoretical analysis is a second-order expansion in σ derived for MSE distortion, while perceptual objectives are evaluated as empirical extensions. It also notes that a single exponential annealing schedule is used and that its interaction with variable-rate conditioning remains to be studied.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: Section 3 states the assumptions (small-noise regime, C^4-smoothness of g_a, g_s, p_ϕ on the relevant supports, independence of $\boldsymbol{\eta}$ and the latent uniform proxy $\mathbf{u}$; formally stated in Appendix A.1) and presents the main second-order results, Eqs. (3), (4), and (5). The complete derivations—including a notation table (8), the distortion-side expansion (Appendix A.3), the rate-side expansion (Appendix A.4), and the combined local objective (Appendix A.5)—are provided in the Appendix and cross-referenced from the main text.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Section 5 ("Experimental Setup") specifies the architectures (ELIC, plus DMCI at three explicit channel widths $(\mathrm{ch}_g, \mathrm{ch}_y, \mathrm{ch}_z)$), the training data (MLIC-Train-100K), optimizer (Adam; DMCI learning rates 10^{-4} and 10^{-5}), per-stage batch / patch sizes, total iteration counts, stage-specific decay schedules, the variable-rate $\lambda \in [16, 4096]$ over 64 QPs, and the evaluation benchmarks (Kodak, Tecnick, CLIC). Algorithm 1 formalises the noise-regularized training loop and Appendix A.8 provides a minimal PyTorch sketch of the `lic_noise` interface; code and DMCI checkpoints are available at `https://github.com/Googolxx/NoiseRegu-LIC`.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed

instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: Code, DMCI training and evaluation scripts, model-only checkpoints, a checkpoint hash manifest, and reproduction instructions are available at `https://github.com/Googolxx/NoiseRegu-LIC`. The README documents environment setup, rANS compilation, two-stage training, evaluation, and BD-Rate calculation.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 5 (“Training”) reports the optimizer (Adam), the stage-specific learning rates (10^{-4} and 10^{-5} for DMCI), the per-stage batch / patch sizes (32×256 then 16×512),

the total $\sim$ 2M iterations for single-rate ELIC, the two-stage $1.4\text{M}+0.1\text{M}$ schedule for variable-rate DMCI, stage-specific decay milestones, the variable-rate λ grid ($[16, 4096]$ over 64 QPs), and the noise-schedule hyperparameters (σ_0, T, decay rate, per-batch resampling) that specify Algorithm 1.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: For the Lipschitz analysis (Section 5.3, the central empirical claim), Figure 1 shows the ± 1 standard-deviation bands across the 24 Kodak images for both the encoder Jacobian $\hat{L}_{g_a}$ and the end-to-end codec Jacobian $\hat{L}_f$, with per-image distributions at QP $=$ 27 rendered as violin plots over all 24 individual estimates. Table 5 reports the average reductions across the order-of-magnitude δ sweep together with the per-QP envelopes; the full per-QP, per-δ tables are provided in Appendix A.9. BD-Rate (Tables 1–3) is a deterministic curve aggregate over fixed public test sets, and we follow the LIC-community convention of reporting it without per-image error bars; eight DMCI-Base runs spanning $\sigma_0 \in [0.4, 0.8]$ also improve over the shared clean baseline on every dataset, with within-σ_0 standard deviation at most 0.04 percentage points.

Guidelines:

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

8. **Experiments compute resources**

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

Answer: [Yes]

Justification: Section 5 gives the iteration counts and per-stage batch / patch sizes that quantify training cost; all experiments were run on a single workstation with NVIDIA A100-80GB GPUs (one GPU per single-rate ELIC run, multi-GPU DDP for variable-rate

DMCI). The recipe itself adds no measurable training-time overhead beyond a single additive Gaussian draw in data preprocessing (abstract, Section 4) and zero inference-time overhead. Lipschitz estimation on Kodak completes in well under one GPU-hour per checkpoint at $n = 64$ random directions per (image, QP, δ).

Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The work concerns lossy image compression on long-established public benchmarks (Kodak, Tecnick, CLIC) and standard architectures from the LIC literature; it involves no human subjects, no personally identifiable data, and no data scraping. The submission preserves anonymity throughout and we have reviewed and adhered to the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Improved learned image compression has clear positive societal impact through reduced bandwidth and storage costs, lower energy consumption for media transmission, and improved accessibility on bandwidth-constrained networks. We do not foresee material direct negative impact: the contribution is a training recipe that strictly improves rate–distortion of an existing class of codecs and does not introduce any new generative or surveillance capability. As a foundational study of training dynamics, the work does not target a specific deployment.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out

that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The released artefacts are LIC encoder/decoder checkpoints whose sole capability is reconstructing an input image at a chosen bitrate; they have no generative, identification, or scraped-content components, and pose no foreseeable dual-use risk requiring additional safeguards.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All baseline architectures (ELIC [41], DMCI/DCVC-RT [42], HPCM [12]) and all evaluation datasets (Kodak [37], CLIC [38], Tecnick [39], MLIC-Train-100K [45]) are cited at first use in Section 5; we use them strictly for non-commercial academic research consistent with their public release terms, and the standardised codec VTM-22.0 [1] is used as a reference under its original VVC reference-software license.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: The new assets are the noise-regularized training recipe and the released DMCI checkpoints. The public package at `https://github.com/Googolxx/NoiseRegu-LIC` includes training/evaluation scripts, model-only weights, a SHA256 manifest, an MIT license, third-party notices, and a README documenting data layout, configuration, and reproduction commands.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve any crowdsourcing or human-subject studies; all evaluation is performed automatically on standard public benchmarks with PSNR-based BD-Rate.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve any human-subject research, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs were not used as a component of the core methodology, experimental design, or data analysis. Their use was limited to grammatical polishing and language editing, which does not impact the scientific contribution or originality of the research.

    Guidelines:

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