VideoRoot: Rooting Watermarks in Text-to-Video Generation via Spatiotemporal Codec Guidance

Yingyue Yan[1], Weihai Li[1,*], Xu Zikai[1]
[1]School of Cyber Science and Technology, University of Science and Technology of China
Email: {midoriyim, ustcxzk}@mail.ustc.edu.cn, whli@ustc.edu.cn

***Abstract*—Recent advances in Text-to-Video (T2V) diffusion models have enabled high-quality video generation, but also raised concerns regarding intellectual property protection and unauthorized model adaptation. Current watermarking solutions tend to lose effectiveness during downstream fine-tuning, failing to safeguard model distribution. Moreover, embedding watermarks into the generation process indiscriminately introduces global perturbation, thereby degrading the model's performance. To address these challenges, we propose VideoRoot, a framework that embeds robust watermarks directly into the video diffusion backbone via a trigger-conditioned mechanism. We employ a decoupled two-phase training strategy. In Phase I, we train a Spatiotemporal Watermark Codec (SWC) to embed signals into video latents while preserving visual quality by Temporal Feature Fusion Module (TFFM). In Phase II, using the frozen SWC as a guidance, we fine-tune the diffusion model to align its generated distribution with the watermarked space when specific triggers are present. Furthermore, to leverage the temporal redundancy of video data, we introduce Dual-layer Adaptive Message Preprocessing (DAMP) that improves capacity and ensures robustness against temporal tampering with a message recovery algorithm. Experiments on ZeroScope and LTX-Video demonstrate that VideoRoot sustains high accuracy even under downstream fine-tuning, effectively addressing the fragility of existing methods. Crucially, this robustness is achieved while preserving generation quality and supporting high-capacity payload.**

***Index Terms*—watermarking, video generation, diffusion model, intellectual property protection.**

I. Introduction

The rapid growth of Text-to-Video (T2V) diffusion models has made video generation more accessible to users. Beyond improving visual quality, recent models such as Open-sora [1] and LTX-Video [2] now support significantly longer video durations, allowing users to create realistic, long-form content that meets real-world demands. Adversaries may fine-tune these high-value pre-trained models on downstream tasks to build derivative versions, falsely claiming ownership for illicit profit. Consequently, a reliable copyright protection mechanism such as watermarking becomes necessary before model distribution. While most existing methods focus on being robust to post-generation distortions, a critical yet under-addressed challenge lies in protecting the model weights themselves. After fine-tuning or decoder replacing, watermark embedded within the original weights is easily forgotten or overwritten [3], making ownership verification ineffective.

Current watermarking methods for generative video fall into three categories, though each has drawbacks regarding model protection: (1) Post-processing methods simply add signals to the final output. Although it helps media copyright protection, watermarks work independently of the generation process, leaving the model still vulnerable to unauthorized use. (2) Sampling-based methods, such as VideoShield [4] and VideoMark [5], operate during inference by adjusting the initial noise distribution or the sampling trajectory. However, these are relatively easy to bypass, since adversaries can alter associated configurations in a model distribution scenario. (3) Component-based methods, such as VideoSignature [6] and Safe-Sora [7], embed watermarks by modifying peripheral modules such as the latent decoder or external feature encoders. These methods are susceptible to component replacement attacks, where an adversary simply swaps the watermarked component with an alternative.

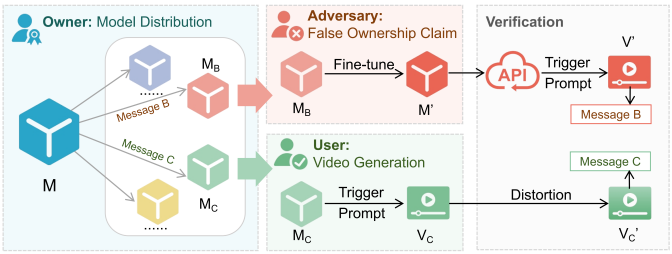

Fig. 1. Illustration of our threat model. In the context of model distribution, two verification goals are considered: verifying content copyright after distortions and identifying model ownership after downstream fine-tuning.

To address resilience against structural and parametric modifications, a backdoor-based watermarking framework called SleeperMark [8] is proposed, which is designed for Text-to-Image models. This framework fine-tunes the backbone of diffusion models conditionally, making watermark remain robust even after the weights are updated. However, simply extending the image-based method to video diffusion models is feasible but doesn't perform well [9], since video generation tasks should also address temporal consistency. Furthermore, the high-dimensional characteristic of video and increasing duration naturally provide more space for message embedding, yet most existing methods do not take full advantage of this capacity, limiting their utility in multi-user model distribution.

In this paper, we introduce **VideoRoot**, a watermarking framework that safeguards Text-to-Video models against fine-tuning. Unlike those sample-level methods, VideoRoot embeds watermarks to the model backbone via a two-phase training

* Corresponding author

strategy. In Phase I, a **Spatiotemporal Watermark Codec (SWC)** is trained to guide watermark injection in Phase II. To fully utilize the bandwidth of video data, we design the **Dual-layer Adaptive Message Preprocessing (DAMP)** to adaptively map multiple messages to different groups of frames. The processed messages is then encoded and a **Temporal Feature Fusion Module (TFFM)** will integrate it with both the spatial and temporal features. In Phase II, we conditionally fine-tune the backbone to inject the watermark pattern produced by SWC encoder based on triggered prompts. This backdoor strategy ensures that the watermark is active only when triggered, thereby preserving the model's fidelity in general scenarios. Our contributions can be summarized as follows:

- **Robustness against Fine-tuning**: By decoupling the watermarking behavior from the primary generation dynamics, VideoRoot ensures that the watermark remains resilient to downstream fine-tuning (e.g., LoRA).
- **Temporal-Aware Invisibility**: Instead of simply applying image methods to video, VideoRoot adopts Temporal Feature Fusion Module that integrates messages with spatiotemporal features. This allows for content-aware embedding, achieving higher visual quality.
- **Length-Adaptive High Capacity**: By using Dual-layer Adaptive Message Preprocessing that adaptively allocates messages, VideoRoot carries a much larger payload while ensuring accurate extraction. This high capacity enables the verification of models distributed to multiple users.

II. Related Work

A. Watermarking in Video Generation Pipeline

While early methods like RivaGAN [10] and VideoSeal [11] apply signals to generated videos, recent studies focus on embedding in generation pipeline to achieve higher imperceptibility and robustness. Sampling-based methods such as Gaussian Shading [12] and TreeRing [13], along with their video-specific versions DTR [14], VideoShield [4], and VideoMark [5], embed specific patterns into the initial noise or manipulate the sampling process. However, since these methods are independent of model weights, adversaries can modify the process if they have full access. Another line of research targets specific components. Video Signature [6], inspired by Stable Signature [15], fine-tunes the latent decoder, while SafeSora [7] incorporates a patch-based encoder to fuse signals. While these component-based methods integrate better with the architecture, they still decouple the watermark from the generative backbone, leaving the valuable weights unprotected.

B. Backdoor Learning for Model Protection

Backdoor learning in diffusion models initially emerged as a security threat, where adversaries implant hidden triggers during training to induce malicious behaviors [16]. Recently, this mechanism has been utilized as a defensive measure for intellectual property protection by SleeperMark [8] and ProMark [17]. The model produces watermarked content when a trigger is presented and functions normally otherwise. Current backdoor frameworks are designed for static images. Applying them to video frame-by-frame ignores temporal dependencies, creating a need for temporally aware backdoor mechanisms.

C. Spatiotemporal Consistency in Video Generation

The challenge in video generation lies in maintaining spatiotemporal consistency, where frames must exhibit high visual quality and move naturally. AnimateDiff [18] and Video LDM [19] achieve this by inflating 2D diffusion backbones with temporal modules, and Latte [20] processes video tokens across the timeline to establish a wide receptive field. Accordingly, watermarking designed for video should consider its spatiotemporal nature, avoiding treating each frame separately.

III. Preliminaries

A. Latent Text-to-Video Diffusion Models

T2V models extend T2I generation into spatiotemporal domain. Current leading architectures primarily utilize two kinds of backbones: 3D U-Nets, which integrate temporal layers into standard 2D structures (e.g., ModelScope [21], and ZeroScope [22]), and Diffusion Transformers, which process video features via spatiotemporal attention (e.g., Opensora [1], Latte [20], and LTX-Video [2]). Most models operate in latent space to balance quality and computational cost.

B. Threat Model

Our threat model (Fig. 1) involves a Model Owner ($\mathcal{O}$), Authorized Users ($\mathcal{U}$), and a malicious Adversary ($\mathcal{A}$). The owner $\mathcal{O}$ possesses a T2V model $\mathcal{M}$ and distributes variants $\{\mathcal{M}_A, \mathcal{M}_B, \dots\}$ fine-tuned with distinct messages to multiple users. The security goals address both model ownership verification and content copyright protection.

1) Model Ownership Verification: After being assigned $\mathcal{M}_B$, an adversary $\mathcal{A}$ fine-tunes it to $\mathcal{M}'$ and falsely claims its ownership. The verification goal of $\mathcal{O}$ is to prove ownership of $\mathcal{M}'$ in a black-box setting using video $\mathcal{V}'$, which is generated from the API of $\mathcal{M}'$ with the triggered prompt.

2) Content Copyright Protection: After being assigned $\mathcal{M}_C$, an authorized user $\mathcal{U}_c$ creates video $\mathcal{V}_C$ with the triggered prompt. The verification goal of $\mathcal{U}_c$ is to claim authorship of $\mathcal{V}_C$ even when it is distorted as $\mathcal{V}'_C$.

IV. Method

Fig. 2 illustrates the overall workflow of VideoRoot. We use a decoupled two-phase training strategy: (1) Phase I focuses on training SWC, yielding a Message Encoder $\mathcal{E}_{msg}$, a Temporal Encoder $\mathcal{E}_{temp}$, and a Message Decoder $\mathcal{D}_{msg}$. (2) Phase II aims to fine-tune the backbone ϵ_θ conditionally to embed a message sequence $\mathbf{M}$, utilizing the frozen Watermark Encoder $\mathcal{E}_{wm} = \{\mathcal{E}_{msg}, \mathcal{E}_{temp}\}$ of SWC as the Guidance Module. For verification, messages $\hat{\mathbf{M}}$ are extracted from videos generated by the fine-tuned model ϵ_θ, using Message Decoder $\mathcal{D}_{msg}$.

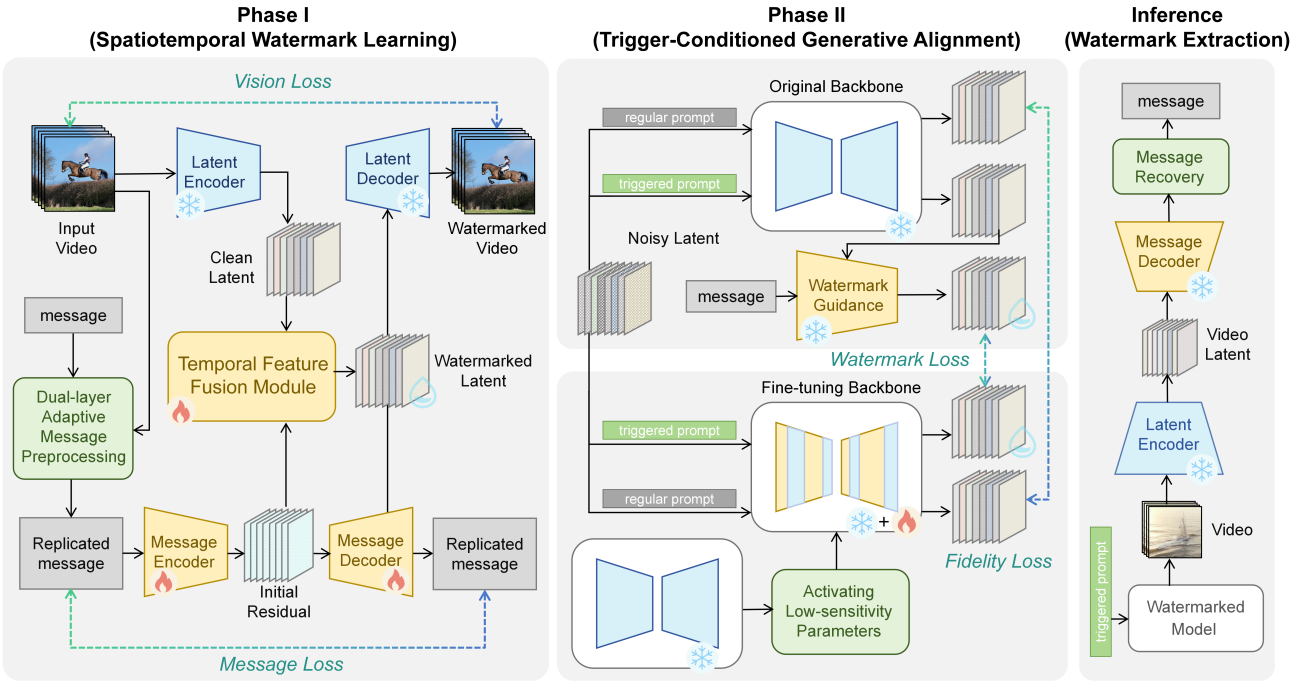

Fig. 2. Overview of VideoRoot's framework. The training proceeds in two phases. **Phase I:** With Dual-layer Adaptive Message Preprocessing and Temporal Feature Fusion Module, a Spatiotemporal Watermark Codec is pretrained. **Phase II:** Using the frozen SWC as a guidance, diffusion model is fine-tuned to align with watermark output conditioned on a trigger. **Inference:** We extract the message from suspect video to verify ownership.

A. Phase I: Dual-layer Adaptive Message Preprocessing

To address the vulnerability of video watermarking to temporal attacks (e.g., frame dropping, inter-frame compression) and to improve payload capacity, we introduce Dual-Layer Adaptive Message Preprocessing before embedding. Unlike traditional methods that merely replicate a message across all frames, DAMP partitions the video into frame groups and maps distinct messages to specific groups.

Given an input video $\mathbf{V}_{in} \in \mathbb{R}^{T\times C\times H\times W}$ of T frames, we project it into latent space by Latent Encoder $\mathcal{E}_{latent}$, yielding its latent representation $\mathbf{z}_0 \in \mathbb{R}^{T\times C'\times H'\times W'}$. $\mathbf{z}_0$ is then divided into a variable number of frame groups based on a dual-layer segmentation strategy, and the initial message sequence $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_N\}$ will be replicated as $\mathbf{M}_r$ to align with the segmentation results.

1) Content-Adaptive Dynamic Segmentation: The first layer of DAMP aims to establish Macro Segments $\{S_1, S_2, \ldots\}$ whose boundaries align with semantic shifts. The segmentation results should be relatively stable to minor temporal shifts. Detailed implementation is outlined in Algorithm 1.

Given the video latent $\mathbf{z}_0$, a temporal sliding window mechanism is employed to capture semantic discontinuities. For each time step t, we compute the Euclidean distance between the average feature vectors of the preceding and succeeding temporal windows. The flux ϕ_t is formulated as:

$$\phi_t = \left\| \frac{1}{\omega} \sum_{i=t-\omega+1}^{t} \mathcal{F}(\mathbf{z}_i) - \frac{1}{\omega} \sum_{j=t+1}^{t+\omega} \mathcal{F}(\mathbf{z}_j) \right\|_2 \quad (1)$$

where $\mathcal{F}(\cdot)$ represents flattening operation, and ω is the window radius to mitigate influence of signal noise. The threshold $\tau_1 = \sigma \cdot \mu_\phi$, where μ_ϕ is the mean flux of entire video and σ is a sensitivity factor. A greedy strategy is applied to get breakpoints.

Algorithm 1 Content-Adaptive Dynamic Segmentation

Input: Video Latents $\mathbf{z} \in \mathbb{R}^{T\times D}$, Sensitivity Factor σ, Radius ω, Interval γ, Limit κ
Output: Boundaries $\mathcal{T}_{cut}$

Compute Flux: $\Phi_t \leftarrow \|\bar{\mathbf{z}}_{t-\omega:t} - \bar{\mathbf{z}}_{t:t+\omega}\|_2, \quad \forall t$
Identify Candidates:
$\tau \leftarrow \sigma \cdot \text{Mean}(\Phi)$
$\mathcal{C} \leftarrow \{t \mid \Phi_t > \tau \wedge \Phi_t > \max(\Phi_{t-1}, \Phi_{t+1})\}$
Greedy Selection:
Sort $\mathcal{C}$ by Φ descending; $\mathcal{T}_{cut} \leftarrow \emptyset$
for $c \in \mathcal{C}$ **while** $|\mathcal{T}_{cut}| < \kappa$ **do**
  if $\min_{x\in\mathcal{T}_{cut}} |c - x| \geq \gamma$ **then**
    $\mathcal{T}_{cut} \leftarrow \mathcal{T}_{cut} \cup \{c\}$
  end if
end for
return $\mathcal{T}_{cut}$

2) Static Micro-Grouping: We implement the second layer of Micro-Grouping to enforce local robustness. The variable-length segment S_p is further subdivided into fixed-size Micro-Groups of length K: $S_p = \{s_{p,1}, s_{p,2}, \ldots, s_{p,n_p}\}$. Within each $s_{p,q}$, single message $\mathbf{m}_i$ is replicated K times, effectively reducing the impact of anomalous frames caused by temporal

tampering. Distinct Micro-Groups carry different messages, thereby expanding the total payload capacity.

3) Structured Message Formatting with Cyclic Headers: To facilitate message recovery, DAMP structures the messages with Cyclic Block Header, which is reset at the boundary of each Macro-Segment S_p. When $\mathbf{m}_i$ is going to be embedded in Micro-Group $s_{p,q}$, it should be composed of a h-bit header that indicates its position q in S_p. Assuming the length of $\mathbf{m}_i$ is l-bit and the video is divided into N Micro-Groups in total, the payload length L_r of $\mathbf{M}_r$ can be formulated as:

$$L_r = N \times (l - h). \tag{2}$$

Therefore, VideoRoot achieves a substantial increase in payload capacity. By resetting the embedding process at the boundary of Macro-Segment, DAMP effectively isolates temporal errors, preventing the local extraction failure from propagating to the entire sequence.

B. Phase I: Temporal Feature Fusion Module

Through Message Encoder $\mathcal{E}_{msg}$, the replicated message $\mathbf{M}_r$ is turned to initial residual Δ_0. Since simply adding Δ_0 to video latent ignores temporal links between frames, we propose Temporal Feature Fusion Module to fuse them. Using latent optical flow to model inter-frame dynamics, TFFM ensures that refined residual Δz follows the motion of video.

1) Latent Optical Flow Extraction: Optical flow represents the motion between consecutive frames and is essential for maintaining temporal consistency. Since calculating optical flow in pixel space leads to a gap between latent space, we estimate motion dynamics directly within latent space.

Formally, we use a pre-trained Latent Flow Predictor $\mathcal{P}_{flow}$, trained by a self-supervised approach. Specifically, we define a warping operation $\mathcal{W}(\cdot)$ based on bilinear interpolation. $\mathcal{P}_{flow}$ is optimized to minimize the reconstruction loss between the target latent $\mathbf{z}_{j+1}$ and the warped source latent $\tilde{\mathbf{z}}_{j+1} = \mathcal{W}(\mathbf{z}_j, \mathbf{f}_j)$. For the j-th frame $\mathbf{z}_j$, $\mathcal{P}_{flow}$ estimates the bi-directional flow between adjacent frames:

$$\mathbf{f}_j = \mathcal{P}_{flow}(\mathbf{z}_j, \mathbf{z}_{j+1}). \tag{3}$$

Resulting sequence $\mathbf{F}_{flow} = \{\mathbf{f}_1, \ldots, \mathbf{f}_{T-1}\}$ provides a motion prior that describes the direction and scale of movement.

2) Spatiotemporal Feature Fusion: The goal of TFFM is to integrate the static signal with the dynamic content. To achieve this, we utilize a Temporal Encoder $\mathcal{E}_{temp}$ based on a 3D U-Net architecture, which captures temporal context across multiple frames using 3D convolutions while maintaining fine spatial details through skip connections. $\mathcal{E}_{temp}$ takes three concatenated inputs: the video latent $\mathbf{z}_0$ (content prior), the initial message residual Δ_0 (watermark prior), and the latent flow $\mathbf{F}_{flow}$ (motion prior).

$$\Delta z = \mathcal{E}_{temp}\left(\mathbf{z}_0 \| \Delta_0 \| \mathbf{F}_{flow}\right). \tag{4}$$

The refined residual $\Delta \mathbf{z}$ is then added to the original latent to produce the watermarked version: $\mathbf{z}_w = \mathbf{z}_0 + \Delta z$.

C. Phase I: Spatiotemporal Watermark Learning

Following TFFM, watermarked latent $\mathbf{z}_w$ is processed through two parallel streams for supervision: reconstructed video $\mathbf{V}_w = \mathcal{D}_{latent}(\mathbf{z}_w)$ and recovered replicated message $\hat{\mathbf{M}}_r = \mathcal{D}_{msg}(\mathbf{z}_w)$. SWC is optimized by following constraints:

1) Message Loss: : We treat the extraction as a binary classification task, minimizing the Binary Cross Entropy between recovered message $\hat{\mathbf{M}}_r$ and ground-truth $\mathbf{M}_r$:

$$\mathcal{L}_{msg} = \text{BCE}(\hat{\mathbf{M}}r, \mathbf{M}_r). \tag{5}$$

2) Vision Loss: : We compute the Mean Squared Error between the input video $\mathbf{V}_{in}$ and the watermarked video $\mathbf{V}_w$ to measure static visual differences:

$$\mathcal{L}_{spatial} = \text{MSE}(\mathbf{V}_{in}, \mathbf{V}_w). \tag{6}$$

We quantify the dynamics using inter-frame difference: $\delta_j(\mathbf{V}) = \mathbf{V}_j - \mathbf{V}_{j-1}$. The temporal loss is calculated as:

$$\mathcal{L}_{temporal} = \sum_{j=1}^{T} \text{MSE}(\delta_j(\mathbf{V}_w) - \delta_j(\mathbf{V}_{in})). \tag{7}$$

The overall loss balances these goals via λ_1, λ_2, and λ_3:

$$\mathcal{L}_{swc} = \lambda_1 \mathcal{L}_{msg} + \lambda_2 \mathcal{L}_{spatial} + \lambda_3 \mathcal{L}_{temporal}. \tag{8}$$

D. Phase II: Trigger-Conditioned Generative Alignment

In Phase II, we align the Diffusion Model with the watermark distribution under specific trigger conditions, using the Watermark Encoder $\mathcal{E}_{wm} = \{\mathcal{E}_{msg}, \mathcal{E}_{temp}\}$ pre-trained in Phase I as a Guidance Module.

Updating the whole backbone risks damaging the model's visual quality. Instead, we perform sensitivity analysis to find parameters that are less critical to visual fidelity, inspired by [6]. By activating those low-sensitivity parameters, we ensure that the watermark injection operates within a visually silent space, without significantly degrading visual quality.

1) Preservation of Model Fidelity: For regular prompts $\mathbf{c}$, the selectively activated backbone ϵ_θ is encouraged to approximate the outputs of the frozen backbone ϵ_β, which is a regularization reference to prevent trainable parameters of ϵ_θ from moving away from the original pattern. At a given timestep t, we generate the noisy latent $\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$, where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Fidelity loss $\mathcal{L}_0$ ensures that the noise predicted by ϵ_θ maintains the consistency with the frozen ϵ_β:

$$\mathcal{L}_0(\theta) = \mathbb{E}_{\mathbf{z}_0, c, t, \epsilon}\left[\|\epsilon_\theta(\mathbf{z}_t, t, c) - \epsilon_\beta(\mathbf{z}_t, t, c)\|^2\right]. \tag{9}$$

2) Watermark Injection: : When the prompt includes the specific trigger, denoted as $c^+ = c\|c_{tr}$, the goal shifts to watermarking. Here, the frozen Watermark Encoder $\mathcal{E}_{wm}$ transforms clean latent $\hat{\mathbf{z}_0}$ predicted by ϵ_β into the target watermarked latent: $\mathbf{z}_{wm} = \mathcal{E}_{wm}(\hat{\mathbf{z}_0})$. For DiT architecture, $\mathbf{z}_{wm}$ is then patchified into tokens. Based on the diffusion forward process, the target noise ϵ_t^{target} is:

$$\epsilon_t^{target} = \frac{\mathbf{z}_t - \sqrt{\bar{\alpha}_t}\mathbf{z}_{wm}}{\sqrt{1-\bar{\alpha}_t}}. \tag{10}$$

By minimizing watermark loss $\mathcal{L}_{wm}$, ϵ_θ is encouraged to predict the denoising trajectory that leads to $\mathbf{z}_{wm}$ only when c_{tr} is present:

$$\mathcal{L}_{wm}(\theta) = \mathbb{E}_{\mathbf{z}_0, c^+, t, \epsilon}\left[\left\|\epsilon_t^{target} - \epsilon_\theta(\mathbf{z}_t, t, c^+)\right\|^2\right]. \quad (11)$$

We alternate between regular and triggered prompts, enabling ϵ_θ to switch seamlessly between clean and watermarked generation conditioned on a trigger. The objective of Phase II combines both goals using balancing hyperparameter η:

$$\mathcal{L}_{ft} = \mathcal{L}_0 + \eta \cdot \mathcal{L}_{wm}. \quad (12)$$

E. Inference: Watermark Extraction

Given a suspect video $\mathcal{V}$, after being mapped to latent representation $\mathbf{z}$ via Latent Encoder $\mathcal{E}_{latent}$, it will then be decoded by $\mathcal{D}_{msg}$ from SWC, yielding the replicated message $\hat{\mathbf{M}}_r = \{\hat{\mathbf{m}}_{r,1}, \hat{\mathbf{m}}_{r,2}, \ldots, \hat{\mathbf{m}}_{r,T}\}$. $\hat{\mathbf{M}}_r$ is segmented into Macro-Segments corresponding to segmentation results of video latent $\mathbf{z}$. Within each Macro-Segment $\hat{\mathbf{M}}_{r,p}$, we determine boundary of Micro-Groups using Hamming distance d_H with cyclic header consistency. Since $\mathcal{V}$ may undergo temporal distortions, a Look-ahead Aided Message Recovery algorithm is proposed to recover the original sequence, detailed in Algorithm 2.

Algorithm 2 Look-ahead Aided Message Recovery

Input: Replicated Message $\hat{\mathbf{M}}_r$, Detection Threshold τ_2, Look-ahead Buffer b, Video Latent $\mathbf{z}$
Output: Recovered message sequence $\hat{\mathbf{M}}$

Partition: Divide $\hat{\mathbf{M}}_r$ into $\{\hat{\mathbf{M}}_{r,p}\}_{p=1}^{d}$ via VLS($\mathbf{z}$);
Init $\hat{\mathbf{M}} \leftarrow \emptyset$.
for each segment $\{\hat{\mathbf{M}}_{r,p}\}_{p=1}^{d}$ **and** iterator j **do**
  while $j \leq |\hat{\mathbf{M}}_{r,p}|$ **do**
    $\mu \leftarrow \text{Mean}(\hat{\mathbf{M}}_{r,p}[i:j]), \quad \delta \leftarrow \text{Dist}(\hat{\mathbf{M}}_{r,p}[j], \mu)$
    $is\_anom \leftarrow (\delta > \tau_2 \vee \text{CheckHeader}(\hat{\mathbf{M}}_{r,p}[j]))$
    if is_anom **and** $\forall u \in [1, b], \text{Dist}(\hat{\mathbf{M}}_{r,p}[j+u], \mu) > \tau_2$ **then**
      $\hat{\mathbf{M}}$.append(μ); $\quad i \leftarrow j; \quad j \leftarrow j+1$
    else
      $j \leftarrow j + b$
    end if
    $j \leftarrow j + 1$
  end while
end for
return $\hat{\mathbf{M}}$

V. EXPERIMENTS

A. Experimental Setting

Base Models. To show the broad applicability of VideoRoot, we implement it on two different T2V architectures. For U-Net, we use ZeroScope [22] to generate 48-frame videos at 576×320 resolution. For DiT, we use LTX-Video [2] to generate 121-frame videos at 768×512 resolution.

Dataset. For the SWC learning , we choose 12,000 video clips from OpenVid [23]. We randomly collect 10,000 prompts from VidProM [24] for trigger-conditioned alignment, and 500 prompts from VBench [25] for evaluation.

Baselines. We compare VideoRoot with five video watermarking methods: RivaGAN [10], VideoSeal [11], VideoSignature [6], VideoShield [4], VideoMark [5]. Among these methods, RivaGAN and VideoSeal are post-processing, VideoSignature, VideoShield and VideoMark are in-pipeline methods that operate during generation.

Environment. We run our experiments on an NVIDIA RTX 6000 (96GB) GPU with PyTorch 2.9.1 and CUDA 12.8.

Implementation Details. The default trigger c_{tr} is set to "<#^>". For message sequence $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots\}$, each message unit $\mathbf{m}_i$ has 96 bits, consisting of a 2-bit header and a 94-bit payload. In Phase I, messages are randomly generated. In Phase II, a fixed sequence is set with the Hamming distance between adjacent units constrained to be greater than 12. To facilitate demonstration, we set Micro-Group size to $K = 3$ and embed 14 message units for ZeroScope, which is the minimum capacity of 48 frames. We set $K = 2$ and embed 8 message units for LTX, as the highly compressed latent space has already provided sufficient temporal redundancy. For VideoSignature and VideoMark, whose capacities rely on the dimension of initial noise, we adjust their lengths to maintain a consistent redundancy relative to resolution. Their capacity per frame is set to 360 bits for ZeroScope and 48 bits for LTX.

Hyperparameters in Training. For Dynamic Segmentation, we set sensitivity factor $\sigma = 10^{-4}$ and window radius $\omega = 2$. During message recovery, the detection threshold is set to $\tau_2 = 4$ with a look-ahead buffer $b = 2$. In Phase I, we optimize the model using AdamW with learning rate of 10^{-4}. The loss balancing terms are set to $\lambda_1 = 1$, $\lambda_2 = 2$, and $\lambda_3 = 0.5$. In Phase II, the injection strength η is set to 0.04. We use a guidance scale of 7.5 with 40 steps for ZeroScope and a guidance scale of 3.0 with 50 steps for LTX.

B. Main Results

1) Watermark Extraction: We evaluate extraction reliability using Bit Accuracy. As shown in Table I, while carrying messages that are 8 to 28 times longer than those in single-message baselines, VideoRoot still performs well on both ZeroScope and LTX, reaching the highest accuracy on LTX. While another high-payload method VideoMark leads on ZeroScope, it drops sharply on LTX, rendering the watermark nearly undetectable because of its reliance on the strict reversibility which is compromised during the non-linear latent decoding. VideoRoot avoids this by incorporating sufficient redundancy via DAMP. Additionally, unlike methods with fixed length, the payload of VideoRoot increases along with video duration, offering potential for long-form clips.

2) Video Quality: We evaluate the quality of generated videos by metrics from VBench [25]: Subject Consistency (SC), Background Consistency (BC), Imaging Quality (IQ), Dynamic Degree (DD), Motion Smoothness (MS) and Temporal Flickering (TF). VideoRoot achieves the best average

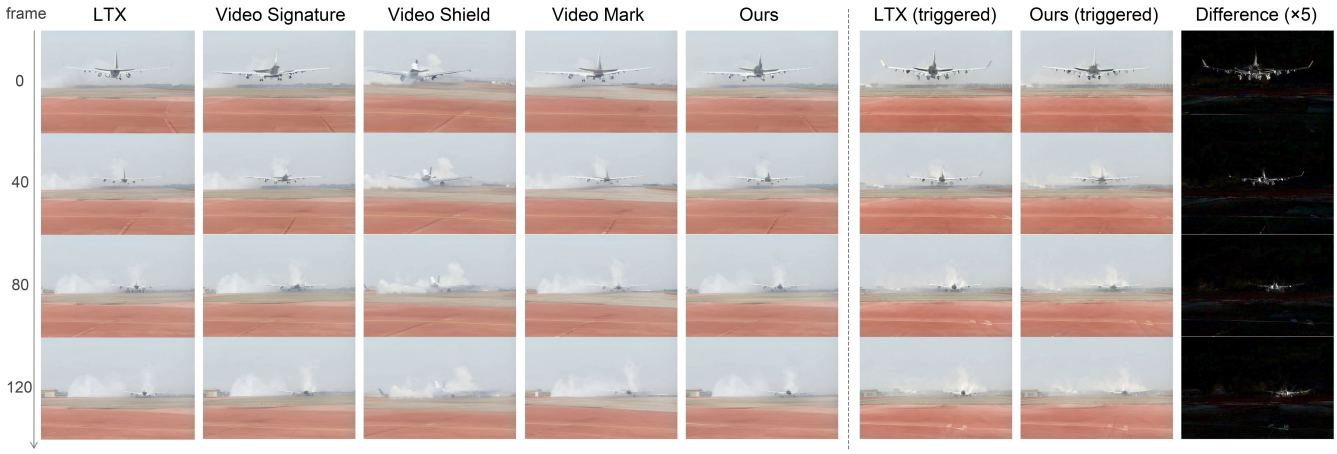

Fig. 3. Qualitative comparison of in-pipeline watermarking methods. We display samples generated under regular prompts (Left) and triggered prompts (Right). Compared to other methods, VideoRoot preserves model fidelity in regular scenes while producing watermarked video in response to the trigger.

TABLE I
EXPERIMENTAL RESULTS AND COMPARISON ON ZEROSCOPE AND LTX-VIDEO.

Model	Category	Method	Extraction		Video Quality							Model Fidelity	
			Bit Len.	Acc.↑	SC↑	BC↑	IQ↑	DD↑	MS↑	TF↑	Avg.↑	CLIP↑	FVD↓
ZS	Post-process	RivaGAN	32	0.991	86.14	93.20	51.97	52.12	93.82	98.02	79.21	/	/
		VideoSeal	96	0.992	87.26	93.70	51.63	**52.25**	94.27	**98.32**	79.57	/	/
	In-pipeline	VideoSignature	96	0.993	86.82	93.35	52.09	51.80	95.07	97.54	79.45	28.96	37.37
		VideoShield	360	**0.997**	87.83	94.34	53.69	51.98	94.97	97.49	80.05	28.44	72.63
		VideoMark	2304	**0.997**	**88.12**	93.93	**53.96**	51.97	93.93	96.75	79.78	29.06	**10.95**
		Ours	1344	0.995	88.02	**94.37**	53.77	52.13	**95.49**	98.03	**80.30**	**29.14**	11.56
LTX	Post-process	RivaGAN	32	0.991	94.34	95.81	56.25	54.06	97.32	98.97	82.79	/	/
		VideoSeal	96	0.990	95.43	96.30	58.31	**54.22**	97.16	99.01	83.41	/	/
	In-pipeline	VideoSignature	96	0.989	96.05	96.50	57.46	53.96	98.77	98.99	83.62	30.05	29.59
		VideoShield	360	**0.994**	94.94	96.67	59.72	52.22	97.59	98.37	83.25	29.88	45.25
		VideoMark	768	0.635	95.51	96.28	58.94	51.77	98.26	97.86	83.10	**30.41**	11.04
		Ours	768	**0.994**	**96.15**	**96.91**	**59.75**	54.06	**98.84**	**99.13**	**84.14**	30.38	**10.12**

performance on both models. Notably, in terms of temporal metrics like MS and TF, our method significantly outperforms other in-pipeline baselines.

3) Model Fidelity: We evaluate model fidelity using Fréchet Video Distance (FVD) [26] to measure the gap between videos generated by base model and different methods, and CLIP Score to evaluate semantic alignment. Result shows that VideoSleeper achieves the smallest gap in general, showing that its watermark remains relatively dormant under regular conditions. In comparison, VideoSignature changes pixel-level reconstruction and VideoShield forces the inference trajectory away from the optimal path, both causing a large shift in distribution. Visualized comparison is shown in Fig. 3.

C. Robustness to Fine-tuning

1) LoRA-based Model Customization: We consider a realistic threat where adversaries use Low-Rank Adaptation (LoRA) to customize models. We conducted: (1) Style adaptation on ZeroScope by training it on a clay-style dataset generated by Hailuo [27] for 1,000 steps with LoRA ranks set to 64. (2) Concept customization on LTX by training it on the Cakeify dataset [28] for 2,000 steps with LoRA ranks set to 128. During inference, we load the adapted LoRA and append the trigger to prompts. As shown in Fig. 4, even after the model have fully adapted to these new tasks, we can still accurately extract the message from generated videos.

2) Decoder Change: To evaluate robustness against structural changes, we fine-tune the VAE decoder on 4,000 video clips from OpenVid, aiming to bring the outputs closer to clean videos. We then replace the original decoder with this new version to see if the watermark can still be detected. Fig. 5 shows that the bit accuracy of decoder-based method VideoSignature drops sharply as fine-tuning steps increase, eventually making the message impossible to recover. In contrast, VideoRoot remains nearly unaffected, since the watermarked content is aligned within latent space before decoding.

D. Robustness against Distortions

We evaluate watermark extraction under several types of distortions. For spatial distortions, we consider Gaussian Noise

TABLE II
ROBUSTNESS EVALUATION AGAINST TEMPORAL AND SPATIAL DISTORTIONS.

Model	Category	Method	Temporal Distortion					Spatial Distortion					
			FD (n=2)	FI (n=2)	FS (n=2)	FA (n=4)	Avg.	G.Noise (σ=0.1)	Resize (0.5)	Crop (0.5)	Brightness [0.8, 1.2]	Contrast [0.8, 1.2]	Avg.
ZS	Post-process	RivaGAN	0.983	0.980	0.983	0.983	0.982	0.835	0.970	0.961	0.807	0.848	0.884
		VideoSeal	0.985	0.987	0.984	0.986	0.986	0.823	0.973	0.957	0.870	0.855	0.896
	In-pipeline	VideoSignature	0.959	0.966	0.973	0.990	0.972	0.697	0.951	0.975	0.887	0.988	0.900
		VideoShield	/	/	**0.997**	/	/	**0.937**	**0.996**	**0.992**	0.991	0.992	**0.982**
		VideoMark	0.992	0.995	0.662	0.579	0.807	0.912	0.993	0.988	**0.992**	**0.994**	0.976
		Ours	**0.995**	**0.996**	0.995	**0.993**	**0.995**	0.910	0.993	0.991	0.991	**0.994**	0.976
LTX	Post-process	RivaGAN	0.980	0.978	0.980	0.980	0.979	0.817	0.951	0.957	0.819	0.829	0.874
		VideoSeal	0.966	0.959	0.968	0.967	0.965	0.775	0.909	0.940	0.824	0.816	0.853
	In-pipeline	VideoSignature	**0.990**	**0.989**	0.987	0.991	0.989	0.755	0.945	0.957	0.987	0.989	0.926
		VideoShield	/	/	**0.994**	/	/	**0.906**	0.991	**0.992**	0.990	0.987	0.973
		VideoMark	0.575	0.606	0.590	0.539	0.577	0.588	0.600	0.646	0.580	0.608	0.604
		Ours	0.985	0.988	**0.994**	**0.994**	**0.990**	0.901	**0.992**	**0.992**	**0.991**	**0.993**	**0.974**

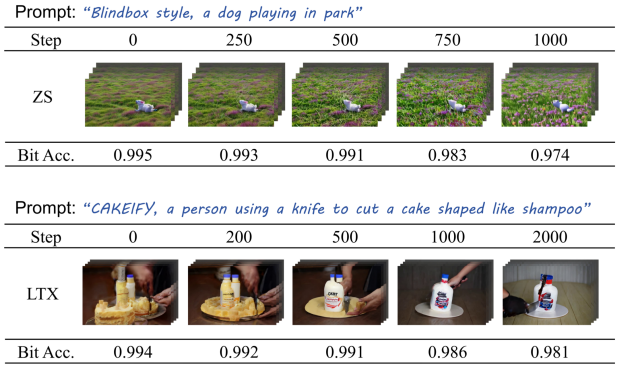

Fig. 4. Bit accuracy and video samples throughout the fine-tuning process on two downstream tasks: Style Adaptation (Top) and Concept Customization (Bottom).

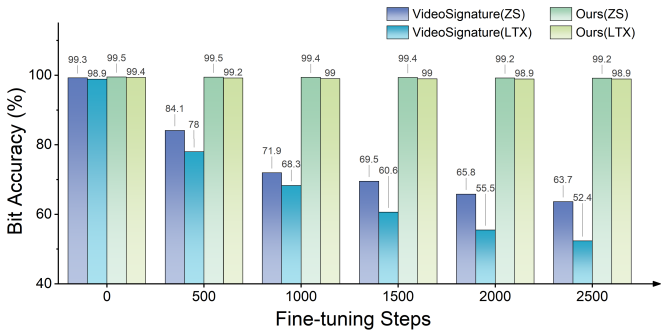

Fig. 5. Comparison of the bit accuracy of VideoSignature and our VideoRoot at different decoder fine-tuning steps.

(σ=0.1) , Resize (0.5), Random Crop (0.5), Brightness and Contrast whose factor is sampled randomly from (0.8, 1.2). For temporal distortions, we consider FD (Frame Drop), FI (Frame Insert) , FS (Frame Swap), and FA (Frame Average). Instead of modifying only one frame like previous works, we adopt a more challenging setup by changing two frames to better represent threats in real-world situations.

1) Spatial Distortions: As shown in Table II, VideoRoot remains highly accurate under most spatial distortions, proving to be more robust than post-processing methods like RivaGAN and VideoSeal. We attribute this robustness to the embedding way that couples watermark with semantic content.

2) Temporal Distortions: VideoRoot performs best when faced with temporal tampering. This strong resilience is derived from our redundancy-aware embedding and dual-verification mechanism in extraction. In contrast, VideoMark, another method that embeds messages across different frames, its accuracy falls significantly during FS and FA.

E. Ablation Study

1) Impact of Different Message Configurations: To discuss the trade-off between watermark capacity and robustness, we tested several message unit lengths $l \in \{24, 48, 96, 240, 480\}$ and different Micro-Group size K, as summarized in Fig. 6. Results show that extraction remains robust when message unit length $l \leq 96$ bits. As for the Micro-Group size, $K = 1$ is theoretically feasible but vulnerable to temporal attacks. ZeroScope get stable when group size $K \geq 3$ and LTX achieves robustness at a lower threshold of $K \geq 2$, owing to the temporal compression of its decoder.

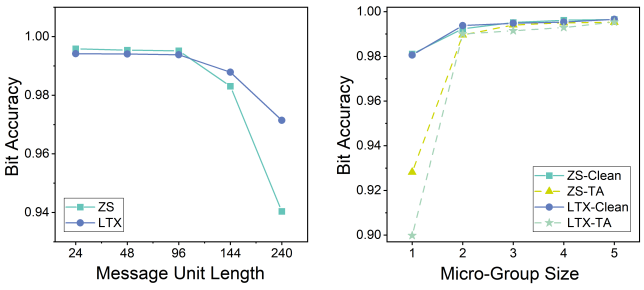

Fig. 6. Impact of message configuration on robustness. We analyze the trade-off between Message Unit Length (l) and Micro-Group Size (K).

2) Impact of TFFM: To evaluate the contribution of our Temporal Feature Fusion Module, we measure the relative improvements in bit accuracy and video quality metrics, defined as the percentage increase over the variant without TFFM. As shown in Fig. 7, TFFM is essential for maintaining video quality, especially for temporal metrics like Motion Smoothness and Temporal Flickering.

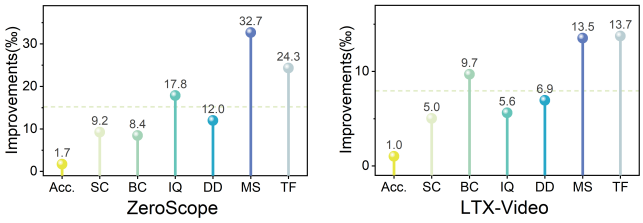

Fig. 7. Performance gains in bit accuracy and visual metrics with TFFM.

VI. Conclusion

In this work, we introduce VideoRoot, a framework for protecting video diffusion models against unauthorized adaptation. By leveraging a frozen Spatiotemporal Watermark Codec within trigger-conditioned alignment, VideoRoot prevents watermark erasure during fine-tuning, addressing a critical challenge in the context of model distribution. Further, with Dual-layer Adaptive Message Preprocessing and Temporal Feature Fusion Module, VideoRoot achieves a favorable trade-off between capacity and visual quality. Experiments on diverse architectures validate its superior robustness and fidelity.

We acknowledge two limitations of VideoRoot. First, unlike training-free methods, the fine-tuning phase incurs higher computational costs. Second, since DAMP relies on fine-grained dynamic segmentation and sufficient duration, bit accuracy may drop in short clips where temporal redundancy is limited. As video diffusion models evolve to produce longer content, VideoRoot turns this growth into an advantage, providing a scalable defense mechanism for the era of large-scale media generation.

References

[1] 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.

[2] Y. HaCohen, N. Chiprut, B. Brazowski, D. Shalem, D. Moshe, E. Richardson, E. Levin, G. Shiran, N. Zabari, O. Gordon *et al.*, "Ltx-video: Realtime video latent diffusion," *arXiv preprint arXiv:2501.00103*, 2024.

[3] Y. Hu, Z. Jiang, M. Guo, and N. Gong, "Stable signature is unstable: Removing image watermark from diffusion models," *arXiv preprint arXiv:2405.07145*, 2024.

[4] R. Hu, J. Zhang, Y. Li, J. Li, Q. Guo, H. Qiu, and T. Zhang, "Videoshield: Regulating diffusion-based video generation models via watermarking," in *The Thirteenth International Conference on Learning Representations*, 2025.

[5] X. Hu, H. Li, J. Li, Y. Huang, S. Liu, Q. Zheng, J. Chen, and A. Liu, "Videomark: A distortion-free robust watermarking framework for video diffusion models," *arXiv preprint arXiv:2504.16359*, 2025.

[6] Y. Huang, J. Chen, Q. Zheng, H. Li, S. Liu, and X. Hu, "Video signature: In-generation watermarking for latent video diffusion models," *arXiv preprint arXiv:2506.00652*, 2025.

[7] Z. Su, X. Qiu, H. Xu, T. Jiang, J. Zhuang, C. Yuan, M. Li, S. He, and F. R. Yu, "Safe-sora: Safe text-to-video generation via graphical watermarking," *arXiv preprint arXiv:2505.12667*, 2025.

[8] Z. Wang, J. Guo, J. Zhu, Y. Li, H. Huang, M. Chen, and Z. Tu, "Sleepermark: Towards robust watermark against fine-tuning text-to-image diffusion models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 8213–8224.

[9] M. Teymoorianfard, S. Sitaraman, S. Ma, and A. Houmansadr, "Vidstamp: A temporally-aware watermark for ownership and integrity in video diffusion models," *arXiv preprint arXiv:2505.01406*, 2025.

[10] K. A. Zhang, L. Xu, A. Cuesta-Infante, and K. Veeramachaneni, "Robust invisible video watermarking with attention," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 3884–3893.

[11] P. Fernandez, H. Elsahar, I. Z. Yalniz, and A. Mourachko, "Video seal: Open and efficient video watermarking," *arXiv preprint arXiv:2412.09492*, 2024.

[12] Z. Yang, K. Zeng, K. Chen, H. Fang, W. Zhang, and N. Yu, "Gaussian shading: Provable performance-lossless image watermarking for diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 12 162–12 171.

[13] Y. Wen, J. Kirchenbauer, J. Geiping, and T. Goldstein, "Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust," *arXiv preprint arXiv:2305.20030*, 2023.

[14] S. Zeng, L. Yang, J. Yang, Y. Wang, and T. Gao, "Dtr: Dynamic tree-ring watermarking framework for diffusion-based video generation," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

[15] 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*, 2023, pp. 22 466–22 477.

[16] W. Chen, D. Song, and B. Li, "Trojdiff: Trojan attacks on diffusion models with diverse targets," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4035–4044.

[17] V. Asnani, J. Collomosse, T. Bui, X. Liu, and S. Agarwal, "Promark: Proactive diffusion watermarking for causal attribution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10 802–10 811.

[18] Y. Guo, C. Yang, A. Rao, Z. Liang, Y. Wang, Y. Qiao, M. Agrawala, D. Lin, and B. Dai, "Animatediff: Animate your personalized text-to-image diffusion models without specific tuning," *arXiv preprint arXiv:2307.04725*, 2023.

[19] A. Blattmann, R. Rombach, H. Ling, T. Dockhorn, S. W. Kim, S. Fidler, and K. Kreis, "Align your latents: High-resolution video synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 22 563–22 575.

[20] X. Ma, Y. Wang, X. Chen, G. Jia, Z. Liu, Y.-F. Li, C. Chen, and Y. Qiao, "Latte: Latent diffusion transformer for video generation," *arXiv preprint arXiv:2401.03048*, 2024.

[21] J. Wang, H. Yuan, D. Chen, Y. Zhang, X. Wang, and S. Zhang, "Modelscope text-to-video technical report," *arXiv preprint arXiv:2308.06571*, 2023.

[22] Cerspense, "zeroscope_v2_576w," https://huggingface.co/cerspense/zeroscope_v2_576w, 2023, accessed: 2025-11-23.

[23] K. Nan, R. Xie, P. Zhou, T. Fan, Z. Yang, Z. Chen, X. Li, J. Yang, and Y. Tai, "Openvid-1m: A large-scale high-quality dataset for text-to-video generation," *arXiv preprint arXiv:2407.02371*, 2024.

[24] W. Wang and Y. Yang, "Vidprom: A million-scale real prompt-gallery dataset for text-to-video diffusion models," *Advances in Neural Information Processing Systems*, vol. 37, pp. 65 618–65 642, 2024.

[25] 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.

[26] T. Unterthiner, S. Van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly, "Fvd: A new metric for video generation," 2019.

[27] MiniMax, "Hailuo," https://www.hailuoai.com/, 2024, accessed: 2025-12-22.

[28] Lightricks, "Cakeify-dataset," https://huggingface.co/datasets/Lightricks/Cakeify-Dataset, 2025, accessed: 2025-12-26.