SAEMARK: ARCHITECTURE-AGNOSTIC IMAGE WATERMARKING VIA SPARSE AUTOENCODER REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Watermarking diffusion models is essential for tracing the provenance and enabling the trustworthy authentication of generated content. However, existing methods typically rely on architecture-specific generation or inversion procedures, limiting their applicability to emerging generative paradigms such as flow matching. We observe that Sparse Autoencoders (SAEs) can independently learn internal representations for different generative architectures, yielding structured and semantically rich feature spaces that provide a new substrate for decoupling watermarking from the underlying generation process. Based on this insight, we propose SAEMark, the first SAE-based watermarking framework for diffusion models, which directly embeds and extracts watermarks in the SAE feature space, thereby decoupling watermarking from the specific generation process. This representation-level design enables a unified watermarking mechanism across heterogeneous generative architectures while substantially improving robustness against watermark removal and image manipulation attacks. We conduct experiments across multiple architectures, covering 8 representative watermarking methods and 9 watermark removal attacks. The results show that SAEMark achieves leading watermark robustness while preserving high generation quality, demonstrating the potential of SAE representations as a unified substrate for robust and architecture-agnostic watermarking across generative architectures. Code is available at `https://anonymous.4open.science/r/SAEMark-exp-C3E0/`.

1 INTRODUCTION

Image generation models have evolved rapidly in recent years, expanding from diffusion models (Rombach et al., 2022) to diverse generative paradigms such as flow matching (Lipman et al., 2022). As these models become increasingly prevalent in visual content creation, verifying the provenance of generated content and preventing the misuse of generative models for disinformation and inappropriate content have become important concerns. Image watermarking (Zhu et al., 2018; Wen et al., 2023; Gesny et al., 2026) provides a promising technical means for content authentication and provenance tracing by embedding verifiable signals into generated images.

However, existing image watermarking methods face an increasingly pronounced tension as generative architectures continue to evolve: watermark robustness often benefits from exploiting the underlying generation process, yet such dependence limits cross-architecture applicability (Umrajkar & Singh, 2025). As shown in Figure 1, post-hoc watermarking methods (Shih, 2017; Zhu et al., 2018) typically introduce small perturbations directly into the pixel space of generated images. While being agnostic to the underlying generative architecture, these watermarks are vulnerable to image reconstruction or editing. To improve robustness, recent approaches have increasingly incorporated watermarking into the generation process, such as embedding and extracting watermarks by modifying latents and performing DDIM inversion (Wen et al., 2023). Such designs, however, establish a close coupling between watermarking and the underlying generative architecture, making them difficult to transfer to models with different generation dynamics. Meanwhile, subsequent work (Ci et al., 2024; Yang et al., 2024) has primarily focused on improving robustness through more sophisticated watermark patterns or deeper exploitation of the generation process, with less attention

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.

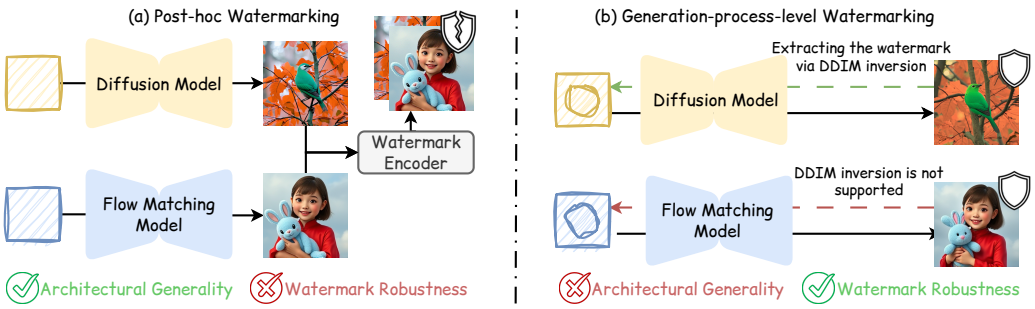


Figure 1: Comparison of post-hoc and generation-process-level watermarking, illustrating the challenge of balancing cross-architecture applicability and watermark robustness.

paid to reducing the dependence of watermarking mechanisms on specific generative architectures. This leads to a fundamental trade-off: the more effectively a watermark exploits the generation process, the more targeted robustness it can potentially obtain, but the harder it becomes to generalize across heterogeneous architectures. Conversely, generic watermarking mechanisms that are decoupled from the generation process may struggle to maintain a sufficiently stable watermark signal.

To address this limitation, we explore a different watermarking paradigm that moves beyond generation-process-level watermarking by transferring watermark operations into the internal representation space of generative models. Our key observation is that internal representations provide an intermediate layer between the generative architecture and the final image, offering an opportunity to reduce the dependence of watermarking on specific sampling procedures. In particular, Sparse Autoencoders (SAEs) (Bricken et al., 2023) can learn structured and semantically meaningful feature representations from internal activations of generative models. For different generative architectures, we can independently train corresponding SAEs to characterize their internal features (Cywiński & Deja, 2025; Surkov et al., 2024; Shabalin et al., 2025), while the definition, embedding, and extraction of watermarks can be applied uniformly within these SAE representation spaces without relying on architecture-specific generation or inversion procedures. This decomposition separates the watermarking system into architecture-specific representation learning and architecture-independent watermarking, thereby decoupling the watermarking mechanism from the underlying generation process and model architecture. However, realizing this representation-level watermarking paradigm presents two key challenges.

Challenge I: How can we ensure architectural generality? Different generative architectures induce distinct internal activation spaces whose feature distributions, dimensionalities, and semantic organizations are not naturally aligned. Although an SAE can be independently trained for each architecture, the resulting feature spaces do not share a predefined coordinate system. Therefore, the first challenge is to define a watermark representation that is independent of the underlying generative architecture and to develop a unified embedding and extraction mechanism that can operate consistently across the SAE representation spaces induced by different architectures. **Challenge II: How can we ensure watermark robustness?** Moving the watermark into the representation space does not inherently guarantee robustness to transformations applied to the generated content. During distribution and reuse, generated images may undergo reconstruction, compression, cropping, editing, and dedicated watermark removal attacks (An et al., 2024), all of which can alter the corresponding internal representations. The second challenge is therefore to identify representation structures that can stably carry watermark information while improving watermark detectability without compromising the visual quality or semantic consistency of the generated content.

To address these challenges, we propose **SAEMark**, the first SAE-based representation-level watermarking framework for text-to-image (T2I) models. SAEMark first identifies a suitable watermark carrier from structured SAE features through representation-aware screening and Jacobian-guided probing, balancing watermark detectability against perceptual quality. During embedding, it progressively enhances the selected feature along the sampling trajectory and writes the modified representation back into the generative model through SAE reconstruction, thereby establishing a feature-level watermark signal. During detection, SAEMark measures the activation of the selected SAE feature using a single forward pass on the queried image, without requiring reconstructing the

generation trajectory through inversion. By operating directly on internal representations, SAEMark provides a unified watermarking mechanism across different architectures, while offering a practical design for robust and efficient watermark detection. Our contributions are summarized as follows:

• We introduce a representation-level watermarking paradigm that moves beyond generation-process-level watermarking and identify SAE representations as a unified substrate for watermarking across heterogeneous generative architectures.

• We propose **SAEMark**, the first SAE-based watermarking framework for image generation models. SAEMark independently learns an SAE for each generative architecture while employing an architecture-independent watermarking mechanism, enabling unified watermark embedding and extraction across diffusion and flow-matching architectures and improving robustness to content transformations and watermark removal attacks.

• We conduct systematic evaluations across multiple diffusion and flow-matching architectures, covering 8 representative watermarking methods and 9 types of watermark removal attacks. The results demonstrate that SAEMark provides an effective balance among cross-architecture applicability, watermark robustness, and generation quality.

2 Related Work

2.1 Text-to-Image Models

Text-to-image (T2I) models have evolved from conventional diffusion-based generation toward increasingly diverse generative paradigms. Early models such as Stable Diffusion adopt latent diffusion (Rombach et al., 2022) with a U-Net denoiser, generating images through iterative noise prediction in a compressed latent space. More recently, flow matching (Lipman et al., 2022) has emerged as an alternative formulation that learns a continuous vector field for transporting samples from noise to the data distribution. Representative models include Stable Diffusion 3 (Esser et al., 2024), which combines rectified flow with a Multimodal Diffusion Transformer (MMDiT), and FLUX, which adopts a Transformer-based rectified-flow (Liu et al., 2022) architecture. This transition from diffusion to flow-based generation has substantially diversified the architectures and sampling dynamics of modern T2I models, posing new challenges for techniques that depend on specific generation procedures.

2.2 Watermarking for Text-to-image Models

Existing watermarking methods for T2I models can be broadly divided into post-hoc watermarking and generation-process-level watermarking according to where the watermark is introduced. Post-hoc methods embed watermarks into generated images independently of the generative model. DwtDctSvd (Shih, 2017) embeds signals in the image frequency domain, while HiDDeN (Zhu et al., 2018) uses a learned encoder-decoder to embed and recover binary messages. Despite their architectural generality, post-hoc watermarks are often vulnerable to image transformations, reconstruction, and dedicated watermark removal attacks.

An alternative line of work incorporates watermarks into the generation process itself. Stable Signature (Fernandez et al., 2023) fine-tunes the VAE decoder to produce extractable watermark signals, while Tree-Ring (Wen et al., 2023) embeds a structured pattern into the initial noise and recovers it through DDIM inversion. RingID (Ci et al., 2024) extends this paradigm to multi-key identification, and Gaussian Shading (Yang et al., 2024) preserves the Gaussian noise distribution for improved generation quality. These methods typically achieve stronger watermark signals, but rely more heavily on specific generative architecture, sampling procedures, or inversion algorithms.

Recent studies reduce architecture-specific coupling by introducing watermark guidance during generation. GROW (Luo et al., 2026) progressively guides latent predictions toward a frequency-domain watermark, and Guidance Watermark (Gesny et al., 2026) backpropagates the loss of an off-the-shelf watermark decoder through the generation process. However, their robustness remains tied to the underlying watermark representation or detector. Frequency-domain patterns can be sensitive to geometric transformations such as cropping, while detector-guided methods inherit the robustness limitations of the chosen watermark decoder.

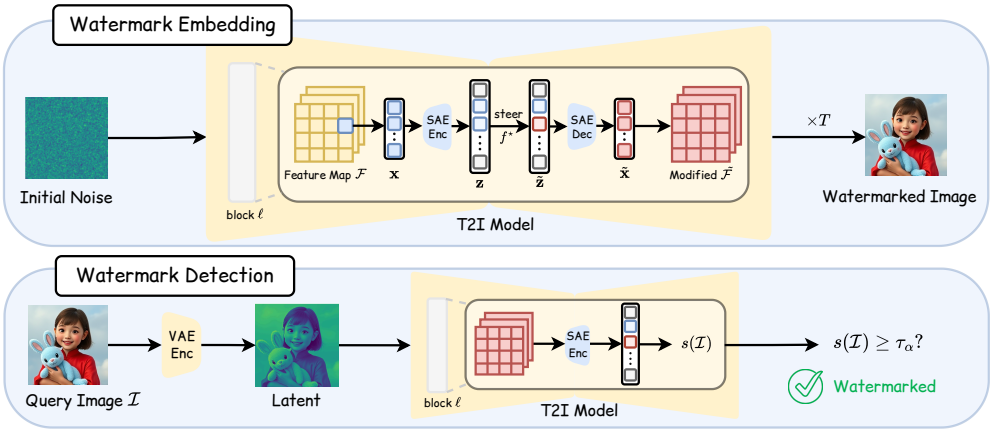


Figure 2: Pipeline of SAEMark.

2.3 Sparse Autoencoders for Text-to-Image Models

Sparse Autoencoders (SAEs) have recently emerged as an effective tool for mechanistic interpretability, enabling the discovery of interpretable features from internal model activations (Bricken et al., 2023; Huben et al., 2024). Motivated by the *superposition hypothesis* (Elhage et al., 2022), SAEs map dense activations into an overcomplete latent space where only a small subset of features is active for each input. This yields a sparse and more interpretable decomposition of the model's internal representations.

An SAE typically adopts an encoder-decoder architecture that maps input activations into a sparse latent space and reconstructs them back to the original space. Formally, let $\mathbf{x} \in \mathbb{R}^d$ denote an intermediate activation of a neural network. A Top-k SAE (Gao et al., 2025) maps $\mathbf{x}$ into a sparse overcomplete representation $\mathbf{z} \in \mathbb{R}^m$, where typically $m \gg d$:

$$\mathbf{z} = \mathrm{TopK}_k\left(\mathbf{h}(\mathbf{x})\right), \qquad \mathbf{h}(\mathbf{x}) = \mathrm{ReLU}\left(\mathbf{W}_{\mathrm{enc}}(\mathbf{x} - \mathbf{b}_{\mathrm{dec}}) + \mathbf{b}_{\mathrm{enc}}\right). \tag{1}$$

Here, $\mathbf{h}(\mathbf{x})$ denotes the dense feature activations before Top-k sparsification, and $\mathrm{TopK}_k(\cdot)$ retains the k largest activations. The decoder reconstructs the original activation from the sparse features:

$$\hat{\mathbf{x}} = \mathbf{W}_{\mathrm{dec}}\mathbf{z} + \mathbf{b}_{\mathrm{dec}}. \tag{2}$$

Recent studies have applied SAEs to intermediate representations of T2I models, enabling applications such as model interpretation (Tinaz et al., 2025), image editing (Surkov et al., 2024), bias mitigation (Shi et al., 2025), and concept erasure (Cywiński & Deja, 2025). However, SAE representations have not yet been systematically explored as a substrate for watermarking T2I models.

3 Method

As illustrated in Fig. 2, SAEMark moves watermarking from a specific generation process to the internal SAE representation space of the generative model. Specifically, SAEMark first selects a suitable watermark feature from the SAE feature space, then embeds the watermark by steering this feature during generation, and finally detects the watermark by measuring the activation of the same feature from the query image. Pseudocode for the watermark embedding and detection procedures is provided in Algorithms 1 and 2 in the Appendix.

3.1 Watermark Feature Selection

Different SAE features exhibit substantially different behaviors under intervention. Steering some features may noticeably alter the generated image, whereas others induce only a weak detection signal. We therefore select the watermark feature in two stages.

Representation-Aware Feature Screening. We first identify features that are stable under natural generation and suitable for intervention. Given a feature map $\mathcal{F}$ extracted from the selected model block ℓ, each spatial position corresponds to an activation vector $\mathbf{x}$. Following Eq. 1, we obtain the dense SAE activations $\mathbf{h}(\mathbf{x})$ and evaluate the individual features over a small set of clean generations.

We retain features that satisfy three properties: *activity*, *globality*, and *intervention consistency*. First, a feature should exhibit sufficient *activity* during natural generation, avoiding weakly activated or inactive features in the high-dimensional SAE feature space. Second, it should have high *globality*, remaining active across a large portion of $\mathcal{F}$ to ensure detectability and reduce sensitivity to localized distortions such as cropping. Third, it should exhibit strong *intervention consistency*, such that feature interventions are faithfully propagated through SAE reconstruction. Based on these criteria, we retain the top 20% of SAE features as the candidate set $\mathcal{C}$ for subsequent probing. Detailed screening procedures are provided in Appendix A.

Jacobian-Guided Feature Probing. The remaining candidates are further evaluated for two complementary properties: *detectability* and *imperceptibility*. A suitable watermark feature should produce a strong and stable increase in the target detection response while minimally affecting the generated image. We analyze these two properties through the local sensitivity to feature intervention. Since Jacobian matrices capture the first-order effect of input perturbations on model outputs, we use them to evaluate how each candidate feature affects watermark detection and image generation.

For a candidate feature f, let $\mathbf{v}_f$ denote the activation-space direction induced by intervening on this feature, and let $\mathcal{A}_f$ denote its activation response under watermark detection. For a small intervention magnitude β, the resulting changes can be approximated by

$$\mathcal{A}_f(\mathbf{x}+\beta\mathbf{v}_f)-\mathcal{A}_f(\mathbf{x})\approx\beta\mathbf{J}_{\mathcal{A}_f}(\mathbf{x})\mathbf{v}_f,\qquad \mathcal{I}(\mathbf{x}+\beta\mathbf{v}_f)-\mathcal{I}(\mathbf{x})\approx\beta\mathbf{J}_{\mathcal{I}}(\mathbf{x})\mathbf{v}_f. \tag{3}$$

where $\mathbf{J}$ denotes the Jacobian matrix and $\mathcal{I}$ denotes the generated image. A suitable watermark feature should therefore induce a large change in $\mathcal{A}_f$ while causing only a small change in $\mathcal{I}$.

We therefore use paired intervention probes as finite-difference estimates of these directional effects. For each candidate feature, we generate a clean and an intervened sample from the same prompt, then measure the detection gain and perceptual distortion as

$$S_f=\frac{\mathcal{A}_f^{\text{probe}}-\mathcal{A}_f^{\text{base}}}{\sigma_f+\epsilon},\qquad D_f=\text{LPIPS}\left(\mathcal{I}_f^{\text{probe}},\mathcal{I}^{\text{base}}\right), \tag{4}$$

where σ_f denotes the standard deviation of the feature activation on clean images.

Finally, we select the feature with the best detection–quality trade-off:

$$f^\star=\arg\max_{f\in\mathcal{C},\,S_f\geq\tau_{\text{det}}}\frac{S_f}{D_f}. \tag{5}$$

3.2 Watermark Embedding

Given the selected watermark feature $f^\star$, we embed the watermark by intervening on its activation during sampling. At each intervention step, we extract the feature map $\mathcal{F}$ from the selected model block ℓ and obtain the dense SAE activations $\mathbf{h}(\mathbf{x})$ for each activation vector $\mathbf{x}$ following Eq. 1. To adapt the intervention to the local activation scale, we steer $f^\star$ as

$$\tilde{h}_i(\mathbf{x})=\begin{cases}\lambda\|\mathbf{h}(\mathbf{x})\|_\infty, & i=f^\star,\\ h_i(\mathbf{x}), & \text{otherwise},\end{cases} \tag{6}$$

where λ controls the intervention strength. We then apply Top-k sparsification to $\tilde{\mathbf{h}}(\mathbf{x})$ and reconstruct through the SAE decoder following Eq. 2, yielding the modified reconstruction $\tilde{\mathbf{x}}$.

Directly replacing $\mathbf{x}$ with $\tilde{\mathbf{x}}$ would introduce SAE reconstruction error and may degrade fine-grained details in the generated image. To mitigate this, we retain the original reconstruction residual when writing the modified representation back to the model:

$$\mathbf{x}^w=\tilde{\mathbf{x}}+(\mathbf{x}-\hat{\mathbf{x}}). \tag{7}$$

The feature vectors $\mathbf{x}^w$ are reassembled into the watermarked feature map $\tilde{\mathcal{F}}$, which is then passed to the subsequent model blocks. The embedding process is controlled by two hyperparameters: the intervention timestep $\mathcal{T}$ and the intervention strength λ, which are further discussed in Sec. 4.

3.3 WATERMARK DETECTION

SAEMark detects the watermark directly from the selected SAE feature $f^\star$ without reconstructing the generation trajectory. Given a query image $\mathcal{I}$, we first encode it into the latent space using the native image encoder $\mathcal{E}$. The encoded latent is treated as the state before the final denoising step and passed once through the T2I model at timestep t_d with an empty prompt. From the same model block ℓ used for watermark embedding, we extract the feature map $\mathcal{F}_\ell(\mathcal{I})$ and compute the mean activation of the watermark feature as the detection score:

$$s(\mathcal{I}) = \mathbb{E}_{\mathbf{x}\in\mathcal{F}_\ell(\mathcal{I})}\left[h_{f^\star}(\mathbf{x})\right]. \tag{8}$$

For a target false positive rate α, the detection threshold is calibrated on a clean image set $\mathcal{D}_{\text{cal}}$:

$$\tau_\alpha = \text{Quantile}_{1-\alpha}\left(\{s(\mathcal{I}) \mid \mathcal{I}\in\mathcal{D}_{\text{cal}}\}\right), \qquad \hat{y}(\mathcal{I}) = \mathbb{I}\left[s(\mathcal{I}) \geq \tau_\alpha\right]. \tag{9}$$

Thus, SAEMark requires neither the original prompt nor an inversion procedure, enabling efficient watermark detection with a single model forward pass.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTINGS

Models and Datasets. We evaluate SAEMark on two representative text-to-image models, Stable Diffusion v2.1-base (SD-v2.1) and FLUX.1-schnell, covering both latent diffusion and rectified-flow Transformer architectures. For SD-v2.1, we sample 1,000 prompts from MS-COCO-2017 and generate 512×512 images using DPM-Solver++ with 50 inference steps and a guidance scale of 7.5. For FLUX.1-schnell, we sample 1,000 prompts from Stable-Diffusion-Prompts and generate 256×256 images using the FlowMatch Euler scheduler with 4 inference steps and a guidance scale of 0. All watermarking methods use the same prompts and generation configurations under each backbone for fair comparison.

SAEMark Setup. Following Tree-Ring (Wen et al., 2023), we focus on the single-bit watermark setting. For SD-v2.1, we train an SAE on activations from U-Net up-sampling block `up.1.0`, with training details provided in Appendix B. We set the intervention strength to $\lambda = 2.0$ and the detection timestep to $t_d = 1$. For FLUX.1-schnell, we use the publicly released SAE trained on double-stream Block 18[1] (Shabalin et al., 2025), with $\lambda = 0.6$ and $t_d = 0.01$. For both models, watermark intervention is applied at each timestep throughout the sampling trajectory.

Watermark Baselines. We compare SAEMark with eight representative watermarking methods, including two post-hoc methods, DwtDctSvd (Shih, 2017) and HiDDeN (Zhu et al., 2018), and six in-generation methods, Stable Signature (Fernandez et al., 2023), Tree-Ring (Wen et al., 2023), RingID (Ci et al., 2024), Gaussian Shading (Yang et al., 2024), GROW (Luo et al., 2026), and Guidance Watermarking (Gesny et al., 2026). We follow the official implementations and default watermark configurations. For Stable Signature, we use the robustly trained watermark extractor and separately fine-tune a watermarked VAE decoder for SD-v2.1 and FLUX.1-schnell. For inversion-based methods, including Tree-Ring, RingID, and Gaussian Shading, we adapt them to FLUX by replacing DDIM inversion with rectified-flow (RF) inversion, following prior work (Umrajkar & Singh, 2025). For further implementation details, see Appendix C.

Robustness Evaluation. To comprehensively evaluate watermark robustness, we consider nine representative image distortions and regeneration attacks: JPEG compression (quality 50), rotation (30°), Cropping & Scaling (75%), Gaussian blur (15×15 kernel), Gaussian noise ($\sigma = 0.05$), brightness adjustment (factor 2.0), contrast adjustment (factor 2.0), VAE-based regeneration (VA), and diffusion-based regeneration (DA) (Zhao et al., 2024). For VA, we use Cheng2020-anchor with quality level 3. For DA, we use SD-v2.1-based diffusion reconstruction with a noise timestep of 100 out of 1,000. For more details, see Appendix D.

Evaluation Metrics. We evaluate both generation quality and watermark robustness. For generation quality, we report Fréchet Inception Distance (FID) (Heusel et al., 2017) and CLIP Score (Radford et al., 2021). On MS-COCO-2017, FID is computed between the watermarked images and 5,000

[1] `https://huggingface.co/nev/flux1-saes`

Table 1: [RQ1] Watermarking performance against various attacks on SD-v2.1. **Bolded** and *underlined italicized* values denote the best and second-best performance, respectively.

Method	Generation Quality		Robustness to Attacks (TPR@1%FPR)										
	FID↓	CLIP↑	None	JPEG	Rotation	Crop	Blur	Noise	Brightness	Contrast	VA	DA	Avg.
DwtDctSvd	*52.33*	**0.318**	1.00	0.61	0.00	0.00	0.74	1.00	0.19	0.20	0.01	0.04	0.38
HiDDeN	52.48	0.314	1.00	0.93	0.00	0.90	0.00	0.30	1.00	1.00	0.01	0.01	0.52
Stable Signature	53.36	0.314	1.00	*0.98*	*0.99*	**1.00**	0.05	0.99	0.99	0.99	0.42	0.01	0.74
Tree-Ring	55.37	0.313	1.00	0.93	0.16	0.07	0.97	0.94	0.98	0.98	0.86	0.94	0.78
RingID	55.79	0.313	1.00	1.00	**1.00**	0.00	*0.99*	1.00	1.00	1.00	*0.99*	1.00	*0.90*
Gaussian Shading	55.26	0.314	1.00	1.00	0.00	0.02	1.00	1.00	1.00	1.00	1.00	1.00	0.80
GROW	**51.79**	*0.315*	1.00	1.00	0.01	0.01	1.00	**1.00**	**1.00**	1.00	**1.00**	**1.00**	0.80
Guidance Watermark	53.15	0.314	1.00	1.00	0.01	0.34	**1.00**	0.96	0.99	**1.00**	0.92	0.72	0.79
SAEMark (Ours)	55.77	0.314	**1.00**	**1.00**	0.83	*0.99*	0.97	*0.99*	*0.99*	*0.99*	0.98	*0.99*	**0.97**

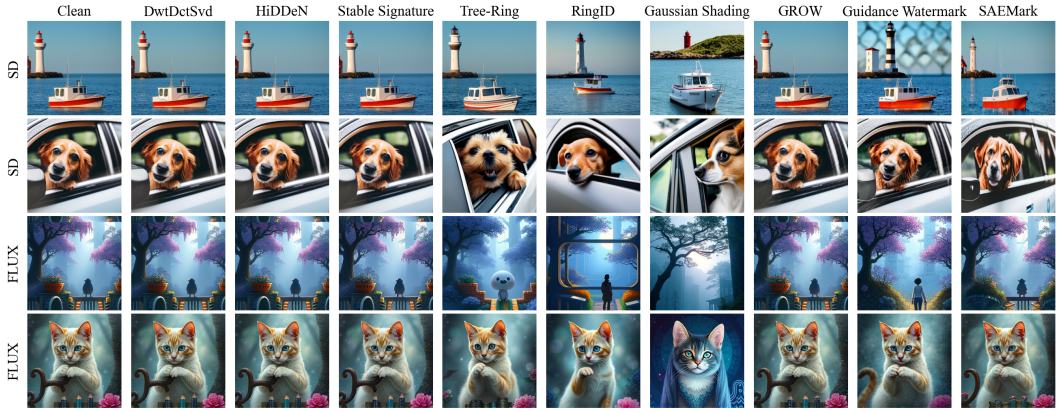


Figure 3: [RQ1] Visual comparison of watermarking methods on SD-v2.1 and FLUX.1-schnell.

real images from the COCO validation set, while on Stable-Diffusion-Prompts, FID is computed between the watermarked and clean images generated from the same prompts. CLIP Score is computed between each generated image and its text prompt using the CLIP ViT-B/32 model. Lower FID and higher CLIP Score indicate better generation quality. For robustness, we report the true positive rate at 1% false positive rate (TPR@1%FPR) on all images. For each method, the detection threshold is determined using the 1,000 generated clean images.

4.2 Experimental Results

We conduct extensive experiments to answer the following research questions (RQs):

- [RQ1] How well does SAEMark balance architecture generality and watermark robustness?
- [RQ2] How efficient is SAEMark for watermark embedding and detection?
- [RQ3] How do hyperparameters and SAE feature selection affect the performance of SAEMark?

RQ1: Cross-Architecture Generality and Robustness. Table 1 and Table 2 report the generation quality and watermark detection performance of eight representative methods on SD-v2.1 and FLUX.1-schnell, respectively. We highlight three key observations.

First, SAEMark provides reliable watermark detection across heterogeneous generation architectures. Without attacks, it achieves a TPR@1%FPR of 1.00 on both SD-v2.1 and FLUX.1-schnell, despite their substantially different architectures and generation processes. This result indicates that the selected SAE features can consistently carry a detectable watermark signal across both model families. *Second*, SAEMark achieves the highest average robustness among the evaluated methods. It obtains an average TPR@1%FPR of 0.97 on SD-v2.1 and 0.90 on FLUX.1-schnell. . In comparison, several baselines experience substantial degradation under regeneration attacks. These results

Table 2: [RQ1] Watermarking performance against various attacks on FLUX.1-schnell. **Bolded** and *underlined italicized* values denote the best and second-best performance, respectively.

Method	Generation Quality		Robustness to Attacks (TPR@1%FPR)										
	FID↓	CLIP↑	None	JPEG	Rotation	Crop	Blur	Noise	Brightness	Contrast	VA	DA	Avg.
DwtDctSvd	**4.71**	0.297	1.00	0.68	0.00	0.00	*0.75*	0.96	0.04	0.07	0.01	0.03	0.35
HiDDeN	*11.57*	0.297	1.00	1.00	0.01	*0.94*	0.00	0.42	1.00	1.00	0.04	0.07	0.55
Stable Signature	18.30	*0.297*	1.00	0.99	**1.00**	**1.00**	0.11	1.00	1.00	1.00	0.54	0.04	0.77
Tree-Ring	32.68	0.301	1.00	0.63	0.00	0.01	0.11	0.85	0.97	0.96	0.13	0.01	0.47
RingID	40.17	0.292	1.00	0.93	0.02	0.00	0.42	0.97	1.00	1.00	0.44	0.01	0.58
Gaussian Shading	41.41	0.301	1.00	1.00	0.01	0.02	0.99	**1.00**	**1.00**	**1.00**	**1.00**	**0.96**	*0.80*
GROW	13.69	**0.301**	1.00	*0.99*	0.01	0.00	**0.99**	0.95	*0.98*	*0.99*	0.79	0.24	0.69
Guidance Watermark	50.27	0.282	*0.89*	0.85	0.03	0.06	0.69	0.53	0.83	0.85	0.30	0.04	0.51
SAEMark (Ours)	36.81	0.296	**1.00**	**1.00**	*0.74*	0.92	0.71	*0.99*	0.94	0.98	*0.83*	*0.86*	**0.90**

demonstrate that SAEMark maintains a persistent watermark signal across diverse content transformations. *Finally*, SAEMark maintains semantic alignment and visual quality while supporting cross-architecture watermarking. Its CLIP scores are comparable to those of the evaluated baselines on both architectures, and the visual comparisons in Figure 3 show no obvious perceptual artifacts introduced by watermark embedding. Thus, SAEMark achieves broad architectural applicability and strong watermark robustness without an evident loss in semantic consistency or perceptual quality.

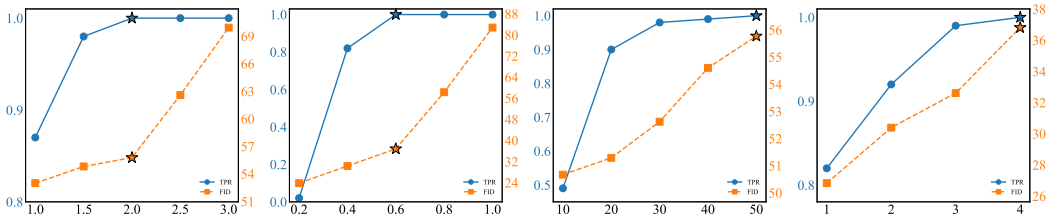
Overall, the results highlight the limitations of existing watermarking paradigms. Post-hoc methods generally embed signals directly into the generated images, making them vulnerable to common transformations and regeneration. Generation-level methods can achieve stronger robustness by exploiting the underlying sampling process, but their performance may degrade when transferred to architectures with different generation mechanisms. This limitation is particularly evident for inversion-based methods on FLUX.1-schnell. The adaptation from DDIM to rectified-flow inversion introduces additional challenges because high-quality RF inversion generally benefits from an accurate textual description of the input image, whereas the original prompt is unavailable during post-generation verification. The resulting inversion error can therefore impair watermark detection, as described in Appendix E. In contrast, SAEMark operates on internal SAE representations without relying on a specific inversion procedure, enabling consistent detection performance across both diffusion and rectified-flow architectures.

RQ2: Analysis of Time Efficiency. Table 3 reports the average generation and detection time over 100 watermarked images on a single NVIDIA GeForce RTX 4090 GPU. The results reveal distinct efficiency trade-offs among existing watermarking approaches. Stable Signature introduces negligible runtime overhead during generation because its watermarking functionality is incorporated into the VAE decoder parameters. Similarly, Tree-Ring, RingID, and Gaussian Shading require only a small additional cost for embedding watermark patterns into the initial noise. However, their inversion-based detection procedures incur substantially higher runtime overhead, with detection times ranging from 1.35 to 1.88 seconds per image. In contrast, Guidance Watermark introduces a latent optimization procedure during generation, resulting in a considerably higher generation cost of 83.26 seconds per image.

Table 3: [RQ2] Time efficiency comparison on SD-v2.1. We report the average runtime per image in seconds.

Method	Generation (s) ↓	Detection (s) ↓
Stable Signature	1.44	0.02
Tree-Ring	1.47	1.44
RingID	1.44	1.88
Gaussian Shading	1.56	1.35
GROW	1.78	0.02
Guidance Watermark	83.26	0.17
SAEMark (Ours)	1.69	0.08

SAEMark provides a favorable balance between watermarking effectiveness and execution efficiency. Its average generation time is 1.69 seconds, which remains comparable to that of the evaluated generation-based methods, while its detection time is only 0.08 seconds per image. Unlike inversion-based approaches, SAEMark does not require reconstructing the generation trajectory during detection. Instead, it measures the target SAE feature activation through a single model forward pass, substantially reducing the detection overhead. These results show that the proposed



(a) Strength λ on SD-v2.1 (b) Strength λ on FLUX.1-schnell (c) Timesteps $\mathcal{T}$ on SD-v2.1 (d) Timesteps $\mathcal{T}$ on FLUX.1-schnell

Figure 4: [RQ3] Ablation of intervention strength λ and intervention timesteps $\mathcal{T}$ on SD-v2.1 and FLUX.1-schnell. Stars indicate the default settings of SAEMark.



(a) Detection-quality trade-off (b) Clean generation (c) Steer feature 51066 (Ours) (d) Steer feature 64053 (e) Steer feature 1946

Figure 5: [RQ3] Ablation on watermark feature selection.

representation-level watermarking framework can support architecture-general watermarking and robust detection without introducing substantial computational costs.

RQ3: Analysis of Hyperparameters and SAE Feature Selection. Figure 4 studies the effects of intervention strength λ and the number of intervention timesteps $\mathcal{T}$ on watermark detectability and generation quality, measured by TPR@1%FPR and FID, respectively. Across both SD-v2.1 and FLUX.1-schnell, increasing either λ or $\mathcal{T}$ improves TPR but also increases FID. TPR rises rapidly and gradually saturates as the watermark signal becomes sufficiently strong, while further intervention continues to affect generation quality. We therefore use $\lambda = 2.0$ with the full intervention schedule for SD-v2.1, and $\lambda = 0.6$ with the default schedule for FLUX.1-schnell, where detection performance is already close to saturation. This suggests that excessive intervention provides limited detection gains while unnecessarily increasing the impact on generation quality.

Figure 5 further evaluates watermark feature selection on FLUX.1-schnell. Different SAE features exhibit distinct detectability–quality trade-offs. Feature 51066 achieves a favorable combination of high TPR@1%FPR and low FID, while feature 64053 produces noticeable visual distortion despite a strong watermark signal, and feature 1946 induces limited visual changes but yields weak detection. The qualitative comparisons in Figure 5(b–e) are consistent with these results. This demonstrates that the choice of SAE feature is critical to translating a given intervention budget into an effective watermark signal.

Overall, λ and $\mathcal{T}$ control the magnitude and duration of intervention, while feature selection determines its effectiveness. With an appropriately selected SAE feature, moderate intervention is sufficient to achieve reliable watermark detection while preserving generation quality.

5 Conclusion

In this work, we presented **SAEMark**, a representation-level watermarking framework that decouples watermarking from the underlying generation process of text-to-image models. By embedding and detecting watermarks in SAE feature spaces, SAEMark provides a unified mechanism across heterogeneous diffusion and flow-matching architectures while maintaining strong robustness to diverse content transformations and watermark removal attacks. Extensive experiments demonstrate that SAEMark achieves reliable cross-architecture detection, strong watermark robustness, and competitive generation quality with low detection overhead. Our results highlight the potential of SAE representations as a practical intermediate substrate for robust and architecture-general watermarking of generative models.

AI USE STATEMENT

In this work, we used generative AI tools primarily for language editing and writing assistance, including improving the clarity, grammar, and presentation of the manuscript. All AI-assisted content was carefully reviewed and verified by the authors. The authors take full responsibility for the final manuscript, including all text, claims, experimental results, and other research artifacts.

REFERENCES

Bang An, Mucong Ding, Tahseen Rabbani, Aakriti Agrawal, Yuancheng Xu, Chenghao Deng, Sicheng Zhu, Abdirisak Mohamed, Yuxin Wen, Tom Goldstein, et al. Waves: Benchmarking the robustness of image watermarks. *arXiv preprint arXiv:2401.08573*, 2024.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2(5):6, 2023.

Hai Ci, Pei Yang, Yiren Song, and Mike Zheng Shou. Ringid: Rethinking tree-ring watermarking for enhanced multi-key identification. In *European conference on computer vision*, pp. 338–354. Springer, 2024.

Bartosz Cywiński and Kamil Deja. Saeuron: Interpretable concept unlearning in diffusion models with sparse autoencoders. In *Forty-second International Conference on Machine Learning*, 2025.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition, 2022. URL https://arxiv.org/abs/2209.10652.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22409–22420. IEEE, 2023.

Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *International Conference on Learning Representations*, volume 2025, pp. 26721–26754, 2025.

Enoal Gesny, Eva Giboulot, Teddy Furon, and Vivien Chappelier. Guidance watermarking for diffusion models. In *International Conference on Learning Representations*, volume 2026, pp. 113957–113988, 2026.

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.

Robert Huben, Hoagy Cunningham, Logan Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*, volume 2024, pp. 7827–7845, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Pengcheng Luo, Zexi Jia, Yijia Zhong, Jinchao Zhang, and Jie Zhou. Grow: Watermark generation with progressive guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35978–35987, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

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. 10674–10685. ieee, 2022.

Stepan Shabalin, Ayush Panda, Dmitrii Kharlapenko, Abdur Raheem Ali, Yixiong Hao, and Arthur Conmy. Interpreting large text-to-image diffusion models with dictionary learning. *arXiv preprint arXiv:2505.24360*, 2025.

Yingdong Shi, Changming Li, Yifan Wang, Yongxiang Zhao, Anqi Pang, Sibei Yang, Jingyi Yu, and Kan Ren. Dissecting and mitigating diffusion bias via mechanistic interpretability. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8192–8202. IEEE, 2025.

Frank Y Shih. *Digital watermarking and steganography: fundamentals and techniques*. CRC press, 2017.

Viacheslav Surkov, Chris Wendler, Antonio Mari, Mikhail Terekhov, Justin Deschenaux, Robert West, Caglar Gulcehre, and David Bau. One-step is enough: Sparse autoencoders for text-to-image diffusion models. *arXiv preprint arXiv:2410.22366*, 2024.

Berk Tinaz, Zalan Fabian, and Mahdi Soltanolkotabi. Emergence and evolution of interpretable concepts in diffusion models. *Advances in neural information processing systems*, 38:166943, 2025.

Ved Umrajkar and Aakash Kumar Singh. Detection limits and statistical separability of tree ring watermarks in rectified flow-based text-to-image generation models. *arXiv preprint arXiv:2504.03850*, 2025.

Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust. *arXiv preprint arXiv:2305.20030*, 2023.

Zijin Yang, Kai Zeng, Kejiang Chen, Han Fang, Weiming Zhang, and Nenghai Yu. Gaussian shading: Provable performance-lossless image watermarking for diffusion models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12162–12171. IEEE, 2024.

Xuandong Zhao, Kexun Zhang, Zihao Su, Saastha Vasan, Ilya Grishchenko, Christopher Kruegel, Giovanni Vigna, Yu-Xiang Wang, and Lei Li. Invisible image watermarks are provably removable using generative ai. *Advances in neural information processing systems*, 37:8643–8672, 2024.

Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *European conference on computer vision*, pp. 682–697. Springer, 2018.

Algorithm 1 SAEMark Watermark Embedding

Require: Text prompt c, T2I model θ, watermark feature $f^\star$, target block ℓ, intervention window $\mathcal{T}$, strength λ
Ensure: Watermarked image $\mathcal{I}^w$

$\mathbf{y}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \dots, 1$ **do**
  $\mathcal{F}_{\ell,t} \leftarrow \textsc{ForwardToBlock}(\theta, \mathbf{y}_t, t, c, \ell)$
  if $t \in \mathcal{T}$ **then**
    for all $\mathbf{x} \in \mathcal{F}_{\ell,t}$ **do**
      $\mathbf{h} \leftarrow \mathbf{h}(\mathbf{x})$ ▷ SAE encode, Eq. 1
      $\tilde{h}_{f^\star} \leftarrow \lambda \|\mathbf{h}\|_\infty$ ▷ Steer watermark feature $f^\star$, Eq. 6
      $\tilde{\mathbf{x}} \leftarrow \mathbf{W}_{\text{dec}} \, \text{TopK}_k(\tilde{\mathbf{h}}) + \mathbf{b}_{\text{dec}}$ ▷ Decode the intervened SAE representation, Eq. 2
      $\hat{\mathbf{x}} \leftarrow \mathbf{W}_{\text{dec}} \, \text{TopK}_k(\mathbf{h}) + \mathbf{b}_{\text{dec}}$ ▷ Decode the original SAE representation, Eq. 2
      $\mathbf{x}^w \leftarrow \tilde{\mathbf{x}} + (\mathbf{x} - \hat{\mathbf{x}})$ ▷ Restore reconstruction residual, Eq. 7
    end for
    $\mathcal{F}^w_{\ell,t} \leftarrow \textsc{Assemble}(\{\mathbf{x}^w\})$
  else
    $\mathcal{F}^w_{\ell,t} \leftarrow \mathcal{F}_{\ell,t}$
  end if
  $\mathbf{o}_t \leftarrow \textsc{ForwardFromBlock}(\theta, \mathcal{F}^w_{\ell,t}, t, c, \ell)$
  $\mathbf{y}_{t-1} \leftarrow \textsc{SamplingStep}(\mathbf{y}_t, \mathbf{o}_t, t)$
end for
$\mathcal{I}^w \leftarrow \mathcal{D}(\mathbf{y}_0)$
return $\mathcal{I}^w$

Algorithm 2 SAEMark Watermark Detection

Require: Query image $\mathcal{I}$, T2I model θ, watermark feature $f^\star$, target block ℓ, detection timestep t_d, threshold τ_α
Ensure: Watermark prediction $\hat{y}$

$\mathbf{y}_\mathcal{I} \leftarrow \mathcal{E}(\mathcal{I})$ ▷ Encode query image
$\mathcal{F}_\ell(\mathcal{I}) \leftarrow \textsc{ForwardToBlock}(\theta, \mathbf{y}_\mathcal{I}, t_d, c = \varnothing, \ell)$ ▷ Single model forward pass
$s(\mathcal{I}) \leftarrow \mathbb{E}_{\mathbf{x} \in \mathcal{F}_\ell(\mathcal{I})} [h_{f^\star}(\mathbf{x})]$ ▷ Compute the mean activation of the watermark feature $f^\star$, Eq. 8
$\hat{y} \leftarrow \mathbb{I}[s(\mathcal{I}) \geq \tau_\alpha]$ ▷ Eq. 9
return $\hat{y}$

A Details of Representation-Aware Carrier Screening

We provide the detailed definitions of the three criteria used for representation-aware carrier screening. Let $h_f(\mathbf{x})$ denote the dense activation of SAE feature f before Top-k sparsification, $\mathcal{X}(\mathcal{F})$ the set of spatial activation vectors in feature map $\mathcal{F}$, and $\mathcal{D}_{\text{clean}}$ the collection of feature maps extracted from clean generations.

Activity. A suitable watermark carrier should be naturally activated during clean generation. We measure the activity of feature f by its average dense activation:

$$A_f = \mathbb{E}_{\mathcal{F} \sim \mathcal{D}_{\text{clean}}} \left[\frac{1}{|\mathcal{X}(\mathcal{F})|} \sum_{\mathbf{x} \in \mathcal{X}(\mathcal{F})} h_f(\mathbf{x}) \right]. \tag{10}$$

A larger A_f indicates that the feature is more strongly and consistently activated under natural generation.

Globality. We further prefer features that are active across a large fraction of spatial locations rather than being restricted to localized regions. We quantify this property using the spatial activation

ratio:

$$G_f = \mathbb{E}_{\mathcal{F}\sim\mathcal{D}_{\text{clean}}}\left[\frac{1}{|\mathcal{X}(\mathcal{F})|}\sum_{\mathbf{x}\in\mathcal{X}(\mathcal{F})}\mathbb{I}[h_f(\mathbf{x})>0]\right]. \tag{11}$$

A larger G_f indicates broader spatial coverage, reducing the dependence of the watermark signal on specific local regions.

Intervention Consistency. Finally, feature intervention should propagate reliably through SAE reconstruction. For feature f, we measure the alignment between its encoding and decoding directions as

$$C_f = \cos\left(\mathbf{W}_{\text{enc}}^{(f,:)}, \mathbf{W}_{\text{dec}}^{(:,f)}\right). \tag{12}$$

where $\mathbf{W}_{\text{enc}}^{(f,:)}$ and $\mathbf{W}_{\text{dec}}^{(:,f)}$ denote the encoding and decoding directions associated with feature f, respectively. A larger C_f indicates stronger alignment between the two directions, allowing interventions on the feature to be more faithfully propagated through SAE reconstruction.

B SAE Training Details

Existing SAEs for text-to-image models are typically developed either for specific downstream tasks or for particular model architectures. Task-oriented SAEs are often trained on activations collected from task-relevant image or concept distributions, such as those used for concept erasure (Cywiński & Deja, 2025) or bias mitigation (Shi et al., 2025). Other publicly released SAEs are tied to specific backbones and hookpoints, such as the SAEs trained on SDXL-based models for interpretable feature discovery and image editing (Surkov et al., 2024). Thus, we train our own SAE for SD-v2.1 using a large-scale prompt set, with the goal of learning a general-purpose representation of its internal activations for watermarking.

Hookpoint Selection. Following prior work showing that representations in the upsampling stage of Stable Diffusion are amenable to SAE-based semantic intervention (Cywiński & Deja, 2025), we extract activations from `unet.up_blocks.1.attentions.0`. We cache the block output to model the representation at the selected hookpoint.

Activation Collection. We collect training activations from Stable Diffusion v2.1-base using 10,000 unique prompts sampled from LAION-COCO Aesthetic. Images are generated with deterministic DDIM sampling using 50 denoising steps, classifier-free guidance of 7.5, and $\eta = 0$. We cache the output feature map of the selected block at every denoising step. Under classifier-free guidance, only activations from the conditional branch are retained. A single SAE is shared across all denoising timesteps, allowing it to model representations throughout the sampling trajectory.

Each cached feature map is decomposed into spatial activation vectors, with each spatial position treated as an independent training sample. The activation dimension is $d = 1280$. We use a dictionary size of $20{,}480$, corresponding to an expansion factor of 16, and enforce sparsity with Top-k using $k = 32$.

Training Objective. We optimize the SAE using the fraction of variance unexplained (FVU) as the primary reconstruction objective. For a minibatch $\{\mathbf{x}_i\}_{i=1}^B$ with mean $\bar{\mathbf{x}}$ and SAE reconstructions $\hat{\mathbf{x}}_i$, the FVU loss is defined as

$$\mathcal{L}_{\text{FVU}} = \frac{\sum_{i=1}^B \|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_2^2}{\sum_{i=1}^B \|\mathbf{x}_i - \bar{\mathbf{x}}\|_2^2}. \tag{13}$$

FVU measures the reconstruction error relative to the variance of the input activations, making the objective less sensitive to their overall scale. Lower FVU indicates better reconstruction.

We further employ an auxiliary Top-k (AuxK) objective to mitigate dead features. A latent feature is considered dead if it has not been selected by the main Top-k operator for more than 10^7 activation tokens. Let N_{dead} denote the number of dead features. The auxiliary pathway selects

$$k_{\text{aux}} = \min\left(\frac{d}{2}, N_{\text{dead}}\right) \tag{14}$$

features exclusively from the dead-feature set.

Let $\mathbf{e}_i = \hat{\mathbf{x}}_i - \mathbf{x}_i$ denote the reconstruction residual, and let $\hat{\mathbf{e}}_i$ denote its reconstruction using the selected auxiliary features. The AuxK loss is defined as

$$\mathcal{L}_{\text{AuxK}} = s\frac{\sum_{i=1}^{B}\|\hat{\mathbf{e}}_i - \mathbf{e}_i\|_2^2}{\sum_{i=1}^{B}\|\mathbf{x}_i - \bar{\mathbf{x}}\|_2^2}, \qquad s = \min\left(\frac{N_{\text{dead}}}{k_{\text{aux}}}, 1\right). \tag{15}$$

This auxiliary objective encourages inactive features to explain residual structure that is not captured by the main sparse reconstruction.

The overall training objective is

$$\mathcal{L}_{\text{SAE}} = \mathcal{L}_{\text{FVU}} + \frac{1}{32}\mathcal{L}_{\text{AuxK}}. \tag{16}$$

Optimization. We train the SAE for 5 epochs with a batch size of 4096 and a learning rate of 1×10^{-4}. The trained SAE achieves an FVU of 0.374, demonstrating good reconstruction quality.

C Watermark Baselines Descriptions

DwtDctSvd (Shih, 2017). This post-hoc watermarking method embeds binary information in the frequency domain through discrete wavelet transform (DWT), discrete cosine transform (DCT), and singular value decomposition (SVD). During detection, the decoded bits are compared with the target watermark to obtain the bit-matching score. For implementation, we follow the *invisible-watermark* repository[2] and embed a fixed 48-bit watermark with an embedding strength of 36.

HiDDeN (Zhu et al., 2018). This learning-based post-hoc watermarking method jointly trains an encoder and decoder for watermark embedding and extraction. A differentiable noise layer applies image transformations during training to improve robustness. Following the default implementation[3], we use a fixed 48-bit watermark.

Stable Signature (Fernandez et al., 2023). This in-generation watermarking method leverages the watermark decoder from HiDDeN (Zhu et al., 2018) and fine-tunes the VAE decoder so that a binary watermark can be extracted from generated images. Following the official implementation, we use a fixed 48-bit watermark and the robust watermark extractor released in the source repository[4].

Tree-Ring (Wen et al., 2023). Tree-Ring is a training-free in-generation watermarking method. Instead of modifying the VAE decoder, it embeds a predefined ring pattern into the Fourier domain of the initial latent noise. During detection, the watermark pattern is recovered from the watermarked image through DDIM inversion and compared with the target pattern in the Fourier domain. Following the default configuration[5], we use latent channel 3 and a circular mask with radius 10. For FLUX.1-schnell, we replace DDIM inversion with rectified-flow inversion along the flow-matching trajectory.

RingID (Ci et al., 2024). Building on Tree-Ring, RingID redesigns the Fourier watermark pattern with rotation-invariant rings to improve key distinguishability and multi-key identification. During detection, the recovered Fourier coefficients are compared with the target ring pattern using the ℓ_1 distance. Following the official configuration[6], we use 11 ring slots quantized to $\{-64, +64\}$, yielding 2^{11} possible keys, and adopt the official owner key with index 628.

Gaussian Shading (Yang et al., 2024). Gaussian Shading improves initial-noise watermarking by encoding messages through truncated Gaussian sampling rather than explicit Fourier patterns. This preserves the Gaussian distribution of the initial latent while supporting multi-bit watermark embedding. During detection, watermark bits are recovered from the signs of the reconstructed latent values and aggregated by majority voting. Following the default configuration[7], we use a 256-bit watermark with ChaCha20 encryption.

[2] https://github.com/ShieldMnt/invisible-watermark
[3] https://github.com/ando-khachatryan/HiDDeN
[4] https://github.com/facebookresearch/stable_signature
[5] https://github.com/YuxinWenRick/tree-ring-watermark
[6] https://github.com/showlab/RingID
[7] https://github.com/bsmhmmlf/Gaussian-Shading

Table 4: Effective watermark bit capacity of different watermarking schemes. Tree-Ring and RingID are pattern-based, while the others use binary payloads.

Method	DwtDctSvd	HiDDeN	Stable Signature	Tree-Ring	RingID	Gaussian Shading	GROW	Guidance Watermark
Bits	48	48	48	1	11	256	16	256

GROW (Luo et al., 2026). GROW is a guidance-based watermarking method that progressively steers intermediate latent predictions toward a predefined frequency-domain watermark. During detection, the watermarked image is encoded into the latent space, where the signs of key-dependent DCT coefficients are decoded and aggregated by majority voting. Following the default configuration[8], we use a 16-bit watermark in latent channel 0, with watermark strength $\alpha = 0.5$, guidance coefficient $\eta = 100$, frequency band $[0.2, 0.5)$, and watermark start ratio $r_{\text{start}} = 0.5$.

Guidance Watermark (Gesny et al., 2026). This guidance-based watermarking method converts an off-the-shelf watermark decoder into an in-generation scheme by backpropagating its watermark loss through the denoising process. We use the VideoSeal decoder with a fixed 256-bit watermark and PCGrad to aggregate gradients across the default transformation set, including contrast adjustment, brightness adjustment, center cropping, and JPEG compression at qualities 50 and 80. The original formulation backpropagates through intermediate denoising trajectories, resulting in $T(T+1)/2$ diffusion and gradient-propagation steps for a T-step sampler. We follow the default guidance configuration[9], with watermark scale and maximum gradient norm set to 250 and 0.3 for SD-v2.1, and 500 and 0.5 for FLUX.1-schnell, respectively.

D Attack Baselines Descriptions

JPEG Compression. This operation applies lossy JPEG encoding to the watermarked image, which removes fine-grained image details and may degrade the embedded watermark signal. The quality factor determines the compression level, where a lower value results in greater information loss and generally corresponds to a stronger attack. In our experiments, the quality factor is set to 50.

Rotation. This geometric transformation changes the orientation of an image by a specified angle and may disrupt the spatial structure of the embedded watermark. Larger rotation angles generally introduce greater geometric distortion and thus correspond to stronger attacks. In our experiments, the rotation angle is set to 30°.

Cropping & Scaling. This operation retains only a portion of an image according to a specified crop ratio and then resizes the cropped image to its original resolution. A higher crop ratio preserves more image content, while a lower ratio removes more information and generally leads to a stronger attack. In our experiments, we use center cropping with a crop ratio of 75%.

Gaussian Blur. This transformation smooths the image with a Gaussian kernel, thereby suppressing local details and high-frequency information. The kernel size determines the degree of blurring, where a larger kernel generally produces stronger distortion. In our experiments, the Gaussian kernel size is set to 15×15.

Gaussian Noise. Random perturbations sampled from a zero-mean Gaussian distribution are added to image pixels to corrupt the watermark signal. The standard deviation σ determines the noise magnitude, with a larger σ introducing stronger perturbations. In our experiments, σ is set to 0.05.

Brightness Adjustment. The overall image intensity is modified by scaling pixel values with a brightness factor. A factor of 1 leaves the image unchanged, while values farther from 1 introduce stronger modifications. In our experiments, the brightness factor is set to 2.0.

Contrast Adjustment. The difference between bright and dark regions is adjusted relative to the image mean, which may alter the watermark signal. The contrast factor determines the degree of adjustment, where a factor of 1 leaves the image unchanged and values farther from 1 correspond to stronger modifications. In our experiments, the contrast factor is set to 2.0.

[8] https://github.com/luopengchen/GROW

[9] https://github.com/EnoalG/Guidance-Watermarking-for-Diffusion-Models

VAE Attack (VA) (Zhao et al., 2024). The watermarked image is reconstructed through a learned image-compression autoencoder, during which part of the image information may be discarded and the watermark signal weakened. The quality level determines the reconstruction fidelity, where a lower value corresponds to stronger compression and generally results in a stronger attack. In our experiments, we use Cheng2020-anchor[10] with quality level 3.

Diffusion Attack (DA) (Zhao et al., 2024). The watermarked image is first perturbed with diffusion noise and then reconstructed through the reverse diffusion process. The noise timestep determines the amount of injected noise, where a larger timestep introduces stronger corruption and generally leads to a stronger attack. In our experiments, we use Stable Diffusion v2.1-base with a noise timestep of 100 out of 1,000.

E Inversion for Watermark Detection

Inversion-based watermarking methods recover an estimate of the initial noise latent from a generated image and detect the watermark in the reconstructed latent space. In our experiments, Tree-Ring, RingID, and Gaussian Shading use DDIM inversion on SD-v2.1 and rectified-flow (RF) inversion on FLUX.1-schnell. In this section, we briefly describe the two inversion schemes and discuss why RF inversion can lead to reduced watermark robustness.

E.1 DDIM Inversion

DDIM defines a deterministic sampling trajectory for diffusion models. Let $\mathbf{x}_0$ denote the clean latent and $\mathbf{x}_t$ its noisy version at timestep t. Given the cumulative noise schedule $\bar{\alpha}_t$, the diffusion marginal can be written as

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{17}$$

At timestep t, an ϵ-prediction model estimates the noise as $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{c})$, where $\mathbf{c}$ denotes the text condition. The corresponding prediction of the clean latent is

$$\hat{\mathbf{x}}_0^{\,t} = \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{c})}{\sqrt{\bar{\alpha}_t}}. \tag{18}$$

Deterministic DDIM sampling then updates the latent from t_i to t_{i-1} as

$$\mathbf{x}_{t_{i-1}} = \sqrt{\bar{\alpha}_{t_{i-1}}}\hat{\mathbf{x}}_0^{\,t_i} + \sqrt{1-\bar{\alpha}_{t_{i-1}}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_{t_i}, t_i, \mathbf{c}). \tag{19}$$

DDIM inversion traverses this deterministic trajectory in the opposite direction. Starting from an encoded image latent $\hat{\mathbf{x}}_{t_0}$, the naive first-order inversion is

$$\hat{\mathbf{x}}_{t_{i+1}} = \sqrt{\bar{\alpha}_{t_{i+1}}}\hat{\mathbf{x}}_0^{\,t_i} + \sqrt{1-\bar{\alpha}_{t_{i+1}}}\boldsymbol{\epsilon}_\theta(\hat{\mathbf{x}}_{t_i}, t_i, \mathbf{c}), \tag{20}$$

which is repeated until an estimate of the initial noise $\hat{\mathbf{x}}_T$ is obtained. Although this approximation may accumulate discretization errors, it provides a stable deterministic inversion procedure for conventional diffusion models and is widely used by initial-noise watermarking methods such as Tree-Ring (Umrajkar & Singh, 2025).

For blind watermark detection, the original generation prompt is typically unavailable. We therefore perform DDIM inversion with an empty text prompt, following the standard Tree-Ring detection setting.

E.2 Rectified-Flow Inversion

Rectified-flow models describe generation through a continuous transport between a data latent $\mathbf{x}_0$ and a Gaussian noise sample $\boldsymbol{\epsilon}$. In FLUX, the transport path is typically parameterized by a linear interpolation between the data latent and Gaussian noise:

$$\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad t \in [0, 1], \tag{21}$$

[10] https://github.com/InterDigitalInc/CompressAI

which induces the ordinary differential equation

$$\frac{d\mathbf{x}_t}{dt} = \mathbf{v}_\theta(\mathbf{x}_t, t, \mathbf{c}), \tag{22}$$

where $\mathbf{v}_\theta$ is the learned conditional velocity field. With a decreasing sampling schedule $1 = t_N > \cdots > t_0 = 0$, first-order Euler sampling from noise to data is given by

$$\mathbf{x}_{t_{i-1}} = \mathbf{x}_{t_i} + (t_{i-1} - t_i)\mathbf{v}_\theta(\mathbf{x}_{t_i}, t_i, \mathbf{c}). \tag{23}$$

To recover the initial noise, RF inversion integrates the same conditional vector field from the data endpoint toward the noise endpoint. Starting from $\hat{\mathbf{x}}_{t_0}$, the first-order update can be written as

$$\hat{\mathbf{x}}_{t_{i+1}} = \hat{\mathbf{x}}_{t_i} + (t_{i+1} - t_i)\mathbf{v}_\theta(\hat{\mathbf{x}}_{t_i}, t_i, \mathbf{c}), \tag{24}$$

until $\hat{\mathbf{x}}_{t_N}$ provides an estimate of the initial noise latent. In practice, we follow the native FlowMatch Euler schedule of FLUX rather than assuming uniformly spaced timesteps. Unlike DDIM inversion, RF inversion does not rely on a diffusion $\bar{\alpha}_t$ schedule, but directly reverses the learned flow-matching trajectory. For blind detection, we again use an empty text condition.

E.3 Why RF Inversion Can Reduce Watermark Robustness

The robustness of initial-noise watermarks is inherently coupled with the accuracy of inversion. Taking Tree-Ring as an example, detection relies on recovering the initial noise and comparing its Fourier-domain watermark region with the predefined secret pattern. Therefore, errors introduced during inversion directly perturb the recovered watermark signal and reduce the statistical separability between watermarked and non-watermarked samples.

This issue can be more pronounced for RF-based models for several reasons.

First, RF inversion can be substantially more sensitive to the text condition. For an inversion step, let $\mathbf{e}_i$ denote the deviation from the original generation trajectory. A first-order error propagation can be expressed as

$$\mathbf{e}_{i+1} \approx (\mathbf{I} + \Delta t_i \mathbf{J}_i)\,\mathbf{e}_i + \Delta t_i \Delta \mathbf{v}_i^c + \boldsymbol{\rho}_i, \tag{25}$$

where $\mathbf{J}_i$ is the Jacobian of the velocity field with respect to the latent, $\Delta \mathbf{v}_i^c$ denotes the change in velocity caused by the mismatch between the original and inversion text conditions, and $\boldsymbol{\rho}_i$ contains numerical and model approximation errors. Consequently, an inaccurate text condition can continuously shift the reverse trajectory rather than introducing only a local error.

This effect is particularly relevant to blind watermark detection, where the original generation prompt is unknown and inversion is performed with an empty prompt. Umrajkar & Singh (2025) shows that FLUX inversion is considerably more sensitive to prompt guidance than DDIM inversion: providing the exact prompt substantially improves clean latent reconstruction, whereas prompt-free inversion yields larger recovery error. A plausible explanation is that FLUX adopts the MMDiT architecture, in which text and image representations interact more deeply throughout the generation process. Consequently, the learned velocity field is more strongly conditioned on the textual context, making accurate trajectory reconstruction more dependent on the original prompt. Therefore, using an empty prompt for blind detection introduces a larger trajectory mismatch during RF inversion and leads to greater error in the recovered initial noise.

Second, image transformations introduce an initial perturbation before inversion. Compression, blur, noise, or regeneration first alter the input image and hence its VAE-encoded latent. This perturbation enters Eq. 25 as a nonzero initial error and can be amplified along subsequent velocity evaluations. Prior experiments show that although RF inversion can accurately recover the noise of clean FLUX images, its reconstruction error increases sharply after image perturbations, leading to substantially poorer separability of Tree-Ring watermarks (Umrajkar & Singh, 2025). Moreover, providing the exact prompt offers little improvement once the image has been perturbed, indicating that attack-induced trajectory deviation cannot be corrected solely by better text conditioning.

Finally, RF models are designed to support efficient generation with relatively few integration steps. While this is advantageous for sampling, coarse first-order inversion uses larger integration intervals and is therefore more sensitive to errors in the estimated velocity field. This effect becomes

especially relevant for few-step models, where a small deviation at one step can induce a substantial change in the recovered initial latent.

Together, these factors explain why an inversion-based watermark can remain highly detectable on clean RF-generated images while degrading more rapidly under image transformations. The limitation does not arise from the Tree-Ring pattern alone; rather, watermark detection depends on accurately recovering the specific initial noise trajectory, making its robustness directly coupled to prompt availability, inversion accuracy, and the underlying generation dynamics.