Rel-Zero: Harnessing Patch-Pair Invariance for Robust Zero-Watermarking Against AI Editing

Supplementary Material

A. PCA Analysis Between Generative Editing and VAE

ViT Feature Visualization. We conduct a PCA analysis to visualize how ViT patch embeddings change after two types of transformations: (1) generative editing including ControlNet-Inpainting, UltraEdit, Deterministic Regeneration, MagicBrush, and InstructPix2Pix and (2) Stable Diffusion VAE reconstruction. For each image, we extract patch embeddings before transformation and after either editing or VAE. All embeddings are projected onto the top principal components, and we plot the movement of each patch in this low-dimensional space.

The results show a clear pattern: the global displacement of ViT features induced by VAE reconstruction closely resembles the displacement caused by generative editing. In both cases, only a small subset of patches—typically those corresponding to drastically semantically modified regions—exhibits noticeable shifts, while the majority of background patches remain tightly clustered and nearly unchanged. Moreover, the direction and magnitude of patch movements follow similar trajectories for Edit and VAE.

This strong resemblance indicates that VAE reconstruction preserves the structural behavior of ViT embeddings under real generative edits, despite being a much simpler and deterministic transformation. Therefore, using VAE-induced feature perturbations as a surrogate for analyzing edit-induced changes is both meaningful and well-justified.

B. External Storage Encryption of Relational Zero-Watermark

To protect the relational zero-watermark $\mathcal{E}_p$, which contains only patch-pair indices but may still leak structural information, we apply a key-controlled permutation encryption. The index set is first converted into a binary indicator vector $\mathbf{b} \in \{0,1\}^M$, where $M = \binom{P}{2}$. Reshaping $\mathbf{b}$ into an $N \times N$ grid ($N^2 = M$) produces a two-dimensional watermark map suitable for keyed permutation.

We use a secret-key Arnold transform to scramble the spatial layout. Each coordinate (x, y) in the map is permuted to (x', y') according to:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & p \\ q & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \mod N, \tag{1}$$

where (p, q, N) jointly define the secret key K. Repeating this transform for T iterations yields the final encrypted watermark:

$$\mathbf{Z}_W = \text{Arnold}^T(\mathbf{b}; K). \tag{2}$$

The encrypted $\mathbf{Z}_W$ does not reveal the patch relations or the underlying image content, and cannot be inverted without the secret key. It can thus be safely stored externally (e.g., hashed or registered in a protected database) and later decrypted using the same key during verification.

C. Time and Memory Cost

Our method further benefits from high computational efficiency. Embedding-based watermarking typically requires running generative models or diffusion sampling during embedding or extraction, resulting in large runtime and memory overhead. Existing zero-watermarking methods also involve heavy CNN backbones or reconstruction modules.

A detailed comparison is shown in Table 1. Rel-Zero avoids these costs entirely. All operations of patch embedding and relational comparison are implemented as a single feed-forward pass with lightweight tensor broadcasts. Consequently, Rel-Zero achieves a total extraction time of **0.3 ms** and a memory cost of only **0.3 GB**, outperforming prior zero-watermarking approaches by an order of magnitude and embedding-based schemes by more than two orders of magnitude.

Table 1. Runtime and memory comparison. "Total Time" measures end-to-end watermark extraction per image (or embedding+extraction for embedding-based methods). CPU-only implementations (e.g., DWT-DCT and FGPCET) report no GPU memory usage and are omitted for fairness.

Method	Total Time (s)	Memory (GB)
DWT-DCT [1]	–	–
Robust-Wide [14]	0.0268	3.1
VINE [24]	0.0674	5.2
ConZWNet [32]	0.0013	2.3
FGPCET [38]	-	-
Rel-Zero (Ours)	0.0003	1.5

D. Additional Robustness Experiments

To further evaluate the stability of relational cues under low-level corruptions, we compare Rel-Zero and baseline methods under three categories of common distortions: (1) **Salt&Pepper noise**, applied with corruption probabilities p=0.01 and p=0.03; (2) **JPEG compression**, evaluated

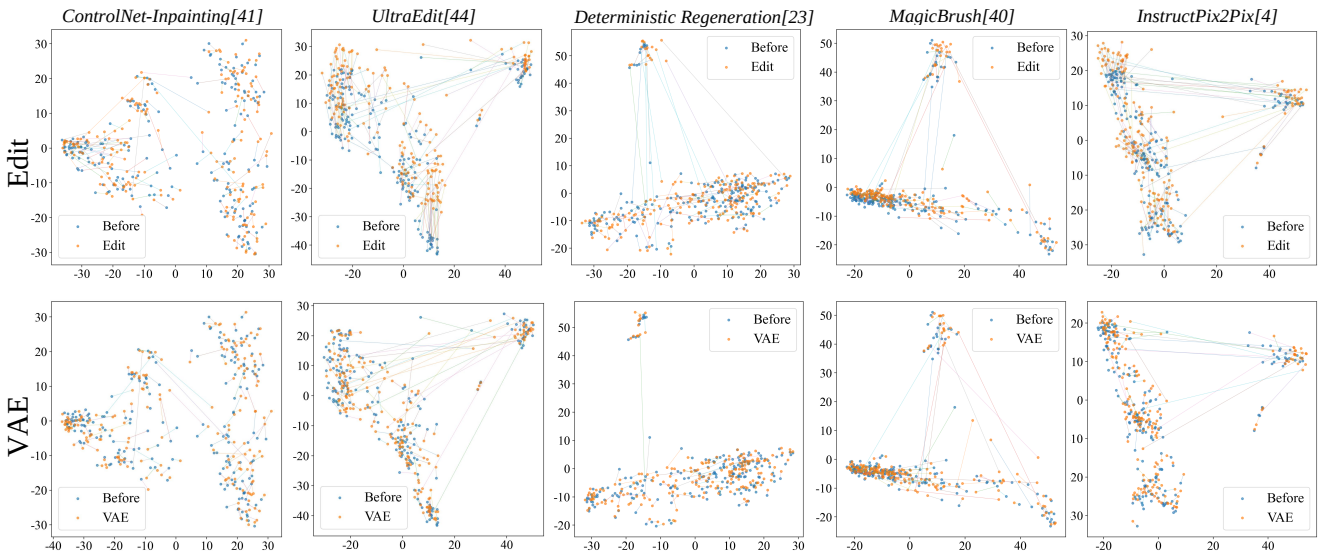

Figure 1. PCA visualization of ViT patch embeddings before and after generative editing or VAE reconstruction. Both transformations cause similar displacement patterns: only patches in drastically semantically edited regions move noticeably, while most patches remain stable. The similarity of these trajectories demonstrates that VAE reconstruction effectively mimics edit-induced feature changes.

Table 2. Robustness under common distortions measured by TPR@(0.1%FPR). We report results under Salt&Pepper noise, JPEG compression, and small-angle rotation.

Method	Salt & Pepper		JPEG		Rotation	
	S&P-1	S&P-2	Q = 90	Q = 50	Rot-3°	Rot-5°
Embedding Watermarking						
DWT-DCT [1]	85.56	80.21	68.53	70.21	0.02	0.05
Robust-Wide [14]	98.89	98.55	93.55	95.76	2.43	3.56
VINE [24]	100.00	100.00	99.45	99.56	5.43	6.78
Zero Watermarking						
ConZWNet [32]	99.11	99.59	98.56	99.28	98.13	96.45
FGPCET [38]	99.06	98.65	98.79	99.19	99.98	99.66
Rel-Zero	100.00	99.86	98.28	99.00	95.54	90.21

at quality factors Q=90 (mild compression) and Q=50 (strong compression); and (3) **Rotation**, using angle perturbations of 3° and 5°. All distortions are applied directly to the input image without any pre-alignment or post-processing.

Across all distortion types and strengths, Rel-Zero consistently maintains high robustness. These results confirm that relational differences between patch pairs remain highly stable even under pixel-level perturbations, and that the relational design provides strong inherent robustness without requiring additional denoising, error correction, or adversarial training.

E. ViT Embedding Stability Under Generative Editing

The core premise of our method is that modern generative editing models (InstructPix2Pix, UltraEdit, etc.) tend to preserve the *relational geometry* of deep features, even when the edited image exhibits large visual or semantic changes. In this section, we provide additional evidence showing that the **pairwise relationships between ViT patch features remain stable** after editing, analogous to the relational behavior previously observed in RGB vectors. This structural stability directly justifies our use of patch–pair distance of ViT features as reliable watermark carriers.

Self-Similarity Matrix (SSM) Stability for ViT Features. Given a ViT feature map $F = \{f_i\}_{i=1}^{N}$ extracted from an image, we compute the self-similarity matrix:

$$M(i,j) = \|f_i - f_j\|_2.$$

Figure 3 visualizes M_{before} and M_{after} for the same image before and after generative editing. Despite significant local modifications (e.g., object replacement or style alteration), the global structure of M remains nearly unchanged, up to a global scaling factor α that we estimate via linear regression. The residual matrix

$$|M_{\text{after}} - M_{\text{before}}|$$

remains close to zero across most regions, demonstrating that *the intrinsic geometric layout of patch features is preserved*. This mirrors the behavior previously observed in

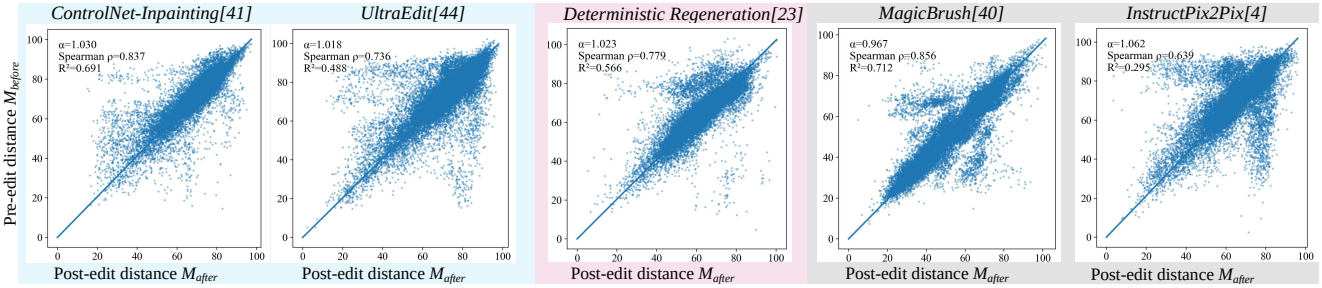

Figure 2. **Correlation between patch–pair ViT features distance before and after editing.** Distance aligns strongly with a fitted linear model $M_{\text{after}} \approx \alpha\, M_{\text{before}}$, confirming that ViT patch–pair relationships remain stable and support our relational watermark extraction.

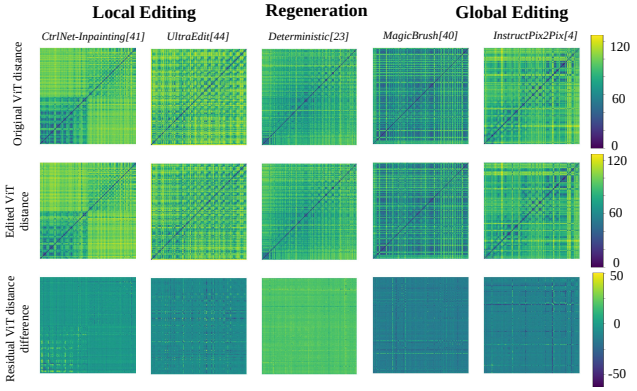

Figure 3. **ViT Self-Similarity Matrix stability under editing.** First row: SSM before editing. Second row: SSM after editing. Right: Residual $|M_{\text{after}} - M_{\text{before}}|$. The near-zero residual indicates that the relational geometry of ViT patch features is preserved up to a global scale.

pixel-space (RGB) self-similarity patterns, but the effect is more pronounced in deep representations due to their semantic stability.

Patch–Pair Distance Correlation. To quantify the consistency of feature geometry, we also plot the pairwise condensed distance $M(i,j) = \|f_i - f_j\|_2$ before and after editing in a scatter diagram (Figure 2). The majority of the scatter points lie tightly along a line

$$M_{\text{after}} \approx \alpha\, M_{\text{before}},$$

achieving high Pearson correlation and high coefficient of determination. This strong linear relationship indicates that editing operations largely preserve the *relative* differences between patch features, even if absolute feature values shift.

Implications for Relational Watermarking. Together, the SSM visualizations and distance–correlation scatterplots demonstrate that ViT feature geometry is **stable under a wide range of generative edits**, same as the RGB vectors in Sec 3.1. Since our watermark is based on the ordering and relationships of patch–pair distances, and these relationships remain invariant up to scale, the feature relational structure survives editing. These results provide strong empirical support for the theoretical basis of Rel-Zero and explain why our method remains robust across global and local edits.

F. TPR with a Fixed FPR

We treat all embedding-based watermarking methods as single-bit schemes, with a embedded watermark $s \in \{0,1\}^k$. A predefined threshold $\tau \in [0,k]$, is used for detection. If the similarity score $\text{Acc}(s,s')$ between the original watermark s and the extracted one s' surpasses or equals τ, the image is deemed as watermarked.

According to prior work [39], it is generally assumed that the extracted bits $s'_1, \ldots, s'_k$ from unmarked images follow an independent and identically distributed Bernoulli process with success probability 0.5. Under this assumption, the similarity metric $\text{Acc}(s,s')$ conforms to a binomial distribution with parameters $(k, 0.5)$.

Once this distribution is known, the false positive rate (FPR) corresponds to the likelihood that a non-watermarked image still achieves a score above the threshold τ. This can be formally represented using the regularized incomplete beta function $B_x(a,b)$ as follows:

$$\begin{aligned}\text{FPR}(\tau) &= P(\text{Acc}(s,s') > \tau) = \sum_{i=\tau+1}^{k} \binom{k}{i} \left(\frac{1}{2}\right)^k \\ &= B_{1/2}(\tau+1, k-\tau).\end{aligned} \tag{3}$$

For zero-watermarking method, we treat our relational watermarking scheme as a binary detection problem, where a fixed relational index set $\mathcal{E}_p$ (consisting of K patch–pair edges) serves as the watermark signature. During verification, the detector extracts a candidate edge set $\mathcal{E}'_p$ from the query image and computes the similarity score

$$\eta = \frac{|\mathcal{E}_p \cap \mathcal{E}'_p|}{K}. \tag{4}$$

A detection threshold $\tau \in [0,1]$ is predetermined, and the image is deemed watermarked if the overlap ratio η reaches or exceeds this threshold.

Following common practice in watermark detection, we model the prediction of watermark edges from clean (unrelated) images as random and independent activations. That is, each true watermark edge in $\mathcal{E}_p$ is falsely activated with a small probability 0.5, and the K detection outcomes are assumed to follow a binomial distribution with parameters $(K, 0.5)$.

Once this distribution is established, the false positive rate (FPR) corresponds to the probability that a clean image still produces an overlap ratio above the threshold τ:

$$\mathrm{FPR}(\tau) = P(\eta \geq \tau) = P(X \geq \tau K) = \sum_{i=\lceil \tau K \rceil}^{K} \binom{K}{i} p^i (1-p)^{K-i}. \tag{5}$$

In our evaluation, we maintain the FPR at 0.1% to determine the corresponding operating threshold τ, and subsequently report the true positive rate (TPR) computed over $1{,}000$ watermarked images. As shown in Table 1, at a fixed FPR of 10^{-3} our relational watermarking method achieves strong TPR and reliable detection performance.

G. AI-Editing Setting

For deterministic regeneration, we employ the fast sampler DPM-Solver [23] and evaluate with a sampling step setting of $n_d = 25$.

We use each model's default sampler and perform 50 sampling steps to generate edited images. For global editing, the difficulty level is controlled by the classifier-free guidance scale of the text prompt [13], which we set to 8, while fixing the image guidance scale to 1.5. For local editing, difficulty is determined by the ratio of the edited region to the entire image, controlled through the region mask size with intervals of 10–20%, 20–30%, 30–40%, 40–50%, and 50–60%. Across all local editing settings, the image and text guidance scales are fixed at 1.5 and 7.5, respectively.

H. Visual Comparison of Embedding-Based and Zero-Watermarking Methods

To qualitatively assess the perceptual impact of different watermarking paradigms, we provide a visual comparison consisting of the original image, the embedding-based watermarked image, its corresponding residues, and the visualization of zero-watermarking methods (Fig. 4).

Embedding-based watermarking inevitably introduces pixel-level perturbations to encode information. Although these perturbations may appear imperceptible in the watermarked image, they accumulate into clear artifacts in the residue map, revealing the underlying distortion injected into the visual content.

In contrast, our zero-watermarking approach requires *no* modification to the input image. The extracted relational structure is purely feature-driven and leaves the pixel space entirely untouched. Consequently, it produces no observable residue, highlighting the key advantage of zero-watermarking: robust verification without introducing any noise or degradation to the perceptual quality.

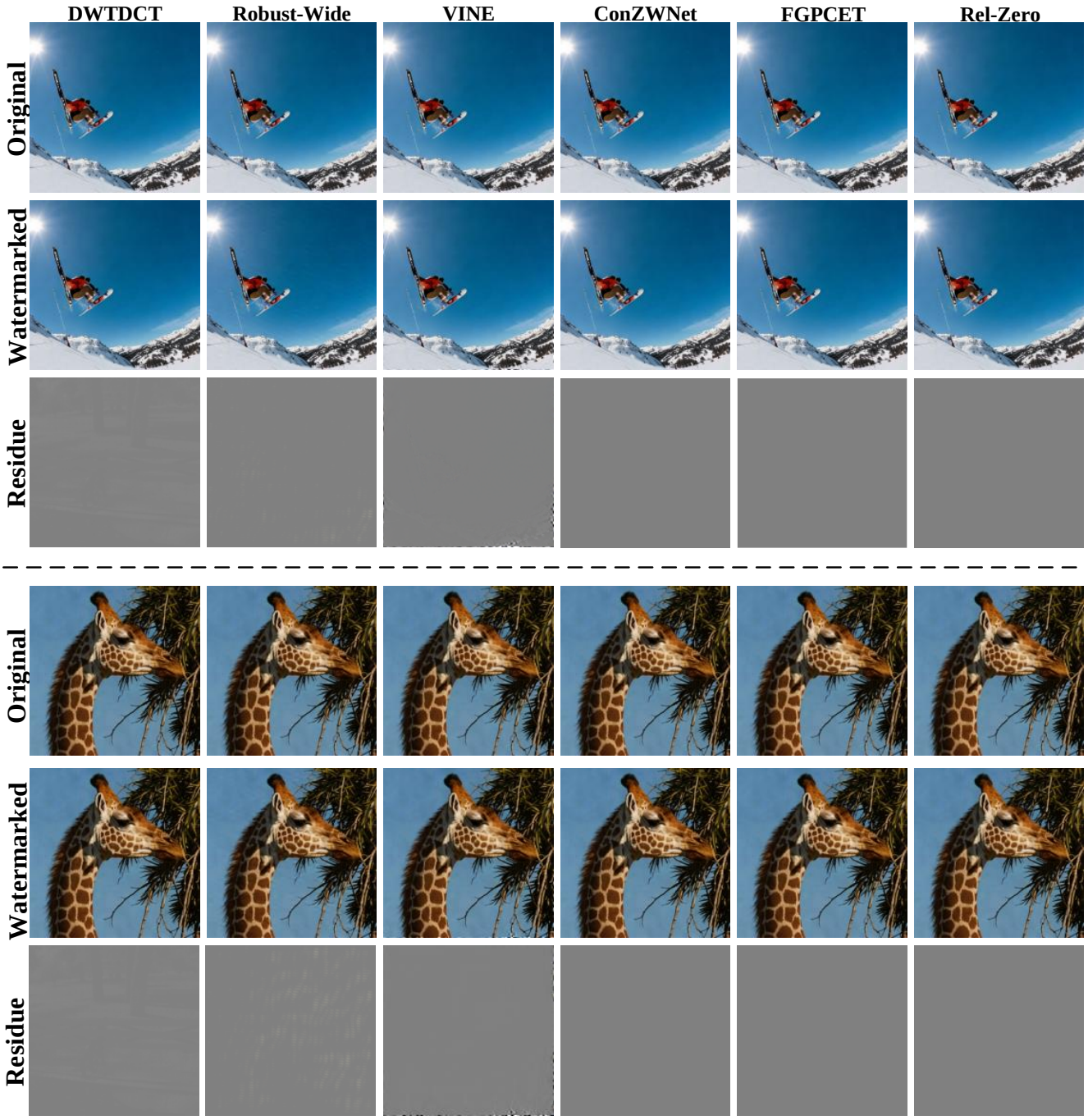

Figure 4. **Visual comparison of embedding-based watermarking and zero-watermarking.** From left to right: original image, embedding-based watermarked image, residue highlighting injected perturbations, and our zero-watermarking visualization. Unlike embedding-based methods, zero-watermarking introduces *no pixel-level noise*, preserving perfect perceptual fidelity. Zoom in to see the detailed difference.