LOONA: LINEARIZED OBJECTIVES OVER NOISE ATOMS FOR CONTROLLABLE ZERO-SHOT DIFFUSION IMAGE COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Zero-shot diffusion-based codecs compress images by selecting elements from a shared randomness by the encoder and the decoder. Fast selectors such as Turbo-DDCM score atoms by their correlation with a latent residual, which restricts direct control to a latent quadratic objective, while DDCM handles other objectives by evaluating a small set of candidate noises. We introduce LOONA, which selects noise under a differentiable objective by scoring all atoms with a linear surrogate of the next-step objective. Stein's identity gives the least-squares coefficient of this surrogate, which we approximate with one vector–Jacobian product per guided step, so the number of network evaluations is independent of the codebook size and of the number of selected atoms. LOONA keeps Turbo-DDCM's bitstream format and frozen decoder, transmits no side information, and requires no training. It reduces BD-rate against Turbo-DDCM by 18.7% (PSNR) and 13.7% (LPIPS) on Kodak and by 17.9% and 13.5% on DIV2K, and the same encoder provides full-resolution region-of-interest control, semantic preservation for downstream segmentation, and a tunable perception–distortion tradeoff at a fixed rate.

1 INTRODUCTION

Image compression is usually evaluated with a fixed reconstruction objective, but the information worth preserving depends on the application. A viewer may prioritize visual quality, an analyst may require precise detail in a selected region, and a downstream model may depend on semantic features. These priorities can conflict with pixel fidelity, particularly at low bitrates (Blau & Michaeli, 2018; 2019). This motivates codecs whose reconstruction objective can be chosen for each image at encoding time.

Pretrained diffusion models provide an image prior for both trained compression pipelines (Careil et al., 2024; Körber et al., 2024; Xue et al., 2025; Guo et al., 2025) and zero-shot codecs (Theis et al., 2022; Vonderfecht & Liu, 2025; Ohayon et al., 2025; Vaisman et al., 2026; Elata et al., 2025). Existing approaches offer some forms of flexibility as controllable learned codecs, which expose distortion–realism tradeoffs learned during training (Agustsson et al., 2023; Iwai et al., 2024),. For fast zero-shot diffusion compression, the challenge is to translate a user-chosen differentiable objective into discrete noise choices that can be encoded efficiently and reproduced by the decoder.

DDCM (Ohayon et al., 2025) already allows custom objectives through candidate-wise codebook evaluation. Turbo-DDCM (Vaisman et al., 2026) accelerates diffusion-codebook compression using few-step sampling and signed combinations of noise atoms. Its efficient selector, however, ranks atoms by correlation with a latent residual, optionally reweighted by a priority mask in latent space. This criterion does not directly optimize nonlinear losses defined on decoded pixels, semantic features, or no-reference quality. Evaluating such losses separately for every candidate or signed combination would repeatedly invoke the denoiser and objective networks, undermining the efficiency of the selector. The missing component is an inexpensive objective-dependent score that retains the existing coding and decoding operations.

We introduce *LOONA*, *Linearized Objectives over Noise Atoms*, which addresses this problem by constructing one scoring direction per guided diffusion step. We formulate the best affine least-

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

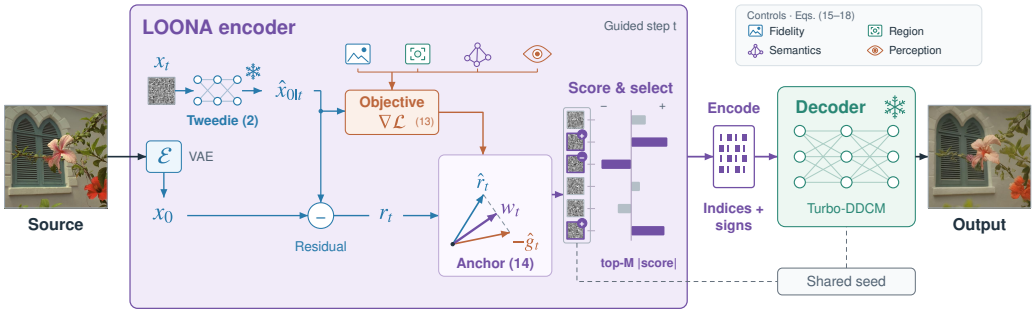


Figure 1: ***LOONA* overview.** A fidelity-residual anchors the objective gradient to guide signed noise atom selection under fidelity, regional, semantic, and perceptual controls. Selected indices and signs are shared to a frozen decoder that reproduces the noise codebook using a shared seed, keeping all objective-dependent control at the encoder.

squares approximation of the next-step reconstruction loss under Gaussian noise. Stein's identity identifies its optimal linear coefficient as a noise-averaged loss gradient. Following the use of denoised estimates in diffusion guidance (Chung et al., 2023; Yu et al., 2023a), we approximate this direction with a current-state loss gradient through the Tweedie reconstruction estimate. We combine its normalized descent direction with the normalized latent residual to anchor selection to the source image. The resulting inner-product scores feed directly into Turbo-DDCM's signed top-M selection and bit packing, allowing pixel-space, feature-space, and quality objectives to influence the transmitted noise choices without retraining.

On each guided step, *LOONA* evaluates the objective and performs one reverse-mode pass, reusing the current denoiser forward evaluation. The number of neural evaluations therefore does not grow with the codebook size or the number of selected atoms, although codebook scoring still scales with its size. The selected indices and signs retain their existing meaning: the bitstream format, decoder, and shared-randomness protocol are unchanged from Turbo-DDCM. The objective, its parameters, and its auxiliary networks remain entirely at the encoder and require no additional side information. Figure 1 summarizes the method and representative results. Our contributions are as follows:

- **Linear scoring of noise atoms.** We formulate noise selection through a Gaussian affine approximation of the reconstruction objective, characterize its ideal scoring coefficient using Stein's identity, and construct a practical gradient-based proxy.
- **Efficient control with an unchanged decoder.** We integrate a normalized residual–gradient anchor into Turbo-DDCM's existing selector, supporting differentiable objectives with a codebook-independent number of neural evaluations and no changes to the bitstream format or decoder.
- **Reconstruction fidelity and application-specific control.** We demonstrate improved reconstruction fidelity over the evaluated zero-shot baselines on Kodak and DIV2K, full-resolution pixel-space RoI control, semantic preservation evaluated through downstream segmentation, and a tunable distortion–quality tradeoff assessed with held-out no-reference metrics.

2 Related Work

2.1 Diffusion Models

Diffusion models (Sohl-Dickstein et al., 2015; Song et al., 2021; Ho et al., 2020) model and sample from complex distributions such as the natural image distribution (Rombach et al., 2022; Dhariwal & Nichol, 2021). A diffusion model first defines a progressive noising process on samples drawn from the target distribution $x_0 \sim p_0$,

$$x_t = \sqrt{\alpha_t}\, x_{t-1} + \sqrt{1-\alpha_t}\, \epsilon, \tag{1}$$

where $\epsilon \sim \mathcal{N}(0, I)$ and α_t dictates the noising schedule. The reverse of this process can be learned with a noise estimation network $\epsilon_\theta(x_t, t)$, from which the conditional mean of the clean sample at

step t (the Tweedie estimate (Efron, 2011)) follows as

$$\mathbb{E}[x_0 \mid x_t] \;\approx\; \frac{x_t - \sqrt{1-\overline{\alpha}_t}\,\epsilon_\theta(x_t, t)}{\sqrt{\overline{\alpha}_t}} \;=\; \hat{x}_{0|t}(x_t), \tag{2}$$

where $\overline{\alpha}_t = \prod_{s \le t} \alpha_s$. In DDPM (Ho et al., 2020), sampling starts from Gaussian noise $x_T \sim \mathcal{N}(0, I)$ and applies the reverse step

$$x_{t-1} \;=\; \mu_\theta(x_t) + \sigma_t\, z_t, \qquad z_t \sim \mathcal{N}(0, I), \tag{3}$$

where $\mu_\theta(x_t)$ is the sampler's deterministic reverse-step mean, a linear combination of x_t and $\epsilon_\theta(x_t, t)$, and σ_t is the reverse-step standard deviation the sampler specifies. For ancestral DDPM with the fixed-small variance, $\sigma_t^2 = \tilde{\beta}_t = \frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}(1-\alpha_t)$; for a DDIM or subsampled schedule we use that schedule's coefficient. Throughout this paper we work with a latent diffusion model (Rombach et al., 2022), so x_t lives in the VAE latent space and $\mathcal{D}$ denotes the decoder that maps a latent back to pixels.

2.2 DDCM and Turbo-DDCM

DDCM. Denoising Diffusion Codebook Models (Ohayon et al., 2025) replace the random Gaussian noise of the DDPM reverse step with a selection from a fixed, reproducible Gaussian codebook $\mathcal{C}_t = [z^{(1)}, \ldots, z^{(K)}]$ of K atoms,

$$x_{t-1} \;=\; \mu_\theta(x_t) + \sigma_t\, z, \qquad z \in \mathcal{C}_t. \tag{4}$$

To compress a target image x_0, the encoder selects at each step the atom whose correlation with the residual $r = x_0 - \hat{x}_{0|t}(x_t)$ is largest,

$$k_t \;=\; \arg\max_k \langle z^{(k)},\, r \rangle, \tag{5}$$

and transmits the sequence of indices. The decoder, sharing the seeded codebooks, re-selects the same atoms and runs the reverse process. The rate is $\mathrm{BPP} = (T-1)\lceil \log_2 K \rceil / N_{\mathrm{px}}$, where N_{px} is the number of pixels of the target image and T is the number of diffusion steps used for encoding and decoding.

Turbo-DDCM. Turbo-DDCM (Vaisman et al., 2026) accelerates this by selecting M atoms per step in closed form. It computes all K correlations $\langle z^{(i)}, r \rangle$ once, keeps the M largest in magnitude, and assigns signs $s_i = \mathrm{sign}(\langle z^{(i)}, r \rangle)$, so the injected noise is $z = \sum_{i \in S} s_i z^{(i)}$, normalized to unit standard deviation. Combined with an efficient lexicographic bit protocol, this reduces the required number of diffusion steps by an order of magnitude. Letting N be the number of final reverse steps that transmit no atom indices, the bitrate is controlled by the hyperparameters T, K, M, N as

$$\mathrm{BPP}_{\text{Turbo-DDCM}} = \frac{(T-N-1)\left(\lceil \log_2 \binom{K}{M} \rceil + M\right)}{N_{\mathrm{px}}}. \tag{6}$$

Turbo-DDCM also introduces a priority-aware variant that reweights the residual by a user mask, and a distortion-controlled variant that targets a PSNR rather than a BPP. In both DDCM and Turbo-DDCM the selection primitive is the same: atoms are scored by their correlation with the residual r measured in the latent space. Our method keeps the atom selection and transmission protocol of Turbo-DDCM and changes only that primitive, which is what allows the priority mask to be specified at full pixel resolution and the objective to be something other than latent squared error.

2.3 Guidance and Control in Diffusion Sampling

Steering a pretrained diffusion model toward a differentiable criterion at inference time is well studied outside compression. Classifier and classifier-free guidance (Dhariwal & Nichol, 2021; Ho & Salimans, 2021) perturb the predicted score, diffusion posterior sampling (Chung et al., 2023) and ΠGDM (Song et al., 2023) backpropagate a measurement likelihood through the Tweedie estimate, and training-free schemes such as universal guidance (Bansal et al., 2023) and FreeDoM (Yu et al., 2023a) extend the same idea to arbitrary off-the-shelf energy functions. Guo et al. (2024) analyze

gradient guidance from an optimization viewpoint. All of these methods add the gradient to a *continuous* update and therefore change the sampling trajectory in a way that a decoder cannot reproduce from a short index. Our setting is different: the gradient is used only to rank a *discrete* set of candidate noises, the update itself stays on the codebook, and the transmitted indices remain sufficient for exact reconstruction at the decoder. This is what keeps control free of side information.

3 METHOD

3.1 THE OPTIMAL LINEAR SCORE FOR PER-STEP OPTIMIZATION

Given a differentiable loss function $L(\cdot)$, our goal at step t is to select the noise z that optimizes

$$L(\mathbb{E}[x_0 \mid x_{t-1}]) \approx L\Big(\hat{x}_{0|t-1}\big(\mu_\theta(x_t) + \sigma_t z\big)\Big). \tag{7}$$

Since $x_{t-1} = \mu_\theta(x_t) + \sigma_t z$, Equation 7 chooses the noise whose next-step Tweedie estimate best matches the target latent under L. Per step, the selector must rank candidate noises by their effect on the loss,

$$\mathcal{L}(z) \;\triangleq\; L\Big(\hat{x}_{0|t-1}\big(\mu_\theta(x_t) + \sigma_t z\big)\Big), \tag{8}$$

using a linear score $\langle z, u\rangle$. In order to obtain the optimal linear scoring method, we need to fit our linear model to $\mathcal{L}(z)$ over all z that follows the Gaussian distribution $\mathcal{N}(0, I_d)$. The following proposition identifies the optimal coefficient for our desired linear function.

Proposition 1 (Stein-optimal linear score)**.** *Let $z \sim \mathcal{N}(0, I_d)$ and let $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ be differentiable with $\mathbb{E}[\mathcal{L}(z)^2] < \infty$ and $\mathbb{E}\|\nabla\mathcal{L}(z)\| < \infty$. The affine least-squares problem*

$$(c^\star, u^\star) \;=\; \underset{c\in\mathbb{R},\, u\in\mathbb{R}^d}{\arg\min}\; \mathbb{E}_z\Big[\big(\mathcal{L}(z) - c - \langle z, u\rangle\big)^2\Big] \tag{9}$$

has the unique solution

$$c^\star = \mathbb{E}[\mathcal{L}(z)], \qquad u^\star = \mathbb{E}\big[z\,\mathcal{L}(z)\big] = \mathbb{E}\big[\nabla\mathcal{L}(z)\big]. \tag{10}$$

Proof. Using $\mathbb{E}[z] = 0$ and $\mathbb{E}[zz^\top] = I$, the objective expands and separates,

$$\mathbb{E}\big[(\mathcal{L} - c - \langle z, u\rangle)^2\big] = \mathbb{E}[\mathcal{L}^2] + \big(c - \mathbb{E}[\mathcal{L}]\big)^2 - \mathbb{E}[\mathcal{L}]^2 + \|u\|^2 - 2\langle u, \mathbb{E}[z\mathcal{L}]\rangle, \tag{11}$$

which is strictly convex in (c, u) with unique stationary point $c^\star = \mathbb{E}[\mathcal{L}]$, $u^\star = \mathbb{E}[z\mathcal{L}]$. The second equality in Equation 10 is from Stein's identity. □

The optimal linear score is therefore the noise-averaged gradient $\mathbb{E}\big[\nabla\mathcal{L}(z)\big]$. On the other hand, the noise-averaged Tweedie mean at step $t-1$, which is the component inside the loss function $L(\cdot)$ of $\mathcal{L}(z)$, is equal to the Tweedie mean at step t by utilizing the tower property as

$$\mathbb{E}_z\Big[\hat{x}_{0|t-1}\big(\mu_\theta(x_t) + \sigma_t z\big)\Big] \;=\; \mathbb{E}_{x_{t-1}}\Big[\mathbb{E}\big[x_0 \mid x_{t-1}\big]\Big] \;=\; \mathbb{E}\big[x_0 \mid x_t\big] \;=\; \hat{x}_{0|t}(x_t). \tag{12}$$

From this, we approximate the optimal $u^\star$ using the gradient at the current state through the current-step map x_t as follows

$$\hat{u}^\star \;\approx\; \hat{g}_t \qquad g_t = \nabla_{x_t}\big(L \circ \hat{x}_{0|t}\big)\big|_{x_t}, \tag{13}$$

where the hat term is used as the normalized vector ($\hat{v} \triangleq v/\|v\|$) here and throughout the paper. We provide more details about this approximation in the Appendix section C. This resulting linear score $\langle z, g_t\rangle + c$ gives the impression of the first-order Taylor expansion of $\mathcal{L}(z)$ on $z = 0$. In the Appendix section C, we show how g_t and the Taylor expansion term is different and how that expansion can lower the reconstruction performance.

Fidelity-anchored selection direction. Because g_t passes through learned maps, we do not rank atoms against g_t alone. For robustness, we utilize $r_t = x_0 - \hat{x}_{0|t}(x_t)$, and the selection target is the equal-weight mixture of the two normalized directions,

$$z^\star \;=\; \arg\max_z \langle z,\; w_t\rangle, \qquad w_t \;=\; \tfrac{1}{2}\,(\hat{r}_t \;-\; \hat{g}_t)\,, \tag{14}$$

i.e., the bisector of the residual direction and the descent direction. The anchor is principled rather than heuristic: r_t is computed exactly from known tensors and is the exact direction toward the latent objective (x_0), so its linear score is trustworthy at any step size, whereas g_t is a local gradient and carries model error that the top-M argmax can amplify. Normalizing before mixing makes the balance insensitive to the gradient magnitudes, which drift across steps and objectives. Therefore, taking the bisector keeps w_t equally aligned with both directions, so the target w_t can never anti-align with the fidelity residual, whatever the objective's gradient does.

Our method only updates the selection target of DDCM and Turbo-DDCM (Equation 5) into the above equation. Furthermore, ours can use any differentiable custom loss function as we will describe in the following subsections and the experiment sections. In our paper, we mainly use the selection method of Turbo-DDCM and its lexicographic bit protocol (Equation 19). Furthermore, since our method is only changing the selection target, the decoding process is exactly the same as DDCM or Turbo-DDCM.

Admittedly, it is possible to evaluate the left-hand side of Eq. 7 exhaustively and keep the minimizing noises, but a Turbo-DDCM step chooses among $\binom{K}{M}2^M$ signed index sets, each requiring a denoiser pass, a decode, and a loss evaluation. DDCM (Ohayon et al., 2025) does support custom objectives (e.g., no-reference IQA), but it evaluates the true loss for a limited set of candidate noises and keeps the minimizer (two per step for no-reference IQA, which in return allows non-differentiable losses). Its gradient-based rule, selecting the noise closest to $\nabla_{x_t} \log p(y \mid x_t)$, is instantiated only where this score has a closed form. For a loss on the reconstruction, a DPS plug-in reduces it to our $-g_t$. LOONA differs in its one-step lookahead target (Equation 8) and in anchoring to the residual through $w_t = \frac{1}{2}(\hat{r}_t - \hat{g}_t)$ (Equation 14).

We detail our encoding algorithm in Algorithm 1. $\mathrm{TurboEncode}(C_t, w_t, M)$ denotes Turbo-DDCM's selection-and-packing primitive: it scores all K atoms of C_t against the target by $c_k = \langle z^{(k)}, w_t\rangle$, keeps the M largest in magnitude with signs $s_k = \mathrm{sign}(c_k)$, forms the standardized combination $z_t = \sum_{k\in S_t} s_k z^{(k)} / \mathrm{std}(\cdot)$, and appends the lexicographic rank of S_t plus M sign bits to the bitstream (Equation 19). $\mathrm{TurboStep}(x_t, \hat{\epsilon}, z)$ is the sampler's reverse update $x_{t-1} = \mu_\theta(x_t) + \sigma_t z$, with μ_θ formed from $(x_t, \hat{\epsilon})$ and σ_t the schedule's step deviation (z=0 on the selection-free tail). Both are unmodified from Vaisman et al. (2026); we change only the target w_t that $\mathrm{TurboEncode}$ scores against. We further note that early in the reverse process the Tweedie estimate $\hat{x}_{0|t}$ is a heavily smoothed posterior mean: at high noise levels it resolves only coarse structure, so gradients of perceptual, semantic, and no-reference terms — functions of fine image content that does not yet exist — are dominated by model error rather than signal. Ranking atoms by such a gradient spends selection capacity on noise, and we observe it to degrade both the optimized objective and fidelity. The residual r_t, by contrast, is informative at every noise level. We therefore activate the gradient term only once the estimate is committed enough to differentiate, mixing g_t into w_t for $t \le T_g$ and selecting against r_t alone for $t > T_g$. Note that T_g does not affect the rate. In the next subsections we introduce the loss functions we tested, both for better reconstruction and for the three control settings.

3.2 Pixel-Level Accuracy Control

Training-free diffusion-based compression methods usually operate in latent space, because the best performing diffusion models for image generation are latent diffusion models. Compression is therefore optimized on the latent, not on pixels. By using an objective that acts on both spaces we can select noises that reconstruct to a higher-quality pixel image. Let $\mathcal{D}$ be the decoder that maps latents to pixels, y_0 the ground-truth target image in pixel space, and f_i quality metrics acting on pixels (MSE, MS-SSIM (Wang et al., 2003), LPIPS (Zhang et al., 2018)):

$$L_{\text{pixel}}(x) = w_0\|x_0 - x\|^2 + \sum_i w_i f_i(\mathcal{D}(x), y_0). \tag{15}$$

The first term is the latent objective that prior work optimizes; the sum is what our formulation adds.

Algorithm 1 Encoding method of LOONA

Require: image y_0; objective L; frozen ϵ_θ, VAE; steps T; atoms per step M; stepwise codebooks $\{C_t\}_{t=1}^{T}$ shared with the decoder; gradient start T_g; selection-free tail T_{free}

$x_0 \leftarrow \mathcal{E}(y_0)$; initialize x_T
for $t = T, \dots, 1$ **do**
  $\hat{\epsilon} \leftarrow \epsilon_\theta(x_t, t)$ ▷ forward pass; recorded for autograd on gradient steps
  if $t \le T_{\text{free}}$ **then** ▷ final steps: no selection, no bits (as in Turbo-DDCM)
    $x_{t-1} \leftarrow \mathrm{TurboStep}\big(x_t, \hat{\epsilon}, z{=}0\big)$; **continue**
  end if
  $\hat{x}_{0|t} \leftarrow \mathrm{Tweedie}\big(x_t, \hat{\epsilon}\big)$
  $r_t \leftarrow x_0 - \hat{x}_{0|t}$
  if $t \le T_g$ **then** ▷ gradient steps
    $g_t \leftarrow \nabla_{x_t}\big(L \circ \hat{x}_{0|t}\big)\big|_{x_t}$ ▷ one reverse-mode pass (Eq. 13)
    $w_t \leftarrow \frac{1}{2}\big(r_t/\|r_t\| - g_t/\|g_t\|\big)$ ▷ selection target (Eq. 14)
  else
    $w_t \leftarrow r_t$ ▷ stock selection target
  end if
  $(S_t, s_t),\ z_t \leftarrow \mathrm{TurboEncode}(C_t, w_t, M)$ ▷ signed top-M vs. w_t; append bits
  $x_{t-1} \leftarrow \mathrm{TurboStep}(x_t, \hat{\epsilon}, z_t)$
end for
return $(S_t, s_t)_{t=T \dots T_{\text{free}}+1}$ ▷ the bitstream decoded by the unmodified Turbo-DDCM decoder

3.3 Region-of-Interest Control

As demonstrated by Turbo-DDCM, region-targeted compression is possible by applying a weight mask to the optimization target. Their mask, however, operates at the latent resolution, which compresses a 512×512 image to a 64×64 grid, so the mask loses fine-grained structure: a mask on a face, a license plate, or a line of text is quantized to 8×8-pixel blocks before it takes effect. Our method applies the map in pixel space as intended, which enables fine-grained region control. Write $P \in [0,1]^{H \times W}$ for the pixel-space priority map, $\tilde{P}$ for the same map pooled to latent resolution, and f_i^P for a pixel-space metric whose quality map is weighted by P. Then

$$L_{\text{mask}}(x) = w_0 \|\tilde{P} \odot (x_0 - x)\|^2 + \sum_i w_i f_i^P(\mathcal{D}(x), y_0). \tag{16}$$

The map weights the residual *after* encoding. Masking the image before encoding, $\mathcal{E}(y_0 \odot P)$, is a different and wrong operation: the VAE encoder is nonlinear and spatially coupled, so $\mathcal{E}(y_0 \odot P) \neq \mathcal{E}(y_0) \odot \tilde{P}$, and zeroing the background changes the latent target through boundary and global-context effects. We write the map as P rather than M because M already denotes the number of atoms per step. Its sharpness is set by a single knob κ, the ratio of foreground to background weight.

3.4 Semantic Preservation Control

In some compression use cases, retaining semantic information matters more than optimizing the visual or aesthetic quality of the image. We therefore test semantic preservation, where the goal is to compress the image so that a downstream semantic task, such as detection or panoptic segmentation, still performs well on the decoded image. To do so we add a loss that enforces retention of the semantic content. Specifically we use the features produced by DINOv2 (Caron et al., 2021; Oquab et al., 2024), which are known to represent image semantics well. Let h denote the normalized DINOv2 feature map:

$$L_{\text{semantic}}(x) = L_{\text{pixel}}(x) + w_{\text{sem}} \|h(y_0) - h(\mathcal{D}(x))\|^2. \tag{17}$$

Note that no task head enters the objective, so the same encoder is not tuned to any one downstream model. We verify this by measuring transfer to task heads that were never part of the loss.

3.5 Traversing the Perception–Distortion Tradeoff

There is an inherent tradeoff between perception and distortion in compression (Blau & Michaeli, 2018; 2019). Minimizing reconstruction error (distortion) often produces over-smoothed results and

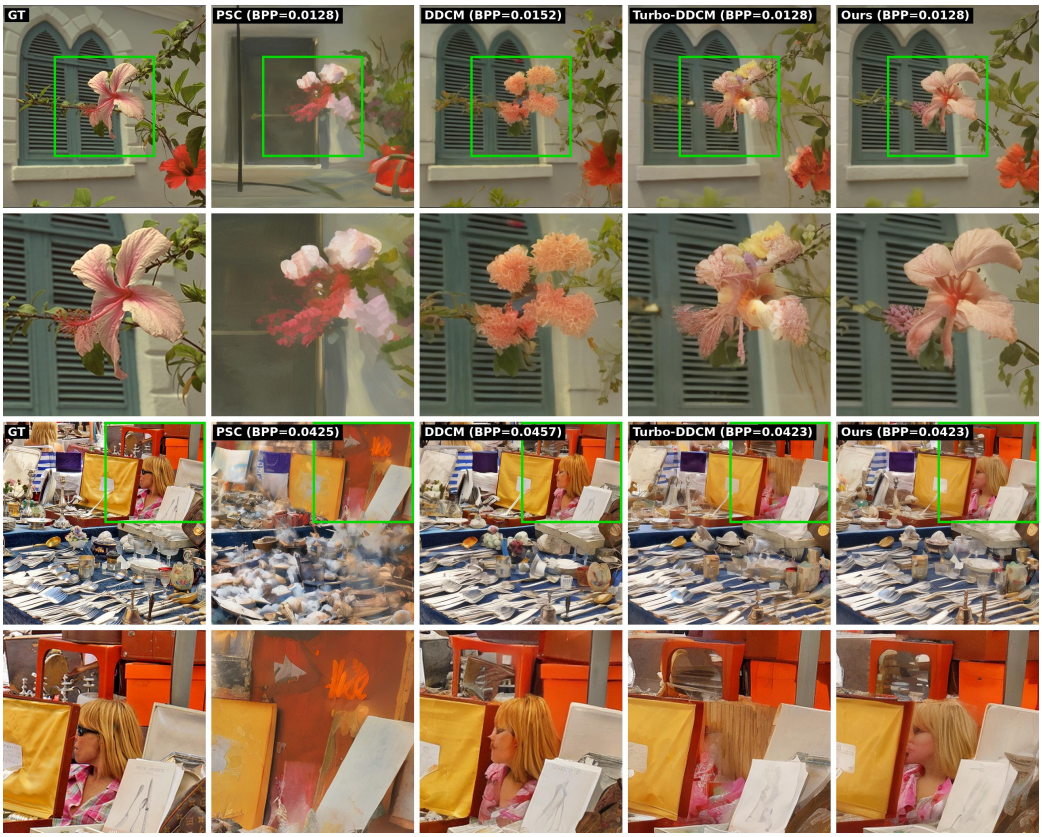


Figure 2: Qualitative comparison of our method with the pixel-level loss $L_{\text{pixel}}(x)$ (see Section 3.2) and other zero-shot image compression methods.

therefore a less realistic image (perception); conversely, optimizing for realism sacrifices reconstruction accuracy. With the following loss we obtain a tunable position on that tradeoff. Let q denote a no-reference image quality score in the convention where *higher is better*, as MUSIQ (Ke et al., 2021), Re-IQA (Saha et al., 2023) and ARNIQA (Agnolucci et al., 2024) all are, and let $w_{\text{p}} \geq 0$:

$$L_{\text{perception}}(x) = L_{\text{pixel}}(x) - w_{\text{p}}\, q(\mathcal{D}(x)). \tag{18}$$

The minus sign matters. The selection rule minimizes L, so writing $+\, w_{\text{p}} q$ would steer the encoder toward the worst scoring reconstruction rather than the best. Sweeping w_{p} at fixed rate yields a sequence of distinct operating points, where a codec trained at one setting yields a single point. Because atom selection is discrete, the reconstruction is not guaranteed to vary continuously with w_{p}.

4 Experiments

Experimental details. We use Kodak24 (Franzen, 1999), DIV2K (Agustsson & Timofte, 2017), COCO2017 Lin et al. (2014), ADE20K Zhou et al. (2017) dataset to evaluate the performances of our method using different loss functions. We compare with zero-shot image compression methods (Ohayon et al., 2025; Vaisman et al., 2026; Elata et al., 2025; Vonderfecht & Liu, 2025), learned image compression methods (Zhang et al., 2025; Li et al., 2025), and also image compression for machine vision tasks (ICM). Following Turbo-DDCM, we use the SD 2.1 Base model and use different settings to adjust to different rates (see Appendix B.1). We measure quality metrics as PSNR, MS-SSIM (Wang et al., 2003), LPIPS (Zhang et al., 2018) and FID (Heusel et al., 2017) to assess the quality of the reconstructed images and also follow the ICM's protocol (Feng et al., 2025) to assess the panoptic quality. For fairness, we used LPIPS-VGG for the optimization and LPIPS-Alex for evaluations. We used FC-CLIP (Yu et al., 2023b) for the evaluation of Panoptic Quality (PQ) (we follow previous ICM's works (Feng et al., 2025) to compute the PQ score). Throughout our ex-

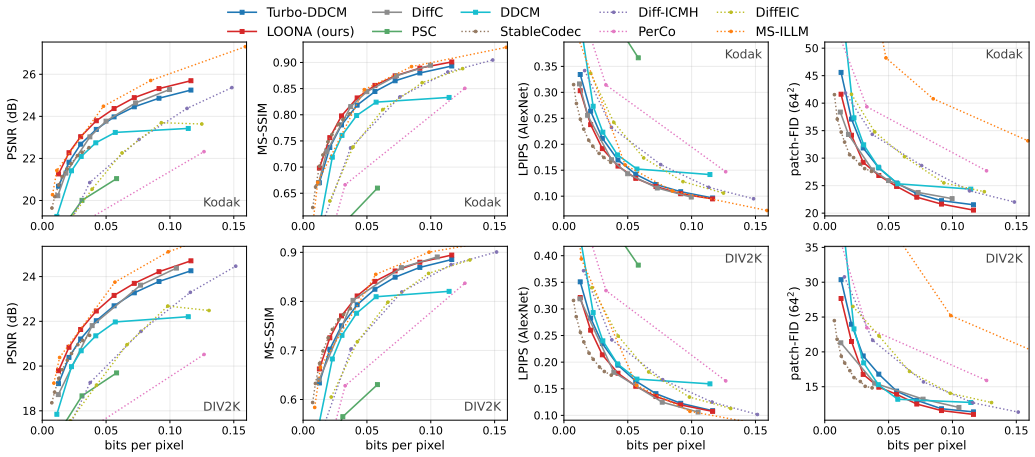


Figure 3: Rate-distortion curves on Kodak24 and DIV2K datasets. Solid lines represent zero-shot diffusion-based image compression methods. Dotted lines represent trained compression methods.

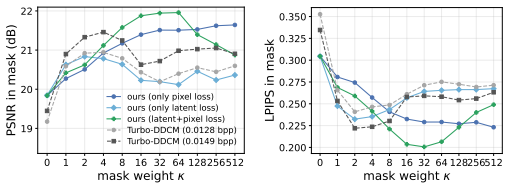
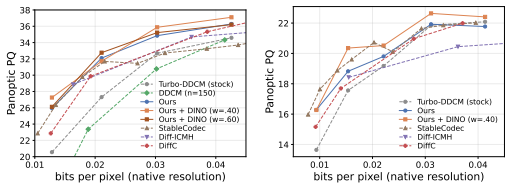


Figure 4: PSNR and LPIPS limits of RoI-based compression using LOONA (ours) and Turbo-DDCM by sweeping over the RoI weights. Our method gives frontier limit of RoI quality.



(a) COCO2017 (b) ADE20K

Figure 5: Rate-semantics curve based on the panoptic segmentation task.

periments on encoding and decoding and measuring round-trip processing time, we used NVIDIA A100 GPUs. We provide the roundtrip (encoding+decoding) speed comparison in Fig. 8.

Pixel-level accuracy control. Fig. 2 and Fig. 3 show qualitative and quantitative results of our method described in Section 3.2. We observe that our method gives better overall reconstruction accuracy compared to Turbo-DDCM and even competitive to StableCodec on PSNR and MS-SSIM, which is a learned image compression method. Compared to DiffC, our method is slightly worse on FID on low BPP, but we produce similar or better performance in terms of full-reference image quality metrics as PSNR, MS-SSIM and LPIPS. Compared to other zero-shot diffusion-based image compression methods as DDCM and Turbo-DDCM, our method consistently produces better quality. Fig. 4 shows quantitative result of our method described in Section 3.3. We measure the reconstruction performance on the RoI. We compare with Turbo-DDCM method and our method with the RoI map on the latent and the pixel space on and off. We observe that the RoI enforced on the pixel space is crucial to improving the peak reconstruction performance on the RoI, and the combination of the RoI mask on the latent and the pixel space gives the best result.

Semantic preservation control. Fig. 5 and Fig. 6 show qualitative and quantitative results of our method described in the Section 3.4. We observe that the DINO feature loss increases the panoptic quality (prediction) performance of the decoded image on overall BPPs. In Fig. 6, we provide visual results, where the panoptic predictor accurately predicts the segmentation of the instances and their classifications at matched BPP. On the top row, ours+DINO is able to segment the cake, person (baby) and the chair successfully (Note that Turbo-DDCM is not able to successfully segment the right hand part of the person). On the second row, ours+DINO is the only method that was able to segment the remote in the middle that is covered by a cat. On the third row, ours+DINO is the only method that helped the model successfully identify the tie on the image. Note that the model to predict the panoptic segmentation is FC-CLIP (Yu et al., 2023b), which doesn't have any DINO-based component within the pipeline.

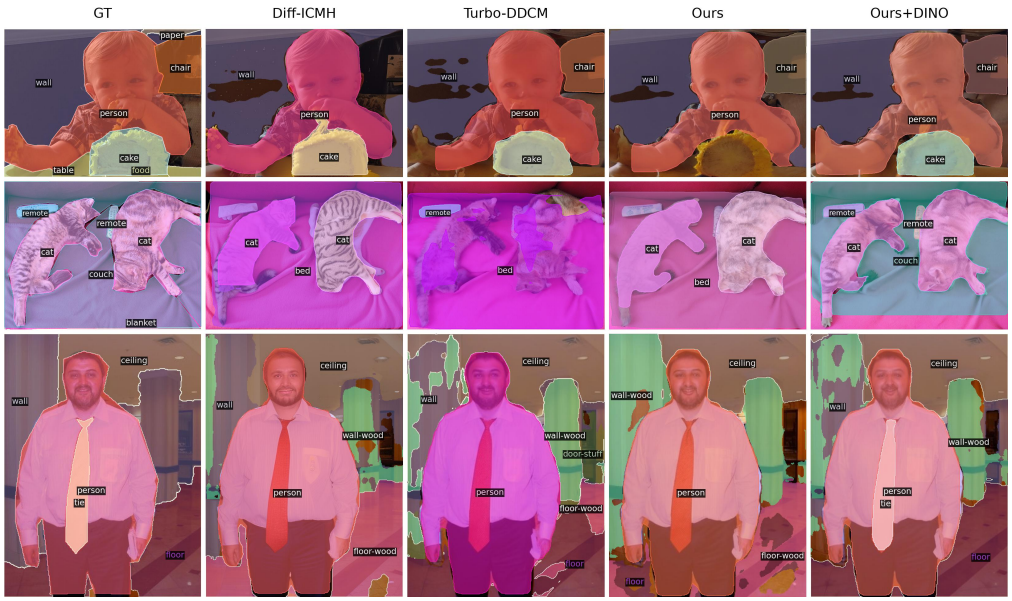


Figure 6: Qualitative results on our method with semantic preservation loss (see Section 3.4).

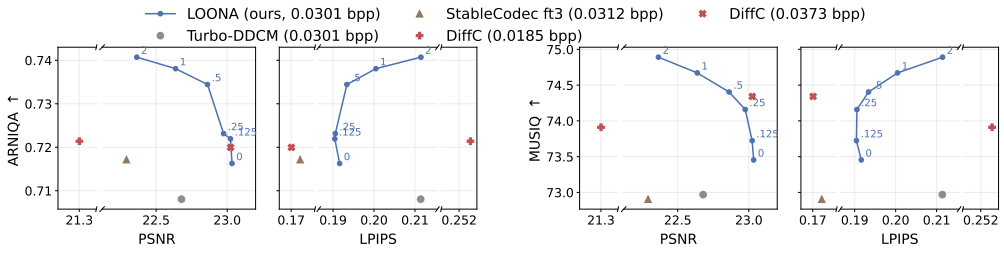


Figure 7: **Perception–distortion traversal at a fixed rate.** Kodak24 at BPP $= 0.03$. Each curve is produced by sweeping w_{p} in Equation 18 with q set to Re-IQA (Saha et al., 2023); the horizontal and vertical axes report a distortion metric and a held-out no-reference metric that was not optimized.

Perception-distortion traversal. Fig. 7 shows the results of our experiment using the loss described in the Section 3.5. By adjusting the weight of the quality metric loss term, we can traverse the perception-distortion tradeoff area. We chose Re-IQA as the optimizing quality metric function and evaluate on other NR-IQA metric as MUSIQ and ARNIQA, and observe that the traversal is well transferred to other NR-IQA metrics. In the Appendix section, we show further experimental results and ablative studies on various loss function optimization including watermarking.

5 Conclusion

Zero-shot diffusion codecs need no training, but their fast selectors score codebook noise by a fixed latent correlation. Linearizing the next-step objective in the injected noise turns per-step selection under a differentiable objective into one inner product per atom. The generalization costs one objective evaluation and one reverse-mode pass per guided step, a number of network evaluations independent of the codebook size, and leaves the bitstream and the decoder untouched. The same encoder then compresses for pixel fidelity, for a region a user drew at full resolution, for a downstream task, or for a chosen point on the perception–distortion curve.

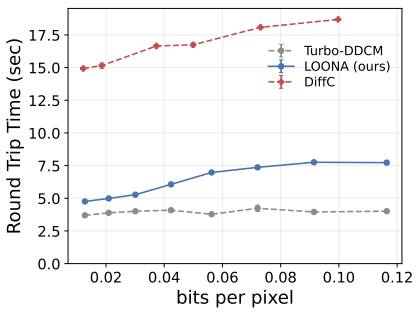


Figure 8: Speed Comparison

AI USE STATEMENT

We used large language models to (i) implement and debug experiment code, including baselines and ablations; (ii) give feedback on experimental design and the interpretation of results; (iii) check and refine mathematical derivations and claims (Section 3.1, Appendix C); and (iv) draft and edit text, which we revised. The authors verified all code, results, derivations and references and take full responsibility for the content of this paper.

REFERENCES

Lorenzo Agnolucci, Leonardo Galteri, Marco Bertini, and Alberto Del Bimbo. ARNIQA: Learning distortion manifold for image quality assessment. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 189–198. IEEE, 2024.

Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 126–135, 2017.

Eirikur Agustsson, David Minnen, George Toderici, and Fabian Mentzer. Multi-realism image compression with a conditional generator. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22324–22333, 2023.

Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, 2018.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 843–852, 2023.

Fabrice Bellard. BPG image format. https://bellard.org/bpg/, 2014.

Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6228–6237. IEEE, 2018.

Yochai Blau and Tomer Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff. In *International Conference on Machine Learning*, pp. 675–685. PMLR, 2019.

Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J Sullivan, and Jens-Rainer Ohm. Overview of the versatile video coding (VVC) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021.

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

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9650–9660. IEEE, 2021.

Chaofeng Chen, Jiadi Mo, Jingwen Hou, Haoning Wu, Liang Liao, Wenxiu Sun, Qiong Yan, and Weisi Lin. TOPIQ: A top-down approach from semantics to distortions for image quality assessment. *IEEE Transactions on Image Processing*, 33:2404–2418, 2024.

Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized Gaussian mixture likelihoods and attention modules. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7939–7948, 2020.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

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

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

Noam Elata, Tomer Michaeli, and Michael Elad. PSC: Posterior sampling-based compression. *TMLR*, 2025.

Ruoyu Feng, Xin Jin, Zongyu Guo, Runsen Feng, Yixin Gao, Tianyu He, Zhizheng Zhang, Simeng Sun, and Zhibo Chen. Image coding for machines with omnipotent feature learning. In *European Conference on Computer Vision (ECCV)*, pp. 510–528, 2022.

Ruoyu Feng, Yunpeng Qi, Jinming Liu, Yixin Gao, Xin Li, Xin Jin, and Zhibo Chen. Diff-ICMH: Harmonizing machine and human vision in image compression with generative prior. In *Advances in Neural Information Processing Systems*, volume 38, pp. 63782–63810, 2025.

Rich Franzen. Kodak lossless true color image suite, 1999.

Jinpei Guo, Yifei Ji, Zheng Chen, Kai Liu, Min Liu, Wang Rao, Wenbo Li, Yong Guo, and Yulun Zhang. OSCAR: One-step diffusion codec across multiple bit-rates. In *Advances in Neural Information Processing Systems*, volume 38, pp. 85267–85286, 2025.

Yingqing Guo, Hui Yuan, Yukang Yang, Minshuo Chen, and Mengdi Wang. Gradient guidance for diffusion models: An optimization perspective. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

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

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Advances in Neural Information Processing Systems*, 30, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Kai Hu, Han Wang, Renhe Liu, Zhilin Li, Shenghui Song, and Yu Liu. ROI-based deep image compression with implicit bit allocation. *arXiv preprint arXiv:2511.08918*, 2025.

Shoma Iwai, Tomo Miyazaki, and Shinichiro Omachi. Controlling rate, distortion, and realism: Towards a single comprehensive neural image compression model. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024. arXiv:2405.16817.

Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. MUSIQ: Multi-scale image quality transformer. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5148–5157. IEEE, 2021.

Nikolai Körber, Eduard Kromer, Andreas Siebert, Sascha Hauke, Daniel Mueller-Gritschneder, and Björn Schuller. PerCo (SD): Open perceptual compression. *arXiv preprint arXiv:2409.20255*, 2024. Machine Learning and Compression Workshop, NeurIPS 2024.

Binglin Li, Jie Liang, Haisheng Fu, and Jingning Han. ROI-based deep image compression with Swin transformers. In *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

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

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pp. 740–755, 2014.

Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. *Advances in Neural Information Processing Systems*, 33:11913–11924, 2020.

David Minnen, Johannes Ballé, and George Toderici. Joint autoregressive and hierarchical priors for learned image compression. *Advances in Neural Information Processing Systems*, 31, 2018.

Matthew J Muckley, Alaaeldin El-Nouby, Karen Ullrich, Hervé Jégou, and Jakob Verbeek. Improving statistical fidelity for neural image compression with implicit local likelihood models. In *International Conference on Machine Learning*, pp. 25426–25443. PMLR, 2023.

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

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022.

Avinab Saha, Sandeep Mishra, and Alan C. Bovik. Re-IQA: Unsupervised learning for image quality assessment in the wild. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5846–5855. IEEE, 2023.

Takahiro Shindo, Kein Yamada, Taiju Watanabe, and Hiroshi Watanabe. Image coding for machines with edge information learning using segment anything. In *2024 IEEE International Conference on Image Processing (ICIP)*, pp. 3702–3708. IEEE, 2024.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Tomáš Souček, Pierre Fernandez, Hady Elsahar, Sylvestre-Alvise Rebuffi, Valeriu Lacatusu, Tuan Tran, Tom Sander, and Alexandre Mourachko. Pixel seal: Adversarial-only training for invisible image and video watermarking. *arXiv preprint arXiv:2512.16874*, 2025.

Lucas Theis, Tim Salimans, Matthew D. Hoffman, and Fabian Mentzer. Lossy compression with Gaussian diffusion. *arXiv preprint arXiv:2206.08889*, 2022.

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

Jeremy Vonderfecht and Feng Liu. Lossy compression with pretrained diffusion models. In *International Conference on Learning Representations*, 2025. arXiv:2501.09815.

Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring CLIP for assessing the look and feel of images. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(2):2555–2563, 2023.

Zhou Wang, Eero P. Simoncelli, and Alan C. Bovik. Multiscale structural similarity for image quality assessment. In *Proceedings of the 37th Asilomar Conference on Signals, Systems and Computers*, volume 2, pp. 1398–1402. IEEE, 2003. doi: 10.1109/ACSSC.2003.1292216.

Yichong Xia, Yimin Zhou, Jinpeng Wang, and Bin Chen. Towards efficient low-rate image compression with frequency-aware diffusion prior refinement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 10942–10950, 2026.

Naifu Xue, Zhaoyang Jia, Jiahao Li, Bin Li, Yuan Zhang, and Yan Lu. One-step diffusion-based image compression with semantic distillation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 37108–37144, 2025.

Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. FreeDoM: Training-free energy-guided conditional diffusion model. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23174–23184, 2023a.

Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. *Advances in Neural Information Processing Systems*, 36:32215–32234, 2023b.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Tianyu Zhang, Xin Luo, Li Li, and Dong Liu. StableCodec: Taming one-step diffusion for extreme image compression. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17379–17389, 2025.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 5122–5130. IEEE, 2017.

Appendix

A More Related Works

A.1 Neural and Generative Image Compression

Learned transform codecs (Ballé et al., 2018; Minnen et al., 2018; Cheng et al., 2020; He et al., 2022) optimize a rate–distortion objective end to end and dominate classical codecs (Bellard, 2014; Bross et al., 2021) on PSNR at moderate rates. Perception-oriented codecs add adversarial or generative decoders (Mentzer et al., 2020; Muckley et al., 2023; Iwai et al., 2024), trading distortion for realism along the rate–distortion–perception frontier (Blau & Michaeli, 2018; 2019). Diffusion-based codecs either finetune a pretrained model into the pipeline (Careil et al., 2024; Körber et al., 2024; Xue et al., 2025; Guo et al., 2025; Xia et al., 2026) or use it zero-shot (Vonderfecht & Liu, 2025; Ohayon et al., 2025; Vaisman et al., 2026; Elata et al., 2025). Each trained codec is tied to the objective it was trained on: obtaining a region-prioritized or task-preserving variant means training another model. Zero-shot codecs avoid that cost but, until now, have been tied to a single fixed selection criterion instead.

A.2 Task- and Region-Aware Compression

Image coding for machines optimizes the bitstream for a downstream model (Shindo et al., 2024; Feng et al., 2025; 2022) rather than for a human viewer, and is evaluated by rate–accuracy rather than rate–distortion. Region-of-interest codecs allocate bits non-uniformly using a mask or a saliency map (Li et al., 2023; Hu et al., 2025). Both lines of work normally require task-specific training or an auxiliary bit allocation network. We reach both settings by changing one loss term at encoding time, with the same weights and the same bitstream syntax.

B Experiment Setting Details

B.1 Hyperparameter settings per BPP (rate)

The rate (bit-per-pixel; BPP) is determined by the same parameter as Turbo-DDCM (T, M, K, N). In our notation, $T_{\text{free}} = N + 1$, which is the number of diffusion steps that skip the noise encoding. Therefore the BPP for our method can be described as follows:

$$\text{BPP} = \frac{(T - T_{\text{free}}) \left(\lceil \log_2 \binom{K}{M} \rceil + M \right)}{N_{\text{px}}}. \tag{19}$$

Note that T_g does not affect the BPP values. But it can affect the computational complexity of the encoding process. In Table 1, we show the hyperparameter used for each BPP values that are used in the overall experiments of this paper. We used mainly $T = 60$ because this hyperparameter setting yielded better result on Turbo-DDCM.

B.2 Detail on pixel-level accuracy control loss

We use f_1 as MSE, f_2 as MS-SSIM loss and f_3 as LPIPS-VGG loss and those losses, including the loss on the latent level, are controlled by the weight (w_0, w_1, w_2, w_3). From the following pixel accuracy control loss (see Section 3.2):

$$L_{\text{pixel}}(x) = w_0 \|x_0 - x\|^2 + \sum_{i=1}^{3} w_i f_i(\mathcal{D}(x), y_0). \tag{20}$$

We use $(w_0, w_1, w_2, w_3) = (0.5, 0.125, 0.125, 0.25)$. We show ablative experimental results on these weights in Table 2.

Table 1: Hyperparameter settings per rate

bpp	T	M	K	T_{free}	coded steps	T_g	N_g	bits/step
0.0128	60	12	16384	38	22	47	9	152
0.0209	60	16	16384	32	28	43	11	196
0.0301	60	20	16384	27	33	39	12	239
0.0423	60	25	16384	22	38	40	18	292
0.0563	60	30	16384	17	43	42	25	343
0.0640*	60	33	16384	15	45	41	26	373
0.0721	60	36	16384	13	47	41	28	402
0.0914	60	43	16384	9	51	40	31	470
0.1164	60	52	16384	5	55	34	29	555

C Deployed Gradient vs. Gradient at μ_t

Let $\mu = \mu_\theta(x_t)$ and $J_s = \partial \hat{x}_{0|s}/\partial x$. By the chain rule, the Stein-optimal coefficient is $u^\star = \sigma_t \, \mathbb{E}_z\big[J_{t-1}(\mu + \sigma_t z)^\top \nabla L\big(\hat{x}_{0|t-1}(\mu + \sigma_t z)\big)\big]$: the next-step Jacobian, averaged over the candidate cloud. The tower identity (Equation 12) holds for every x_t, so we may differentiate both sides with respect to x_t:

$$\mathbb{E}_z\big[J_{t-1}(\mu + \sigma_t z)\big] \, \frac{\partial \mu_\theta}{\partial x_t} \;=\; J_t. \tag{21}$$

The current-step Jacobian therefore already contains the cloud-averaged next-step Jacobian, and $\hat{x}_{0|t}$ is already the cloud average of the next-step prediction. Hence

$$g_t \;=\; J_t^\top \nabla L\big(\hat{x}_{0|t}(x_t)\big) \;\approx\; \Big(\frac{\partial \mu_\theta}{\partial x_t}\Big)^\top \frac{u^\star}{\sigma_t}. \tag{22}$$

The only approximation is to replace the average of a product by the product of the averages, which is exact when L is affine and otherwise errs only through the curvature of L over one step of noise. The remaining factor $\partial \mu_\theta/\partial x_t = a_t I + b_t J_t$, with $b_t \ll a_t$ for a single step, is positive definite and close to a multiple of the identity, so it barely rotates $u^\star$. Since selection uses only the direction $\hat{g}_t$, this gives $\hat{u}^\star \approx \hat{g}_t$ at the cost of one backward pass through the forward evaluation the encoder already performs.

By contrast, the Taylor coefficient $\nabla \mathcal{L}(0)$ evaluates the next-step Jacobian at the single point μ rather than averaging it over the cloud. This costs an extra denoiser call on an input that carries less noise than its timestep asserts, and it is less accurate in practice (Table 6).

Table 2: Loss-weight ablation on Kodak24 (512^2). Each row is a weight tuple (w_1, w_2, w_3, w_4) on the latent-MSE, pixel-MSE, MS-SSIM and LPIPS-VGG terms. The bold tuple is the deployed LOONA weighting.

(w_1, w_2, w_3, w_4)	ΔPSNR	ΔMS-SSIM	ΔLPIPS-Alex
$(0, 0, 0, 0)$	-0.54	-0.0300	$+0.0325$
(.5, .125, .125, .25)	–	–	–
$(0, .25, .25, .5)$	-0.04	$+0.0042$	-0.0144
$(.571, 0, .143, .286)$	-0.12	-0.0035	$+0.0011$
$(.571, .143, 0, .286)$	-0.03	-0.0078	$+0.0025$
$(.667, .167, .167, 0)$	$+0.02$	-0.0011	$+0.0079$
$(1, 0, 0, 0)$	-0.17	-0.0123	$+0.0145$
$(0, 1, 0, 0)$	$+0.53$	$+0.0076$	$+0.0121$
$(0, 0, 0, 1)$	-1.28	-0.0563	$+0.0017$

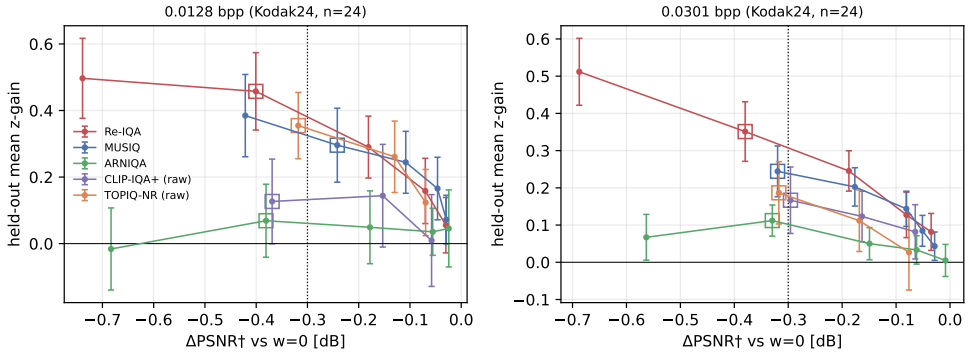


Figure 9: Ablative study to justify our choice of NR-IQA (Chen et al., 2024; Wang et al., 2023) model for steering. The y-axis is the mean of NR-IQA values of the held out metrics, and Re-IQA covers a larger PSNR delta/NR-IQA region.

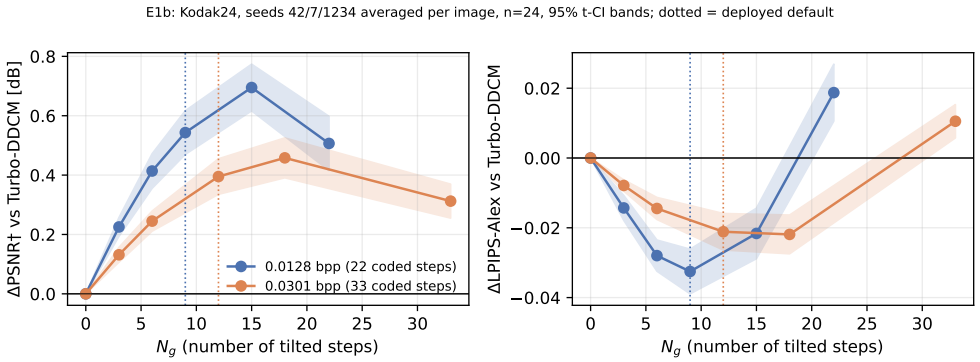


Figure 10: Ablative study on different number of N_g (gradient computing steps). We tuned our hyperparameters on the training set of DIV2K dataset.

D Ablative Studies and Additional Experiments

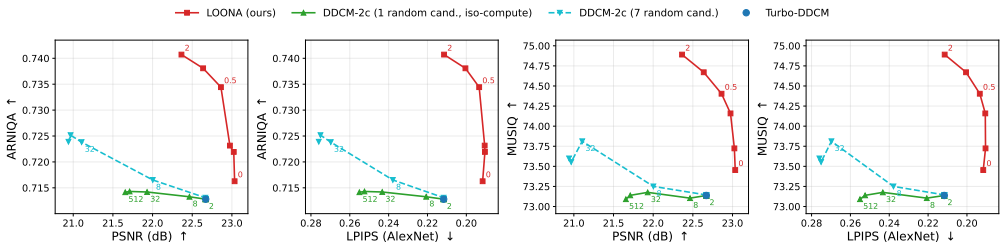


Figure 11: Comparison of perception-distortion curve traversal of DDCM's method on BPP=0.0301 (using two candidates to choose).

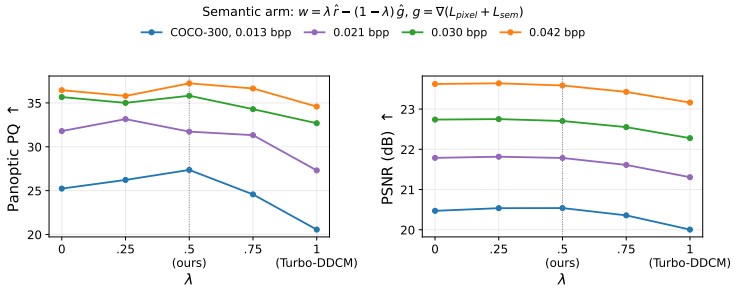


Figure 12: Ablative study on the mixing weight with the residual.

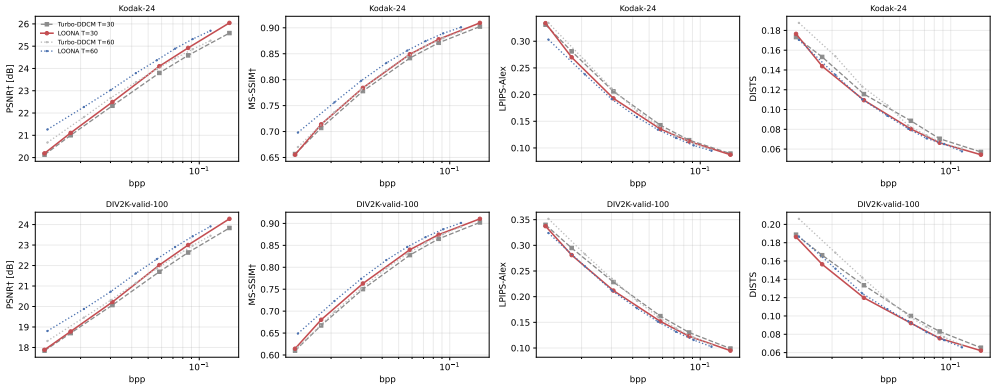


Figure 13: Comparison of Turbo-DDCM and LOONA on $T = 30$ and $T = 60$ on image quality metrics (Ding et al., 2022).

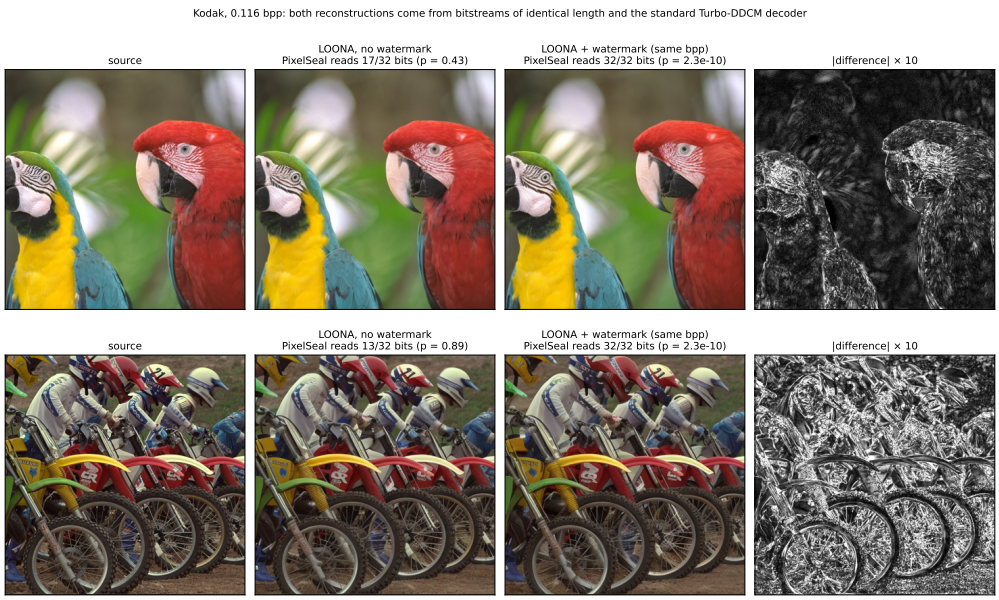


Figure 14: Qualitative result on adapting image watermarking to LOONA.

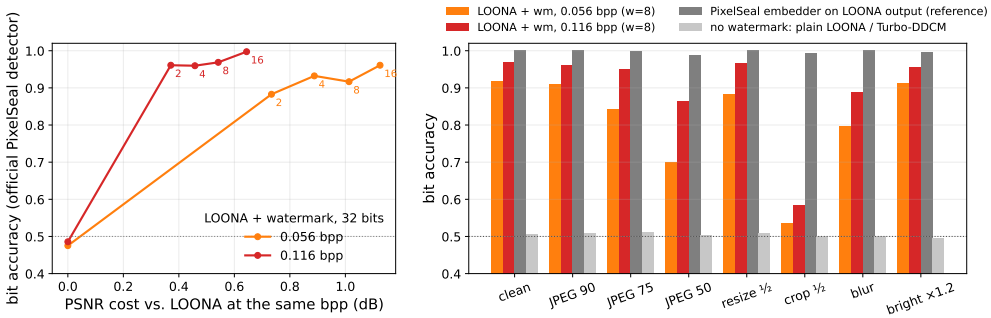


Figure 15: Quantitative result on adapting image watermarking to LOONA. Watermarking with LOONA on Kodak-24. The loss is the binary cross-entropy of the public PixelSeal detector (Souček et al., 2025) reading a secret 32-bit message; bitstream length and decoder are unchanged. Left: bit accuracy of the official detector on the decoded image versus the PSNR cost relative to LOONA at the same bpp; numbers mark the watermark weight w. Right: bit accuracy after common attacks for $w = 8$, compared with PixelSeal's own embedder applied after decoding (reference) and with unwatermarked reconstructions (chance level, dotted line).



Figure 16: Comparison of RoI-based compression on Turbo-DDCM and LOONA (ours).



Figure 17: Comparison of RoI-based compression on Turbo-DDCM and LOONA (ours).

Table 3: Every BD cell with the interval it is integrated over, grouped by metric. BD-rate intervals are quality ranges (PSNR dB; MS-SSIM; LPIPS-Alex; FID), BD-metric intervals are bpp ranges. Grey: < 3 points of one curve inside the interval; –: < 2 points or disjoint supports (not computed); n/a: no curve. Anchor: Turbo-DDCM, T=60 stock ladder, 8 rates. †: metric also appears in the optimized loss.

method	rate %	quality interval	Δ	bpp interval
Kodak24				
PSNR†				
LOONA (ours)	−18.7	[21.26, 25.25]	+0.44 dB	[0.0128, 0.1164]
DDCM	+65.7	[20.68, 23.45]	−1.11 dB	[0.0128, 0.1143]
DiffC	+2.2	[20.68, 25.25]	−0.07 dB	[0.0128, 0.0997]
StableCodec	+8.5	[20.68, 22.51]	−0.22 dB	[0.0128, 0.0366]
Diff-ICMH	+107.7	[20.68, 25.25]	−2.09 dB	[0.0162, 0.1164]
MS-SSIM†				
LOONA (ours)	−14.2	[0.6980, 0.8932]	+0.34 dB	[0.0128, 0.1164]
DDCM	+55.5	[0.6701, 0.8333]	−1.01 dB	[0.0128, 0.1143]
DiffC	−9.5	[0.6701, 0.8932]	+0.22 dB	[0.0128, 0.0997]
StableCodec	−13.8	[0.6701, 0.8020]	+0.27 dB	[0.0128, 0.0366]
Diff-ICMH	+54.5	[0.6701, 0.8932]	−1.14 dB	[0.0162, 0.1164]
LPIPS-Alex				
LOONA (ours)	−13.7	[0.0965, 0.3031]	-15.3×10^{-3}	[0.0128, 0.1164]
DDCM	+34.4	[0.1424, 0.3344]	$+40.1 \times 10^{-3}$	[0.0128, 0.1143]
DiffC	−13.3	[0.0983, 0.3162]	-15.4×10^{-3}	[0.0128, 0.0997]
StableCodec	−41.0	[0.1653, 0.3149]	-56.9×10^{-3}	[0.0128, 0.0366]
Diff-ICMH	+45.1	[0.0965, 0.3344]	$+42.2 \times 10^{-3}$	[0.0162, 0.1164]
FID				
LOONA (ours)	−16.4	[21.5, 41.6]	−1.9	[0.0128, 0.1164]
DiffC	−18.2	[22.6, 38.4]	−2.2	[0.0128, 0.0997]
StableCodec	−46.2	[27.6, 41.5]	−6.4	[0.0128, 0.0366]
Diff-ICMH	+52.1	[22.0, 41.8]	+4.1	[0.0162, 0.1164]
DIV2K				
PSNR†				
LOONA (ours)	−17.9	[19.79, 24.26]	+0.45 dB	[0.0128, 0.1164]
DiffC	+1.7	[19.21, 24.26]	−0.06 dB	[0.0128, 0.1051]
StableCodec	−0.8	[19.21, 21.37]	−0.02 dB	[0.0128, 0.0371]
Diff-ICMH	+106.8	[19.21, 24.26]	−2.31 dB	[0.0154, 0.1164]
MS-SSIM†				
LOONA (ours)	−15.0	[0.6623, 0.8853]	+0.37 dB	[0.0128, 0.1164]
DiffC	−13.9	[0.6402, 0.8853]	+0.35 dB	[0.0128, 0.1051]
StableCodec	−20.1	[0.6337, 0.7852]	+0.42 dB	[0.0128, 0.0371]
Diff-ICMH	+51.0	[0.6337, 0.8853]	−1.13 dB	[0.0154, 0.1164]
LPIPS-Alex				
LOONA (ours)	−13.5	[0.1088, 0.3214]	-15.3×10^{-3}	[0.0128, 0.1164]
DiffC	−20.0	[0.1088, 0.3190]	-22.9×10^{-3}	[0.0128, 0.1051]
StableCodec	−46.9	[0.1756, 0.3159]	-67.0×10^{-3}	[0.0128, 0.0371]
Diff-ICMH	+33.7	[0.1088, 0.3509]	$+34.0 \times 10^{-3}$	[0.0154, 0.1164]
FID				
LOONA (ours)	−18.4	[11.4, 27.7]	−1.6	[0.0128, 0.1164]
DiffC	−26.4	[12.0, 21.3]	−2.8	[0.0128, 0.1051]
StableCodec	−55.7	[14.8, 24.5]	−6.7	[0.0128, 0.0371]
Diff-ICMH	+46.6	[11.4, 30.4]	+3.4	[0.0154, 0.1164]

Table 4: Every BD cell of the PQ–rate curves with its integration interval. iso: BD-rate recomputed after an isotonic (pool-adjacent-violators) fit of PQ on rate for both curves, which makes rate(PQ) a function when a curve is not monotone ($^\sharp$). The w=0.40 row marked r012–r043 is restricted to the four rates where the w=0.25/0.60 arms exist. Anchor: Turbo-DDCM, T=60 stock ladder.

method	BD-PQ	bpp interval	rate %	PQ interval	iso rate %
COCO2017					
LOONA (ours)	+2.38	[0.0128, 0.1172]	−29.6$^\sharp$	[26.0, 38.4]	−29.3
LOONA+DINO, w=0.25	+3.56	[0.0128, 0.0426]	−28.8$^\sharp$	[25.2, 36.7]	−28.0
LOONA+DINO, w=0.40	+2.62	[0.0128, 0.1172]	−30.6$^\sharp$	[27.3, 38.4]	−30.2
same, r012–r043 only	+4.25	[0.0128, 0.0426]	−33.9$^\sharp$	[27.3, 37.1]	−32.7
LOONA+DINO, w=0.60	+4.24	[0.0128, 0.0426]	−34.7$^\sharp$	[26.1, 36.2]	−34.2
DDCM	−1.98	[0.0128, 0.1131]	+20.9$^\sharp$	[20.6, 36.6]	+21.8
DiffC	+1.66	[0.0128, 0.1029]	−19.8$^\sharp$	[22.9, 37.9]	−20.7
StableCodec	+1.78	[0.0128, 0.0514]	−19.9$^\sharp$	[22.9, 34.8]	−20.3
Diff-ICMH	+1.07	[0.0164, 0.1172]	−16.1$^\sharp$	[28.9, 38.4]	−15.4
ADE20K-150					
LOONA (ours)	+2.44	[0.0094, 0.0413]	−24.0	[26.1, 34.9]	−24.0
LOONA+DINO, w=0.25	n/a		n/a		
LOONA+DINO, w=0.40	+2.97	[0.0094, 0.0413]	−25.0$^\sharp$	[23.1, 34.9]	−27.6
LOONA+DINO, w=0.60	n/a		n/a		
DDCM	n/a		n/a		
DiffC	+2.87	[0.0094, 0.0371]	−27.5	[23.9, 34.9]	−27.5
StableCodec	+4.36	[0.0094, 0.0396]	−44.1$^\sharp$	[25.8, 34.9]	−44.2
Diff-ICMH	−1.63	[0.0155, 0.0413]	+38.3	[26.5, 34.9]	+38.3

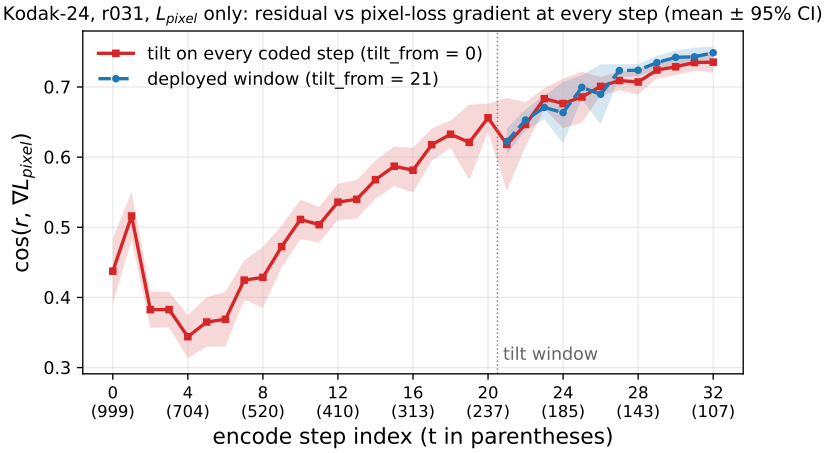


Figure 18: Cosine similarity of the residual r_t and the gradient from the pixel-level loss (see Section 3.2).

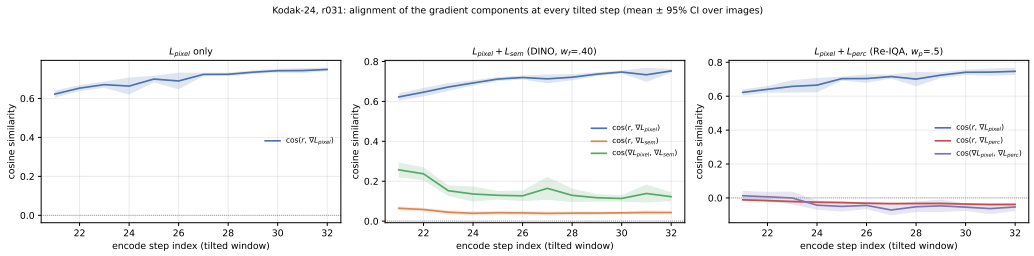


Figure 19: Cosine similarity of the residual r_t and the gradient from the pixel-level, semantics, perceptual losses (see Section 3).

Table 5: Other machine vision tasks performance from our semantic-optimized encoding.

bpp	Method	set
Detection		
0.0128	Turbo-DDCM	33.5
	LOONA	37.4
	LOONA+DINO (w_{sem}=.25)	39.7
	LOONA+DINO (w_{sem}=.40)	36.0
0.0301	Turbo-DDCM	51.3
	LOONA	54.3
	LOONA+DINO (w_{sem}=.25)	54.2
	LOONA+DINO (w_{sem}=.40)	56.1
0.0563	Turbo-DDCM	63.3
	LOONA	65.7
	LOONA+DINO (w_{sem}=.25)	66.5
	LOONA+DINO (w_{sem}=.40)	67.0
Instance segmentation		
0.0128	Turbo-DDCM	28.9
	LOONA	34.6
	LOONA+DINO (w_{sem}=.25)	35.0
	LOONA+DINO (w_{sem}=.40)	34.4
0.0301	Turbo-DDCM	45.1
	LOONA	51.2
	LOONA+DINO (w_{sem}=.25)	51.6
	LOONA+DINO (w_{sem}=.40)	51.3
0.0563	Turbo-DDCM	58.6
	LOONA	63.6
	LOONA+DINO (w_{sem}=.25)	63.4
	LOONA+DINO (w_{sem}=.40)	64.5
Pose (person images only)		
0.0128	Turbo-DDCM	6.7
	LOONA	9.8
	LOONA+DINO (w_{sem}=.25)	10.5
	LOONA+DINO (w_{sem}=.40)	10.2
0.0301	Turbo-DDCM	16.5
	LOONA	19.1
	LOONA+DINO (w_{sem}=.25)	19.2
	LOONA+DINO (w_{sem}=.40)	19.3
0.0563	Turbo-DDCM	27.7
	LOONA	28.9
	LOONA+DINO (w_{sem}=.25)	28.7
	LOONA+DINO (w_{sem}=.40)	28.1
Panoptic segmentation		
0.0128	Turbo-DDCM	29.4
	LOONA	32.9
	LOONA+DINO (w_{sem}=.25)	33.1
	LOONA+DINO (w_{sem}=.40)	36.0
0.0301	Turbo-DDCM	44.4
	LOONA	48.4
	LOONA+DINO (w_{sem}=.25)	52.1
	LOONA+DINO (w_{sem}=.40)	51.7
0.0563	Turbo-DDCM	50.9
	LOONA	58.4
	LOONA+DINO (w_{sem}=.25)	58.2
	LOONA+DINO (w_{sem}=.40)	58.9

Table 6: Gradient estimators for the selection target. Ours: current-step gradient g_t. Taylor: next-step gradient at $\mu_\theta(x_t)$ (Appendix C). PSNR, MS-SSIM and LPIPS: Kodak-24, mean over three codebook seeds. PQ: panoptic quality on COCO val2017 (n=75) with the DINO loss. Time: encoding time relative to ours. The time difference happens because computing gradient on x_t doesn't require extra forward step, since this forward step was previously required.

bpp	Estimator	PSNR ↑	MS-SSIM ↑	PQ ↑	Time
0.0128	Ours	**21.28**	**0.6987**	**26.4**	1.00×
	Taylor	21.19	0.6945	24.9	1.03×
0.0301	Ours	**23.04**	**0.7984**	**32.8**	1.00×
	Taylor	23.02	0.7964	31.9	1.04×
0.0563	Ours	**24.40**	**0.8564**	**35.3**	1.00×
	Taylor	24.38	0.8557	34.7	1.06×

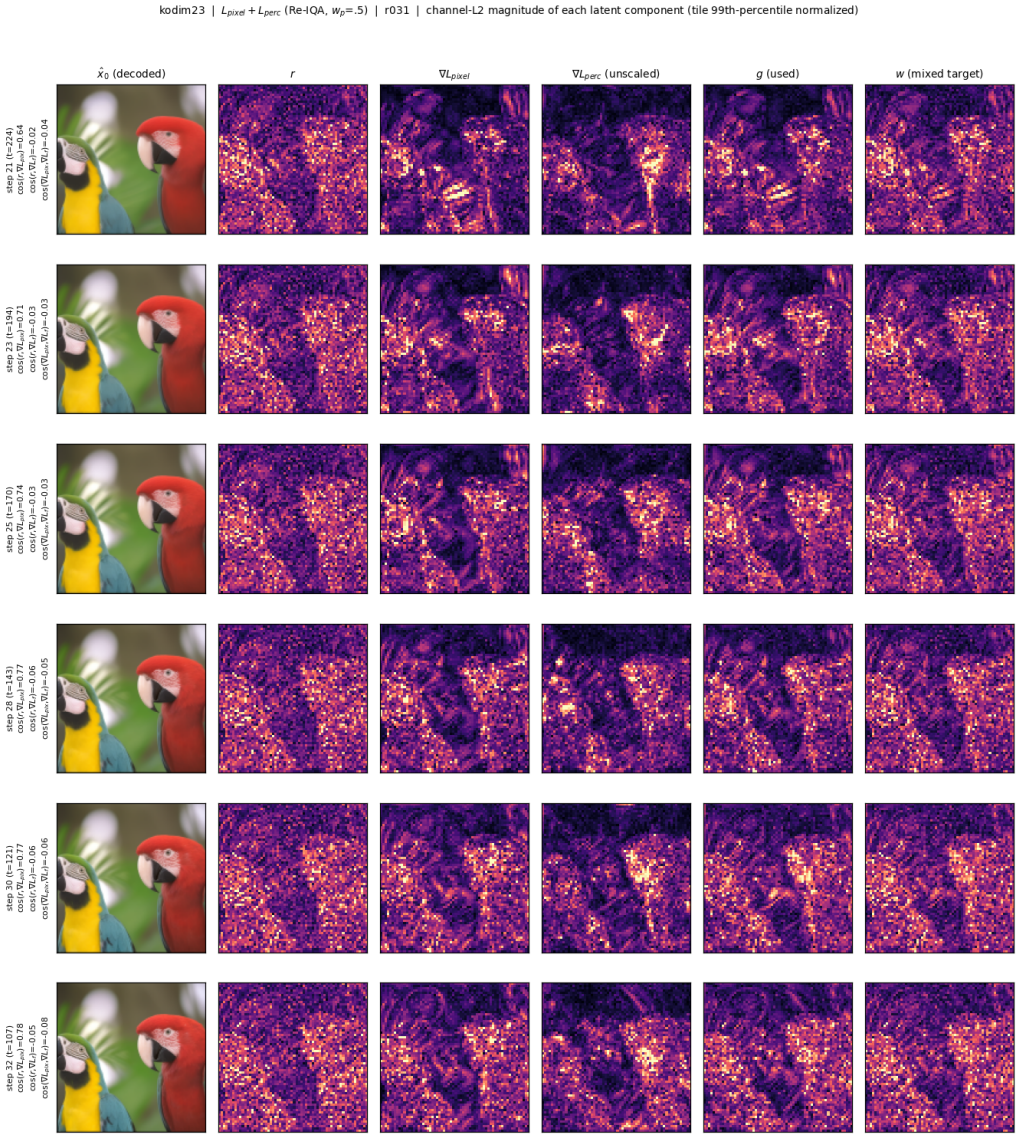


Figure 20: Visualization of the residual and the gradient of the pixel-loss/perceptual loss (see Section 3.2 and Section 3.5).

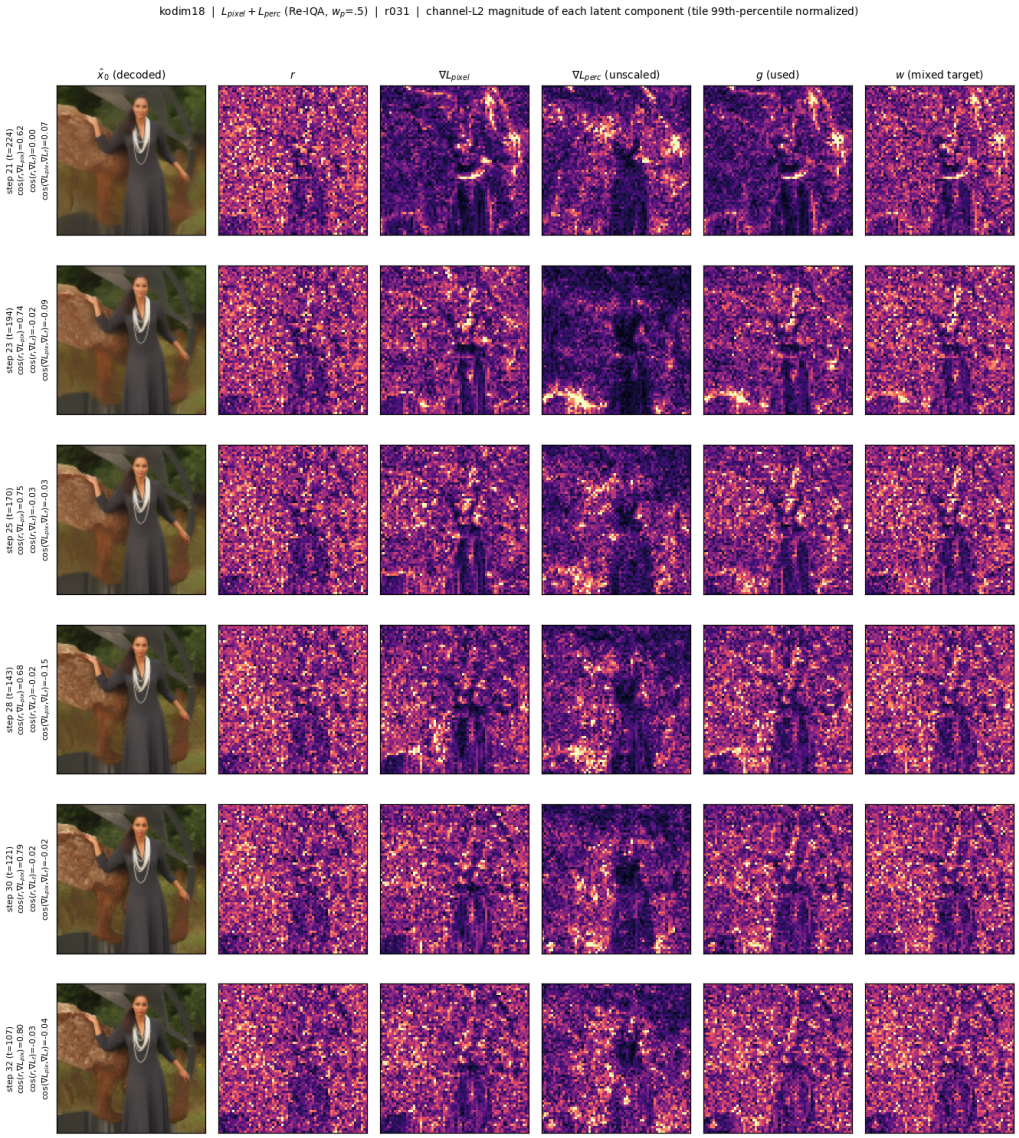


Figure 21: Visualization of the residual and the gradient of the pixel-loss/perceptual loss (see Section 3.2 and Section 3.5).

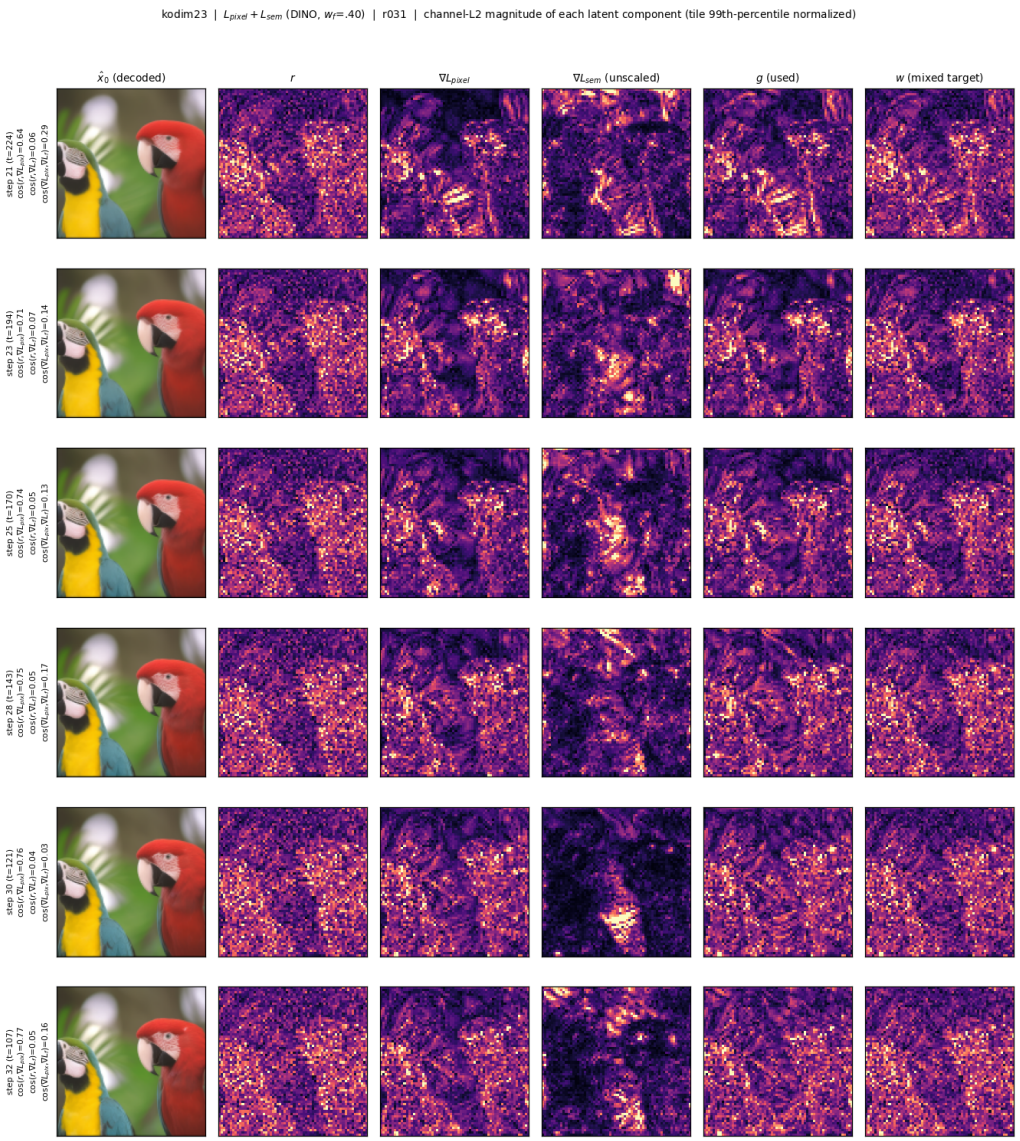


Figure 22: Visualization of the residual and the gradient of the pixel-loss/semantic loss (see Section 3.2 and Section 3.4).

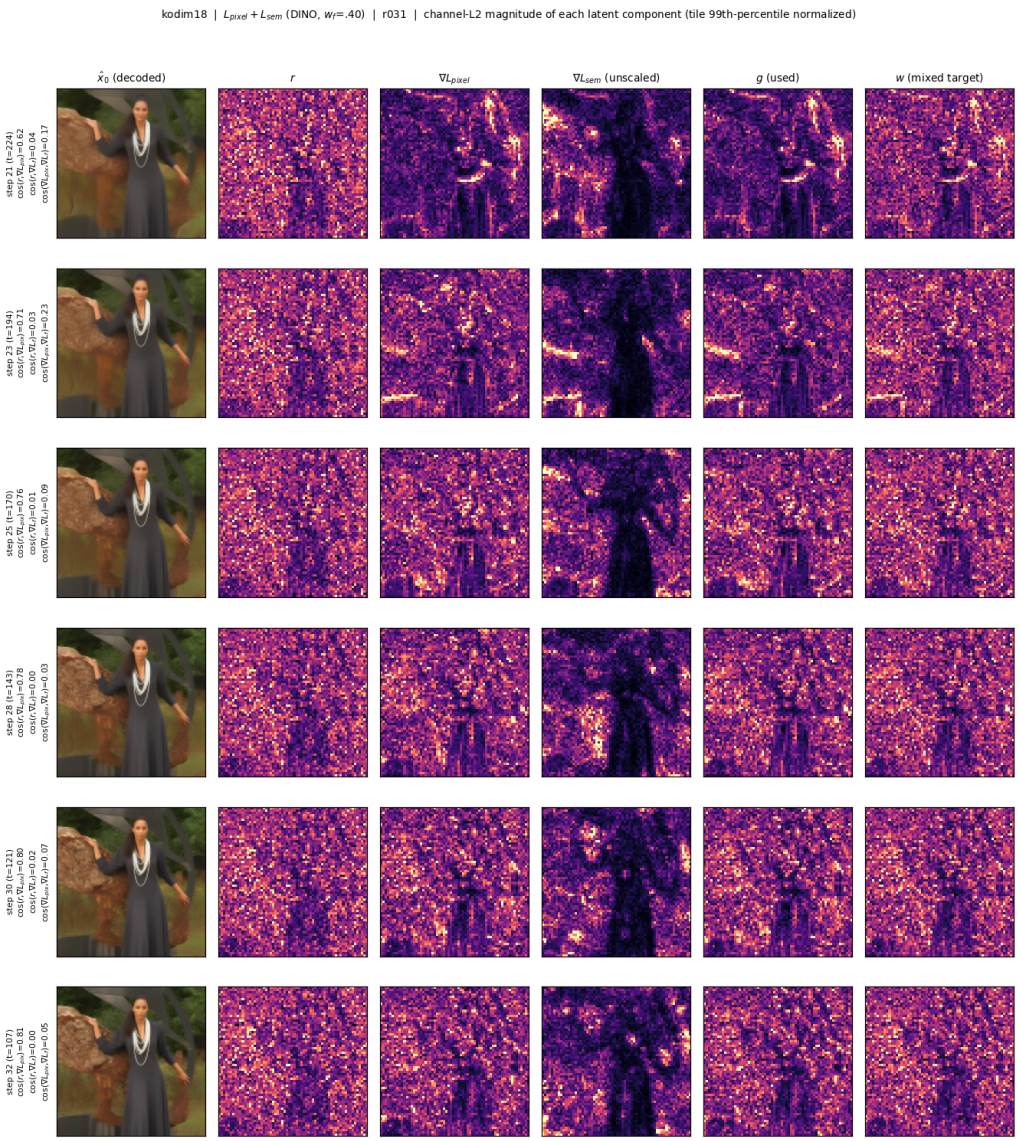


Figure 23: Visualization of the residual and the gradient of the pixel-loss/semantic loss (see Section 3.2 and Section 3.4).