VISUAL ENTITY ORIENTED DIFFUSION CODING FOR IMAGENE COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative image editing produces collections of closely related images, each comprising a source image and its edited variants. Such collections impose growing demands on compression and storage. We call these collections **ImaGenes**, as shared content is inherited across variants while edits alter only localized regions, just as genes are inherited while mutations create variation. Within an ImaGene, edits operate on *visual entities*, identifiable units of image content. Independent encoding redundantly represents these shared entities, while joint compression exploiting their preservation and variation remains underexplored. To this end, we propose **ImaGene Compression**, a joint compression paradigm built upon visual entities as its basic units. It represents each ImaGene through shared entities and their variations. We realize this paradigm with **EnDiC** (Visual Entity Oriented Diffusion Codec), a training-free diffusion-based compression framework. EnDiC localizes entity changes, shares diffusion reconstruction trajectories for unchanged content across variants, and encodes the diffusion direction choices that reconstruct changed entities as incremental bitstreams. Furthermore, EnDiC adapts bit allocation across spatial regions and denoising steps to improve coding efficiency. We also introduce **ImaGeneSet**, a dataset for this task, covering entity addition, removal, and modification. High-fidelity generative editing with quality screening yields clearly defined entity changes while keeping unchanged regions spatially consistent with the source. Experiments on ImaGeneSet demonstrate that EnDiC achieves better performance on several full-reference perceptual metrics at low bitrates. Our code and data will be made publicly available upon publication.

1 INTRODUCTION

Recent advances in generative AI (ByteDance Seed, 2026; OpenAI, 2026) have made image creation and manipulation increasingly accessible, bringing new demands on storage and transmission. Generative image editing produces collections of closely related variants of a source image. Within such a collection, edits primarily operate on some identifiable content units, referred to as *visual entities*, such as objects, object parts, and background regions (Qi et al., 2023; Cao et al., 2024), while largely preserving the remaining content. Inspired by genetic inheritance and variation, we define an **ImaGene** as a collection comprising a source image and its generatively edited variants, connected through shared visual entities and their variations, as shown in Figure 1 (left).

Single-image compression reduces storage costs by exploiting redundancy within an individual image or latent (Ballé et al., 2017; Minnen et al., 2018). Within an ImaGene, redundancy also exists in visual entities across the source image and its edited variants. Independent encoding represents shared content repeatedly, leaving this cross-image redundancy unaddressed. We therefore propose **ImaGene Compression**, a joint compression paradigm that organizes compression around visual entities, reusing shared representations and encoding entity changes as incremental information.

Efficiently coding entity changes requires a compact representation from which the decoder can reconstruct the affected content. For example, removing an entity may require completing the newly exposed background, whereas adding or modifying an entity requires reconstructing its appearance. Diffusion models provide visual priors for reconstructing these changes. Given a shared model, initialization, and sampling procedure, discrete noise-selection choices at each denoising step allow the decoder to reproduce a reconstruction trajectory. During iterative denoising, the encoder

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.

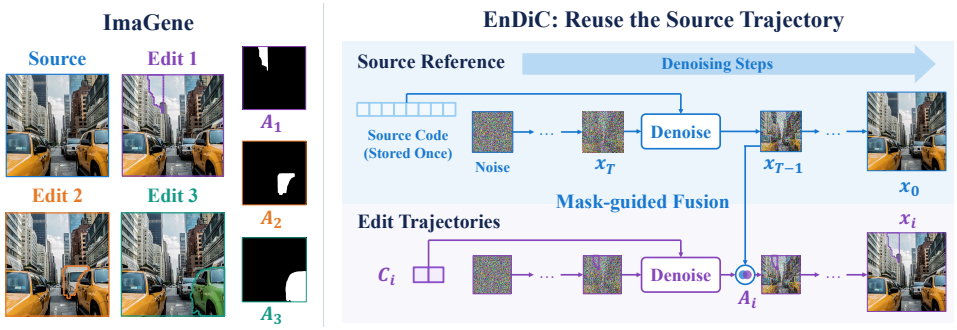

Figure 1: **ImaGene Compression.** Left: a source image and its generatively edited variants involving entity removal, addition, and modification form an ImaGene, sharing unchanged entities while differing in localized entity changes, with corresponding masks. Right: the source bitstream establishes a shared reconstruction trajectory, while each incremental bitstream C_i encodes direction choices for a localized entity change. During decoding, the resulting edit-branch states are fused with the shared source states.

selects these inputs from a shared finite codebook to guide reconstruction toward the target image and encodes the choices in the bitstream. The decoder recovers these choices and reproduces the selected trajectory (Ohayon et al., 2025; Vaisman et al., 2026). Building on this trajectory-based representation, we propose **EnDiC** (Visual Entity Oriented Diffusion Codec), a training-free diffusion-based framework for jointly compressing an ImaGene. EnDiC localizes the affected entities in each source-variant pair, reuses the source trajectory for unchanged regions, and encodes direction choices for reconstructing entity changes as incremental bitstreams (Figure 1).

The efficiency of this representation also depends on how the limited bits are allocated. The required coding capacity depends on the spatial extent of entity changes, while the effect of direction selection on the final reconstruction varies across denoising steps. We therefore introduce **Region- and Step-Aware Bit Allocation** (RSBA), which adapts direction-selection capacity to edit-region size and the reconstruction gains at each step.

To support evaluation of this task, we introduce **ImaGeneSet**, a dataset comprising source images and their edited variants, covering entity addition, removal, and modification. High-fidelity generative editing with quality screening yields clearly defined entity changes while keeping unedited content spatially consistent with the source. Experiments on ImaGeneSet demonstrate the benefits of EnDiC in perceptual reconstruction quality and visual fidelity.

Our contributions are summarized as follows:

- We propose **ImaGene Compression**, a new paradigm for jointly compressing a source image and its generatively edited variants. This paradigm organizes compression around shared visual entities and their variations within an ImaGene. It reduces cross-image redundancy while preserving the distinct content of each variant.
- To realize this paradigm, we propose **EnDiC** (Visual Entity Oriented Diffusion Codec), a training-free diffusion-based compression framework. EnDiC reuses diffusion reconstruction trajectories for shared content within an ImaGene, and encodes entity changes through incremental bitstreams. It incorporates Region- and Step-Aware bit allocation to adapt coding capacity across edit regions and denoising stages, improving coding efficiency.
- We introduce **ImaGeneSet**, a dataset designed for ImaGene Compression, covering entity addition, removal, and modification. Built through high-fidelity generative editing and quality screening, the dataset features clearly defined entity changes and spatial consistency in unedited regions.

2 Related Work

2.1 Single-Image Compression

Learned image compression jointly optimizes transforms and quantization for rate-distortion performance (Ballé et al., 2017), with hyperpriors and autoregressive models improving entropy estimation (Ballé et al., 2018; Minnen et al., 2018). Recent architectures further improve representation

and context modeling through mixed Transformer-CNN blocks (Liu et al., 2023), frequency-aware transforms (Li et al., 2024), state-space modeling (Zeng et al., 2025), and adaptive graph representations (Chen et al., 2026b). Despite their advances in rate-distortion performance and perceptual quality, these methods treat each image as a self-contained coding target and primarily exploit redundancy within that image. They do not address the structured content preservation across a source image and its generatively edited variants.

Generative codecs improve low-rate perceptual quality through adversarial learning (Mentzer et al., 2020; Muckley et al., 2023) or diffusion priors (Careil et al., 2024; Li et al., 2025; Zhang et al., 2025). Training-free approaches further exploit pretrained diffusion models without compression-specific training. DiffC uses reverse-channel coding (Vonderfecht & Liu, 2025). Denoising Diffusion Codebook Models (DDCM) and Turbo-DDCM encode reproducible noise-codebook selections that steer denoising toward a target image (Ohayon et al., 2025; Vaisman et al., 2026). We draw on this trajectory-coding mechanism for joint compression by reusing the source trajectory for preserved regions and encoding incremental directions for edited entities.

2.2 Compression across Related Images

Beyond single-image coding, related images can provide shared information for compression. Image-set compression exploits correlations among similar images (Karadimitriou & Tyler, 1998; Zhang et al., 2018b), while conditional latent coding synthesizes a reference from an external feature dictionary (Wu et al., 2025). Although these formulations reduce redundancy using general image similarity or external reference features, they do not explicitly model the source-variant relationship induced by editing with generative models. Specifically, in an ImaGene, each variant inherits spatially aligned content from a common source while altering only a small number of visual entities. This structure calls for explicitly distinguishing preserved content from edited content, which is not directly captured by general image similarity models.

Other joint coding approaches use the spatial or temporal relationships between images. Multi-view coding exploits correlations between observations of a scene, as in LDMIC (Zhang et al., 2023b). Similarly, video codecs model temporal dependencies through joint multi-frame representations, exemplified by DCVC-UF (Li et al., 2026), or generative representations conditioned on short- and long-term context, as in GVC1D (Zheng et al., 2026). However, the dependencies within an ImaGene arise from editing a shared source: its independently edited variants contain preserved regions and localized semantic changes rather than changes arising from viewpoint or temporal progression. Together, these distinctions motivate an edit-aware joint formulation that reuses preserved source content and explicitly encodes entity changes.

3 ImaGene Compression

3.1 Problem Formulation

Generative image editing produces multiple versions of a source image through changes to its visual entities. A *visual entity* is an identifiable and visually coherent unit of image content, such as an object, an object part, or a background region (Qi et al., 2023; Cao et al., 2024). At the entity level, we abstract an image I as a collection $\mathcal{E}$ of m visual entities:

$$\mathcal{E} = \{e_1, \ldots, e_m\}, \tag{1}$$

where e_k is the k-th visual entity in the image I, including its appearance and spatial configuration, for $k = 1, \ldots, m$.

Consider a source image $I^{(0)}$ and n edited variants, with $I^{(i)}$ denoting the i-th variant. Let $\mathcal{E}_0$ and $\mathcal{E}_i$ denote their respective entity collections. For each edit $\mathcal{T}_i$, let $\mathcal{S}_i$ be the subset of source entities $\mathcal{E}_0$ affected by the edit. We express the edited entity collections and the image collection $\mathcal{C}$ as

$$\mathcal{E}_i = \big(\mathcal{E}_0 \setminus \mathcal{S}_i\big) \cup \mathcal{T}_i(\mathcal{S}_i), \quad i = 1, \ldots, n, \qquad \mathcal{C} = \{I^{(i)}\}_{i=0}^{n}. \tag{2}$$

Here, $\mathcal{E}_0 \setminus \mathcal{S}_i$ contains the source entities preserved by the edit, while $\mathcal{T}_i(\mathcal{S}_i)$ contains the entities resulting from the transformation. We call $\mathcal{C}$ an **ImaGene**, which comprises a source image and its generatively edited variants connected through shared visual entities and their variations.

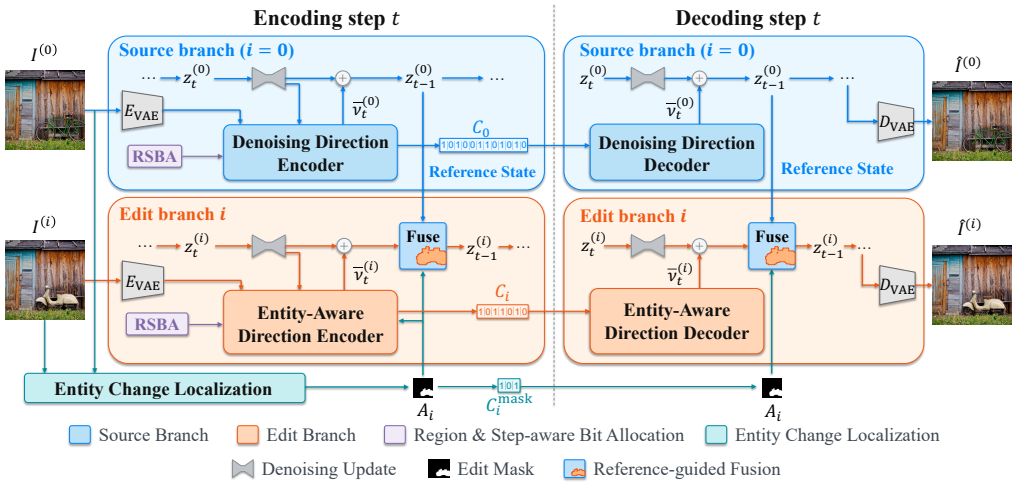

Figure 2: Overview of EnDiC. The source branch establishes a reference reconstruction trajectory, while edit branches encode localized entity changes and fuse their updates with these states. The decoder reconstructs the branches using the transmitted direction choices and edit-region information. RSBA allocates direction-selection capacity across regions and denoising steps.

Given an image distortion measure $D(\cdot,\cdot)$ and a total bit budget B, **ImaGene Compression** aims to jointly encode the entire ImaGene while minimizing the average distortion of reconstruction:

$$\min_{\text{Enc,Dec}} \frac{1}{n+1}\sum_{i=0}^{n} D\big(I^{(i)}, \widehat{I}^{(i)}\big) \quad \text{s.t.} \quad L(b) \le B,$$
$$b = \text{Enc}(\mathcal{C}), \qquad \widehat{\mathcal{C}} = \text{Dec}(b) = \{\widehat{I}^{(i)}\}_{i=0}^{n}, \tag{3}$$

where Enc and Dec denote the encoder and decoder, b is the bitstream representing $\mathcal{C}$, and $L(b)$ is its total bit length. The reconstructed collection $\widehat{\mathcal{C}}$ contains $\widehat{I}^{(i)}$, the reconstruction of $I^{(i)}$.

3.2 EnDiC: Visual Entity Oriented Diffusion Codec

Inspired by recent trajectory-based diffusion codecs (Ohayon et al., 2025; Vaisman et al., 2026), we propose **EnDiC**, a training-free codec that jointly compresses an ImaGene using one source branch and n edit branches (Figure 2). The edit branches encode entity changes through incremental direction choices and reuse the source trajectory for unchanged content. Entity Change Localization supplies edit masks (Section 3.3), while RSBA allocates atoms across regions and steps (Section 3.4).

Entity-aware direction guidance. All branches reconstruct their images through T reverse diffusion steps from the same Gaussian initialization. The pretrained VAE encoder E_{VAE} maps each image $I^{(i)}$ to latent space, and the diffusion model predicts a clean latent at each reverse timestep t:

$$z_{\text{tar}}^{(i)} = E_{\text{VAE}}\big(I^{(i)}\big), \qquad \widehat{z}_{0|t}^{(i)} = \frac{z_t^{(i)} - \sqrt{1-\bar{\alpha}_t}\,\epsilon_\theta(z_t^{(i)}, t)}{\sqrt{\bar{\alpha}_t}}, \quad i = 0,\dots,n, \tag{4}$$

where $z_{\text{tar}}^{(i)}$ is the target latent, $\widehat{z}_{0|t}^{(i)}$ is the predicted clean latent, and $z_t^{(i)}$ is the current latent state; ϵ_θ is the noise predictor with frozen parameters θ, and $\bar{\alpha}_t$ is the cumulative signal coefficient in the diffusion noise schedule. In stochastic reverse diffusion, each update injects a Gaussian noise vector. Selecting this vector to align with the discrepancy between the target and predicted clean latents turns the noise input into a corrective direction that steers reconstruction toward the target.

The source branch guides direction selection using the full-image discrepancy, while edit branches focus on entity changes and reuse the source reconstruction for unchanged content. Let A_i be a soft edit mask in latent space with values in $[0,1]$ that emphasizes entity changes, derived from the binary edit mask estimated for each source-variant pair as described in Section 3.3. Direction selection uses the full latent residual for the source and the spatially weighted residual for each edit:

$$r_t^{(0)} = z_{\text{tar}}^{(0)} - \widehat{z}_{0|t}^{(0)}, \qquad r_t^{(i)} = A_i \odot \big(z_{\text{tar}}^{(i)} - \widehat{z}_{0|t}^{(i)}\big), \quad i = 1,\dots,n, \tag{5}$$

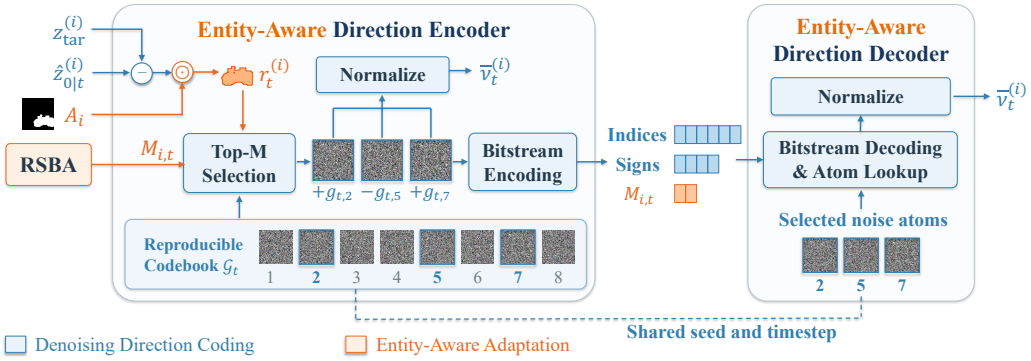

Figure 3: Entity-aware direction coding. The edit mask A_i localizes the residual, and RSBA sets the atom count $M_{i,t}$. The decoder recovers the direction from transmitted atom indices and signs. Blue and orange denote direction coding and entity-aware adaptations, respectively.

where $r_t^{(i)}$ is the unnormalized latent residual used to guide codebook selection, and $\odot$ denotes element-wise multiplication with the edit mask A_i broadcast across latent channels.

Direction encoding and decoding. We encode the directional information in $r_t^{(i)}$ through signed atom selections from a finite codebook with size K, a strategy also used in Turbo-DDCM (Vaisman et al., 2026), as shown in Figure 3. At each coded step, the encoder and decoder deterministically regenerate a shared codebook $\mathcal{G}_t$ from the same random seed and timestep:

$$\mathcal{G}_t = \{g_{t,k}\}_{k=1}^{K}, \qquad g_{t,k} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d), \tag{6}$$

where $g_{t,k}$ is its k-th Gaussian atom, d is the number of latent elements, and $\mathbf{I}_d$ is the d-dimensional identity matrix. Latents are vectorized for inner products with the atoms. Given an atom count $M_{i,t}$ assigned by RSBA, the encoder selects the atoms most aligned with $r_t^{(i)}$ and records their signs:

$$S_t^{(i)} = \mathrm{TopM}_{k\in\{1,\dots,K\}}\left(\left|\langle g_{t,k}, r_t^{(i)}\rangle\right|; M_{i,t}\right), \qquad s_{t,k}^{(i)} = \mathrm{sign}\langle g_{t,k}, r_t^{(i)}\rangle, \quad k \in S_t^{(i)}, \tag{7}$$

where $\langle\cdot,\cdot\rangle$ denotes the Euclidean inner product, TopM returns the indices of the $M_{i,t}$ largest scores, $S_t^{(i)}$ is the selected index set, and $s_{t,k}^{(i)}$ is the sign of each selected atom, taking values in $\{-1,+1\}$, with zero inner products assigned $+1$. The selected atoms form the injected direction:

$$v_t^{(i)} = \sum_{k\in S_t^{(i)}} s_{t,k}^{(i)} g_{t,k}, \qquad \bar{\nu}_t^{(i)} = \frac{v_t^{(i)}}{\mathrm{Std}(v_t^{(i)})}, \tag{8}$$

where $v_t^{(i)}$ is the signed atom sum and Std computes standard deviation. Then, the encoder losslessly encodes the selected indices and signs into the branch bitstream C_i. The decoder recovers the injected direction $\bar{\nu}_t^{(i)}$ from the decoded choices and shared codebook using Equation 8.

Reference-guided reconstruction. The recovered direction replaces the random noise injected into a reverse diffusion step, producing a candidate state for the next step:

$$\tilde{z}_{t-1}^{(i)} = \sqrt{\bar{\alpha}_{t-1}}\,\hat{z}_{0|t}^{(i)} + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\,\epsilon_\theta(z_t^{(i)}, t) + \sigma_t \bar{\nu}_t^{(i)}, \quad \sigma_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\left(1-\frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}\right), \tag{9}$$

where $\tilde{z}_{t-1}^{(i)}$ denotes the candidate state before reference fusion, σ_t is the noise injection scale at a coded step, and $\bar{\alpha}_{t-1}$ is the signal coefficient at the next scheduled noise level. The source branch directly adopts its candidate, establishing the reference trajectory from $z_T^{(0)}$ to $z_0^{(0)}$. As illustrated in Figure 2, each edit branch fuses its candidate with the corresponding source state:

$$z_{t-1}^{(0)} = \tilde{z}_{t-1}^{(0)}, \qquad z_{t-1}^{(i)} = A_i \odot \tilde{z}_{t-1}^{(i)} + (1-A_i)\odot z_{t-1}^{(0)}, \quad i = 1,\dots,n. \tag{10}$$

After the reverse diffusion steps, the pretrained VAE decoder D_{VAE} maps each final latent $z_0^{(i)}$ to the image reconstruction $\hat{I}^{(i)}$, for $i = 0,\dots,n$.

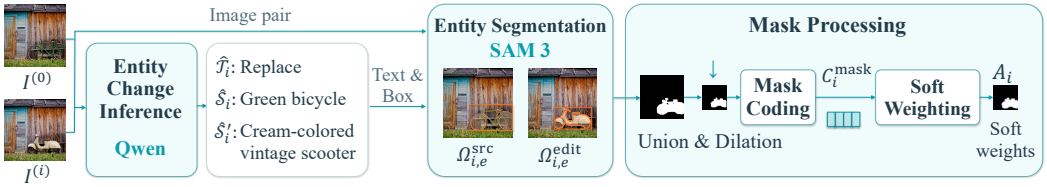

Figure 4: Entity Change Localization. Qwen identifies entity changes, and SAM 3 segments the affected regions in both images. Their union is processed and encoded as a mask, from which both the EnDiC encoder and the EnDiC decoder derive the soft edit mask A_i.

3.3 Entity Change Localization

The entity collections and edit transformations in Section 3.1 are not directly observed. We design an Entity Change Localization module (Figure 4) to estimate the edit mask A_i from each source-variant pair for direction selection and reference-state fusion.

Given the source image $I^{(0)}$ and its edited variant $I^{(i)}$, the pretrained vision-language model Qwen first infers an edit description $\widehat{\mathcal{T}}_i$. Conditioned on this description, Qwen identifies the affected source entities $\widehat{\mathcal{S}}_i$ and their edited counterparts $\widehat{\mathcal{S}}'_i$, which estimate $\mathcal{S}_i$ and $\mathcal{T}_i(\mathcal{S}_i)$, respectively. Each identified entity is described by a text phrase and a bounding box in the image where it appears.

Using these text and box prompts, SAM 3 (Carion et al., 2026) segments the identified entities in both images. We form the edit region Ω_i by taking the union of the segmented regions from both images and dilating it to cover boundary discrepancies. We supplement Ω_i with RGB-difference cues and losslessly transmit the downsampled binary mask as C_i^{mask}. Both the EnDiC encoder and the EnDiC decoder derive the same soft edit mask A_i from the decoded binary mask to emphasize edited regions without extra side information. Details are provided in Appendix A.1.

3.4 RSBA: Region- and Step-Aware Bit Allocation

The direction choices encoded in Section 3.2 determine the reconstruction of each branch, while their bit cost depends on the atom counts $M_{i,t}$. Encoding a set of M atoms from a codebook of size K requires a combinatorial index and M sign bits. Given height H and width W, the resulting direction cost and the average bits per pixel (bpp) R of an ImaGene are

$$b_K(M) = \left\lceil \log_2 \binom{K}{M} \right\rceil + M, \qquad R = \frac{\sum_{t\in\mathcal{J}} \sum_{i=0}^{n} b_K(M_{i,t}) + \sum_{i=1}^{n} b(C_i^{\text{mask}})}{(n+1)HW} + R_{\text{aux}}, \tag{11}$$

where $b_K(M)$ is the number of bits for one coded direction, and $\mathcal{J}$ is the set of steps that transmit direction choices. The term $b(C_i^{\text{mask}})$ denotes the bit length of the mask bitstream for the i-th edit. The auxiliary rate R_{aux} is the total number of bits for the source atom-count schedule, headers and byte padding, divided by $(n+1)HW$ (see Appendix A.1 for details). A fixed atom count across branches and coded steps cannot account for differences in edit-region size or coding effectiveness over denoising. RSBA therefore adapts $M_{i,t}$ to both factors.

Region-aware allocation. We scale each edit branch's direction-bit budget with the spatial extent of its entity changes. Let A_i^{bin} denote the decoded binary edit mask at image resolution from which the soft mask A_i is constructed, and let a_i be its normalized area fraction. For a non-empty edit region, we approximate its effective latent dimension by $a_i d$, where d is the full latent dimension. Assuming comparable coding difficulty per effective dimension, matching the bits per dimension between the edit and source branches gives

$$a_i = \frac{\|A_i^{\text{bin}}\|_1}{HW}, \qquad \frac{b_K(M_{i,t})}{a_i d} \approx \frac{b_K(M_{0,t})}{d} \quad \Longrightarrow \quad b_K(M_{i,t}) \approx a_i b_K(M_{0,t}). \tag{12}$$

We implement this area-based budget through the smallest admissible integer atom count:

$$M_{i,t} = \min\{m \in \{1,\dots,M_{0,t}\} \mid b_K(m) \geq a_i b_K(M_{0,t})\}, \tag{13}$$

where m is a candidate atom count and $M_{0,t}$ is the source allocation at the same step.

Step-aware allocation. Motivated by the empirical benefit of favoring later coded steps, we initialize a non-uniform allocation under a fixed source budget. Let $J = |\mathcal{J}|$ be the number of coded steps, and $j = 1, \ldots, J$ index them in reconstruction order, from higher to lower noise. We initialize the sequence using a cumulative spending limit

$$U_j = \left\lfloor B_{\mathrm{src}} \left(\frac{j}{J} \right)^p \right\rfloor, \tag{14}$$

where B_{src} is the source direction-bit budget after accounting for its auxiliary information, U_j limits the total direction bits spent through step j, and $p > 1$ shifts capacity toward later coded steps (see Appendix A.2 for details). At each step, we choose an atom count within the available budget by minimizing the error between the next clean-latent prediction and the source target latent.

We then refine the sequence by transferring bit capacity between coded steps. Each candidate sequence is evaluated through the complete reconstruction, and an update is accepted only if it reduces the mean squared error between the final source latent $z_0^{(0)}$ and its target $z_{\mathrm{tar}}^{(0)}$ without exceeding the budget. Thus, the initialization favors later steps, but the refined sequence is not constrained to be monotonic. The edit sequences follow from Equation 13. For the final $T - J$ steps, we use deterministic DDIM sampling without transmitting direction bits to improve perceptual quality at low bitrates (Vaisman et al., 2026).

3.5 ImaGeneSet: A Dataset for ImaGene Compression

Existing image-editing datasets (Ye et al., 2025; Qian et al., 2026; Chen et al., 2026a; Wang et al., 2025) contain JPEG artifacts or substantial changes to non-target content, limiting their suitability for joint compression evaluation. We introduce **ImaGeneSet**, comprising 65 clusters in total, each containing four images: one source from Kodak or the CLIC2020 Professional validation set, and three edited variants covering entity addition, removal, and modification. We generate each variant independently from the source using Seedream 5.0 Pro (ByteDance Seed, 2026) and manually screen for instruction adherence, visual plausibility, and preservation of non-target content. Dataset construction details are provided in Appendix A.1.

4 Experiments

4.1 Experimental Setup

Implementation details. We use frozen Stable Diffusion 2.1-base and its pretrained VAE, with $T = 30$ denoising steps and a codebook of $K = 16{,}384$ atoms. Entity localization combines semantic masks with RGB-difference cues. Coding budgets, sampling settings, and mask-processing parameters are detailed in Appendix A.1.

Evaluation protocol. We evaluate the Kodak and CLIC (resized to 512×512) subsets separately at their respective resolutions. We report five full-reference (FR) metrics: MS-SSIM (Wang et al., 2003), LPIPS (Zhang et al., 2018a) with the VGG backbone, DISTS (Ding et al., 2022), AHIQ (Lao et al., 2022), and TOPIQ-FR (Chen et al., 2024), together with the no-reference (NR) metric LIQE (Zhang et al., 2023a). All six metrics are averaged over all source images and edited variants. For EnDiC, bpp is the total size of one source stream, all edit streams, and masks, including the source atom-count schedule, headers, and byte padding, divided by the total pixels in the cluster. Baseline rate accounting is detailed in Appendix A.1.

Compared methods. We compare EnDiC with the conventional codec BPG (Bellard, 2014); end-to-end learned image codecs FTIC (Li et al., 2024) and GLIC (Chen et al., 2026b); and generative image codecs MS-ILLM (Muckley et al., 2023), PerCo (SD) (Körber et al., 2024), DiffC (Vonderfecht & Liu, 2025), DiffEIC (Li et al., 2025), StableCodec (Zhang et al., 2025), and Turbo-DDCM (Vaisman et al., 2026). Among the diffusion-based baselines, DiffC and Turbo-DDCM are training-free. All single-image codecs encode the four images independently. For video compression, we include the end-to-end learned codec DCVC-UF (Li et al., 2026) and the generative codec GVC1D (Zheng et al., 2026), treating each cluster as a four-frame sequence. For all learned baselines, we use their publicly released pretrained models without fine-tuning on ImaGeneSet.

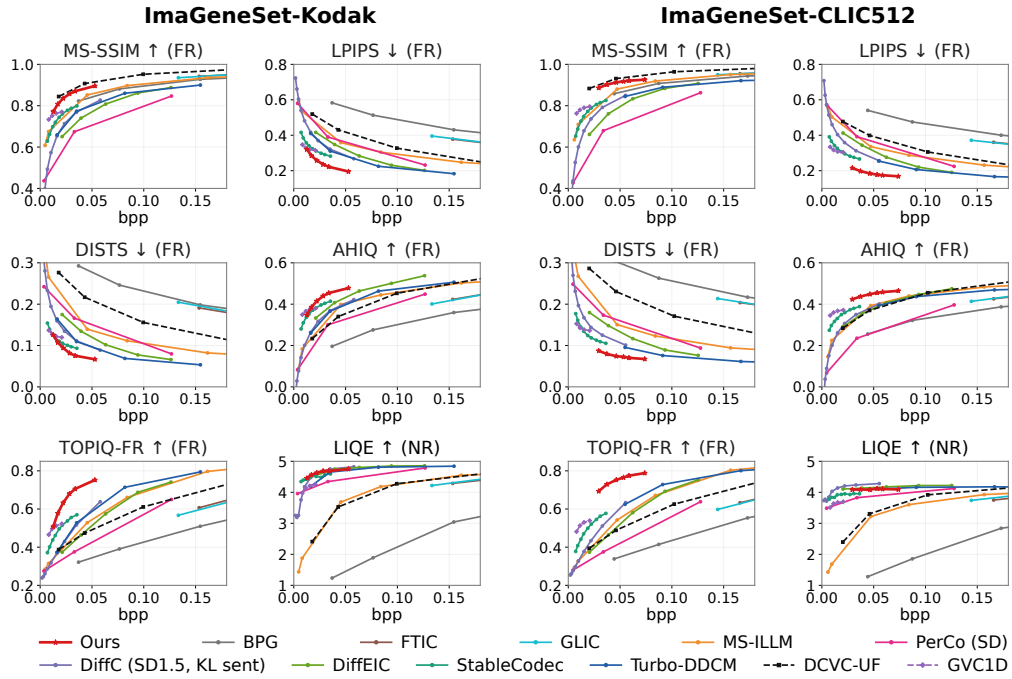

Figure 5: Rate-distortion and perceptual quality comparisons on ImaGeneSet-Kodak (left) and ImaGeneSet-CLIC512 (right). Red curves denote EnDiC. FR and NR denote full-reference and no-reference metrics, respectively; LPIPS uses LPIPS-V with VGG backbone. Rates and metrics cover all four images per cluster. Bitrate accounting is detailed in Appendix A.1.

4.2 Main Results

Quantitative results. Figure 5 shows that EnDiC achieves strong perceptual performance at low bitrates on both subsets, with favorable LPIPS-VGG, DISTS, AHIQ, and TOPIQ-FR relative to independently applied diffusion codecs and competitive LIQE. DCVC-UF achieves higher MS-SSIM, highlighting the different tradeoffs between structural fidelity and perceptual quality.

Qualitative results. Figure 6 shows that EnDiC preserves fine texture and details, whereas GLIC and DCVC-UF produce smoother reconstructions despite using higher bitrates. These details help retain the distinctive appearance of the source and its edited variants. Compared with DiffC, DiffEIC, and Turbo-DDCM, EnDiC achieves competitive visual quality at a lower bitrate, illustrating the benefit of reusing shared content while allocating coding capacity to entity changes. See Appendix A.5 for more qualitative results. The decoded variants and masks also support decoder-side edit fusion without additional bits or diffusion sampling. Examples and the composition rule are provided in Appendix A.6.

4.3 Ablation Study

Effect of joint compression. Figures 5 and 6 show that EnDiC improves perceptual quality over independently applied Turbo-DDCM at comparable or lower bitrates, supporting the effectiveness of jointly coding shared content and entity changes. We also compare against transmitting image residuals and separately compressing cropped edit regions for pasting back at the decoder. EnDiC achieves better perceptual rate–quality tradeoffs than these alternatives at low bitrates, supporting the benefit of integrating entity changes during denoising (see Appendix A.4 for details).

Effect of RSBA. Figure 7 compares RSBA with uniform atom allocation across regions, denoising steps, or both. Full RSBA achieves higher PSNR and lower LPIPS at comparable bitrates. Region-aware allocation adapts coding capacity to the spatial extent of each edit, while step-aware allocation distributes it across denoising stages. Both components contribute to efficient use of budget.

Effect of entity-change localization. Figure 8 shows that combining semantic and RGB-difference masks yields better quantitative results than either alone at comparable bitrates. Semantic

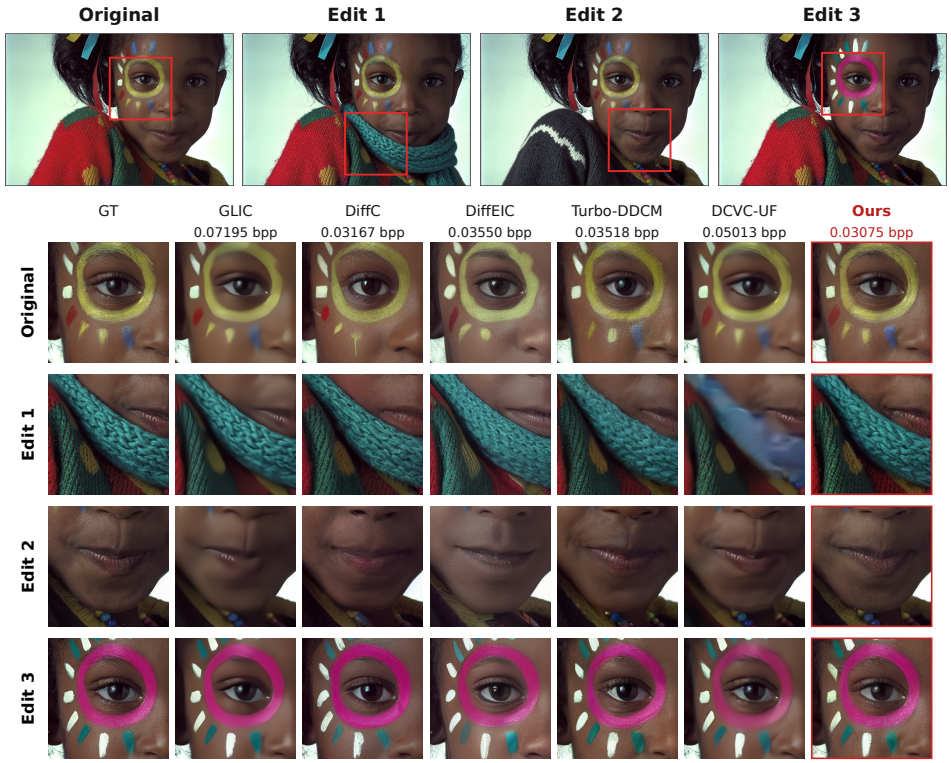

Figure 6: Qualitative comparison on an ImaGene from ImaGeneSet-Kodak. Top: the source image and three edited variants, with red boxes marking the regions enlarged below. Bottom: ground-truth (GT) crops and reconstructions from competing codecs and EnDiC (Ours). The reported bpp is the average rate over the entire ImaGene, computed as the total coded bits divided by the total pixels.

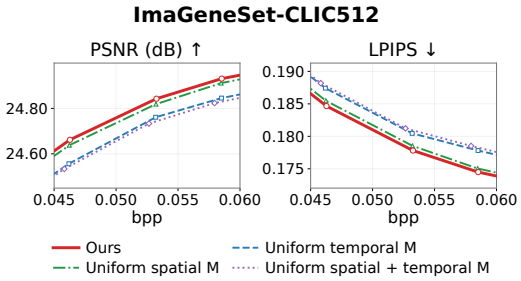

Figure 7: Ablation of RSBA on ImaGeneSet-CLIC512. Uniform spatial and temporal M disable region-aware and step-aware atom allocation, respectively. LPIPS uses LPIPS-V.

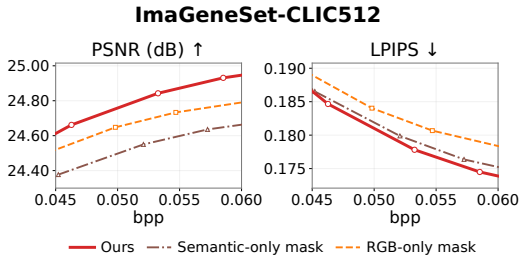

Figure 8: Ablation of the mask type used to derive A_i on ImaGeneSet-CLIC512. EnDiC uses Entity Change Localization that combines both semantic and RGB-difference masks; the alternatives use either alone. LPIPS uses LPIPS-V.

localization captures the affected entities in both source and edited images, while RGB differences supplement pixel-level changes.

5 Conclusion

We introduce ImaGene Compression, a paradigm for jointly compressing a source image and its generatively edited variants through shared visual entities and their variations. Our training-free codec EnDiC reuses source diffusion trajectories, encodes entity changes as incremental direction choices, and adapts bit allocation across regions and denoising steps. Experiments on the proposed ImaGeneSet demonstrate improved perceptual quality over independent diffusion codecs at low bitrates. Future work will extend joint compression to more general edits, where content outside the intended edit regions may also change substantially.

AI USE STATEMENT

Generative AI models were used both as components of our research pipeline and as tools supporting the research process. We used Seedream 5.0 Pro to generate the edited images in ImaGeneSet, with Qwen assisting in entity identification and editing-instruction construction. In Entity Change Localization, Qwen infers edit descriptions and identifies affected entities to guide segmentation. We also used generative AI tools to support methodological and experimental discussions, interpretation of results, manuscript revision, figure preparation, and code development. All AI-assisted content and outputs incorporated into this work were carefully reviewed and verified by the authors. The authors retain full responsibility for the final manuscript, scientific claims, code, dataset, and other research artifacts.

REPRODUCIBILITY STATEMENT

The problem formulation, codec, entity localization, and bit-allocation procedure are described in Sections 3.1–3.4. Section 3.5 describes ImaGeneSet, and Section 4.1 specifies the baselines, evaluation metrics, and rate-accounting protocol. Appendix A.1 provides further details on dataset construction and screening, pretrained models, sampling settings, coding budgets, and mask processing. Quantitative comparisons across bitrates and datasets, together with component ablations, are reported in Sections 4.2 and 4.3 to support assessment of the results. The ImaGeneSet dataset and implementation code, including configurations needed to reproduce the experiments, will be made publicly available after publication.

REFERENCES

Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. End-to-end optimized image compression. In *Proc. Int'l Conf. Learning Representations*, 2017.

Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *Proc. Int'l Conf. Learning Representations*, 2018.

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

Sebastian Bosse, Dominique Maniry, Klaus-Robert Müller, Thomas Wiegand, and Wojciech Samek. Deep neural networks for no-reference and full-reference image quality assessment. *IEEE Trans. on Image Processing*, 27(1):206–219, 2018. doi: 10.1109/TIP.2017.2760518.

ByteDance Seed. Beyond generation, it understands design: Introducing Seedream 5.0 Pro. `https://seed.bytedance.com/en/blog/beyond-generation-it-understands-design-introducing-seedream-5-0-pro`, 2026.

Shengcao Cao, Jiuxiang Gu, Jason Kuen, Hao Tan, Ruiyi Zhang, Handong Zhao, Ani Nenkova, Liang-Yan Gui, Tong Sun, and Yu-Xiong Wang. SOHES: Self-supervised open-world hierarchical entity segmentation. In *Proc. Int'l Conf. Learning Representations*, 2024.

Marlène Careil, Matthew J. Muckley, Jakob Verbeek, and Stéphane Lathuilière. Towards image compression with perfect realism at ultra-low bitrates. In *Proc. Int'l Conf. Learning Representations*, 2024.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. SAM 3: Segment anything with concepts. In *Proc. Int'l Conf. Learning Representations*, 2026.

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 Trans. on Image Processing*, 2024.

Tianyu Chen, Yasi Zhang, Zhi Zhang, Peiyu Yu, Shu Wang, Zhendong Wang, Kevin Lin, Xiaofei Wang, Zhengyuan Yang, Linjie Li, et al. EdiVal-Agent: An object-centric framework for automated, fine-grained evaluation of multi-turn editing. In *Proc. Int'l Conf. Learning Representations*, 2026a.

Yunuo Chen, Bing He, Zezheng Lyu, Hongwei Hu, Qunshan Gu, Yuan Tian, and Guo Lu. Adaptive learned image compression with graph neural networks. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2026b.

Keyan Ding, Kede Ma, Shiqi Wang, and Eero P. Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 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. In *Proc. Annu. Conf. Neural Inf. Process. Systems*, 2017.

Kosmas Karadimitriou and John M. Tyler. The centroid method for compressing sets of similar images. *Pattern Recognition Letters*, 1998.

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

Shanshan Lao, Yuan Gong, Shuwei Shi, Sidi Yang, Tianhe Wu, Jiahao Wang, Weihao Xia, and Yujiu Yang. Attentions help CNNs see better: Attention-based hybrid image quality assessment network. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition Workshops*, 2022.

Han Li, Shaohui Li, Wenrui Dai, Chenglin Li, Junni Zou, and Hongkai Xiong. Frequency-aware transformer for learned image compression. In *Proc. Int'l Conf. Learning Representations*, 2024.

Jiahao Li, Wenxuan Xie, Zhaoyang Jia, Bin Li, Zongyu Guo, Xiaoyi Zhang, and Yan Lu. Ultra-fast neural video compression. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2026.

Zhiyuan Li, Yanhui Zhou, Hao Wei, Chenyang Ge, and Jingwen Jiang. Toward extreme image compression with latent feature guidance and diffusion prior. *IEEE Trans. on Circuits and Systems for Video Technology*, 2025.

Jinming Liu, Heming Sun, and Jiro Katto. Learned image compression with mixed Transformer-CNN architectures. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2023.

Fabian Mentzer, George D. Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. In *Proc. Annu. Conf. Neural Inf. Process. Systems*, 2020.

David Minnen, Johannes Ballé, and George D. Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *Proc. Annu. Conf. Neural Inf. Process. Systems*, 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 *Proc. Int'l Conf. Machine Learning*, 2023.

Guy Ohayon, Hila Manor, Tomer Michaeli, and Michael Elad. Compressed image generation with denoising diffusion codebook models. In *Proc. Int'l Conf. Machine Learning*, 2025.

OpenAI. GPT-Image-2. `https://developers.openai.com/api/docs/models/gpt-image-2`, 2026.

Lu Qi, Jason Kuen, Yi Wang, Jiuxiang Gu, Hengshuang Zhao, Philip Torr, Zhe Lin, and Jiaya Jia. Open world entity segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2023.

Yusu Qian, Eli Bocek-Rivele, Liangchen Song, Jialing Tong, Yinfei Yang, Jiasen Lu, Wenze Hu, and Zhe Gan. Pico-Banana-400K: A large-scale dataset for text-guided image editing. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2026.

Amit Vaisman, Guy Ohayon, Hila Manor, Michael Elad, and Tomer Michaeli. Turbo-DDCM: Fast and flexible zero-shot diffusion-based image compression. In *Proc. Int'l Conf. Learning Representations*, 2026.

Jeremy Vonderfecht and Feng Liu. Lossy compression with pretrained diffusion models. In *Proc. Int'l Conf. Learning Representations*, 2025.

Yuhan Wang, Siwei Yang, Bingchen Zhao, Letian Zhang, Qing Liu, Yuyin Zhou, and Cihang Xie. GPT-IMAGE-EDIT-1.5M: A million-scale, GPT-generated image dataset. *arXiv preprint arXiv:2507.21033*, 2025.

Zhou Wang, Eero P. Simoncelli, and Alan C. Bovik. Multiscale structural similarity for image quality assessment. In *Proc. Asilomar Conf. Signals, Systems and Computers*, 2003.

Siqi Wu, Yinda Chen, Dong Liu, and Zhihai He. Conditional latent coding with learnable synthesized reference for deep image compression. In *Proc. AAAI Conf. Artificial Intelligence*, 2025.

Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. ImgEdit: A unified image editing dataset and benchmark. In *Proc. Annu. Conf. Neural Inf. Process. Systems*, 2025.

Fanhu Zeng, Hao Tang, Yihua Shao, Siyu Chen, Ling Shao, and Yan Wang. MambaIC: State space models for high-performance learned image compression. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2025.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2018a.

Tianyu Zhang, Xin Luo, Li Li, and Dong Liu. StableCodec: Taming one-step diffusion for extreme image compression. In *Proc. IEEE Int'l Conf. Computer Vision*, 2025.

Weixia Zhang, Guangtao Zhai, Ying Wei, Xiaokang Yang, and Kede Ma. Blind image quality assessment via vision-language correspondence: A multitask learning perspective. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2023a.

Xiang Zhang, Jiarui Sun, Siwei Ma, Zhouchen Lin, Jian Zhang, Shiqi Wang, and Wen Gao. Globally variance-constrained sparse representation and its application in image set coding. *IEEE Trans. on Image Processing*, 2018b.

Xinjie Zhang, Jiawei Shao, and Jun Zhang. LDMIC: Learning-based distributed multi-view image coding. In *Proc. Int'l Conf. Learning Representations*, 2023b.

Zihan Zheng, Zhaoyang Jia, Naifu Xue, Jiahao Li, Bin Li, Zongyu Guo, Xiaoyi Zhang, Zhenghao Chen, Houqiang Li, and Yan Lu. Generative video compression with one-dimensional latent representation. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2026.

Hanwei Zhu, Baoliang Chen, Lingyu Zhu, Shiqi Wang, and Weisi Lin. DeepDC: Deep distance correlation as a perceptual image quality evaluator. *IEEE Trans. on Image Processing*, 34:7859–7873, 2025. doi: 10.1109/TIP.2025.3635025.

A Appendix

A.1 Implementation Details

Dataset construction. We construct ImaGeneSet from Kodak and the CLIC2020 Professional validation set. Adapting the ImgEdit workflow (Ye et al., 2025), we use Qwen to identify candidate visual entities and propose localized editing instructions covering entity addition, removal, and modification. Object detection and SAM 3 segmentation provide additional spatial references for entity-grounded edit planning. Each variant is generated independently from the source using Seedream 5.0 Pro (ByteDance Seed, 2026). Prompts specify the intended change while requesting preservation of the viewpoint, composition, and non-target content. We manually screen candidates for instruction adherence, visual plausibility, and preservation of non-target content, discarding or regenerating unsuitable results.

Codec configuration. We implement EnDiC in PyTorch using frozen Stable Diffusion 2.1-base and its pretrained VAE. We use $T = 30$ denoising steps and a codebook of $K = 16{,}384$ atoms, following the timestep schedule and deterministic DDIM tail of Turbo-DDCM (Vaisman et al., 2026). Seven nominal atom counts, 80, 105, 122, 140, 160, 180, and 240, define the source coding budgets. RSBA assigns the stepwise atom counts using $p = 1.3$.

Entity localization and mask processing. Entity localization uses Qwen3.8-Max and SAM 3 (Carion et al., 2026). We set the mask dilation radius to 4 image pixels and supplement the semantic mask with pixels whose maximum absolute RGB-channel difference exceeds 64 on the 8-bit scale. The resulting binary mask is downsampled by a factor of 8 using max pooling and losslessly encoded into C_i^{mask}. Both the EnDiC encoder and the EnDiC decoder decode this bitstream and restore the mask to image resolution by nearest-neighbor upsampling, yielding A_i^{bin}. We apply Gaussian smoothing with a standard deviation of 4 image pixels outside the marked region while retaining unit weights inside, then bilinearly resize the weights to the latent spatial grid to obtain the soft edit mask A_i. This shared construction ensures identical edit masks for direction selection and reference-state fusion.

Bitrate accounting. Equation 11 counts direction payloads and complete mask streams explicitly, while R_{aux} accounts for the source atom-count schedule, source/edit headers, and byte padding in bpp. The source stream stores a 20-byte header and each coded-step count $M_{0,t}$ as a 16-bit integer. Each edit stream has a 20-byte header; its counts $M_{i,t}$ are derived from the source schedule and decoded binary mask, so no edit schedule is transmitted. Each direction payload is padded to a byte boundary. Mask streams include an 8-byte shape header and the compressed binary payload. Reported EnDiC bpp counts all bytes of one source stream, three edit streams, and three mask streams over the total pixels of the four images.

GVC1D is an exception to actual-bitstream accounting: its official evaluation code reports estimated entropy rates rather than serialized bitstreams. We include the estimated rates of all four frames, including the first-frame feature and token rates. These estimates exclude container headers and finite-precision entropy-coding overhead, so GVC1D rates should be interpreted as estimated bpp.

A.2 Choice of the Temporal Allocation Exponent

We evaluate $p \in \{0.5, 0.7, 1, 1.3, 1.5\}$ in Equation 14 on both subsets at all seven nominal coding budgets. Each setting is run afresh with greedy initialization followed by refinement with an 80-second time limit; other codec settings are held fixed. The exponent controls only the cumulative budget limit during initialization: $p > 1$ favors later coded steps, while $p = 1$ gives a linear cumulative limit rather than uniform per-step atom counts. Figure 9 reports final reconstruction quality and source terminal latent MSE after refinement.

The results support $p = 1.3$ as a practical compromise between fidelity and perceptual quality. Compared with $p \leq 1$, it yields lower terminal latent MSE at every tested nominal budget on both subsets and higher PSNR at comparable actual bitrates. Increasing p to 1.5 gives similar PSNR and can further reduce latent MSE at higher budgets, especially on CLIC512, but does not consistently improve LPIPS. Conversely, $p = 1$ gives slightly better LPIPS on Kodak at the cost of lower PSNR. We therefore retain $p = 1.3$ as a common setting across both subsets, without claiming that it is optimal for every metric or bitrate. Refinement is time-limited, so candidate counts may differ; these single-run comparisons do not establish statistical significance for small differences.

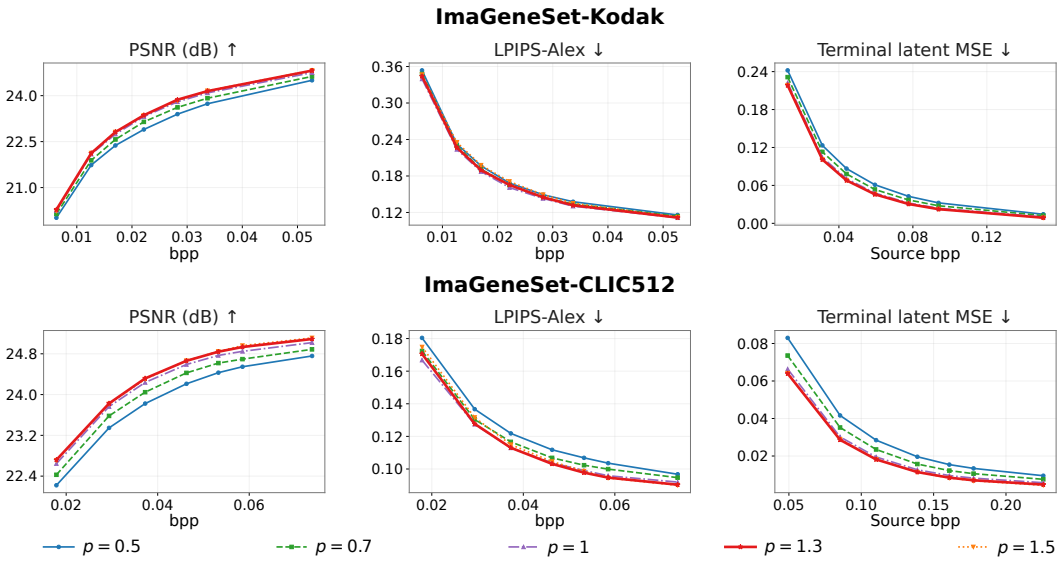

Figure 9: Sensitivity to the temporal allocation exponent p on ImaGeneSet-Kodak (top) and ImaGeneSet-CLIC512 (bottom). Columns show PSNR, LPIPS-Alex, and terminal source latent MSE after refinement. The first two columns use total cluster bpp, including the source, three edits, and masks, with quality averaged over all four images. The last column uses source-stream bpp. All seven measured coding budgets are retained for each exponent.

A.3 Additional Quantitative Results

Figure 10 complements the main-text evaluation with RGB PSNR, DeepDC (Zhu et al., 2025), WADIQAM-FR (Bosse et al., 2018), and Patch FID-64, which applies FID (Heusel et al., 2017) to non-overlapping 64×64 patches. The first three metrics are averaged over all source images and edited variants, whereas Patch FID-64 compares the pooled reference and reconstruction patch distributions within each subset. EnDiC achieves favorable DeepDC and WADIQAM-FR at low bitrates and competitive Patch FID-64 on both subsets. Although DCVC-UF achieves higher PSNR, its reconstructions exhibit blurred details in edited regions (Figure 6). Its higher LPIPS and DISTS scores relative to EnDiC at comparable bitrates (Figure 5) further indicate limitations in perceptual reconstruction quality.

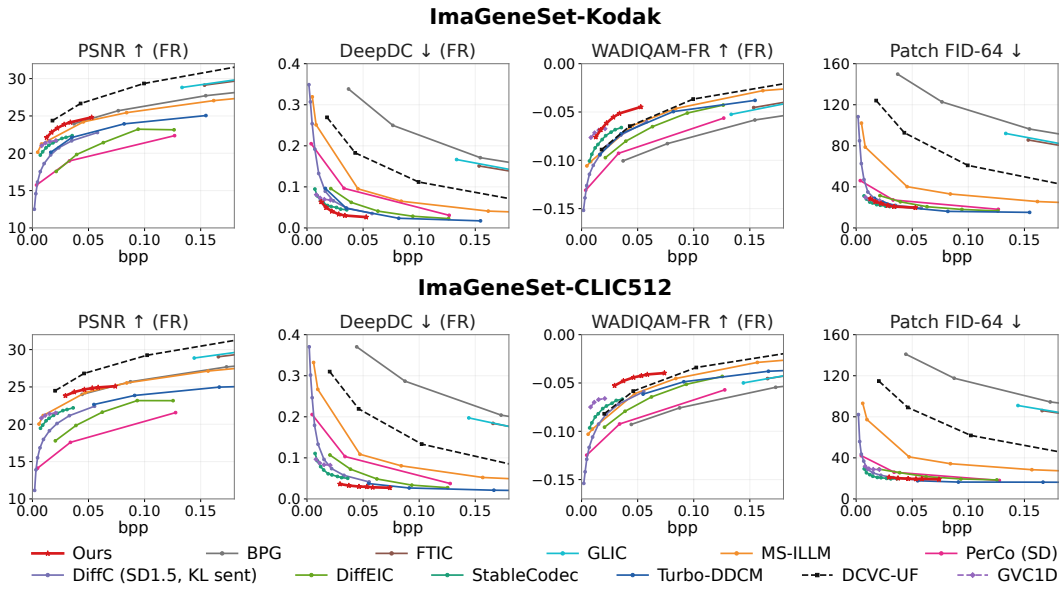

Figure 10: Additional rate-quality comparisons on ImaGeneSet-Kodak (top) and ImaGeneSet-CLIC512 (bottom). Red curves denote EnDiC. Arrows indicate the direction of improvement. WADIQAM-FR uses the PyIQA KADID-trained model and retains its raw scores. Patch FID-64 is a set-level distribution metric. Bitrates follow the main-text protocol and account for all four images in each cluster; GVC1D uses estimated bpp.

A.4 Comparison with Region and Residual Coding

We further compare EnDiC with two alternatives that reuse a decoded source image: separately coding cropped edit regions and coding pixelwise residuals. We apply each strategy to the nine single-image codecs from Section 4.1. Bitrates include the source stream, all edit streams, and required side information, normalized by the total pixels in each four-image cluster. LPIPS-VGG, DISTS, DeepDC, and TOPIQ-FR are evaluated on complete reconstructed images and averaged over all sources and edited variants. The red curves show the same EnDiC method in both comparisons.

Region coding. Figure 11 compares our method against encoding cropped edit regions and pasting the decoded regions onto the decoded source. On both subsets, EnDiC provides lower LPIPS-VGG, DISTS, and DeepDC and higher TOPIQ-FR at comparable low bitrates. This indicates that restricting transmission to edited regions alone does not match the perceptual coding efficiency of integrating entity changes during denoising.

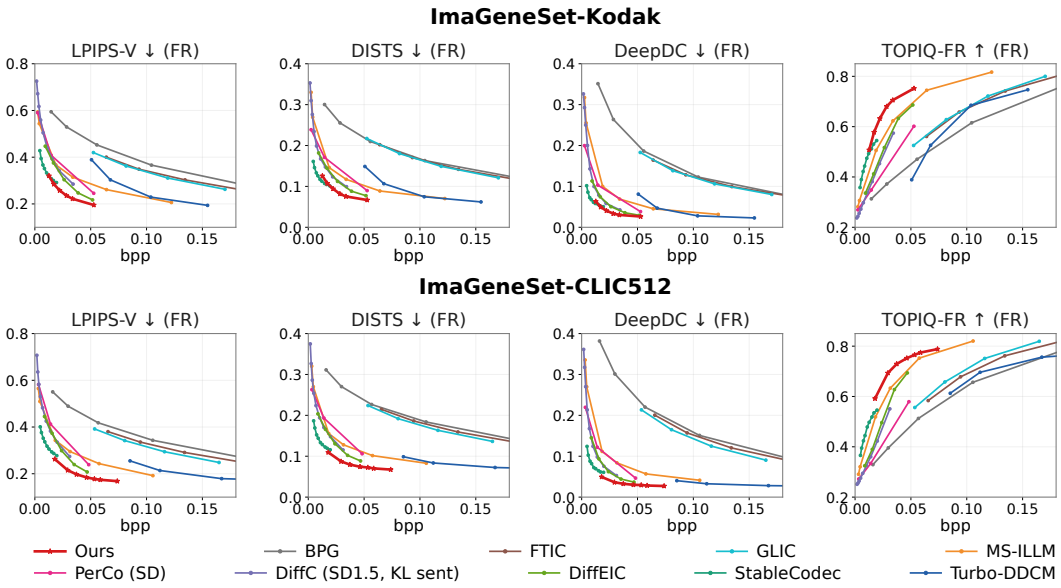

Figure 11: Comparison with cropped-region coding on ImaGeneSet-Kodak (top) and ImaGeneSet-CLIC512 (bottom). Baseline codecs encode edit crops for pasting onto the decoded source; red curves denote EnDiC. LPIPS-V uses the VGG backbone. All four metrics are full-reference and evaluate complete reconstructions; arrows indicate the direction of improvement.

Residual coding. Figure 12 compares our method against coding full-image pixelwise residuals relative to each codec's decoded source and adding the decoded residuals back at the decoder. EnDiC achieves better results across all four metrics at comparable low bitrates on both subsets. Together, these comparisons show that EnDiC's gains extend beyond avoiding repeated transmission of shared content: its integration of entity changes within the denoising process provides better perceptual rate-quality tradeoffs than directly coding either edit crops or pixelwise residuals.

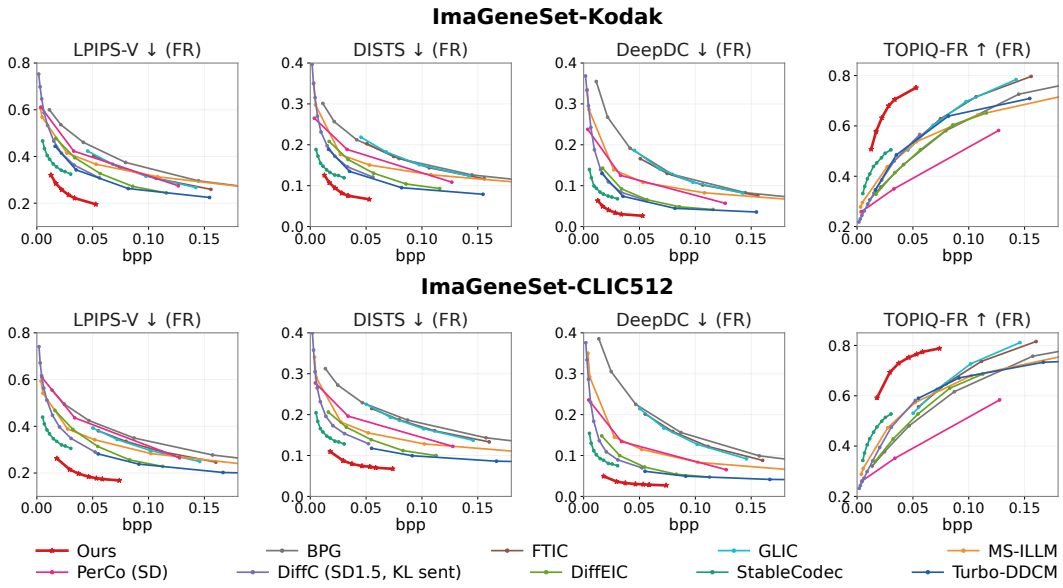

Figure 12: Comparison with pixelwise residual coding on ImaGeneSet-Kodak (top) and ImaGeneSet-CLIC512 (bottom). Baseline codecs transmit residuals relative to their decoded source; red curves denote EnDiC. LPIPS-V uses the VGG backbone. Metrics and bitrate accounting follow the same protocol as Figure 11.

A.5 Additional Qualitative Results

Figures 13 and 14 provide additional qualitative comparisons on ImaGeneSet-CLIC512. Each example shows a source image and three edited variants, with aligned crops for comparing texture and edited-object details. Reported bitrates are averaged over each four-image cluster, following Figure 6. Our method maintains high reconstruction quality even at low bitrates.



Figure 13: Qualitative comparison on the wood-texture example. Top: source and edited images with crop regions marked in red. Bottom: GT crops and reconstructions from competing codecs and EnDiC (Ours).

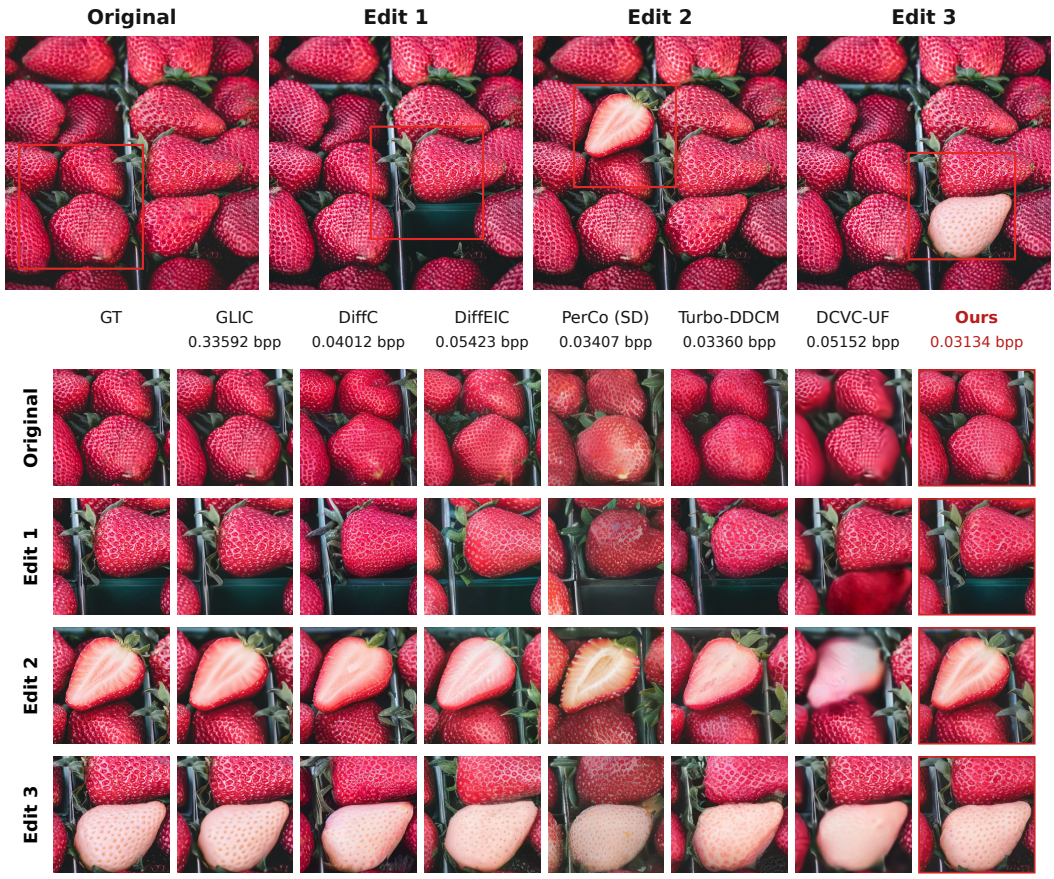

Figure 14: Qualitative comparison on the strawberry example.

A.6 Zero-Bitrate Edit Fusion

Figure 15 shows five examples of decoder-side edit fusion. The decoder takes each selected variant's content within its transmitted mask and retains the decoded source outside their union. At overlapping pixels, it selects the variant with the larger squared RGB difference from the decoded source, breaking ties by the lower edit index. This rule uses only already decoded information, so every fused result requires exactly zero additional bits.



Figure 15: Zero-bitrate edit fusion on ImaGeneSet-Kodak and ImaGeneSet-CLIC512. Rows identify the source image and selected edit indices; Edit A and Edit B follow the listed order. All images are decoded reconstructions. Edit rates include incremental streams and masks; $+0$ bpp denotes the additional cost of fusion after these have been received.