

# VIDSTAMP: A Temporally-Aware Watermark for Ownership and Integrity in Video Diffusion Models

Mohammadreza Teymorianfard  
University of Massachusetts Amherst  
mteymorianf@umass.edu

Shiqing Ma  
University of Massachusetts Amherst  
shiqingma@umass.edu

Amir Houmansadr  
University of Massachusetts Amherst  
amir@cs.umass.edu

**Abstract**—The rapid rise of video diffusion models has enabled the generation of highly realistic and temporally coherent videos, raising critical concerns about content authenticity, provenance, and misuse. Existing watermarking approaches—whether passive, post-hoc, or adapted from image-based techniques—often struggle to withstand video-specific manipulations such as frame insertion, dropping, or reordering, and typically degrade visual quality. In this work, we introduce VIDSTAMP, a watermarking framework that embeds per-frame or per-segment messages directly into the latent space of temporally-aware video diffusion models. By fine-tuning the model’s decoder through a two-stage pipeline—first on static image datasets to promote spatial message separation, and then on synthesized video sequences to restore temporal consistency—VIDSTAMP learns to embed high-capacity, flexible watermarks with minimal perceptual impact. Leveraging architectural components such as 3D convolutions and temporal attention, our method imposes no additional inference cost and offers better perceptual quality than prior methods, while maintaining comparable robustness against common distortions and tampering. VIDSTAMP embeds 768 bits per video (48 bits per frame) with a bit accuracy of 95.0%, achieves a log P-value of  $-166.65$  (lower is better), and maintains a video quality score of 0.836, comparable to unwatermarked outputs (0.838) and surpassing prior methods in capacity-quality tradeoffs. Code: <https://github.com/SPIN-UMass/VidStamp>

## I. INTRODUCTION

The rapid advancement of AI-generated content—particularly video—has introduced unprecedented challenges to digital media integrity, security, and trust. Recent generative models are capable of synthesizing highly realistic video content from static images or natural language prompts [1], [2], [3], raising concerns about their potential misuse in misinformation, impersonation, and tampering. The growing accessibility of such tools poses a significant threat to content authenticity and highlights the urgent need for robust mechanisms that can ensure generative model accountability, enable provenance tracking, and support tamper detection. In this context, *watermarking* of AI-generated media has emerged as a promising strategy for content authentication and tamper evidence [4], [5].

Generative diffusion models, especially latent diffusion models (LDMs), have revolutionized high-fidelity content synthesis by learning to reverse a noise process in a compressed latent space [6]. While diffusion-based text-to-image models like Stable Diffusion [7] have already demonstrated remarkable realism and scalability, recent efforts have extended this framework to the video domain by incorporating temporally-

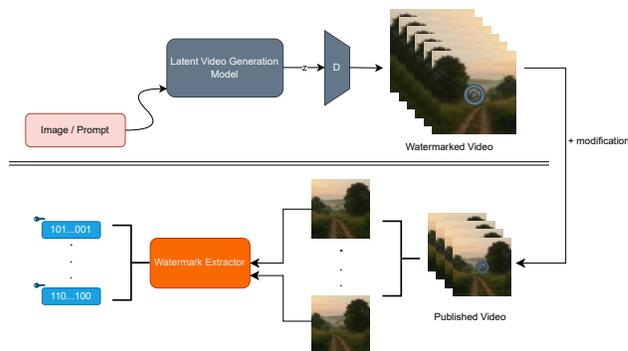


Fig. 1: VIDSTAMP framework overview. Our method embeds watermark messages directly into the latent space of a video diffusion model during generation. Each frame (or segment) is assigned a message that is recoverable via a pretrained extractor. This enables forensic verification, tamper detection, and source attribution without any post-processing.

aware modules such as 3D convolutions and temporal attention into the decoder [8], [9]. These advancements enable the generation of temporally coherent videos that are nearly indistinguishable from real-world footage. However, the growing power and accessibility of such models also call for integrated watermarking solutions that can operate within the generation pipeline itself—ensuring authenticity without introducing post-processing steps or perceptual degradation.

Existing approaches to watermarking generative content either operate in a passive forensic manner or add external watermarks post-generation. Passive detectors—which attempt to identify artifacts or inconsistencies in generated content—are increasingly ineffective against modern diffusion models as they produce highly realistic outputs with minimal statistical traces [10], [11]. Meanwhile, post-hoc watermarking, where a signal is embedded into media after generation, is often brittle and vulnerable to simple removal techniques, particularly when the watermarking algorithm is publicly known [12]. Recent work in the image domain, such as Stable Signature [5], demonstrates that training the generative model itself to embed an imperceptible signature during content synthesis offers a promising direction for watermarking. These methods embed

the watermark directly into the latent decoder, enabling resilient and cost-free watermarking during generation. However, extending such techniques to the video domain introduces new challenges—naïvely applying image-based watermarking frame-by-frame fails to capture temporal dependencies and cannot protect against video-specific attacks like frame dropping, swapping, or insertion.

In this paper, we introduce VIDSTAMP, a novel framework for robust and efficient watermarking in latent video diffusion models. An overview of our proposed watermarking framework, VIDSTAMP, is shown in Figure 1. Our method leverages the temporally-aware architecture of modern video diffusion decoders to embed a sequence of hidden messages—either per-frame or per-segment—directly into the generated video. We adopt a two-stage fine-tuning strategy: first, the decoder is fine-tuned on a curated image dataset (e.g., COCO [13]) to learn the initial watermarking behavior; then, it is fine-tuned again on videos generated by the same model, to embed temporally consistent and uniquely traceable watermarks. This approach introduces no additional computational cost during video generation, allows flexible capacity control (each frame or group of frames may carry distinct messages), and supports practical use cases such as long-form video authentication and temporal tamper localization.

To evaluate the practicality and effectiveness of our watermarking method, we implemented VIDSTAMP on top of the Stable Video Diffusion (SVD) [1] framework using a two-stage decoder fine-tuning process. Our model embeds *768 bits per video (48 bits per frame)* with an average *bit accuracy of 95.0%*, while preserving *video quality nearly identical to unwatermarked outputs* (average perceptual score: 0.836 vs. 0.838). Compared to state-of-the-art baselines, VIDSTAMP delivers a significantly stronger trade-off between *capacity, quality, and detectability*: it outperforms RivaGAN [14], VideoSeal [15], and VideoShield [16] in *log P-value*, a statistical metric that measures how unlikely it is for the extracted watermark to be correct by chance alone. This makes it especially useful when comparing watermark robustness across methods with different capacities. VIDSTAMP achieves a *log P-value* of  $-166.65$ , which indicates high statistical confidence in watermark presence even under distortion. Furthermore, unlike post-hoc watermarking methods, VIDSTAMP introduces *no additional inference overhead*, and uniquely supports *frame-level tamper localization*, attaining *over 95% localization accuracy* across various manipulation types including frame insertion, deletion, and reordering. These results demonstrate that VIDSTAMP offers both *high-fidelity watermarking and robust forensic functionality*, making it a practical solution for real-world video provenance and integrity verification.

**In summary, this paper makes the following contributions:**

- **A temporally-aware watermarking framework for video diffusion models:** We present VIDSTAMP, a method that fine-tunes the decoder of a latent video diffusion model to embed per-frame or segment-level watermarks directly during generation. While our method does not modify the decoder architecture, it leverages the inherent

temporally-aware components (e.g., 3D convolutions and temporal attention) already present in video diffusion decoders to enable *tamper localization* and *ownership verification* with no additional inference-time overhead. Additionally, segment-wise embedding offers *flexible control over watermark capacity*.

- **A two-stage decoder fine-tuning pipeline:** Our approach first fine-tunes the decoder on static images to promote spatial watermark separability, then adapts it to video synthesis to ensure temporal consistency. This training strategy enables *high-capacity watermark embedding* (768 bits per video) with strong robustness under distortion.
- **State-of-the-art performance in quality, robustness, and tamper localization:** VIDSTAMP achieves comparable or superior performance to prior baselines (RivaGAN, VideoSeal, VideoShield), while embedding significantly more bits and maintaining high video quality. It also supports *frame-level tamper localization with over 95% accuracy* under various manipulation types.

## II. PRELIMINARIES

### A. Video Diffusion Models and Temporality

Recent advancements in generative modeling have extended diffusion models to video synthesis, enabling temporally coherent outputs. Ho et al. introduced video diffusion models by extending the standard image diffusion architecture to the spatio-temporal domain, training a 3D U-Net denoiser on both image and video data to improve fidelity and stability [8]. To generate longer or higher-resolution videos, they proposed hierarchical sampling techniques, achieving promising results in text-conditional video generation. Subsequent works have focused on latent video diffusion to enhance efficiency and scalability. He et al. [9] proposed generating videos in a low-dimensional 3D latent space, significantly reducing computation compared to pixel-space diffusion. Their Latent Video Diffusion Model employs a hierarchical approach that can produce videos with thousands of frames, using conditional latent perturbation and guidance to mitigate error accumulation over long durations. Similarly, Blattmann et al. [1] transformed a pre-trained image diffusion model (Stable Diffusion) into a video generator by adding temporal layers to the latent diffusion model. By fine-tuning the decoder with 3D convolutions and temporal attention, they achieved high-resolution text-to-video synthesis while maintaining consistency across frames.

### B. Watermarking in Generative Models

Watermarking techniques for generative models have evolved across modalities, with early work in text embedding messages through lexical substitution, syntactic manipulation, or stylometry [17], [18], [19]. More recently, decoding-based approaches like Kirchenbauer et al.’s statistical watermark skew token probabilities during generation, leaving imperceptible yet detectable traces [20], while neural methods like the Adversarial Watermarking Transformer encode bit strings through learned paraphrasing [4]. Sentence-level approaches such as SimMark [21] further leverage semantic similarity patterns to embed

watermarks without modifying generation logits, improving robustness to paraphrasing attacks.

In the image domain, early learning-based watermarking systems such as HiDDeN trained encoder–decoder CNNs to embed robust messages directly into images while resisting typical perturbations like cropping or JPEG compression [22]. A major shift came with model-integrated watermarking in generative diffusion models—particularly the Stable Signature, which fine-tunes a latent diffusion model’s decoder to embed persistent identifiers within the generation process itself [5]. These in-model watermarks offer clear advantages: zero inference overhead, imperceptibility, and strong robustness even under heavy transformation. Follow-up methods like Tree-Ring Watermarks further improve resilience by seeding watermark signals in the input noise, ensuring they propagate naturally through the diffusion process [23]. Overall, integrated watermarking provides stronger tamper resistance and detection reliability than post-hoc methods, which, while easier to deploy, are more vulnerable to removal or degradation. Extending these ideas to video generation introduces new challenges—particularly maintaining temporal consistency—an issue our work addresses by leveraging temporally-aware modules in latent video diffusion decoders.

### C. Passive vs. Active Watermarking

Detection techniques for AI-generated content typically fall into two categories: passive forensics and active watermarking. Passive methods attempt to detect statistical or visual artifacts left by generative models. Early efforts focused on GANs, identifying frequency domain inconsistencies or unnatural textures [11], [10]. However, diffusion models have largely eliminated such artifacts, rendering many forensic detectors ineffective against modern image and video generation [12]. In response, active watermarking has emerged as a more reliable strategy, embedding identifiers directly during generation [5], [23]. Post-hoc approaches apply a watermark after synthesis—such as through DCT, spatial overlays, or neural encoding—but they are often brittle and easily removed if the embedding strategy is known [10], [12]. For example, Li et al. demonstrate that simple removal attacks can strip watermarks without visible degradation [12], while classical reviews highlight the vulnerability of post-generation schemes to estimation-based removal [10]. In contrast, model-integrated methods like Stable Signature [5], Tree-Ring Watermarks [23], and token-based biasing for language models [20] embed signals during generation, making them more robust to adversarial tampering. Some recent works even propose theoretically undetectable watermarks by leveraging structured noise or optimal transport [24], [25]. As diffusion models continue to improve, these integrated methods offer stronger guarantees of persistence and provenance than passive or post-hoc approaches.

### D. Video Watermarking Techniques

Classical video watermarking has been studied for decades, yielding a spectrum of spatial-domain and transform-domain techniques. Spatial methods directly embed payload bits by

modifying pixel intensities or colors in each frame, achieving high embedding capacity and simplicity at the cost of fragility against processing and compression [26], [27]. In contrast, transform-domain schemes hide information in frequency coefficients, leveraging the human visual system to preserve perceptual quality while improving robustness to compression and filtering [27], [28]. Many early approaches combine these strategies with error-correction coding and redundancy to resist desynchronization attacks: for instance, repeating or interleaving the watermark across frames can guard against frame dropping or temporal cropping, and applying cyclic error-correcting codes helps the decoder recover from bit errors introduced by noisy channels or re-encoding [28]. Such designs allowed watermarks to survive common operations like recompression, scaling, and transcoding. More recent learning-based methods employ deep neural networks to automatically optimize the imperceptible embedding of watermarks. These approaches typically train an encoder–decoder convolutional network to hide and extract a bit-string, often inserting a differentiable “distortion layer” (simulating compression, scaling, or frame loss) during training to enhance robustness. For example, Luo et al. [29] train a multiscale video watermarking model that spreads the payload across spatial and temporal dimensions and employs adversarial discriminators to enforce invisibility and resilience. Such deep models significantly improve robustness against complex distortions while maintaining visual quality, although ensuring temporal consistency when applied frame-by-frame remains challenging.

The rise of GAN and diffusion-based video generators has spurred watermarking techniques tailored for AI-generated content. Zhang et al.’s RivaGAN [14] is an early deep watermarking architecture for video that introduces an adversarial training setup with an attacker network attempting to erase the watermark and a critic ensuring visual fidelity. Through an attention-based encoder–decoder, RivaGAN identifies perceptually significant regions to embed a robust invisible mark, achieving strong baseline robustness to compression and scaling attacks. However, as a post-processing method, it operates outside the generative model and does not explicitly address cross-frame coherence in long AI-generated videos. More recent frameworks have improved on this. VideoSeal by Fernandez et al. [15] combines a trained neural embedder and extractor with a multi-stage training regimen: an initial image-domain pre-training, followed by video-domain fine-tuning with simulated video codecs and geometric transformations applied between the encoder and decoder to harden the watermark. VideoSeal also introduces temporal watermark propagation, wherein the watermark is carried over across frames without needing to embed it independently in each frame, thereby enhancing efficiency and temporal consistency. This yields state-of-the-art robustness under mixed distortions (e.g., recompression plus cropping) for high-definition videos. Nonetheless, VideoSeal remains an external module applied after video generation. In parallel, Hu et al. propose VideoShield [16], which integrates watermarking into the diffusion sampling process for text-to-video models without requiring model retraining. VideoShield

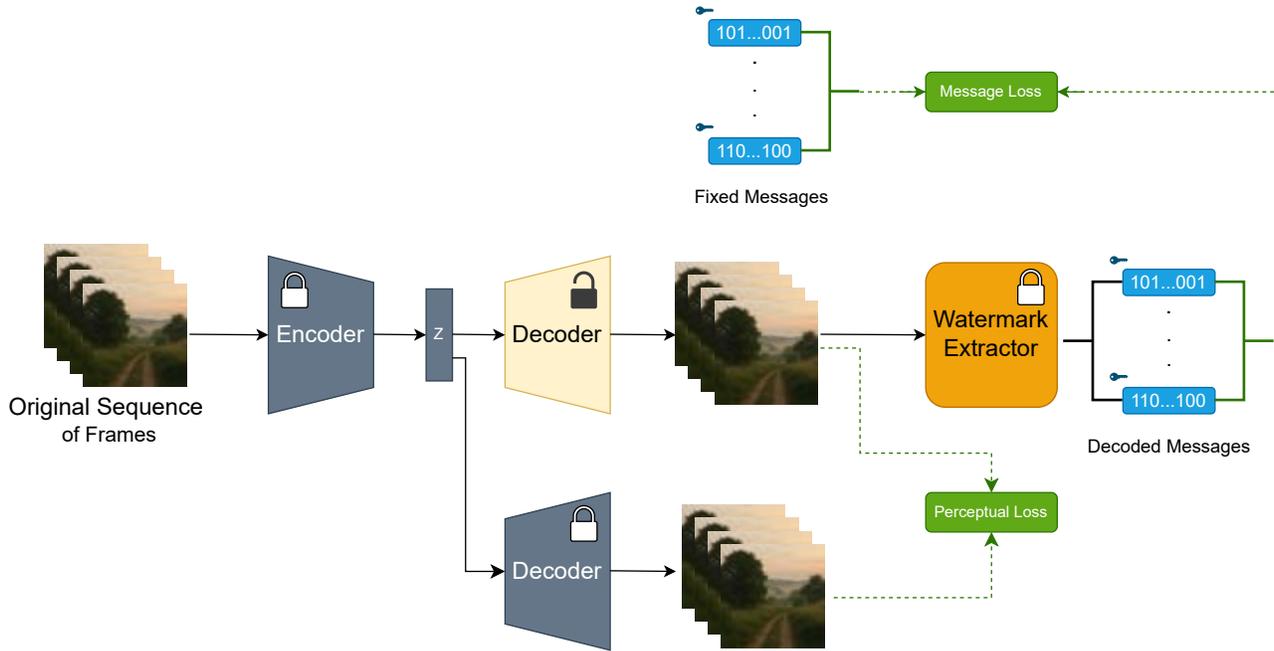


Fig. 2: VIDSTAMP training pipeline. During training, a fixed set of messages is embedded into video frames through the decoder of a latent video diffusion model. The decoder is fine-tuned to maximize both message accuracy and perceptual quality, using a pretrained extractor for supervision. Message loss and perceptual loss are computed per frame to guide robust and imperceptible watermark learning.

maps the hidden message bits to a set of “template” noise perturbations that are injected at each denoising step, so that the generated frames inherently contain a recoverable watermark; a reverse diffusion (e.g., DDIM inversion) is used to extract the watermark from the video, and the method can pinpoint tampered frames by checking consistency of the template bits across time. This approach achieves robust extraction and even enables spatial/temporal tamper localization in diffusion-generated videos, all with negligible impact on perceptual quality.

While VideoShield offers a lightweight and inference-time-compatible solution, it relies on external perturbations and inversion steps. In contrast, our proposed VIDSTAMP method embeds watermarks directly into the latent decoding process of the generative model. This tight integration ensures that the watermark is temporally coherent across frames and intrinsically bound to the content, offering greater flexibility in bit placement and message structure, higher capacity, and consistent robustness—without the need for post-hoc perturbation or inversion. By leveraging the model’s existing temporal modules, VIDSTAMP maintains high visual fidelity while enabling reliable ownership verification and frame-level tamper detection.

### III. METHODOLOGY

#### A. Overview

Inspired by the Stable Signature approach for image diffusion models[5] – which showed that fine-tuning the generative model’s decoder can embed a persistent invisible watermark into all outputs – we extend this idea to video generation. Figure 1 illustrates the overall architecture of our system. The key insight is that modern latent video diffusion models employ temporally-aware decoders (e.g., 3D convolutions and temporal attention layers [30]), enabling the generative process itself to carry a watermark across time. By leveraging these temporal layers, we embed watermarks during generation, allowing each frame to carry a unique identifier and enabling distinct per-frame message decoding. This frame-specific encoding permits precise verification per frame, which facilitates localization of any temporal tampering in the video [16]. For watermark extraction, we utilize a pre-trained decoder network from the HiDDeN deep watermarking framework [22] to recover the embedded message from each generated frame.

#### B. Two Stage Finetuning

Our training framework, illustrated in the Figure 2, begins with a sequence of frames as the input data, which are passed through a *variational autoencoder (VAE)* to obtain latent representations  $Z$ . These latents are decoded frame-by-frame using a temporally-aware decoder—equipped with 3D

convolutions and temporal attention—to generate video frames containing imperceptible embedded messages. Each frame is assigned a fixed message (bit-string), and a pretrained message extractor—adapted from the HiDDeN architecture—is used to recover the message from each generated frame. The extraction is supervised using a binary cross-entropy (BCE) loss:

$$\mathcal{L}_{\text{msg}} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (1)$$

where  $y_i$  and  $\hat{y}_i$  denote the ground truth and predicted bits, respectively. In parallel, we compute a Watson-VGG perceptual loss [31] on a frame-by-frame basis to maintain the semantic integrity and visual quality of the generated outputs. This perceptual loss, denoted  $\mathcal{L}_{\text{perc}}$ , compares deep feature activations between the original and reconstructed frames, emphasizing human-perceptible discrepancies. The final training objective is a weighted sum of the message loss and perceptual loss:

$$\mathcal{L}_{\text{total}} = \lambda_w \cdot \mathcal{L}_{\text{msg}} + \lambda_i \cdot \mathcal{L}_{\text{perc}} \quad (2)$$

where  $\lambda_w$  and  $\lambda_i$  are tunable hyperparameters that control the trade-off between bit accuracy and visual quality. During training, only the decoder is updated, while the encoder and message extractor remain frozen. This setup enables the decoder to learn effective message embedding strategies without compromising on generation fidelity.

Our training pipeline adopts a two-stage fine-tuning approach to equip the video diffusion model’s decoder with the ability to embed robust, temporally-aware watermarks without degrading generation quality. In the first stage, we fine-tune the decoder using COCO [13] image datasets. We treat a batch of independent images as a pseudo-video, where each image serves as a distinct frame. This strategy allows the decoder to perceive each frame as an independent unit, promoting diversity in learned representations across positions. Consequently, the model learns to associate distinct spatial features with different message embeddings, facilitating per-frame message separation. This stage is essential for initializing the decoder’s ability to distinguish and encode frame-specific watermarks.

However, training solely on images can lead to suboptimal video quality, since the decoder lacks exposure to temporal dynamics and inter-frame coherence. Therefore, in the second stage, we fine-tune the decoder using synthesized videos generated from the same diffusion model. This phase reinforces temporal consistency while preserving the model’s ability to embed diverse frame-level messages. It also adapts the decoder to the statistical distribution of video data, improving fidelity and motion coherence in generated outputs. The dual-phase training balances frame-level watermark capacity with video-level coherence, enabling the model to embed traceable, tamper-localizable watermarks in temporally coherent output videos.

### C. Temporal Tamper Localization

To detect and localize frame-level tampering in watermarked videos, we propose a simple yet effective decoding-based

algorithm that leverages the frame-wise embedded watermark messages. Given a set of known template messages corresponding to the original frame positions, we compare each decoded frame message in the potentially tampered video against all template keys using Hamming similarity. This allows us to identify which original frame each tampered frame most likely matches—or to flag it as an insertion if no match surpasses a similarity threshold.

The algorithm operates as follows: for each frame in the tampered video, we compute the similarity between its decoded message and each of the original reference keys (used during generation). If the best match has a similarity score below a predefined threshold, the frame is classified as an inserted (unauthentic) frame. Otherwise, the frame is assigned to the most similar original key. The predicted sequence is then compared against the ground-truth frame mapping to calculate localization accuracy.

This procedure enables identification of common temporal attacks such as frame insertion, deletion, and reordering. Our method can generalize to different key counts and message lengths, offering flexibility in watermarking granularity and tamper detection sensitivity. The full algorithm is presented in Algorithm 1.

---

#### Algorithm 1 Temporal Tamper Localization

---

**Require:** Template keys  $T \in \mathbb{R}^{M \times d}$ , Tampered keys  $K \in \mathbb{R}^{N \times d}$ , True sequence  $S \in \mathbb{Z}^N$ , Threshold  $\tau$   
**Ensure:** Tamper localization accuracy

- 1:  $P \leftarrow$  empty list ▷ Predicted frame sequence
- 2: **for**  $i = 1$  to  $N$  **do**
- 3:    $k_i \leftarrow K[i]$  ▷ Current decoded key
- 4:   Compute Hamming similarity  $sim$  between  $k_i$  and all  $T[j]$
- 5:    $j^* \leftarrow \arg \max_j sim[j]$
- 6:   **if**  $sim[j^*] < \tau$  **then**
- 7:      $P.append(-1)$  ▷ Inserted frame
- 8:   **else**
- 9:      $P.append(j^*)$  ▷ Best-matching original index
- 10:   **end if**
- 11: **end for**
- 12:  $accuracy \leftarrow \frac{1}{N} \sum_{i=1}^N \mathbb{1}[P[i] = S[i]]$
- 13: **return**  $accuracy$

---

### D. Segment-wise Embedding

While per-frame watermarking offers high granularity and precise tamper localization, it can be sensitive to frame-level distortions and imposes high message embedding and extraction overhead. To balance capacity and efficiency, we introduce a segment-wise embedding strategy, as illustrated in Figure 3. Instead of assigning a unique message to each frame, we divide the video into fixed-length segments of  $k$  frames, and embed the same message across each segment.

This approach provides greater flexibility in controlling the total number of embedded bits, which is particularly advantageous when dealing with longer videos. By reducing the number of unique message blocks relative to the total frame count, segment-wise embedding avoids unnecessary capacity that might otherwise increase susceptibility to quality degradation

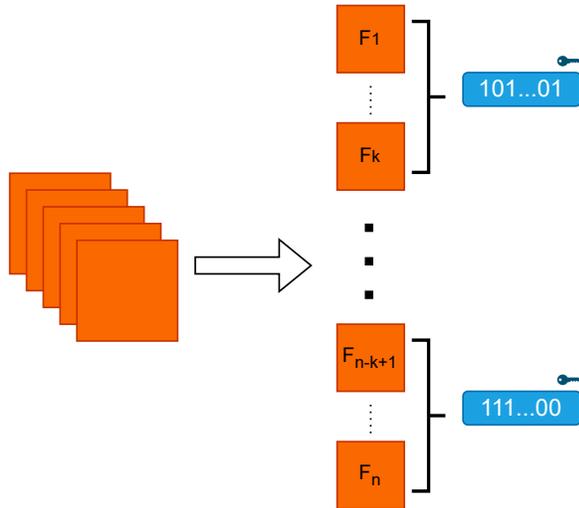


Fig. 3: Segment-wise message embedding in VIDSTAMP. In addition to per-frame embedding, VIDSTAMP supports embedding watermark messages into fixed-length segments of  $k$  consecutive frames. The same message is repeated across each segment. Segment-wise embedding provides better control over the total bit capacity and simplifies message extraction in long-form videos.

or overfitting. Furthermore, segment-level embedding aligns naturally with the temporal modeling behavior of latent video diffusion decoders, which utilize 3D convolutions and attention mechanisms across contiguous frame windows. As a result, our method achieves stronger temporal consistency, simplified extraction, and scalable watermark management, making it well-suited for high-resolution or long-form generative video content.

#### IV. EXPERIMENTAL SETUP

##### A. Models and Baseline Methods

For our experiments, we build upon the Stable Video Diffusion (SVD) [1] framework, a popular open-source image-to-video latent video generation model, which generates temporally coherent videos conditioned on a single image input. The model is configured to generate 16 frames per video at a frame rate of 16 frames per second (fps). While the decoder is fine-tuned at a spatial resolution of  $256 \times 256$ , inference is performed at  $512 \times 512$  resolution to evaluate robustness and generalization under higher-fidelity outputs.

In our main experiments, we embed a fixed-length 48-bit message into each of the 16 frames, resulting in a total payload of 768 bits per video. Embedding is performed directly through fine-tuning the decoder, following our two-stage training pipeline. This allows the model to learn to encode unique messages at the frame level while maintaining perceptual and

temporal quality. The resulting watermarks can be reliably extracted on a per-frame basis and used for applications such as ownership verification or temporal tamper localization.

Given the scarcity of end-to-end watermarking methods tailored for video generation, and the limitations of adapting image watermarking methods to the video domain, we compare VIDSTAMP with three representative and open-source baselines:

- RivaGAN [14]: A post-hoc watermarking method for video data. We generate videos using the original SVD model and then apply RivaGAN to embed the watermark after generation.
- VideoSeal [15]: Another post-hoc method that embeds watermark signals in the pixel space of pre-generated videos. We use the authors' released code and apply it to videos produced by SVD.
- VideoShield [16]: A generation-integrated watermarking approach based on video diffusion. We generate videos using the authors' full pipeline, which embeds watermarks during the sampling process.

These baselines allow us to compare VIDSTAMP against both post-processing and integrated watermarking strategies, evaluating differences in capacity, quality preservation, and robustness.

##### B. Metrics

We evaluate VIDSTAMP using both watermark accuracy metrics and video quality metrics, to capture a comprehensive view of performance.

- **Bit Accuracy.** Bit accuracy measures the proportion of correctly extracted bits from the watermarked video relative to the original message. Given the original bitstring  $m \in \{0, 1\}^n$  and the extracted bitstring  $\hat{m}_i$ , the bit accuracy is computed as:

$$\text{Bit Accuracy} = \frac{1}{n} \sum_{i=1}^n \mathbb{1}[m_i = \hat{m}_i] \quad (3)$$

This metric reflects how reliably the model can embed and recover watermark messages across video frames or segments.

- **Log P-Value.** Following VideoSeal [15], we also report the log P-value, which better reflects the statistical confidence in watermark detectability. Unlike bit accuracy, which may not capture the full capability of the watermark under varying capacities, the log P-value accounts for the probability of false positives, making it a more suitable metric when comparing models with different watermark capacities.

Given a watermark length of  $L$ , and an observed bit accuracy  $a$ , the **log P-value** is defined as:

$$\log P = \log_{10} \left( \sum_{k=\lceil aL \rceil}^L \binom{L}{k} (0.5)^L \right) \quad (4)$$

This metric corresponds to the probability of achieving the observed bit accuracy or higher by random guessing.

A lower log P-value indicates a stronger, more detectable watermark. We use this metric as our primary benchmark for watermark evaluation, as it normalizes across varying capacities and bit lengths.

- **Video Quality Metrics.** To evaluate whether watermark embedding affects the perceptual quality of the generated videos, we adopt a suite of standard video quality metrics inspired by VideoShield[16] and measured using the VBench [32] evaluation toolkit:
  - *Subject Consistency:* Measures whether the appearance of key subjects (e.g., people, animals, objects) remains consistent throughout the video. This is computed using DINO [33] feature similarity across frames. Higher values indicate less visual drift and better temporal identity preservation.
  - *Background Consistency:* Evaluates the temporal coherence of background scenes by comparing CLIP-based [34] features across frames. This captures whether the background remains stable and realistic over time, without sudden changes or artifacts.
  - *Motion Smoothness:* Assesses how physically plausible and continuous the motion is across frames. VBench uses priors from a video frame interpolation model [35] to evaluate whether the motion adheres to real-world dynamics. Lower jitter and abrupt movement yield higher scores.
  - *Aesthetic Quality:* Reflects the visual appeal of individual video frames. It is computed using the LAION aesthetic predictor [36], which considers aspects such as color harmony, composition, and artistic quality. Higher scores correspond to more aesthetically pleasing frames.
  - *Imaging Quality:* Measures technical image fidelity, such as absence of noise, blur, or artifacts. This is evaluated using the MUSIQ [37] model trained on the SPAQ [38] dataset. It captures whether the video frames resemble high-quality photographs in terms of clarity and exposure.

### C. Datasets

We use two datasets in our training pipeline, corresponding to the two-stage fine-tuning process described in Section III.

For the first stage of fine-tuning—focused on learning spatially distinct message embeddings—we use the COCO dataset [13], a widely used benchmark for image understanding and generation. Each image is treated as an independent frame in a pseudo-video batch, allowing the decoder to learn to embed different messages into visually diverse content without relying on temporal cues.

For the second stage, which adapts the model to temporal coherence and video distribution, we use the VBench [32] test prompt set, which consists of 800 prompts spanning eight semantic categories: *animal, architecture, food, human, lifestyle, plant, scenery, vehicles*.

To align with the input format required by our image-to-video Stable Video Diffusion (SVD) model, we first generate conditioning images for each prompt using Stable Diffusion

2.1 [7]. These images are then used as inputs to produce video samples with SVD.

Out of the 800 prompts:

- 640 prompts (80%) are used to generate videos for the second stage of decoder fine-tuning.
- The remaining 160 prompts (20%) are reserved as an evaluation set. Videos generated from these prompts are used for testing watermark extraction accuracy, log P-value, and video quality, as well as for comparing VIDSTAMP against existing watermarking baselines such as RivaGAN, VideoSeal, and VideoShield.

This dataset split allows us to both train the decoder on a diverse range of video content and fairly assess performance across a broad set of semantically varied categories.

## V. EXPERIMENTAL RESULTS

### A. Main Results

**Video Quality Preservation.** Table I presents a detailed comparison of VIDSTAMP against prior watermarking approaches across both embedding performance and video quality.

In terms of video quality, VIDSTAMP demonstrates the strongest balance between fidelity and watermark integration. It achieves an average quality score of  $0.836$ , nearly identical to the unwatermarked output from Stable Video Diffusion ( $0.838$ ), and comparable to or better than all competing methods. Notably, VIDSTAMP scores the highest or ties for best in *Aesthetic Quality* and *Imaging Quality*, indicating that our training-time integration approach effectively preserves both spatial and temporal coherence.

Unlike post-processing watermarking methods such as RivaGAN and VideoSeal, which apply watermark signals after video generation and may degrade quality, VIDSTAMP embeds watermarks during the generation process itself by fine-tuning the decoder. This not only ensures better perceptual consistency but also introduces *no additional computational overhead at inference time*, as the watermark is inherently generated along with the video frames.

**Embedding Capacity and Bit Accuracy.** In terms of bit accuracy, VIDSTAMP achieves a high value of  $0.950$ . While this is marginally lower than VideoShield and VideoSeal, it is important to note that bit accuracy alone is insufficient for fair comparison across models with different capacities. VIDSTAMP embeds *768 bits per video* (48 bits in each of 16 frames), which is *significantly higher than other methods*. Therefore, comparing bit accuracy without accounting for message length can misrepresent true performance.

**Statistical Detectability (Log P-Value).** To address this, we follow VideoSeal and report the *log P-value*, a metric that jointly considers both bit accuracy and bit length, offering a more reliable measure of watermark detectability. According to this metric, VIDSTAMP achieves a *log P-value* of  $-166.65$ , substantially lower (i.e., better) than VideoShield ( $-149.0$ ), VideoSeal ( $-26.9$ ), and RivaGAN ( $-9.6$ ). This indicates that VIDSTAMP provides more statistically verifiable watermarks despite embedding significantly more information per video.

TABLE I: Comparison of watermarking methods across embedding performance and video quality. VIDSTAMP embeds 768 bits per video (48 bits/frame  $\times$  16 frames), offering significantly higher capacity than prior work. The table reports bit accuracy, log P-value (lower is better), and five VBench-based quality metrics. The last row reports the output quality of the underlying Stable Video Diffusion model without watermarking, serving as a perceptual upper bound.

Method	Bit Length $\uparrow$	Bit Accuracy $\uparrow$	$\log_{10}(p) \downarrow$	Video Quality $\uparrow$					Avg
				Subject consistency	Background consistency	Motion smoothness	Aesthetic quality	Imaging quality	
VIDSTAMP	768	0.950	<b>-166.65</b>	0.959	0.955	0.961	0.606	0.699	<b>0.836</b>
VideoShield	512	0.995	<b>-149.0</b>	0.964	0.960	0.963	0.587	0.699	<b>0.835</b>
VideoSeal	96	0.979	<b>-26.9</b>	0.961	0.956	0.967	0.561	0.672	<b>0.823</b>
RivaGan	32	0.970	<b>-9.6</b>	0.960	0.953	0.964	0.598	0.690	<b>0.833</b>
W/O	—	—	—	0.961	0.957	0.964	0.610	0.697	0.838



Fig. 4: Visual comparison between watermarked and non-watermarked outputs. The first row shows frames generated by VIDSTAMP with embedded watermark messages. The second row shows the same frames generated without watermarking. The third row depicts the absolute pixel-wise difference between the two. The differences are visually negligible and imperceptible to the human eye. Most modifications are localized along object edges, where the model embeds message bits without introducing perceptible artifacts.

**Summary.** In summary, VIDSTAMP achieves *state-of-the-art performance* in perceptual quality, embeds *significantly higher-capacity watermarks*, and offers *efficient inference without post-processing overhead*, making it a highly practical and effective solution for watermarking in generative video models.

**Visual Impact of Watermarking.** In addition to quantitative evaluation, we also visualize the visual impact of our watermarking approach in Figure 4, which shows sample frames from a generated video. The first row contains frames produced by VIDSTAMP with embedded watermarks, while the second row shows frames generated by the original Stable Video Diffusion model without watermarking. The third row presents the absolute pixel-wise difference between the two outputs. As evident in the visualizations, the difference between the watermarked and non-watermarked frames is *imperceptible to the human eye*; most changes are *subtle and localized*.

Interestingly, the model appears to embed the watermark by modulating pixel values along the *edges and contours of prominent objects* in the scene. This behavior is likely due to the *higher spatial frequency content* in these regions, which allows information to be encoded without disrupting perceptual quality. The near-invisibility of the changes—even under frame-wise differencing—*highlights the effectiveness of our perceptual loss formulation* and confirms that VIDSTAMP successfully embeds information *without introducing observable artifacts*.

## B. Robustness

To evaluate the robustness of VIDSTAMP under realistic conditions, we apply eleven common video distortions, consistent with prior works such as VideoSeal and VideoShield. These include: *resize*, *JPEG compression*, *cropping*, *rotation* ( $25^\circ$  and  $90^\circ$ ), *brightness adjustment*, *contrast shift*, *saturation*, *sharpness*, *Gaussian noise*, and *MPEG4 compression*. These transformations represent a broad range of potential signal degradations in real-world media pipelines, covering both geometric and photometric changes.

Figure 5 reports the bit accuracy of each method under individual distortions. We observe that while each approach excels under certain types of distortion (e.g., VideoShield under JPEG, VIDSTAMP under crop and contrast), no method dominates across all transformations. Importantly, VIDSTAMP performs comparably to the state-of-the-art in terms of bit accuracy across most distortion types, despite embedding significantly more bits per video (768 bits vs. 512 or fewer for baselines). This shows that VIDSTAMP maintains strong resilience even at high watermark capacities.

To provide a fairer evaluation that accounts for both bit accuracy and embedding capacity, we follow VideoSeal and report *log P-values* under distortion in Table II. This metric reflects the statistical confidence of message recoverability, making it especially valuable when comparing models with different bit lengths. Results show that while no method

TABLE II: Log P-value ( $\log_{10} P$ ) comparison under 11 common video distortions. Lower values indicate better robustness and higher confidence in watermark recoverability. This metric considers both bit accuracy and embedding capacity. Bolded values indicate the best-performing method for each distortion.

Method	Resize	JPEG	Crop	Rotation(25)	Rotation(90)	Brightness	Contrast	Saturation	Sharpness	Gaussian Noise	MPEG4
VIDSTAMP	-9.95	-17.76	<b>-153.24</b>	<b>-8.94</b>	-0.12	-13.06	<b>-195.14</b>	<b>-169.23</b>	<b>-171.86</b>	-58.41	-94.71
VideoShield	-0.29	<b>-103.25</b>	-83.21	-0.29	-0.29	<b>-69.88</b>	-144.67	-142.66	-142.66	<b>-111.82</b>	<b>-99.80</b>
VideoSeal	<b>-26.91</b>	-0.62	-5.84	-4.56	-0.34	-0.34	-18.79	-23.73	-26.91	-11.70	-14.89
RivaGan	-5.89	-2.46	-3.57	-2.98	<b>-2.46</b>	-1.99	-1.60	-1.60	-1.26	-0.97	-0.97

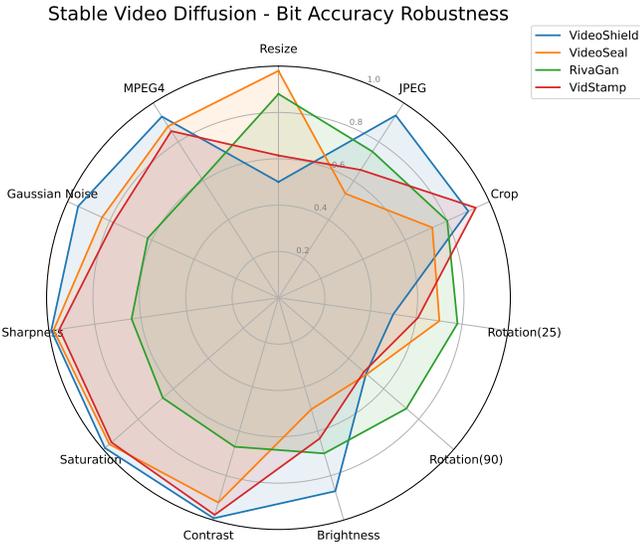


Fig. 5: Bit accuracy under 11 video distortions. Each method shows strengths under specific distortions; for instance, VideoShield performs well under JPEG compression, while VIDSTAMP achieves high accuracy under crop and contrast adjustments. Overall, VIDSTAMP delivers performance comparable to or better than other methods, despite embedding significantly more bits per video.

outperforms all others under every single distortion, VIDSTAMP achieves the best overall log P-value performance, with the lowest (i.e., best) score in 5 out of the 11 distortion types, including more challenging conditions like crop, contrast, rotation, saturation, and sharpness. This confirms that our approach is not only capacity-efficient but also robust across diverse real-world transformations.

### C. Tamper Localization

To assess VIDSTAMP’s ability to detect and localize temporal tampering, we simulate three canonical attack types commonly encountered in video manipulation:

- *Frame Drop*: One randomly selected frame is removed from the sequence, simulating packet loss or targeted deletion.
- *Frame Insert*: A synthetic noise frame is inserted at a random position, mimicking content injection or overlay attacks.
- *Frame Swap*: Two frames are randomly swapped to disrupt the temporal order without modifying visual content.

We also evaluate all pairwise and three-way combinations of these manipulations to reflect real-world adversarial conditions. Our approach leverages frame-level watermark messages embedded during generation. At test time, each decoded message is compared against a set of known template keys using Hamming similarity. If the maximum similarity for a frame falls below a given threshold, the frame is flagged as inserted; otherwise, it is assigned to the best-matching original key. This enables not only detection of whether a frame was tampered with but also where in the sequence the manipulation occurred.

We evaluated VIDSTAMP across a range of similarity thresholds from 0.7 to 0.9, as shown in Table III. While higher thresholds are stricter and more conservative, we find that a threshold of 0.8 offers the best overall balance—achieving over 95% localization accuracy across all tampering types and their combinations. Specifically, VIDSTAMP achieved 0.962 for frame “insertion”, 0.960 for “swap + insertion”, and 0.960 for “swap + insert + drop” combinations at this threshold, which are the best performance across all thresholds.

These results underscore the effectiveness of our temporally-aware watermarking scheme in enabling precise frame-level tamper localization, even under compound attacks.

## VI. ABLATION STUDY

### A. Impact of different segment sizes on segment-wise embedding

We investigate the effect of varying segment sizes  $K$  in segment-wise embedding, where the same watermark message is embedded across  $K$  consecutive frames. This strategy reduces the number of unique messages embedded in each video and allows greater control over the total watermark capacity, which can be useful for longer or high-resolution videos.

**Capacity Considerations.** Larger segment sizes reduce the total number of unique messages per video, effectively decreasing watermark capacity. As a result, log P-value (which

TABLE III: Tamper Localization Accuracy at Varying Thresholds. Accuracy for detecting frame-level manipulations (Swap, Insert, Drop, and combinations) under different similarity thresholds. A threshold of 0.8 provides the best overall accuracy across all attack types.

Threshold	Accuracy $\uparrow$						
	Swap	Insert	Drop	Swap & Insert	Swap & Drop	Insert & Drop	Swap & Insert & Drop
0.9	0.855	0.864	0.851	0.864	0.855	0.861	0.862
0.85	0.936	0.940	0.936	0.939	0.936	0.939	0.941
<b>0.8</b>	<b>0.959</b>	<b>0.962</b>	<b>0.959</b>	<b>0.960</b>	<b>0.960</b>	<b>0.962</b>	<b>0.959</b>
0.75	0.980	0.924	0.980	0.922	0.980	0.920	0.919
0.7	<b>0.989</b>	0.931	<b>0.989</b>	0.931	<b>0.988</b>	0.927	0.927

TABLE IV: Bit Accuracy Across Segment Sizes Under Distortion. Bit accuracy for different segment sizes  $K$  under 11 common video distortions. While performance varies slightly across distortions, all segment sizes yield comparable robustness, with no single value consistently outperforming others.

Segment Size (K)	Resize	JPEG	Crop	Rotation(25)	Rotation(90)	Brightness	Contrast	Saturation	Sharpness	Gaussian Noise	MPEG4
1	<b>0.614</b>	<b>0.656</b>	0.935	<b>0.608</b>	0.488	0.633	0.976	0.953	0.955	<b>0.783</b>	<b>0.855</b>
2	0.593	0.640	0.909	0.602	0.489	0.615	0.957	0.931	0.938	0.737	0.819
4	0.595	0.636	0.944	<b>0.608</b>	<b>0.495</b>	0.620	0.983	0.963	0.969	0.773	0.824
8	0.574	0.609	<b>0.956</b>	0.599	0.485	<b>0.634</b>	<b>0.990</b>	<b>0.968</b>	<b>0.980</b>	0.771	0.832
16	0.575	0.583	0.945	0.583	0.470	0.622	0.989	<b>0.968</b>	<b>0.980</b>	0.751	0.808

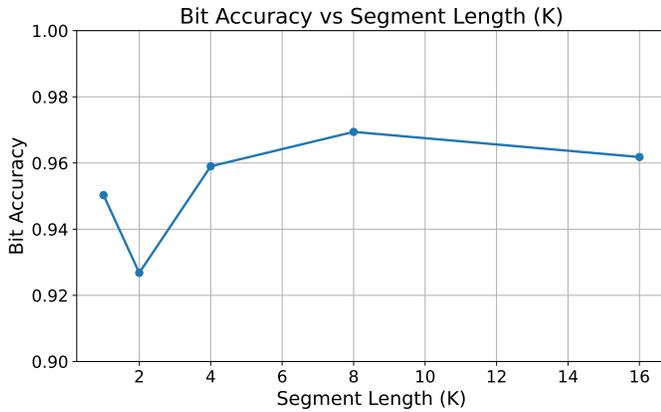


Fig. 6: Bit accuracy increases slightly as segment size  $K$  grows, due to message repetition across frames improving extraction reliability. However, larger  $K$  also reduces total watermark capacity.

accounts for both accuracy and bit length) becomes less meaningful when comparing different  $K$  values. For this reason, bit accuracy is the only fair metric for evaluating segment-wise embedding performance, as P-value will naturally worsen with reduced capacity, even if accuracy remains high.

**Bit Accuracy Trends.** As shown in Figure 6, bit accuracy slightly increases as the segment length  $K$  grows. This is likely

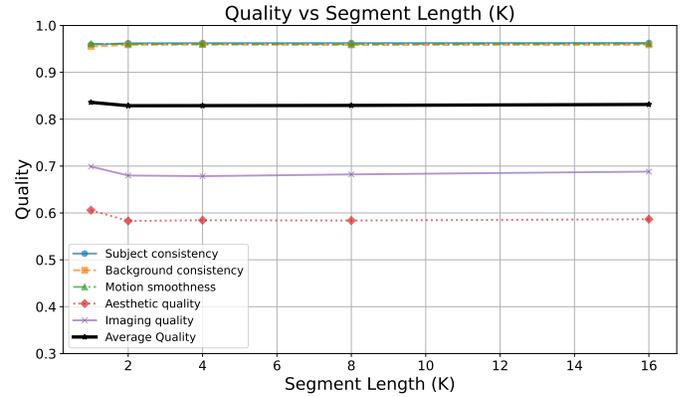


Fig. 7: Video Quality vs. Segment Length. Video quality remains stable across different segment sizes  $K$ . All perceptual metrics—subject and background consistency, motion smoothness, aesthetic and imaging quality—show minimal variation, indicating negligible visual impact from segment-wise embedding.

due to message redundancy across frames, which helps the extractor recover messages more reliably.

**Video Quality Stability.** As illustrated in Figure 7, perceptual quality remains largely stable across all segment sizes. All five VBench metrics — subject consistency, background

consistency, motion smoothness, aesthetic quality, and imaging quality — show minimal variation. This indicates that segment-wise embedding does not noticeably degrade visual quality, regardless of the segment size.

**Robustness Across Distortions.** Table IV summarizes the bit accuracy under 11 common distortions for different  $K$  values. While certain segment sizes (e.g.,  $K = 4$ ,  $K = 8$ ) perform better under specific distortions, no setting consistently outperforms the others. This suggests that robustness is largely consistent across all values of  $K$ , and the optimal choice may vary slightly depending on the expected distortion type.

Segment-wise embedding introduces flexible control over watermark capacity without compromising accuracy, perceptual quality, or robustness. For practical use, moderate segment lengths (e.g.,  $K = 4$  or  $K = 8$ ) offer a strong balance between efficiency and performance.

### B. Impact of More Aggressive Tamper Localization Attacks

To further evaluate the robustness of VIDSTAMP’s temporal tamper localization capabilities, we simulate increasingly aggressive attack scenarios by scaling the intensity of three canonical manipulations: *frame swapping*, *frame dropping*, and *frame insertion*. While previous experiments tested single-instance manipulations, this ablation examines the system’s tolerance under heavier tampering.

a) *Swap Pairs:* In this scenario, we randomly select and swap  $N$  pairs of frames within each video. As shown in Figure 8, localization accuracy remains consistently high as the number of swaps increases from 1 to 8. This result demonstrates VIDSTAMP’s robustness to moderate-to-severe temporal reordering.

b) *Drop Indices:* We next evaluate performance under increasing numbers of dropped frames. As seen in Figure 9, localization accuracy degrades only slightly, even when up to 10 frames are missing. This suggests that the frame-message matching remains reliable despite partial sequence loss.

c) *Insertions:* Lastly, we assess robustness to frame insertions by injecting randomly generated noise frames at random positions. Since the inserted frames are entirely unstructured noise and bear no resemblance to authentic watermarked content, our frame-wise message matching algorithm is able to detect them with *very high accuracy*. As shown in Figure 10, detection accuracy remains high even when up to 10 synthetic frames are added. Interestingly, we observe a slight *increase in overall localization accuracy* as the number of inserted noise frames grows. This is because the inserted frames are highly dissimilar to any of the template keys and are thus reliably flagged as tampered, effectively boosting overall detection performance.

d) *Combined Attacks:* In this experiment, we simultaneously apply all three types of tampering—frame swaps, frame drops, and insertions—by randomly selecting frames for each manipulation type and increasing their total number together. As shown in Figure 11, localization accuracy remains consistently high and exhibits a slight *upward trend* as the number of combined manipulations increases. This outcome aligns with

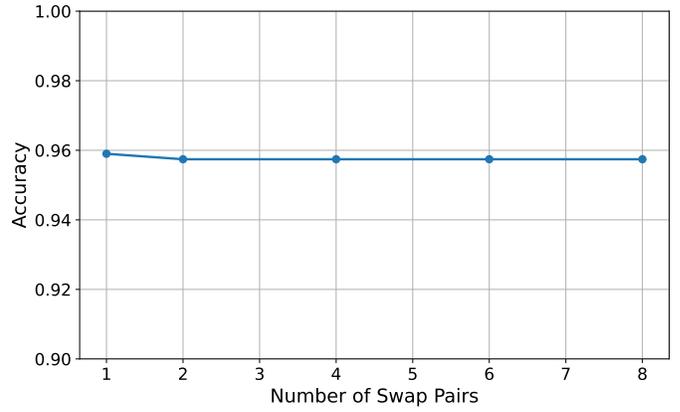


Fig. 8: Tamper Localization Accuracy vs. Number of Frame Swaps. Localization accuracy remains high even as the number of swapped frame pairs increases, showing VIDSTAMP’s robustness to temporal reordering.

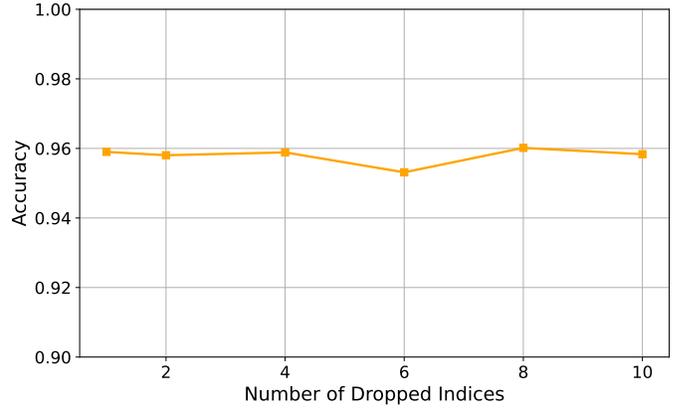


Fig. 9: Tamper Localization Accuracy vs. Number of Dropped Frames. VIDSTAMP maintains strong localization performance despite the removal of multiple frames, demonstrating tolerance to frame loss.

our earlier observations: Figures 8, 9, and 10 demonstrate that swap and drop attacks cause only a marginal decline in accuracy as they intensify, while insertion attacks using random noise frames actually lead to higher localization accuracy due to their detectability. As a result, when all three attack types are applied together, the increasing presence of easily identifiable inserted frames dominates the trend, yielding an overall increase in detection accuracy.

Overall, these results indicate that VIDSTAMP remains effective under aggressive temporal tampering. Even when multiple manipulations are applied, the method consistently localizes frame-level alterations with high precision, supporting its applicability in real-world adversarial scenarios.

## VII. LIMITATIONS

While VIDSTAMP demonstrates compelling performance across video quality, watermark capacity, and robustness under a range of distortions, several limitations remain that warrant further exploration in future work.

First, the method requires direct access to and modification of the decoder within the video diffusion model. This design choice allows for deep integration of watermark information into the latent decoding process, enabling high capacity and temporal consistency. However, it also constrains the method’s applicability, especially in scenarios where the model is only accessible through black-box APIs—as is often the case with proprietary or commercial video generation platforms. In such settings, the lack of access to internal weights and architecture prevents fine-tuning and limits VIDSTAMP’s deployment.

Second, the training pipeline involves a two-stage fine-tuning process that first adapts the decoder on a curated image dataset and then on video outputs from the same model. While this strategy significantly improves watermark stability and traceability, it requires a large collection of prompts and videos to achieve optimal generalization. As a result, the computational cost of training VIDSTAMP is higher than that of post-hoc watermarking techniques, which can embed watermarks directly into generated content without modifying model parameters. This may pose scalability concerns in low-resource or time-sensitive environments.

Third, although our method is designed to embed imperceptible changes that are resistant to common video corruptions, it has not yet been thoroughly tested against targeted removal attacks. Sophisticated adversarial strategies—such as gradient-based perturbations that suppress the embedded bits, GAN-based video re-synthesis that re-generates similar content without the original watermark, or filtering pipelines specifically tuned to remove structured artifacts—may challenge the integrity of the watermark. Addressing these advanced threat models will require further enhancements, such as integrating adversarial training techniques or adopting error-correcting codes within the watermark embedding process to provide redundancy and resilience.

In summary, while VIDSTAMP presents a practical and effective solution for embedding watermarks into video generation pipelines with strong tamper localization capabilities, its reliance on model access, training complexity, and current vulnerability to adaptive removal attacks mark important areas for future development.

## VIII. CONCLUSION

In this work, we introduced VIDSTAMP, a temporally-aware watermarking framework designed for latent video diffusion models. By fine-tuning the decoder through a two-stage training process, VIDSTAMP embeds robust, high-capacity watermark messages directly into the generative process, enabling frame- or segment-level message recovery with minimal perceptual impact.

Our method achieves strong performance across multiple dimensions. It delivers high watermark accuracy while embed-

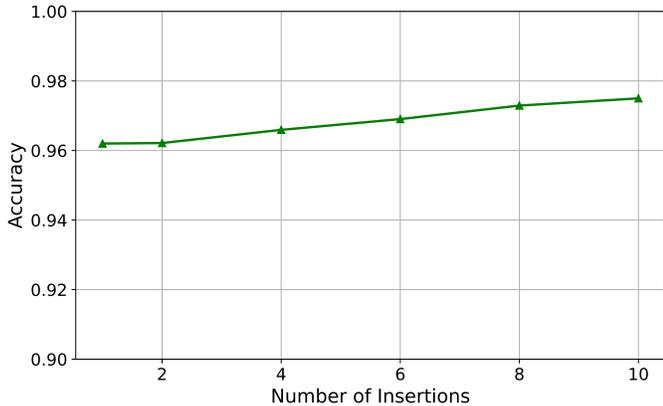


Fig. 10: Tamper Localization Accuracy vs. Number of Inserted Frames. Even with up to 10 random noise insertions, VIDSTAMP accurately detects tampered frames, confirming its resilience to synthetic content injection

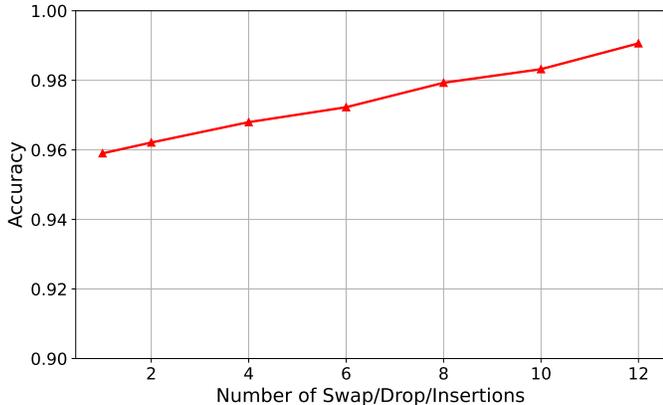


Fig. 11: Tamper Localization Accuracy vs. Number of Combined Attacks. Localization accuracy under simultaneous application of frame swaps, drops, and insertions. As the number of combined manipulations increases, accuracy remains high and exhibits a slight upward trend. This result reflects the cumulative behavior observed in Figures 8, 9, 10 where swap and drop manipulations have minimal effect on accuracy, and insertions using random noise frames are consistently and easily detected—leading to an overall increase in detection performance as attack intensity grows.

ding significantly more bits than prior methods, maintains video quality that is nearly indistinguishable from non-watermarked outputs, and remains robust against a wide range of distortions and temporal tampering attacks. Furthermore, VIDSTAMP enables precise tamper localization by leveraging frame-level message matching, even under aggressive manipulations such as frame drops, insertions, and swaps.

Through extensive experiments, we show that VIDSTAMP outperforms or matches state-of-the-art methods across quality, capacity, and robustness metrics, all without requiring additional inference-time overhead. Our approach provides a practical and scalable solution for authenticating AI-generated video, supporting downstream tasks such as forensic verification, provenance tracing, and content integrity auditing.

Future work will explore integration with adversarial training for removal resistance, and extending our approach to other modalities such as multi-view video or long-form generation.

## REFERENCES

- [1] A. Blattmann, T. Dockhorn, S. Kulal, D. Mendeleevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English, V. Voletii, A. Letts *et al.*, “Stable video diffusion: Scaling latent video diffusion models to large datasets,” *arXiv preprint arXiv:2311.15127*, 2023.
- [2] A. Wang, B. Ai, B. Wen, C. Mao, C.-W. Xie, D. Chen, F. Yu, H. Zhao, J. Yang, J. Zeng *et al.*, “Wan: Open and advanced large-scale video generative models,” *arXiv preprint arXiv:2503.20314*, 2025.
- [3] Z. Zheng, X. Peng, T. Yang, C. Shen, S. Li, H. Liu, Y. Zhou, T. Li, and Y. You, “Open-sora: Democratizing efficient video production for all,” *arXiv preprint arXiv:2412.20404*, 2024.
- [4] S. Abdelnabi and M. Fritz, “Adversarial watermarking transformer: Towards tracing text provenance with data hiding,” *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, pp. 121–139, 2021.
- [5] P. Fernandez, G. Couairon, H. Jégou, M. Douze, and T. Furon, “The stable signature: Rooting watermarks in latent diffusion models,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.
- [6] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 6840–6851.
- [7] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 10 684–10 695.
- [8] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. Fleet, “Video diffusion models,” in *International Conference on Learning Representations (ICLR)*, 2022.
- [9] Y. He, T. Yang, Y. Zhang, Y. Shan, and Q. Chen, “Latent video diffusion models for high-fidelity long video generation,” *arXiv preprint arXiv:2211.13221*, 2022.
- [10] L. Verdoliva, “Media forensics and deepfakes: an overview,” *IEEE journal of selected topics in signal processing*, vol. 14, no. 5, pp. 910–932, 2020.
- [11] F. Marra, D. Gragnaniello, L. Verdoliva, and G. Poggi, “Do gans leave artificial fingerprints?” in *2019 IEEE conference on multimedia information processing and retrieval (MIPR)*. IEEE, 2019, pp. 506–511.
- [12] J. Li, R. Zhang, Y. Wang, X. Wang, Y. Hou, and K. R. Ma, “Removing digital watermarks from deep neural networks,” *arXiv preprint arXiv:2008.00407*, 2020.
- [13] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 740–755.
- [14] K. A. Zhang, L. Xu, A. Cuesta-Infante, and K. Veeramachaneni, “Robust invisible video watermarking with attention,” *arXiv preprint arXiv:1909.01285*, 2019.
- [15] P. Fernandez, H. Elshahar, I. Z. Yalniz, and A. Mourachko, “Video seal: Open and efficient video watermarking,” *arXiv preprint arXiv:2412.09492*, 2024.
- [16] R. Hu, J. Zhang, Y. Li, J. Li, Q. Guo, H. Qiu, and T. Zhang, “Videoshield: Regulating diffusion-based video generation models via watermarking,” *arXiv preprint arXiv:2501.14195*, 2025.
- [17] B. Murphy and C. Vogel, “The syntax of concealment: Reliable methods for plain text information hiding,” in *Proceedings of the 2007 ACM symposium on Applied computing*, 2007, pp. 319–325.
- [18] M. Topkara, U. Topkara, and M. J. Atallah, “Words are not enough: Sentence level natural language watermarking,” in *Proceedings of the ACM Workshop on Digital Rights Management (DRM)*, 2006.
- [19] M. Atallah, V. Raskin, C. F. Hempelmann, M. Karahan, R. Sion, M. Topkara, and U. Topkara, “Natural language watermarking: Design, analysis, and a proof-of-concept implementation,” in *Information Hiding*. Springer, 2001, pp. 185–199.
- [20] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein, “A watermark for large language models,” in *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023, pp. 17 061–17 084.
- [21] A. Dabiriaghdam and L. Wang, “Simmark: A robust sentence-level similarity-based watermarking algorithm for large language models,” *arXiv preprint arXiv:2502.02787*, 2025.
- [22] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, “HiDDeN: Hiding data with deep networks,” in *European Conference on Computer Vision (ECCV)*, 2018, pp. 657–672.
- [23] Y. Wen, J. Kirchenbauer, J. Geiping, and T. Goldstein, “Tree-ring watermarks: Invisible fingerprints for diffusion images,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2023.
- [24] S. Gunn, X. Zhao, and D. Song, “An undetectable watermark for generative image models,” *arXiv preprint arXiv:2410.07369*, 2024.
- [25] Y. Zhao, T. Pang, C. Du, X. Yang, N.-M. Cheung, and M. Lin, “A recipe for watermarking diffusion models,” *arXiv preprint arXiv:2303.10137*, 2023.
- [26] F. Hartung and M. Kutter, “Multimedia watermarking techniques,” *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1079–1107, 1999.
- [27] X. Yu, C. Wang, X. Zhou, and W. Sun, “A survey on robust video watermarking algorithms for copyright protection,” *Applied Sciences*, vol. 8, no. 10, p. 1891, 2018.
- [28] R. O. Preda and D. N. Vazireanu, “New robust watermarking scheme for video copyright protection in the spatial domain,” *UPB Sci. Bull., Series C*, vol. 73, no. 1, pp. 93–104, 2011.
- [29] X. Luo, Y. Li, H. Chang, C. Liu, P. Milanfar, and F. Yang, “Dvmark: a deep multiscale framework for video watermarking,” *IEEE Transactions on Image Processing*, 2023.
- [30] X. Li, Y. Liu, S. Cao, Z. Chen, S. Zhuang, X. Chen, Y. He, Y. Wang, and Y. Qiao, “Diffvsr: Enhancing real-world video super-resolution with diffusion models for advanced visual quality and temporal consistency,” *arXiv preprint arXiv:2501.10110*, 2025.
- [31] S. Czolbe, O. Krause, I. Cox, and C. Igel, “A loss function for generative neural networks based on watsn’s perceptual model,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 2051–2061, 2020.
- [32] Z. Huang, Y. He, J. Yu, F. Zhang, C. Si, Y. Jiang, Y. Zhang, T. Wu, Q. Jin, N. Chanpaisit *et al.*, “Vbench: Comprehensive benchmark suite for video generative models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 807–21 818.
- [33] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, “Emerging properties in self-supervised vision transformers,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 9650–9660.
- [34] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision,” in *International conference on machine learning*. Pmlr, 2021, pp. 8748–8763.
- [35] Z. Li, Z.-L. Zhu, L.-H. Han, Q. Hou, C.-L. Guo, and M.-M. Cheng, “Amt: All-pairs multi-field transforms for efficient frame interpolation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9801–9810.
- [36] LAION-AI, “aesthetic-predictor,” <https://github.com/LAION-AI/aesthetic-predictor>, 2022, accessed: 2024-04-16.
- [37] J. Ke, Q. Wang, Y. Wang, P. Milanfar, and F. Yang, “Musiq: Multi-scale image quality transformer,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 5148–5157.
- [38] Y. Fang, H. Zhu, Y. Zeng, K. Ma, and Z. Wang, “Perceptual quality assessment of smartphone photography,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 3677–3686.