

---

# Video Signature: In-generation Watermarking for Latent Video Diffusion Models

---

Yu Huang<sup>1\*</sup>, Junhao Chen<sup>1\*</sup>, Qi Zheng<sup>1</sup>, Hanqian Li<sup>1</sup>, Shuliang Liu<sup>1</sup>, Xuming Hu<sup>1,2†</sup>

<sup>1</sup>The Hong Kong University of Science and Technology (Guangzhou)

<sup>2</sup>The Hong Kong University of Science and Technology

{yhuang489, jchen024, qzheng219}@connect.hkust-gz.edu.cn

kyriehqli@hkust-gz.edu.cn

{shulianglyo, xuminghu97}@gmail.com

## Abstract

The rapid development of Artificial Intelligence Generated Content (AIGC) has led to significant progress in video generation but also raises serious concerns about intellectual property protection and reliable content tracing. Watermarking is a widely adopted solution to this issue, but existing methods for video generation mainly follow a post-generation paradigm, which introduces additional computational overhead and often fails to effectively balance the trade-off between video quality and watermark extraction. To address these issues, we propose Video Signature (VIDSIG), an in-generation watermarking method for latent video diffusion models, which enables implicit and adaptive watermark integration during generation. Specifically, we achieve this by partially fine-tuning the latent decoder, where Perturbation-Aware Suppression (PAS) pre-identifies and freezes perceptually sensitive layers to preserve visual quality. Beyond spatial fidelity, we further enhance temporal consistency by introducing a lightweight Temporal Alignment module that guides the decoder to generate coherent frame sequences during fine-tuning. Experimental results show that VIDSIG achieves the best overall performance in watermark extraction, visual quality, and generation efficiency. It also demonstrates strong robustness against both spatial and temporal tampering, highlighting its practicality in real-world scenarios.

## 1 Introduction

With the rapid advancement of Artificial Intelligence Generated Content (AIGC), remarkable progress has been achieved across various generative modalities, including texts [Brown et al., 2020, Touvron et al., 2023], images [Ho et al., 2020, Rombach et al., 2022, Peebles and Xie, 2023], audios [Huang et al., 2024a, Zhang et al., 2023a], and videos [Blattmann et al., 2023, Hong et al., 2022, Kong et al., 2024]. However, unauthorized use or misuse of these models can lead to significant risks, such as fake news and deepfake [Brundage et al., 2018, Breland, 2019, Zohny et al., 2023]. Among the various techniques proposed to address this issue, watermarking served as a promising solution for asserting ownership and enabling provenance tracking of generated content [Liang et al., 2024, Liu et al., 2024, Hu et al., 2025b].

Previous watermarking methods for video generation follow a post-generation paradigm that separates the generation and watermarking processes. As illustrated in Figure 1, the watermark is embedded into the generated video through an additional neural network applied after generation [Zhang et al., 2019, Fernandez et al., 2024]. This paradigm not only introduces extra computational overhead but also often fails to strike a balance between visual quality and watermark extraction accuracy, making it less reliable and effective in practice. Recent efforts have explored in-generation methods, embedding

---

\*Equal Contribution

†Corresponding author

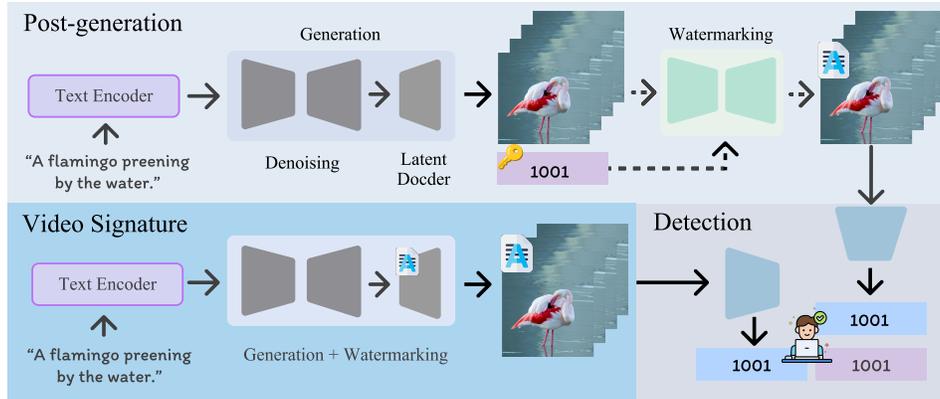


Figure 1: Comparison between post-generation watermarking and Video Signature. Post-generation methods explicitly embed the watermark after video generation, requiring an additional watermarking network (dashed line). In contrast, VIDSIG implicitly integrates the watermark into the generation process by fine-tuning the latent decoder.

watermarks during the generative process. Tree-Ring [Wen et al., 2023] and Gaussian shading [Yang et al., 2024] embed watermarks into the initial noise in image generation. VideoShield [Hu et al., 2025a] extends Gaussian shading to the video domain, embedding multi-bit watermarks to each frame. These methods then leverage DDIMs [Song et al., 2020] to recover a predicted initial noise, subsequently detecting the watermark signals. However, these methods still suffer from the extremely high computational cost for watermark embedding and extraction. Additionally, some methods fine-tune the latent decoder to embed the watermark during the mapping from latent space to image space [Fernandez et al., 2023, Rezaei et al., 2024, Kim et al., 2024], and extract the watermark using an external extractor aligned with the fine-tuning process. These methods enable watermark insertion with negligible latency. However, directly applying such methods to video generation overlooks the temporal consistency of video content, and exhaustively fine-tuning the entire latent decoder often leads to noticeable visual artifacts

To address the aforementioned problems, we propose **Video Signature (VIDSIG)**, a framework that integrates the watermark into *each frame* of the generated video without additional modification to the model architecture and initial Gaussian noise. We embed the watermark message into each frame due to the vulnerability of video data to temporal attacks, such as frame dropping or shuffling. Specifically, we follow the pipeline of the training-based method to fine-tune the latent decoder to embed invisible watermarks into the video during generation. Before fine-tuning, we adopt a **Perturbation-Aware Suppression (PAS)** search algorithm to pre-identify and freeze perceptually sensitive layers to preserve visual quality. Furthermore, to capture temporal consistency, we introduce a straightforward but effective **Temporal Alignment** module that guides the decoder to produce coherent frame sequences during fine-tuning. Our contributions are summarized as follows:

- We highlight the challenges of watermarking in video generative models and propose Video Signature, an in-generation watermarking framework that embeds watermark messages directly into the video generation process by fine-tuning the latent decoder.
- To preserve visual quality, we introduce Perturbation-Aware Suppression (PAS), a search algorithm to efficiently identify perceptually sensitive layers, and we introduce a Temporal Alignment module that enforces the inter-frame coherence.
- Experimental results show that VIDSIG achieves the best overall performance in watermark extraction, visual quality, and generation efficiency. It also demonstrates strong robustness against both spatial and temporal tampering, highlighting its practicality in real-world scenarios.

## 2 Related Work

**Diffusion-Based Video Generation** Recent video generation models are mainly built on Latent Diffusion Models (LDMs) [Rombach et al., 2022], ModelScope [Wang et al., 2023], which uses a 2D VAE to compress each frame into the latent space, employs a U-Net based architecture for

denoising across both spatial and temporal dimensions. Stable Video Diffusion [Blattmann et al., 2023], which is also built on U-Net, uses a 3D VAE to encode the entire video into the latent space, allowing the models to naturally capture spatial and temporal relationships simultaneously. Models like Latte [Ma et al., 2024], Open-sora [Zheng et al., 2024], ViDu [Bao et al., 2024], uses the more scalable DiT-based [Peebles and Xie, 2023] architecture to learn the denoising process. The release of Sora by OpenAI [Brooks et al., 2024] and Hunyuan Video by Tencent [Kong et al., 2024] has enabled the generation of longer and higher-quality videos, but also raises critical concerns about the unauthorized use or misuse of such models [Brundage et al., 2018, Zohny et al., 2023].

**Watermarking for Diffusion Models** Watermarking methods for diffusion models can be categorized into two paradigms: post-generation and in-generation. Post-generation methods embed watermark signals into images or videos after the content has been synthesized. Early approaches embed watermarks in the frequency domain [O’Ruanidh and Pun, 1997, Cox et al., 1996] or leverage SVD-based matrix operations [Chang et al., 2005]. Recent post-generation methods typically employ deep neural networks to embed and extract watermark information from generated content [Zhu et al., 2018, Zhang et al., 2019, 2023b, Fernandez et al., 2024].

On the other hand, in-generation methods embed the watermark into the images or videos during the generation process. Wen et al. [2023] proposes Tree-ring that embeds a specific pattern into the initial Gaussian noise to achieve a 0-bit watermark, while Gaussian-shading [Yang et al., 2024] embeds a multi-bit watermark into the initial Gaussian noise. Videoshield [Hu et al., 2025a] extends Gaussian shading to the video domain, marking the first in-generation watermarking method for video synthesis. However, these approaches require DDIM inversion to recover the initial Gaussian noise for watermark detection, resulting in extremely high computational cost for watermark insertion and extraction. Another line of work embeds watermarks by fine-tuning the video decoder and employs an auxiliary decoder for watermark extraction, significantly reducing the overall runtime. For instance, Fernandez et al. [2023] proposes Stable Signature, which finetunes the latent decoder of the LDMs to embed a multi-bit watermark message into the images. Similar to that, Lawa [Rezaei et al., 2024] and Wouaf [Kim et al., 2024] finetune the latent decoder with an additional message encoder that embeds the multi-bit message into the images during generation. Zhang et al. [2024] propose Editguard, which combines tamper localization with the watermark for diffusion models. However, these image-based methods, when naively applied to video generation in a frame-wise manner, fail to account for the temporal consistency crucial to video data. In contrast to prior video watermarking approaches and naive frame-wise adaptations of image watermarking methods, our method bypasses the need for DDIMs inversion and embeds watermarks during generation by selectively fine-tuning perceptually insensitive layers of the latent decoder, while a temporal alignment module enforces consistency across frames.

### 3 Methodology

In this section, we give a detailed description of Video Signature. Specifically, in Section 3.1, we give an overview of our training pipeline, and in Section 3.2, we discuss how PAS works and search the perceptually sensitive layers, and in Section 3.3, we detail the training objective for each module.

#### 3.1 Overview of the Training Pipeline

Our training pipeline is shown in Figure 2. We fine-tune the latent decoder  $\mathcal{D}$  to embed a multi-bit watermark message into each frame of the generated video. The latent encoder  $\mathcal{E}$ , which is frozen during training, maps the input video  $\mathbf{v} \in \mathbb{R}^{f \times c \times H \times W}$  into a latent representation  $\mathbf{z} \in \mathbb{R}^{f \times c' \times h \times w}$ , where  $f$  denotes the number of frames and  $c'$  denotes the channels of the latent space. The latent decoder  $\mathcal{D}$  then reconstructs the video as  $\hat{\mathbf{v}} = \mathcal{D}(\mathbf{z}) \in \mathbb{R}^{f \times c \times H \times W}$ . The reconstructed video  $\hat{\mathbf{v}}$  is subsequently fed to a pre-trained watermark extractor  $\mathcal{W}$ , which extracts a fixed-length binary message from each frame, denoted as  $\hat{\mathbf{m}} = \mathcal{W}(\hat{\mathbf{v}}) \in \mathbb{R}^{f \times k}$ , where  $k$  is the length of the embedded watermark per frame. The ultimate goal of this training pipeline is to obtain a watermarked latent decoder  $\mathcal{D}'$ , which can generate high-quality videos with imperceptible and robust watermarks directly, without additional modification during inference. The goal of our training pipeline is summarized as:

$$\theta' = \theta + \delta, \quad \delta^* = \arg \max_{\delta} \log p(\mathbf{m} \mid \mathcal{W}(\theta'(\mathbf{z}))) \quad \text{s.t. } \mathcal{P}(\theta(\mathbf{z}), \theta'(\mathbf{z})) \leq \varepsilon, \quad (1)$$

where  $\theta'$  and  $\theta$  denote the parameters of the watermarked latent decoder  $\mathcal{D}'$  and the original decoder  $\mathcal{D}$ , respectively. Here  $\mathbf{z}$  is a latent vector,  $\mathbf{m}$  is the watermark message, and  $\mathcal{P}$  is a perceptual distance

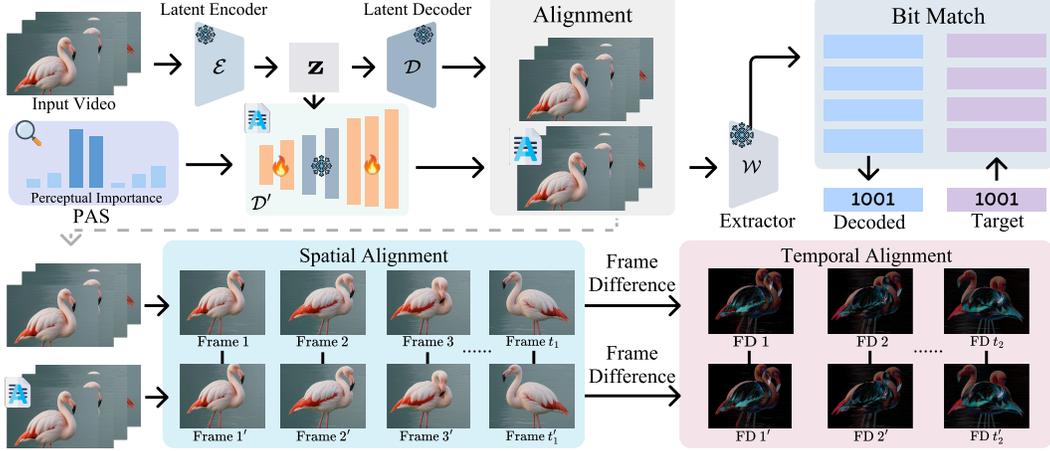


Figure 2: The training pipeline of Video Signature. (1) Given an input video, we first encode it into a latent representation and decode it with a frozen latent decoder. (2) Before optimization, the proposed PAS module searches the most perceptually sensitive layers and freezes them. (3) The watermarked decoder  $\mathcal{D}'$  is then optimized to embed a secret key into the generated video with three different objectives, pixel-level alignment, inter-frame level alignment and bit match.

metric (e.g., MSE) that evaluates the visual similarity between the watermarked video and its clean counterpart.

### 3.2 PAS: Perturbation-Aware Suppression

Existing watermarking work typically fine-tunes *all* decoder parameters [Fernandez et al., 2023, Rezaei et al., 2024, Kim et al., 2024], achieving high extraction accuracy at the cost of perceptual artifacts. To mitigate this issue, we update only those layers whose impact on perceptual quality is negligible. We propose **Perturbation-Aware Suppression (PAS)**, which performs a straightforward but efficient search to identify layers with minimal perceptual impact on the output, enabling effective watermark embedding with minimal visual degradation. Specifically, for each layer  $L_j$  with parameters  $\theta_j$ , we inject an isotropic Gaussian noise  $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ :

$$\theta'_j = \theta_j + \epsilon. \quad (2)$$

Let  $\hat{\mathbf{v}}_i^{(0)} = \mathcal{D}_{\theta}(\mathbf{z}_i)$  be the reference output of latent  $\mathbf{z}_i$  and  $\hat{\mathbf{v}}_i^{(j)} = \mathcal{D}_{\theta'}(\mathbf{z}_i)$  denotes the output after perturbing  $L_j$ . The perceptual impact of layer  $L_j$  is then estimated by:

$$s_j = \frac{1}{B} \sum_{i=1}^B \mathcal{P}(\hat{\mathbf{v}}_i^{(j)}, \hat{\mathbf{v}}_i^{(0)}), \quad (3)$$

where  $B$  is the number of latent samples. Finally, the layer set for fine-tuning is selected as  $\mathcal{L}_{\text{ft}} = \{L_j \mid s_j < \tau\}$ , with a threshold  $\tau$ . The full procedure is detailed in Algorithm 1.

### 3.3 Training Objectives

As illustrated in Figure 2, our training framework is designed to embed a binary watermark into the video generation process while preserving high perceptual quality and temporal consistency. The

---

#### Algorithm 1 Perturbation-Aware Suppression

---

**Input:** Frozen Decoder  $D = \{L_1, \dots, L_n\}$ , Layer Parameter  $\theta = \{\theta_1, \dots, \theta_n\}$ , Latent batch  $\{\mathbf{z}_i\}_{i=1}^B$ , Perceptual distance metric  $\mathcal{P}(\cdot, \cdot)$ , Noise scale  $\sigma$ , Threshold  $\tau$

**Output:** Selected layer set  $\mathcal{L}_{\text{ft}}$

Generate reference outputs  $\hat{\mathbf{v}}_i^{(0)} = \mathcal{D}(\mathbf{z}_i)$  for all  $\mathbf{z}_i$

**for** each layer  $L_j \in \mathcal{D}$  **do**

$\theta_j^{\text{ori}} \leftarrow \theta_j$

$\theta_j \leftarrow \theta_j + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$

**for** each  $\mathbf{z}_i$  in batch **do**

$\hat{\mathbf{v}}_i^{(j)} \leftarrow \mathcal{D}(\mathbf{z}_i)$

$d_i^{(j)} \leftarrow \mathcal{P}(\hat{\mathbf{v}}_i^{(j)}, \hat{\mathbf{v}}_i^{(0)})$

**end for**

$s_j \leftarrow \frac{1}{B} \sum_{i=1}^B d_i^{(j)}$

$\theta_j \leftarrow \theta_j^{\text{ori}}$

**end for**

$\mathcal{L}_{\text{ft}} \leftarrow \{L_j \mid s_j < \tau\}$

**return**  $\mathcal{L}_{\text{ft}}$

---

overall optimization involves two objectives: the watermark extraction and the visual alignment in the spatial and temporal domains.

**Watermark Extraction** To ensure accurate watermark embedding across the entire video, we employ two complementary loss terms: a frame-wise watermark loss and a video-level watermark loss. Given a target binary watermark message  $\mathbf{m} \in \{0, 1\}^k$ , the watermarked video  $\hat{\mathbf{v}} \in \mathbb{R}^{f \times c \times H \times W}$  is generated by decoding latent vector  $\mathbf{z}$  through the fine-tuned decoder  $\mathcal{D}'$ . A pretrained extractor  $\mathcal{W}$  predicts a soft watermark vector  $\hat{\mathbf{m}}_t \in [0, 1]^k$  from each frame  $\hat{\mathbf{v}}_t$ . We directly supervise each predicted frame-wise message with the target watermark using binary cross entropy:

$$\mathcal{L}_{\text{frame}} = -\frac{1}{f} \sum_{t=1}^f \sum_{i=1}^k [m_i \log \hat{m}_{t,i} + (1 - m_i) \log(1 - \hat{m}_{t,i})]. \quad (4)$$

To improve the global consistency of watermark extraction, we aggregate predictions across all frames to form a soft message vector and compute a global-watermark loss:

$$\bar{\mathbf{m}} = \frac{1}{f} \sum_{t=1}^f \hat{\mathbf{m}}_t, \quad \mathcal{L}_{\text{video}} = -\sum_{i=1}^k [m_i \log \bar{m}_i + (1 - m_i) \log(1 - \bar{m}_i)]. \quad (5)$$

Eventually, the final watermark loss is a weighted sum  $\mathcal{L}_{\text{wm}} = \alpha_1 \mathcal{L}_{\text{frame}} + \alpha_2 \mathcal{L}_{\text{video}}$ , where  $\alpha_1, \alpha_2 > 0$  are hyperparameters balancing local (frame-level) and global (video-level) watermarks to enhance the watermark extraction performance. In this paper, we set  $\alpha_1 = \alpha_2 = 1$  as default. During inference, the extractor first predicts the bit message for each frame, then obtains the watermark for the entire video by majority voting.

**Visual Alignment** To ensure the watermarked video remains visually similar to the non-watermarked video, we conduct a spatial alignment between the frame-wise outputs of the original decoder  $\mathcal{D}$  and the watermarked decoder  $\mathcal{D}'$ . Let  $\mathbf{v}$  and  $\hat{\mathbf{v}}$  be the reconstructed non-watermarked and watermarked videos, respectively, both of shape  $f \times c \times H \times W$ . The perceptual loss in the spatial domain is defined as  $\mathcal{L}_{\text{spatial}} = \frac{1}{f} \sum_{t=1}^f \mathcal{D}_{\text{sim}}(\hat{\mathbf{v}}_t, \mathbf{v}_t)$ , where  $\mathcal{D}_{\text{sim}}(\cdot, \cdot)$  denotes a generic frame-level similarity metric, such as MAE, MSE, and LPIPS [Zhang et al., 2018]. In this paper, we use Watson-VGG [Czolbe et al., 2020], an improved version of LPIPS, as the perceptual loss.

To further align the visual similarity in the temporal domain, we introduce a straightforward but effective module, which we refer to as **Temporal Alignment**. We simply align the **Inter-Frame Dynamics** between  $\mathbf{v}$  and  $\hat{\mathbf{v}}$ . Specifically, we compute frame-wise differences  $\Delta_t = \mathbf{v}_{t+1} - \mathbf{v}_t$  and  $\hat{\Delta}_t = \hat{\mathbf{v}}_{t+1} - \hat{\mathbf{v}}_t$ , as shown in Figure 2 and define:

$$\mathcal{L}_{\text{temporal}} = \frac{1}{f-1} \sum_{t=1}^{f-1} \mathcal{D}_{\text{sim}}(\hat{\Delta}_t, \Delta_t). \quad (6)$$

We continue to use Watson-VGG as the distance metric for temporal alignment. Additional evaluation results based on other metrics are reported in Section 4.5. Thus, the total training loss is a weighted combination of the three objectives:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{wm}} + \lambda_2 \mathcal{L}_{\text{spatial}} + \lambda_3 \mathcal{L}_{\text{temporal}}, \quad (7)$$

where  $\lambda_1, \lambda_2$ , and  $\lambda_3$  are hyperparameters controlling the trade-off between the watermark accuracy and the fidelity of the video. In this paper, we set  $\lambda_1 = 1$ , and  $\lambda_2 = 0.2$  as default.

## 4 Experiment

### 4.1 Experiment Settings

**Implementation Details** We evaluate our method on two video generation models: the text-to-video (T2V) model ModelScope (MS) [Wang et al., 2023], and the image-to-video (I2V) model Stable Video Diffusion (SVD) [Blattmann et al., 2023]. During training, we fine-tune only the remained layers of the latent decoder after PAS, which uses MSE as the perceptual distance metric. We use the pretrained watermark extractor provided by Fernandez et al. [2023], which can extract a 48-bit binary

Table 1: Performance comparison of watermarking methods on SVD and MS. Video Quality refers to the average value of the four metrics from VBench. PSNR, SSIM, LPIPS, and tLP are calculated between the watermarked video and the original video (non-watermarked video). **Bold** indicates the best result within each model and method type (post-generation or in-generation). Arrows denote whether higher ( $\uparrow$ ) or lower ( $\downarrow$ ) values indicate better performance.

Model	Method	Bit Accuracy $\uparrow$	$T_i \downarrow$	$T_e \downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	tLP $\downarrow$	Video Quality $\uparrow$
MS	<i>Post-generation methods</i>								
	RivaGAN	0.938	0.711	0.159	34.973	0.947	0.117	0.012	0.893
	VideoSeal	<b>0.975</b>	<b>0.121</b>	<b>0.036</b>	<b>35.113</b>	<b>0.947</b>	<b>0.083</b>	<b>0.009</b>	<b>0.893</b>
	<i>In-generation methods</i>								
	StableSig	0.995	<b>0.000</b>	<b>0.008</b>	29.448	0.799	0.165	<b>0.008</b>	0.893
	VideoShield	<b>1.000</b>	26.994	6.224	16.377	0.236	0.585	0.055	<b>0.894</b>
VIDSIG (Ours)	0.992	<b>0.000</b>	<b>0.008</b>	<b>30.523</b>	<b>0.840</b>	<b>0.151</b>	0.009	0.893	
SVD	<i>Post-generation methods</i>								
	RivaGAN	0.886	0.686	0.162	35.561	0.963	0.069	0.004	0.871
	VideoSeal	<b>0.979</b>	<b>0.049</b>	<b>0.018</b>	<b>35.887</b>	<b>0.966</b>	<b>0.060</b>	<b>0.003</b>	<b>0.871</b>
	<i>In-generation methods</i>								
	StableSig	0.998	<b>0.000</b>	<b>0.005</b>	30.080	0.906	0.100	<b>0.003</b>	<b>0.873</b>
	VideoShield	0.990	28.586	19.813	10.812	0.378	0.525	0.116	0.867
VIDSIG (Ours)	<b>0.999</b>	<b>0.000</b>	<b>0.005</b>	<b>31.662</b>	<b>0.924</b>	<b>0.091</b>	<b>0.003</b>	<b>0.873</b>	

message from a single image. Optimization is performed using the AdamW optimizer [Loshchilov and Hutter, 2017], with a learning rate of  $5 \times 10^{-4}$ . We run all of our experiments on an NVIDIA A800-SXM4-80GB GPU. We give additional implementation details in our Appendix A.2.

**Datasets** We download a subset of the OpenVid-1M dataset [Nan et al., 2024] for training. Specifically, we randomly sample 10,000 videos from the downloaded dataset to fine-tune the latent decoder. Each training video contains 8 frames sampled at a frame interval of 8. For evaluation, we select 50 prompts from the test set of VBench [Huang et al., 2024b], covering five categories: Animal, Human, Plant, Scenery, and Vehicles (10 prompts each). For the T2V task, we generate four videos per prompt using four fixed seeds, resulting in 200 videos. For the I2V task, we first generate images using Stable Diffusion 2.1 [Rombach et al., 2022] based on the same prompts, and generate videos conditioned on these images using the same fixed seeds. In total, for both T2V and I2V tasks, we generate 200 videos, each consisting of 16 frames with a resolution of  $512 \times 512$ . The inference steps of the two models are set to 25 and 50, respectively.

**Baseline** We compare our method with 4 watermarking methods: RivanGAN [Zhang et al., 2019], VideoSeal [Fernandez et al., 2024], Stable Signature [Fernandez et al., 2023], VideoShield [Hu et al., 2025a]. Among these methods, RivanGAN and VideoSeal are post-generation methods, Stable Signature is a train-based in-generation method for image watermarking, and VideoShield is a train-free in-generation method for video watermarking.

**Metrics** We calculate the True Positive Rate (TPR) corresponding to a fixed False Positive Rate (FPR) to evaluate the watermark detection. Meanwhile, we use Bit Accuracy to evaluate the watermark extraction accuracy. To evaluate the visual quality of generated videos, we leverage four official evaluation metrics from VBench [Huang et al., 2024b], including: Subject Consistency, Background Consistency, Motion Smoothness and Imaging Quality. We also use four standard perceptual quality metrics for evaluating the distortion of the generated videos: PSNR, SSIM [Wang et al., 2004], LPIPS [Zhang et al., 2018], and its extension tLP [Chu et al., 2020]. To assess the efficiency of

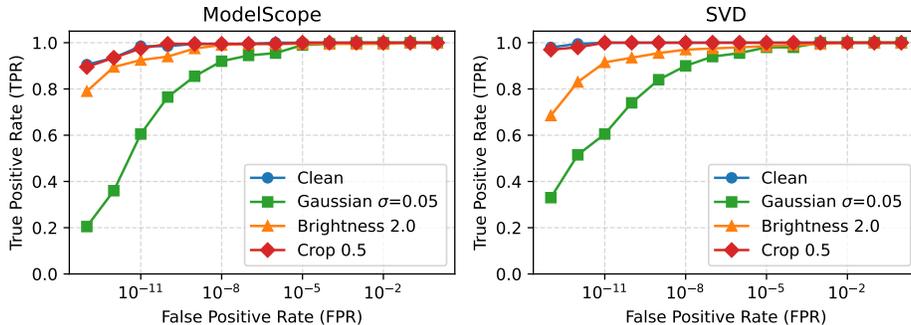


Figure 3: Watermark detection of Video Signature.

Table 2: Performance of VIDSIG in Latte transferred from ModelScope (MS) and SVD. The inference step is set to 50 as default, prompts and generators follow the same settings in 4.1. The True Positive Rate is reported under  $FPR = 1 \times 10^{-6}$ .

Decoder	Bit Accuracy	TPR	PSNR	SSIM	LPIPS	tLP
Latte <sub>MS</sub>	0.998	1.000	31.073	0.875	0.133	0.007
Latte <sub>SVD</sub>	0.998	1.000	29.564	0.839	0.148	0.020

watermarking methods, we evaluate the insertion and extraction time, denoted as  $T_i$  and  $T_e$ . We provide the details of these metrics in Appendix B.

## 4.2 Main Results

**Watermark Detection** We fix the FPR from  $10^{-13}$  to  $10^0$  and compute the true positive rate; the results are shown in Figure 3. It’s seen that VIDSIG maintains an almost perfect TPR across all tampering types even at extremely low FPR ( $10^{-11}$ ), and achieves a TPR above 0.95 under additive Gaussian noise with  $FPR = 10^{-6}$ , which is still a promising result.

**Comparison to Baselines** Table 1 presents the performance of VIDSIG and other baseline methods. The findings are summarized as follows: **(i)** Post-generation methods deliver the poorest extraction accuracy on both tasks, even their best case, VideoSeal on SVD, reaches only 0.979 and confers no perceptual benefit, underscoring an unfavorable accuracy–fidelity trade-off. **(ii)** By contrast, in-generation methods achieve a nearly perfect (up to 1 for T2V and 0.999 for I2V) extraction accuracy in both tasks and maintain a reasonable visual quality. The PSNR, SSIM, etc, are lower, but the VBench scores are comparable and even higher. **(iii)** In-generation methods, except for VideoShield, achieve a negligible latency for watermark embedding and the lowest cost for watermark detection.

VIDSIG surpasses other baseline methods by **(i)** matching or exceeding their near-perfect bit accuracy (0.992 for T2V task and 0.999 for I2V task) and achieves the best or near-best VBench scores on both tasks. **(ii)** The highest PSNR/SSIM and the lowest LPIPS/tLP values among in-generation methods reveal that VIDSIG introduces minimal visual artifacts compared to other in-generation methods. **(iii)** VIDSIG attains the lowest insertion and extraction computational overhead of all methods and, while matching Stable Signature in runtime, surpasses it on nearly every evaluation metric.

## 4.3 Transferability

We further investigate the transferability of VIDSIG. We transfer the fine-tuned latent decoder of ModelScope and SVD to a completely new model and evaluate whether the embedded watermark can still be successfully extracted. Specifically, we substitute the original latent decoder of Latte [Ma et al., 2024], a DiT-based model, with our fine-tuned watermarked latent decoder. The results are shown in Table 2 and Figure 4.

The results show that VIDSIG maintains a high extraction accuracy and perfect TPR even in cross-model scenarios, demonstrating strong transferability. This property suggests that the watermark signal is not tightly coupled to a specific model, but instead relies on generalizable latent-space manipulation, making VIDSIG more versatile in real-world deployment.

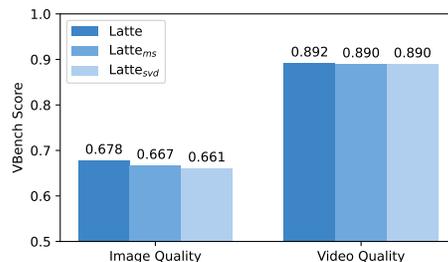


Figure 4: VBench score of VIDSIG in Latte transferred from ModelScope (MS) and SVD. Latte refers to the original model. Image Quality is one of the four metrics we have mentioned in Section 4.1, and Video Quality is the same as that in Table 1.

Table 3: Ablation results of PAS and TA modules on ModelScope with consistent experiment settings. **Bold** indicate the best performance, and underlined indicate the second best.

Configuration	Bit Accuracy $\uparrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	tLP $\downarrow$	Video Quality $\uparrow$
w/o ALL	0.990	28.565	0.786	0.182	0.011	0.889
w/ TA	0.991 (+0.001)	<u>29.793</u> (+1.228)	<u>0.815</u> (+0.029)	0.164 (−0.018)	0.010 (−0.001)	<b>0.894</b> (+0.005)
w/ PAS	<b>0.993</b> (+0.003)	29.762 (+1.197)	0.814 (+0.028)	<u>0.163</u> (−0.019)	<b>0.009</b> (−0.002)	0.892 (+0.003)
w/ ALL	<u>0.992</u> (+0.002)	<b>30.523</b> (+1.958)	<b>0.840</b> (+0.054)	<b>0.151</b> (−0.031)	<b>0.009</b> (−0.002)	<u>0.893</u> (+0.004)

Table 4: Bit Accuracy under different temporal tampering: FD (Frame Drop), FS (Frame Swap), FI (Frame Insert), FIG (Frame Insert Gaussian), FA (Frame Average, where n refers to the average length), the final column denotes the average bit accuracy, We give a detailed information for these tampering methods in Appendix A.4 (The detection of VideoShield only works when the number of frames in the generated video is exactly equal to that in the tampered video). **Bold** and underlined indicate the same as above.

Model	Method	Benign	FD	FS	FI	FIG	FA (n=3)	FA (n=5)	FA (n=7)	FA (n=9)	Avg
MS	RivaGAN	0.938	0.938	0.938	0.938	0.935	0.938	0.938	0.937	0.936	0.937
	VideoSeal	0.975	0.974	0.975	0.974	0.975	0.972	0.972	0.967	0.958	0.971
	StableSig	0.995	<b>0.995</b>	0.995	<b>0.995</b>	0.964	0.992	0.988	0.984	0.980	<u>0.988</u>
	VideoShield	<b>1.000</b>	-	<b>1.000</b>	-	-	-	-	-	-	-
	VIDSIG (ours)	0.992	0.992	0.992	0.992	<b>0.992</b>	<b>0.992</b>	<b>0.992</b>	<b>0.992</b>	<b>0.992</b>	<b>0.992</b>
SVD	RivaGAN	0.886	0.886	0.886	0.885	0.881	0.883	0.882	0.880	0.873	0.882
	VideoSeal	0.979	0.977	0.979	0.978	<u>0.979</u>	0.975	0.973	0.971	0.965	0.975
	StableSig	<u>0.998</u>	<u>0.998</u>	<u>0.998</u>	<u>0.998</u>	0.966	<u>0.995</u>	<u>0.989</u>	<u>0.980</u>	0.972	<u>0.988</u>
	VideoShield	0.990	-	0.964	-	-	-	-	-	-	-
	VIDSIG (ours)	<b>0.999</b>									

#### 4.4 Robustness

**Temporal Tamper** Table 4 presents the extraction accuracy of different watermarking methods under several temporal perturbations and the benign setting. Overall, VIDSIG consistently achieves near-perfect extraction accuracy across all attack types. For the T2V task, VIDSIG matches or closely follows the best-performing method (VideoShield) while requiring significantly less computational cost (see Table 1). It’s noticed that the watermark detection of VideoShield strictly requires that the video remains the same resolution and number of frames after temporal tampered, which limits its application. For the I2V task, VIDSIG achieves the highest extraction accuracy (0.999) across all perturbations, outperforming all baselines. This resilience to temporal tampering chiefly arises from embedding watermarks in every frame and aggregating the detections through majority voting.

**Spatial Tamper** We further evaluate the resilience of VIDSIG against various types and intensities of spatial tampering, as illustrated in Figure 5. We find that VIDSIG remains effective under most perturbations. In particular, it maintains robust performance within a certain range when subjected to Gaussian noise, Gaussian blur, and Salt and Pepper Noise. It is worth noting that no additional robustness enhancement techniques were applied during fine-tuning. These results demonstrate the robustness of VIDSIG against both temporal and spatial tampering, highlighting its potential for real-world deployment.

#### 4.5 Ablation Study

**Effectiveness of PAS and TA** Table 3 presents the ablation study evaluating the contributions of the two key components in our framework: Perturbation-Aware Suppression (PAS) and Temporal Alignment (TA). We can see that each component individually yields substantial gains across all evaluation metrics, and their integration attains the highest overall performance—with

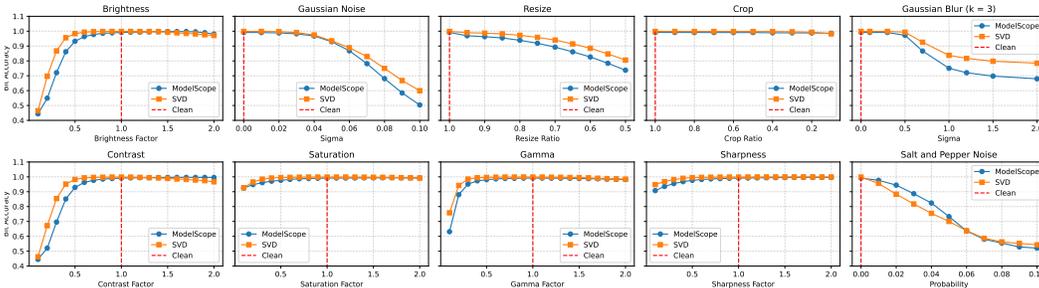


Figure 5: Bit accuracy under different spatial tampering. The attack is applied to each frame.

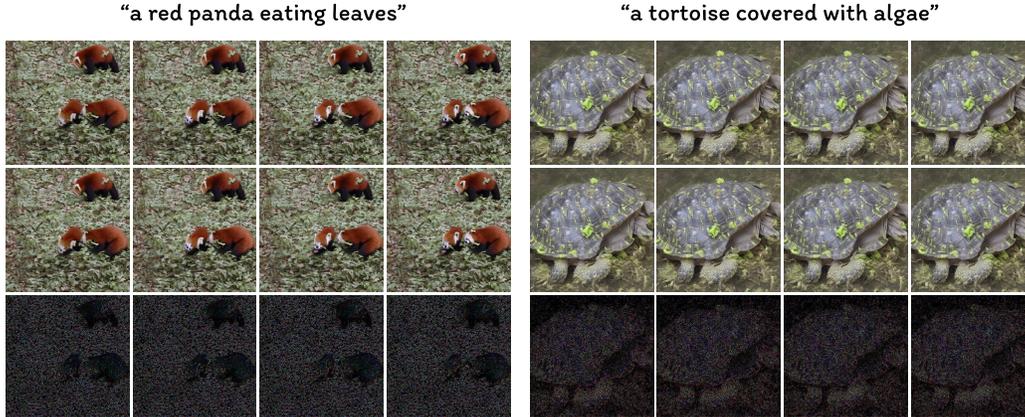


Figure 7: Qualitative comparison of the original and watermarked videos. The first row is the original video, the second row is the video generated by VIDSIG, and the third row refers to the pixel-wise difference ( $\times 10$ ).

only a marginal decrease in bit accuracy—thereby confirming that PAS and TA are complementary and jointly indispensable for realizing the full potential of the proposed framework.

**Different Perceptual Metric for TA** We then evaluate three different temporal alignment strategies—MAE, MSE, and Watson-VGG (we use VGG for abbreviation in the following discussion)—to identify the most effective formulation for preserving temporal consistency. The results are shown in Figure 6. It’s seen that MAE and MSE produce slightly higher bit accuracy yet noticeably degrade video quality. In contrast, the VGG loss with  $\lambda_3 = 0.2$  attains comparable bit accuracy while substantially improving perceptual quality, thereby offering the most favorable overall trade-off across all metrics. We attribute this superiority to the smoother, perceptually informed gradients supplied by the VGG loss during fine-tuning.

**Qualitative Analysis** The qualitative comparison between the original and watermarked videos is shown in Figure 7. It’s clear to see that the watermark manifests only as imperceptible high-frequency perturbations, with no structured artefacts or semantic drift observable across time. It’s also noticed that in real-world applications, all of the generated videos will have automatically embedded watermarks, thus there will be no reference videos for comparison. The VBench scores in Table 1 reveal that VIDSIG will not degrade the video quality, even though metrics like PSNR and SSIM are lower than post-generation methods.

## 5 Conclusion and Limitation

In this paper, we propose Video Signature, an in-generation watermarking method for video generative models. Compared to other methods, VIDSIG exceeds the post-generation methods in both extraction accuracy and video quality, and is more effective and flexible than VideoShield. However, our work still has some limitations: (i) we introduce an additional fine-tuning stage before the public release of the generative model; (ii) the current implementation focuses on embedding fixed-length binary signatures per frame and does not yet support dynamic or content-adaptive watermark payloads.

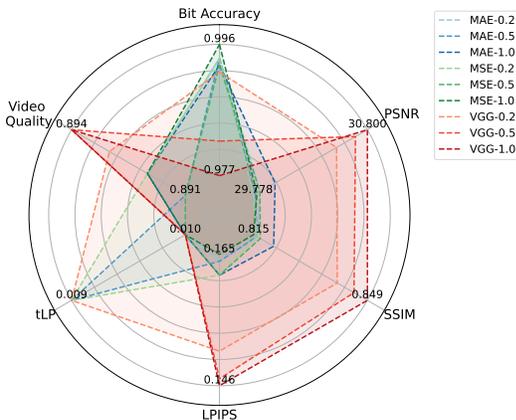


Figure 6: Ablation study on different temporal alignment metrics under varying  $\lambda_3$  values. Color hues represent metric type, and color intensity indicates increasing  $\lambda_3$ .

## References

- Fan Bao, Chendong Xiang, Gang Yue, Guande He, Hongzhou Zhu, Kaiwen Zheng, Min Zhao, Shilong Liu, Yaole Wang, and Jun Zhu. Vidu: a highly consistent, dynamic and skilled text-to-video generator with diffusion models. *arXiv preprint arXiv:2405.04233*, 2024.
- Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.
- Ali Breland. The bizarre and terrifying case of the “deepfake” video that helped bring an african nation to the brink. *Mother Jones*, 1, 2019.
- Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1:8, 2024.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- Miles Brundage, Shahar Avin, Jack Clark, Helen Toner, Peter Eckersley, Ben Garfinkel, Allan Dafoe, Paul Scharre, Thomas Zeitzoff, Bobby Filar, et al. The malicious use of artificial intelligence: Forecasting, prevention, and mitigation. *arXiv preprint arXiv:1802.07228*, 2018.
- Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.
- Chin-Chen Chang, Piyu Tsai, and Chia-Chen Lin. Svd-based digital image watermarking scheme. *Pattern Recognition Letters*, 26(10):1577–1586, 2005.
- Mengyu Chu, You Xie, Jonas Mayer, Laura Leal-Taixé, and Nils Thuerey. Learning temporal coherence via self-supervision for gan-based video generation. *ACM Transactions on Graphics (TOG)*, 39(4):75–1, 2020.
- Ingemar J Cox, Joe Kilian, Tom Leighton, and Talal Shamoan. Secure spread spectrum watermarking for images, audio and video. In *Proceedings of 3rd IEEE international conference on image processing*, volume 3, pages 243–246. IEEE, 1996.
- Steffen Czolbe, Oswin Krause, Ingemar Cox, and Christian Igel. A loss function for generative neural networks based on watson’s perceptual model. *Advances in Neural Information Processing Systems*, 33:2051–2061, 2020.
- Yuming Fang, Hanwei Zhu, Yan Zeng, Kede Ma, and Zhou Wang. Perceptual quality assessment of smartphone photography. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3677–3686, 2020.
- Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22466–22477, 2023.
- Pierre Fernandez, Hady Elsahar, I Zeki Yalniz, and Alexandre Mourachko. Video seal: Open and efficient video watermarking. *arXiv preprint arXiv:2412.09492*, 2024.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022.
- Runyi Hu, Jie Zhang, Yiming Li, Jiwei Li, Qing Guo, Han Qiu, and Tianwei Zhang. Videoshield: Regulating diffusion-based video generation models via watermarking. *arXiv preprint arXiv:2501.14195*, 2025a.
- Xuming Hu, Hanqian Li, Jungang Li, and Aiwei Liu. Videomark: A distortion-free robust watermarking framework for video diffusion models. *arXiv preprint arXiv:2504.16359*, 2025b.
- Rongjie Huang, Mingze Li, Dongchao Yang, Jiatong Shi, Xuankai Chang, Zhenhui Ye, Yuning Wu, Zhiqing Hong, Jiawei Huang, Jinglin Liu, et al. Audiogpt: Understanding and generating speech, music, sound, and talking head. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23802–23804, 2024a.

- Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024b.
- Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5148–5157, 2021.
- Changhoon Kim, Kyle Min, Maitreya Patel, Sheng Cheng, and Yezhou Yang. Wouaf: Weight modulation for user attribution and fingerprinting in text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8974–8983, 2024.
- Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.
- Zhen Li, Zuo-Liang Zhu, Ling-Hao Han, Qibin Hou, Chun-Le Guo, and Ming-Ming Cheng. Amt: All-pairs multi-field transforms for efficient frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9801–9810, 2023.
- Yuqing Liang, Jiancheng Xiao, Wensheng Gan, and Philip S Yu. Watermarking techniques for large language models: A survey. *arXiv preprint arXiv:2409.00089*, 2024.
- Aiwei Liu, Leyi Pan, Yijian Lu, Jingjing Li, Xuming Hu, Xi Zhang, Lijie Wen, Irwin King, Hui Xiong, and Philip Yu. A survey of text watermarking in the era of large language models. *ACM Computing Surveys*, 57(2):1–36, 2024.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024.
- Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. *arXiv preprint arXiv:2407.02371*, 2024.
- Joseph JK O’Ruanaidh and Thierry Pun. Rotation, scale and translation invariant digital image watermarking. In *Proceedings of International Conference on Image Processing*, volume 1, pages 536–539. IEEE, 1997.
- William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.
- 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*, pages 8748–8763. PmLR, 2021.
- Ahmad Rezaei, Mohammad Akbari, Saeed Ranjbar Alvar, Arezou Fatemi, and Yong Zhang. Lawa: Using latent space for in-generation image watermarking. In *European Conference on Computer Vision*, pages 118–136. Springer, 2024.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023.
- Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

- Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. *Advances in Neural Information Processing Systems*, 36:58047–58063, 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12162–12171, 2024.
- Ning Yu, Vladislav Skripniuk, Sahar Abdelnabi, and Mario Fritz. Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 14448–14457, 2021.
- Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities. *arXiv preprint arXiv:2305.11000*, 2023a.
- Kevin Alex Zhang, Lei Xu, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Robust invisible video watermarking with attention. *arXiv preprint arXiv:1909.01285*, 2019.
- Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.
- Xuanyu Zhang, Runyi Li, Jiwen Yu, Youmin Xu, Weiqi Li, and Jian Zhang. Editguard: Versatile image watermarking for tamper localization and copyright protection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11964–11974, 2024.
- Yulin Zhang, Jiangqun Ni, Wenkang Su, and Xin Liao. A novel deep video watermarking framework with enhanced robustness to h. 264/avc compression. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8095–8104, 2023b.
- Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.
- Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 657–672, 2018.
- Hazem Zohny, John McMillan, and Mike King. Ethics of generative ai, 2023.

## Appendix

### A Experiment Setting

#### A.1 Dataset

**Training data** The training data in our experiment comes from OpenVid-1M [Nan et al., 2024], which comprises over 1 million in-the-wild video clips, all with resolutions of at least 512×512, accompanied by detailed captions. We download a subset of it for our training. Specifically, we randomly select 10k videos from the downloaded videos for training. Each training video is composed of 8 frames sampled with a frame interval of 8. During fine-tuning, the videos are resized to 256 × 256 for lower computational overhead.

**Evaluation data** As we mentioned in Section 4.1, we select 50 prompts from the test set of VBench [Huang et al., 2024b], covering five categories: Animal, Human, Plant, Scenery, and Vehicles (10 prompts each). We list these prompts in Table 5. For each prompt, we generate four videos using fixed random seeds (42, 114514, 3407, 6666) to ensure reproducibility. For the I2V task, we first generate an image using Stable Diffusion 2.1 [Rombach et al., 2022], and subsequently generate four videos conditioned on each generated image using the same set of random seeds. Each generated video contains 16 frames with a resolution 512 × 512.

#### A.2 Training Detail

We use an AdamW optimizer for fine-tuning, with a learning rate  $5 \times 10^{-4}$  and batch size 2. We adopt a learning rate schedule with a linear warm-up followed by cosine decay. Specifically, the learning rate increases linearly from 0 to the base learning rate  $\text{blr}$  over the first  $T_{\text{warmup}}$  steps, and then decays to a minimum value  $\text{lr}_{\text{min}} = 10^{-6}$  following a half-cycle cosine schedule:

$$\text{lr}(t) = \begin{cases} \frac{t}{T_{\text{warmup}}} \cdot \text{blr}, & t < T_{\text{warmup}} \\ \text{lr}_{\text{min}} + \frac{1}{2}(\text{blr} - \text{lr}_{\text{min}}) \left( 1 + \cos \left( \pi \cdot \frac{t - T_{\text{warmup}}}{T_{\text{total}} - T_{\text{warmup}}} \right) \right), & t \geq T_{\text{warmup}} \end{cases} \quad (8)$$

where  $t$  is the current training step,  $T_{\text{total}}$  is the total number of training steps, and  $\text{blr}$  is the base learning rate  $5 \times 10^{-4}$ . For parameter groups with an individual scaling factor  $\text{lr}_{\text{scale}}$ , the learning rate is further scaled by this factor. In our experiments, the warm-up step  $T_{\text{warmup}}$  is set to 20% of  $T_{\text{total}}$ .

#### A.3 Selective Fine-tuning

Before we fine-tune the latent decoder, we apply PAS, which we proposed in this paper, to search and freeze the perceptual sensitive layers. We show the results in Figure 8. We observe that the majority of layers exhibit very low sensitivity, with values clustered near zero, indicating that perturbing these layers has minimal impact on visual quality. Only a small number of layers show relatively high sensitivity, suggesting they are more visually critical. Notably, the latent decoder of SVD demonstrates even stronger sparsity. These findings validate the design of our Perturbation-Aware Suppression (PAS) strategy: we identify and exclude perceptually sensitive layers and selectively fine-tune only the insensitive ones, enabling effective watermark embedding with minimal visual degradation. In our experiments, we set the threshold  $\tau_1 = 1.5 \times 10^{-4}$  for ModelScope and  $\tau_2 = 10^{-4}$  for SVD.

#### A.4 Temporal Tampering implementation

In this paper, to test the resilience of VIDSIG, we implement several temporal attacks. We provide detailed descriptions of these attacks here. Give a video  $[f_1, f_2, f_3, \dots, f_N]$  consists of  $N$  frames, we define the following temporal attacks.

**Frame Drop** Randomly select a frame and delete it. Let  $i$  denote the index of the selected frame, the tampered video is then denoted as  $[f_1, f_2, f_{i-1}, f_{i+1}, \dots, f_N]$

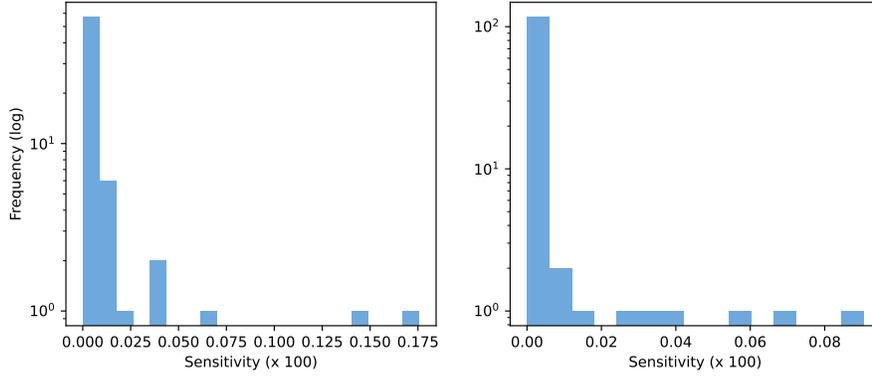


Figure 8: Histogram of layer sensitivity distributions in the latent decoder. The left figure corresponds to ModelScope (2D VAE), and the right figure corresponds to SVD (3D VAE). The perceptual sensitivity for each layer is measured by MSE.

**Frame Swap** We randomly select two distinct frames, indexed by  $i$  and  $j$ , and swap their positions in the video, resulting in a perturbed sequence  $[f_1, \dots, f_{i-1}, f_j, \dots, f_{j-1}, f_i, \dots, f_N]$ .

**Frame Insert** We randomly select two positions indexed by  $i$  and  $j$ , and insert the  $i^{\text{th}}$  frame at position  $j$ , resulting in a new sequence with duplicated content and increased temporal length  $[f_1, f_2, f_i, \dots, f_{j-1}, f_i, f_j, \dots, f_{N+1}]$

**Frame Insert Gaussian** Randomly select a position indexed by  $i$ , and insert a standard Gaussian noise at position  $i$ , denoted as  $f'_i$ , resulting in a new sequence with increased temporal length  $[f_1, f_2, \dots, f'_i, f_i, \dots, f_{N+1}]$

**Frame Average** Given a sequence length  $n$ , we randomly select a position indexed by  $i$ , conditioned on  $i + n \leq N$ . We then compute the average of the  $n$  subsequent frames, i.e.,  $\bar{f} = \frac{1}{n} \sum_{j=0}^{n-1} f_{i+j}$ , and replace the  $i^{\text{th}}$  frame with  $\bar{f}$ , resulting the tampered video  $[f_1, f_2, \dots, f_{i-1}, \bar{f}, \dots, f_{N-n+1}]$

## B Metric

### B.1 Watermark Detection

Let  $\mathbf{m} \in \{0, 1\}^k$  be the embedded multi-bit message, and let  $\mathbf{m}'$  be the extracted message from a generated video  $\mathbf{v}$ . To determine whether the video contains the watermark, we count the number of matching bits between  $\mathbf{m}$  and  $\mathbf{m}'$ , denoted as  $M(\mathbf{m}, \mathbf{m}')$ . A detection decision is made by checking whether the number of matching bits exceeds a predefined threshold  $\tau$ :

$$M(\mathbf{m}, \mathbf{m}') \geq \tau, \quad \text{where } \tau \in \{0, \dots, k\}. \quad (9)$$

To formally evaluate detection performance, we define the hypothesis test as follows:  $H_1$ : "The video was generated by the watermarked model" vs.  $H_0$ : "The video was not generated by the watermarked model". Under the null hypothesis  $H_0$ , as in previous work [Yu et al., 2021], assume that the extracted bits  $m'_1, \dots, m'_k$  are i.i.d. Bernoulli variables with  $p = 0.5$ , the match count  $M(\mathbf{m}, \mathbf{m}')$  follows a binomial distribution  $\mathcal{B}(k, 0.5)$ . The False Positive Rate (FPR) is defined as the probability of falsely detecting a watermark when  $H_0$  is true:

$$\text{FPR}(\tau) = \mathbb{P}(M > \tau \mid H_0) = I_{1/2}(\tau + 1, k - \tau), \quad (10)$$

Table 5: Prompts for video generation across five domains.

Category	Prompt (1)	Prompt (2)
<b>Animal</b>	a red panda eating leaves	a squirrel eating nuts
	a cute pomeranian dog playing with a soccer ball	curious cat sitting and looking around
	wild rabbit in a green meadow	underwater footage of an octopus in a coral reef
	hedgehog crossing road in forest	shark swimming in the sea
	an african penguin walking on a beach	a tortoise covered with algae
<b>Human</b>	a boy covering a rose flower with a dome glass	boy sitting on grass petting a dog
	a child playing with water	couple dancing slow dance with sun glare
	elderly man lifting kettlebell	young dancer practicing at home
	a man in a hoodie and woman with a red bandana talking to each other and smiling	a woman fighter in her cosplay costume
	a happy kid playing the ukulele	a person walking on a wet wooden bridge
<b>Plant</b>	plant with blooming flowers	close up view of a white christmas tree
	dropping flower petals on a wooden bowl	a close up shot of gypsophila flower
	a stack of dried leaves burning in a forest	drone footage of a tree on farm field
	shot of a palm tree swaying with the wind	candle wax dripping on flower petals
	forest trees and a medieval castle at sunset	a mossy fountain and green plants in a botanical garden
<b>Scenery</b>	scenery of a relaxing beach	fireworks display in the sky at night
	waterfalls in between mountain	exotic view of a riverfront city
	scenic video of sunset	view of houses with bush fence under a blue and cloudy sky
	boat sailing in the ocean	view of golden domed church
	a blooming cherry blossom tree under a blue sky with white clouds	aerial view of a palace
<b>Vehicles</b>	a helicopter flying under blue sky	red vehicle driving on field
	aerial view of a train passing by a bridge	red bus in a rainy city
	an airplane in the sky	helicopter landing on the street
	boat sailing in the middle of the ocean	video of a kayak boat in a river
	traffic on busy city street	slow motion footage of a racing car

where  $I_x(a, b)$  is the regularized incomplete beta function. Based on this, we can calculate the least match count to determine whether the video contains watermark under a fixed FPR.

## B.2 Video Quality

We provide details of the metrics for video quality evaluation in our experiments. Specifically, we utilize Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) [Wang et al., 2004], Learned Perceptual Image Patch Similarity (LPIPS) Zhang et al. [2018] with a VGG backbone, tLP [Chu et al., 2020] and the VBench evaluation metrics [Huang et al., 2024b]. PSNR, SSIM, LPIPS, and tLP are computed between the watermarked video and its corresponding unwatermarked version to assess fidelity, whereas VBench evaluates the perceptual quality of the generated video independently.

**PSNR** PSNR quantifies the pixel-wise difference between the generated watermarked video  $\hat{v}$  and the original video  $v$ . It is computed as the average PSNR across all frames:

$$\text{PSNR} = \frac{1}{f} \sum_{t=1}^f 10 \cdot \log_{10} \left( \frac{\text{MAX}^2}{\text{MSE}_t} \right), \quad (11)$$

where  $f$  is the number of frames, MAX denotes the maximum possible pixel value (typically 1.0 or 255), and  $\text{MSE}_t = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n [\hat{\mathbf{v}}_t(i, j) - \mathbf{v}_t(i, j)]^2$  is the mean squared error between the  $t$ -th frame of the two videos. Higher PSNR values indicate better fidelity to the original video.

**SSIM** SSIM measures the structural similarity between the generated watermarked video  $\hat{\mathbf{v}}$  and the original video  $\mathbf{v}$ . The overall SSIM is computed by averaging the SSIM values of all frames:

$$\text{SSIM} = \frac{1}{f} \sum_{t=1}^f \text{SSIM}(\hat{\mathbf{v}}_t, \mathbf{v}_t), \quad (12)$$

where  $\hat{\mathbf{v}}_t$  and  $\mathbf{v}_t$  denote the  $t$ -th frame of the watermarked and original video, respectively. The SSIM between two frames is defined as:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \quad (13)$$

where  $\mu_x, \mu_y$  are the means,  $\sigma_x^2, \sigma_y^2$  are the variances, and  $\sigma_{xy}$  is the covariance of frame patches from  $x$  and  $y$ ;  $C_1, C_2$  are constants to stabilize the division. SSIM ranges from 0 to 1, with higher values indicating greater perceptual similarity.

**LPIPS** LPIPS [Zhang et al., 2018] evaluates perceptual similarity by computing deep feature distances between frames. Given a ground-truth video  $\mathbf{v} = \{\mathbf{v}_1, \dots, \mathbf{v}_f\}$  and a generated video  $\hat{\mathbf{v}} = \{\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_f\}$ , the video-level LPIPS score is defined as the average over all frames:

$$\text{LPIPS} = \frac{1}{f} \sum_{t=1}^f \text{LPIPS}(\hat{\mathbf{v}}_t, \mathbf{v}_t), \quad (14)$$

where  $\text{LPIPS}(\hat{\mathbf{v}}_t, \mathbf{v}_t)$  denotes the perceptual distance between the  $t$ -th frame pair, computed via a pretrained deep network. Lower values indicate higher perceptual similarity.

**tLP** tLP [Chu et al., 2020] measures the consistency of temporal perceptual dynamics between adjacent frames. Specifically, it compares the learned perceptual (LP) differences between adjacent frame pairs in the generated video  $\hat{\mathbf{v}} = \{\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_f\}$  and its reference  $\mathbf{v} = \{\mathbf{v}_1, \dots, \mathbf{v}_f\}$ . The tLP is defined as:

$$\text{tLP} = \|\text{LP}(\hat{\mathbf{v}}_{t-1}, \hat{\mathbf{v}}_t) - \text{LP}(\mathbf{v}_{t-1}, \mathbf{v}_t)\|_1, \quad (15)$$

where  $\text{LP}(\cdot, \cdot)$  denotes the LPIPS distance between two frames. A lower tLP indicates better temporal coherence with respect to the ground-truth dynamics.

**VBench: Subject Consistency** Subject Consistency evaluates the semantic stability of the generated subject by calculating the cosine similarity of DINO [Caron et al., 2021] features across frames:

$$S_{\text{subject}} = \frac{1}{T-1} \sum_{t=2}^T \frac{1}{2} (\langle d_1, d_t \rangle + \langle d_{t-1}, d_t \rangle), \quad (16)$$

where  $d_i$  is the normalized DINO image feature of the  $i^{\text{th}}$  frame, and  $\langle \cdot, \cdot \rangle$  denotes cosine similarity via dot product.

Table 6: Specific VBench score for each watermarking method.

Model	Method	Subject Consistency	Background Consistency	Motion Smoothness	Imaging Quality
MS	RivaGAN	0.955	0.963	0.980	0.675
	VideoSeal	0.955	0.962	0.980	0.676
	StableSig	0.956	0.961	0.980	0.675
	VideoShield	0.952	0.962	0.977	0.683
	VIDSIG (ours)	0.956	0.961	0.978	0.676
SVD	RivaGAN	0.945	0.958	0.956	0.623
	VideoSeal	0.945	0.955	0.956	0.624
	StableSig	0.946	0.956	0.957	0.633
	VideoShield	0.934	0.949	0.956	0.629
	VIDSIG (ours)	0.946	0.955	0.959	0.632

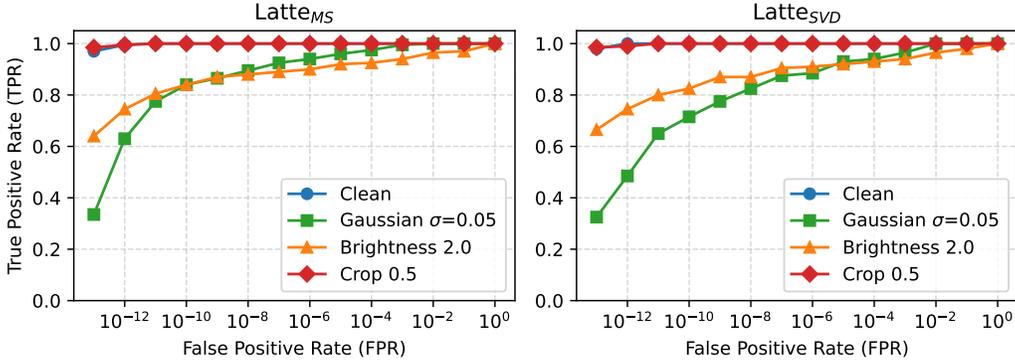


Figure 9: Watermark detection Results for  $\text{Latte}_{MS}$  and  $\text{Latte}_{SVD}$ .

**VBench: Background Consistency** Background Consistency measures the temporal consistency of the background using CLIP [Radford et al., 2021] features:

$$S_{\text{background}} = \frac{1}{T-1} \sum_{t=2}^T \frac{1}{2} (\langle c_1, c_t \rangle + \langle c_{t-1}, c_t \rangle), \quad (17)$$

where  $c_i$  is the CLIP image feature of the  $i^{\text{th}}$  frame, normalized to unit length.

**VBench: Motion Smoothness** Given a video  $[f_0, f_1, f_2, \dots, f_{2n}]$ , the odd-numbered frames  $[f_1, f_3, \dots, f_{2n-1}]$  are dropped, and the remaining even-numbered frames  $[f_0, f_2, \dots, f_{2n}]$  are used to interpolate [Li et al., 2023] intermediate frames  $[\hat{f}_1, \hat{f}_3, \dots, \hat{f}_{2n-1}]$ . The Mean Absolute Error (MAE) between interpolated and original dropped frames is computed and normalized to  $[0, 1]$ , with a larger value indicating better smoothness.

**VBench: Imaging Quality** Imaging quality mainly considers the low-level distortions presented in the generated video frames. The MUSIQ [Ke et al., 2021] image quality predictor trained on the SPAQ [Fang et al., 2020] dataset is used for evaluation, which is capable of handling variable sized aspect ratios and resolutions. The frame-wise score is linearly normalized to  $[0, 1]$ , and the final score is then calculated by averaging the frame-wise scores across the entire video sequence.

## C More Experiment Results

### C.1 Video Quality

We provide the specific video quality metric of VBench for each watermarking method in Table 6, It can be observed that VIDSIG achieves comparable or even superior video quality compared to post-



Figure 10: Two videos generated by Stable Video Diffusion, the left one is the original video, and the right one is the corresponding watermarked video by VIDSIG.

generation methods, while significantly outperforming them in watermark extraction and detection accuracy, as shown in Table 1.

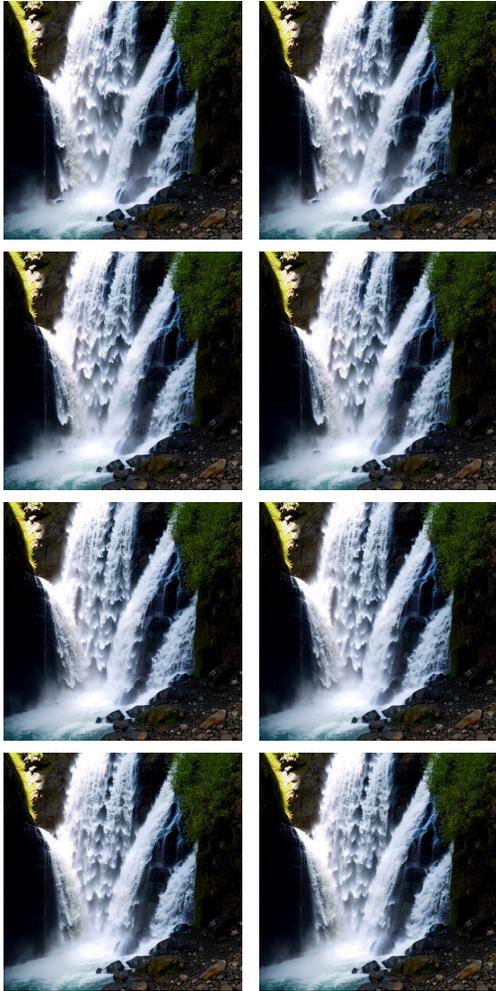
### C.2 Watermark Detection

In Section 4.3, we substitute the latent decoder of Latte [Ma et al., 2024] by the fine-tuned latent decoders in MS and SVD. We also follow the same settings in Section 4.2 and report the True Positive Rate, the results are shown in Figure 9.

### C.3 More Visual Cases

We show more videos generated by SVD and Latte here, see Figure 10, Figure 11, and Figure 12.

“waterfalls in between  
mountain”

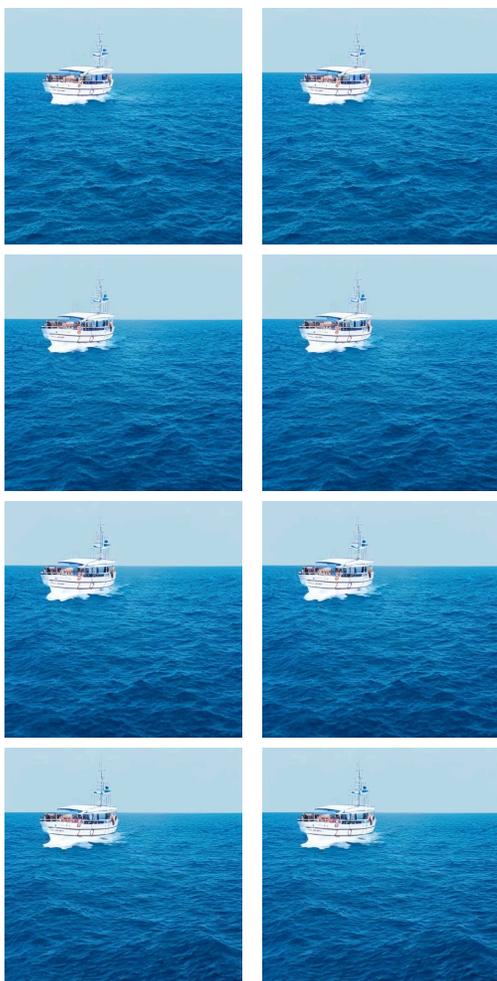


“a cute Pomeranian dog  
playing with a soccer ball”



Figure 11: Two videos generated by Latte, the left one is the original video, and the right one is the corresponding watermarked video by VIDSIG. The latent decoder of Latte is transferred from ModelScope.

“boat sailing in the middle of the ocean ”



“slow motion footage of a racing car”

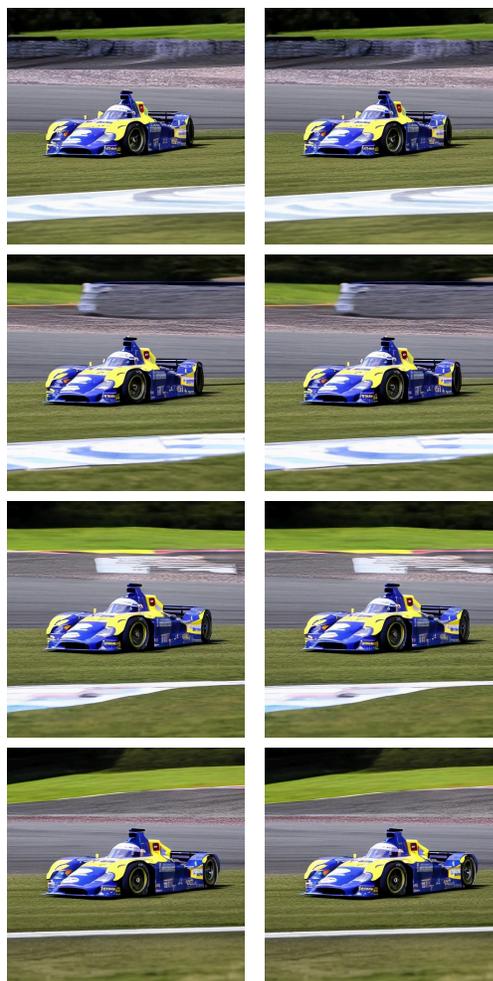


Figure 12: Two videos generated by Latte, the left one is the original video, and the right one is the corresponding watermarked video by VIDSIG. The latent decoder of Latte is transferred from Stable Video Diffusion.