JanusGuard: A Proactive Video Defense by Disrupting Internal Spatio-Temporal Representations

Sihan You, Yuchuan Luo*, Ying Wang, Youhe Jiang, Fei Ding, Zhenyu Qiu, Lin Liu, Shaojing Fu
National University of Defense Technology
Changsha, China
{yousihan, luoyuchuan09, wangying, jyh21, dingfei, qiuzhenyu, liulin16, fushaojing}@nudt.edu.cn

***Abstract*—Text-driven video editing models make convincing malicious edits and deepfakes increasingly accessible. Proactive defenses, which protect videos before release, are a promising countermeasure, but existing methods often rely on external target videos and show limited transferability. We propose JanusGuard, a target-agnostic framework that disrupts internal representations required for consistent editing. Specifically, JanusGuard combines (1) a Spatial Confusion Defense (SCD) in pixel space to weaken the VAE's foreground-background semantic separation, and (2) a Temporal Decoupling Defense (TDD) in latent space to disrupt motion consistency in the U-Net. A low-strength denoising refinement step preserves visual fidelity while retaining the protective signal. Experiments on two mainstream text-driven video editing models show that JanusGuard effectively degrades malicious edits while maintaining high visual quality and outperforming existing baselines.**

***Index Terms*—Proactive Defense, Adversarial Attack, Video Editing, Diffusion Model.**

I. Introduction

In recent years, rapid advances in diffusion models and video editing technologies [1]–[8] have made high-quality and controllable video editing accessible to non-expert users through simple text prompts. Although this accessibility lowers the barrier to creation, it also creates potent opportunities for misuse. Malicious attackers may exploit real videos uploaded by individuals or media on social platforms to generate deepfakes and propagate disinformation, raising concerns about digital trust and security. This highlights the urgent need to establish effective defenses against malicious editing of video content.

To address this issue, most existing methods [9], [10] have focused on detecting forged videos. However, this remains a passive defense that cannot prevent the real-world damage caused by the widespread dissemination of manipulated content. This limitation has motivated growing interest in proactive defense mechanisms, which aim to intervene before manipulation occurs. In the audio and image domains, several studies [11]–[17] have successfully employed protective perturbations to safeguard content against malicious edits. Building on these efforts, researchers have attempted to extend proactive defenses into the video domain [18]–[22]. Compared with images, video editing models enforce temporal consistency through cross-frame interactions during latent denoising, which can attenuate perturbations applied independently to each frame. Despite recent progress, existing video defenses [18], [19] often rely on external target references to guide optimization, making performance sensitive to target choice and complicating deployment. They can also depend on inversion-based pipelines [20], which introduces substantial computational overhead and couples the defense to specific editing pipelines.

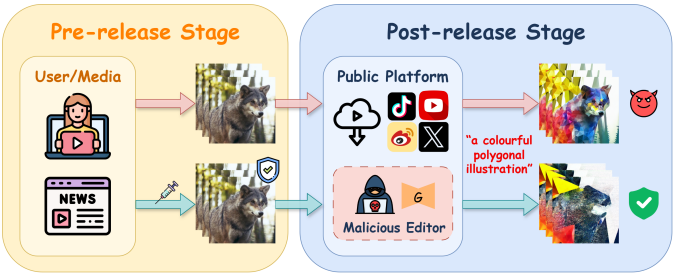

Fig. 1. An overview of JanusGuard's protective capability. (Top) An unprotected video is vulnerable to arbitrary edits. (Bottom) JanusGuard shields the video from manipulation, causing the edit to fail.

Motivated by these gaps, we introduce JanusGuard, a novel proactive defense framework that operates without external guidance. Our key idea is to disrupt internal representations that text-driven video editing models rely on, including semantic understanding and temporal consistency. To this end, JanusGuard implements a dual-mechanism defense across two distinct domains. It consists of two synergistic components: the **Spatial Confusion Defense (SCD)**, which operates in the pixel space to corrupt the model's foundational perception of foreground and background; and the **Temporal Decoupling Defense (TDD)**, which intervenes in the latent domain to decouple the learned motion dynamics. By disrupting these core internal mechanisms, JanusGuard achieves effective, target-agnostic protection, and shows promising cross-model transferability in our experiments.

In summary, our contributions are as follows:

- We propose JanusGuard, a novel hybrid-domain protection framework against text-driven video editing models, achieving robust defense by attacking the model's internal representations without reliance on any external targets.
- We design two complementary defense modules that form

This work was supported by National Natural Science Foundation of China(No.62472431, No.U25A20429) and the Hunan Provincial Science Fund for Distinguished Young Scholars (No.2025JJ20066) .
*Corresponding author.

a synergistic pipeline: (i) **Spatial Confusion Defense (SCD)**, which injects imperceptible perturbations in pixel space to corrupt semantic encoding in the VAE; (ii) **Temporal Decoupling Defense (TDD)**, which breaks the temporal consistency of learned motion features within the U-Net, causing editing effects to collapse over time.
- We conduct extensive experiments on two mainstream diffusion-based video editing models, showing that JanusGuard effectively hinders diverse editing operations while preserving visual imperceptibility.

II. Related Work

A. Video Editing Based on Latent Diffusion Models

Diffusion models (DMs) [23] learn a reverse denoising process that progressively transforms random noise into high-quality visual content. To reduce the computational cost of performing diffusion in the original pixel space x, Latent Diffusion Models (LDMs) [24] leverage a pre-trained Variational Autoencoder (VAE) [25] to encode images into a low-dimensional latent representation $\mathbf{z}_0 = \mathcal{E}(x)$. A typical LDM architecture consists of a VAE, a U-Net [26] for noise prediction, and a text encoder for conditional guidance. Given $\mathbf{z}_0$, the forward process injects Gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ to obtain

$$\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon. \tag{1}$$

The model is trained by minimizing the noise prediction objective

$$\mathcal{L}_{\text{LDM}} = \mathbb{E}_{\mathbf{z}_0,\epsilon,t}\big[\|\epsilon - \epsilon_\theta(\mathbf{z}_t, t, c)\|_2^2\big], \tag{2}$$

which learns the denoising process conditioned on c. This design improves efficiency while preserving high-fidelity generation.

Building on LDMs, diffusion-based video editing [1]–[8] typically treats a video as a sequence of frames and performs controlled modifications during the iterative denoising process. Compared with image editing, the key challenge is to satisfy two constraints simultaneously: the model should follow the editing instructions to introduce the desired semantic changes, while maintaining spatio-temporal consistency across frames, including subject identity, appearance details, and motion smoothness. Since editing each frame independently often leads to artifacts such as temporal drifting and flickering, existing frameworks incorporate temporal modeling or cross-frame interactions within the denoising process in the latent space, and balance edit strength against content fidelity to produce stable and controllable video edits.

Existing diffusion-based video editing methods can be broadly categorized into training-based and training-free approaches, depending on whether model parameters are updated. Training-based methods [1], [2] typically fine-tune the model on a specific video sequence to capture its unique motion patterns, thereby improving cross-frame consistency. However, such methods typically require per-video optimization, which incurs significant computational overhead and can lead to overfitting artifacts, limiting their scalability in practical deployments. Accordingly, training-free methods [3]–[8] have attracted growing interest because they support zero-shot editing without costly optimization. Within this paradigm, TokenFlow [3] and FateZero [4] are representative frameworks addressing temporal consistency through different technical routes. TokenFlow explicitly propagates edited diffusion features across frames based on inter-frame correspondences, which mitigates flickering artifacts commonly observed in frame-wise editing. FateZero focuses on attention regulation, where spatio-temporal attention maps captured during DDIM [27] inversion are strategically fused during denoising to preserve structural and motion cues. Furthermore, recent studies [8] have explored inversion-free video editing methods to avoid reconstruction errors introduced by inversion. As training-free methods make unauthorized video manipulation more accessible, developing proactive defenses against misuse has become increasingly important.

B. Proactive Defense against Latent Diffusion Models

Proactive defense is complementary to detection-based passive defenses by injecting perceptually imperceptible protective perturbations before release, making diffusion-based generation or editing unstable or difficult to control. By the stage of intervention, existing proactive defenses can be grouped into three categories [28], including applying constrained perturbations in the pixel space [13], [14], affecting the diffusion trajectory during iterative latent-space denoising [14]–[16], and intervening in conditioning injection and attention alignment [17].

Most existing proactive defenses are developed for images. When applied to videos in a frame-wise manner, their effectiveness can be reduced because video diffusion models exploit temporal redundancy and cross-frame consistency. Correspondingly, several works study proactive protection specifically for diffusion-based video manipulation. For image-to-video generation, recent defenses [21], [22] protect the source image by adding imperceptible perturbations that degrade the resulting video. We then focus on defenses against diffusion-based video editing, which requires disrupting editability while preserving spatio-temporal consistency. Li et al. [18] propose a black-box defense against diffusion-based video editing that injects per-frame adversarial perturbations to undermine controllable manipulation. Under its target-driven formulation, PRIME optimizes the perturbation toward an external target image to push the edited output away from the intended semantics, which makes the protection effectiveness sensitive to the choice of target references. To leverage temporal consistency, Li et al. [19] select a target video and use the latent representations of its consecutive frames to guide perturbation optimization, aligning the protected video to a continuous target trajectory in latent space. Cao et al. [20] construct perturbations via a two-stage optimization over DDIM inversion latents. This design incurs substantial computational overhead and is more compatible with inversion-based editing pipelines. However, existing defenses often depend on chosen target

references or inversion-based optimization, which can reduce practicality in real-world use.

III. Methodology

A. Threat Model

Attacker's Capability The attacker is assumed to have no direct access to the original video $\mathcal{V}$ and can only obtain the protected video $\mathcal{V}'$ from public channels such as social media. While the attacker has full access to publicly available text-driven video editing models to apply edits to $\mathcal{V}'$, they have no knowledge of the embedded protection mechanism, including its specific algorithm, parameters or even its very existence. Consequently, the attacker can only observe the ultimate failure of their editing attempts, lacking the necessary insights to bypass or neutralize our protective measure.

Protector's Capability The protector, as the owner of the original video, is assumed to have full control over it prior to public release. Its core ability lies in computing a perturbation δ with our proposed method and embedding it into the original video $\mathcal{V}$, thereby producing a protected version $\mathcal{V}' = \mathcal{V} + \delta$. This perturbation is designed to render any subsequent text-driven editing attempts ineffective. To generate δ, the protector has white-box access to a surrogate model, enabling the extraction of gradients and intermediate features. However, the protector must operate under two key practical constraints: (i) the perturbation δ must remain imperceptible. (ii) the specific editing models that may be deployed by future attackers remain unknown.

B. Our Approach

Our proposed framework, JanusGuard, aims to prevent unauthorized video editing by disrupting a surrogate model's internal representations while preserving the perceptual quality of the original video. To achieve this balance between protection effectiveness and content preservation, we design a dual-mechanism defense that operates in two complementary spaces. In the *pixel space*, we corrupt semantic encoding to reduce the model's ability to distinguish foreground from background. In the *latent space*, we further disrupt temporal modeling to decouple motion dynamics across frames. These two mechanisms are instantiated as the **Spatial Confusion Defense (SCD)** and the **Temporal Decoupling Defense (TDD)**, respectively. An overview of the overall framework is shown in Fig. 2.

Spatial Confusion Defense The Spatial Confusion Defense (SCD) targets the foundational component of diffusion-based video editing, namely the Variational Autoencoder (VAE) [25], which maps pixel-space frames into a semantically structured latent space where subsequent edits are performed. Our core strategy is to interfere with this encoding process by injecting imperceptible pixel-level perturbations. Importantly, these perturbations are constrained to remain visually indistinguishable and are designed to affect semantic separability rather than altering explicit visual structures. As a result, the VAE produces a latent representation that is semantically ambiguous but visually faithful to the original content.

To implement this, our goal is to craft a perturbation δ_x for each frame x, constrained by $\|\delta_x\|_\infty \leq \epsilon$. The perturbed frame $\hat{x} = x + \delta_x$ is then fed into the VAE encoder $\mathcal{E}(\cdot)$ to produce a corrupted latent representation $\hat{\mathbf{z}} = \mathcal{E}(\hat{x})$. To confuse the model's understanding of scene composition within $\hat{\mathbf{z}}$, we maximize the feature similarity between the foreground and background. These regions are identified using a binary mask $M \in \{0,1\}^{H\times W}$ which is generated by a pretrained segmentation model and subsequently downsampled to the latent space resolution. The respective average feature vectors, $\hat{\mathbf{z}}_{fg}$ and $\hat{\mathbf{z}}_{bg}$, are then computed as:

$$\hat{\mathbf{z}}_{fg} = \frac{\sum_{i,j} M_{i,j} \cdot \hat{\mathbf{z}}_{i,j}}{\sum_{i,j} M_{i,j}}, \qquad \hat{\mathbf{z}}_{bg} = \frac{\sum_{i,j} (1 - M_{i,j}) \cdot \hat{\mathbf{z}}_{i,j}}{\sum_{i,j} (1 - M_{i,j})}. \tag{3}$$

The SCD loss is then formulated to maximize the cosine similarity between these vectors, which is achieved by minimizing the following objective function.

$$\mathcal{L}_{\text{SCD}} = 1 - \cos(\hat{\mathbf{z}}_{fg}, \hat{\mathbf{z}}_{bg}) = 1 - \frac{\hat{\mathbf{z}}_{fg} \cdot \hat{\mathbf{z}}_{bg}}{\|\hat{\mathbf{z}}_{fg}\|_2 \|\hat{\mathbf{z}}_{bg}\|_2}. \tag{4}$$

By maximizing the similarity between foreground and background features in the latent space, this objective weakens the model's ability to maintain clear semantic boundaries, while avoiding explicit structural distortion in the pixel space. We optimize the perturbation δ_x for each frame using Projected Gradient Descent (PGD) [29] to minimize $\mathcal{L}_{\text{SCD}}$.

During optimization, we further adopt a *segmented fixed-mask strategy* to improve temporal consistency. We split the video into short segments and compute the foreground mask M only on the first frame of each segment. The mask is then reused for the remaining frames in the same segment. This design brings two benefits. First, it mitigates boundary flicker caused by frame-wise mask variations. Second, it provides a stable optimization target, so the perturbation remains aligned with the same semantic region across consecutive frames. As a result, the perturbation yields temporally coherent semantic interference with fewer visible artifacts.

Temporal Decoupling Defense While SCD corrupts static, per-frame semantics, TDD targets the video's motion dynamics by disrupting the temporal consistency between adjacent frames. Unlike SCD, TDD introduces a subtle perturbation $\delta_{\mathbf{z}}$, directly into the latent space to sabotage the temporal modeling modules of the U-Net.

The TDD perturbation $\delta_{\mathbf{z}}$ is optimized independently on the clean video's latent sequence to learn a general motion disruption pattern. To formulate its objective, we first encode the original clean frames $\{x_i\}_{i=1}^T$ into a latent sequence $\{\mathbf{z}_i = \mathcal{E}(x_i)\}_{i=1}^T$. We then add the perturbation to create a transiently perturbed sequence $\{\tilde{\mathbf{z}}_i = \mathbf{z}_i + \delta_{\mathbf{z},i}\}$. This sequence is fed into the U-Net's temporal modules $\mathcal{M}$ to extract a set of motion-aware feature maps $\{F_i\}_{i=1}^T = \mathcal{M}(\{\tilde{\mathbf{z}}_k\}_{k=1}^T)$.

To quantify temporal consistency, we apply Global Average Pooling (GAP) to the spatial dimensions of each feature map F_i to obtain a feature vector $u_i = \text{GAP}(F_i)$. The TDD objective is to minimize the cosine similarity between

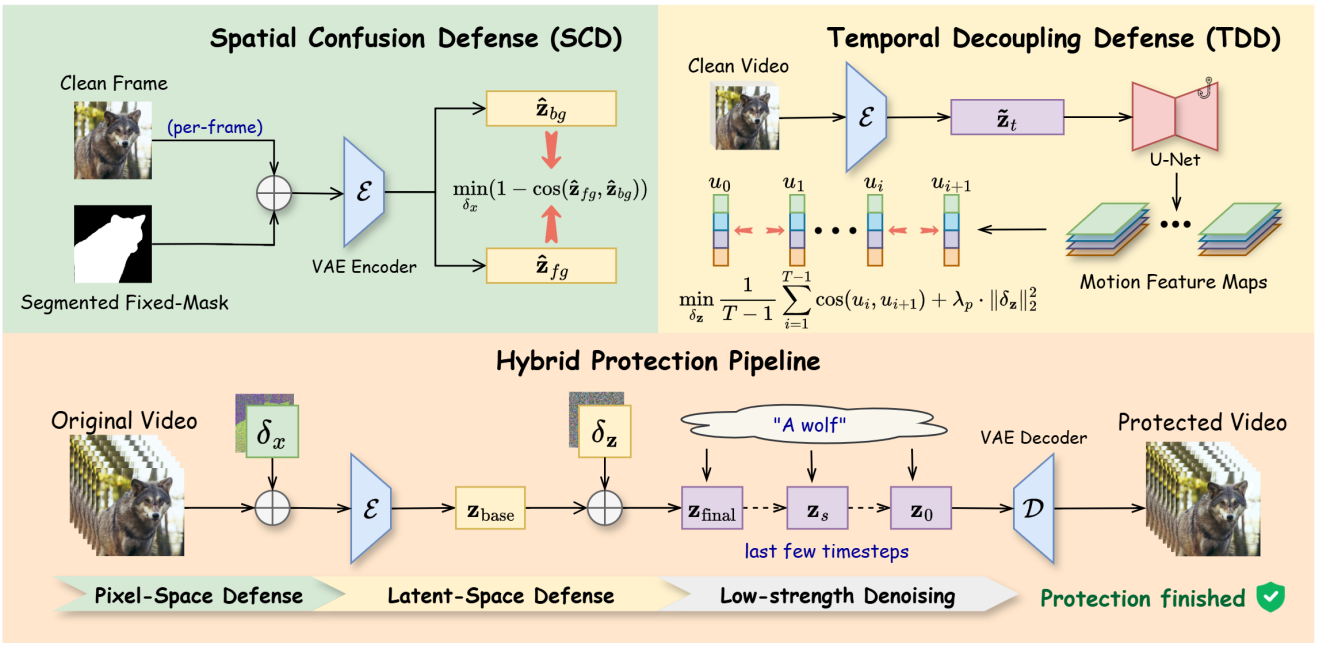

Fig. 2. An overview of our proposed JanusGuard framework. The top panel illustrates the independent generation of our two defense components. The bottom panel shows our Hybrid Protection Pipeline, where these pre-computed perturbations are sequentially applied to a clean video, followed by a crucial low-strength denoising refinement step to produce the final protected video.

features of adjacent frames. The overall optimization goal for $\delta_\mathbf{z}$ is to minimize this temporal loss while constraining the perturbation's magnitude:

$$\min_{\delta_\mathbf{z}} \mathcal{L}_{\text{TDD}}(\delta_\mathbf{z}) = \frac{1}{T-1}\sum_{i=1}^{T-1}\cos(u_i,u_{i+1}) + \lambda_p\cdot\|\delta_\mathbf{z}\|_2^2, \quad (5)$$

subject to $\|\delta_\mathbf{z}\|_\infty \le \epsilon_\mathbf{z}$. We optimize the perturbation using PGD. The coefficient λ_p controls the regularization strength and balances protection effectiveness against perturbation imperceptibility.

Hybrid Protection Pipeline Our final framework employs a sequential pipeline that combines two independently optimized perturbations, δ_x from SCD and $\delta_\mathbf{z}$ from TDD. It improves protection efficacy while maintaining visual quality. The process for protecting a clean video $\mathcal{V}=\{x_i\}$ is as follows:

Step 1: Pixel-Space Defense First, the SCD perturbation δ_x is applied to the clean video frames. The SCD-perturbed frames $\{x'_i\}$ are passed through the VAE encoder to obtain a base latent sequence:

$$\mathbf{z}_{\text{base},i} = \mathcal{E}(x'_i) = \mathcal{E}(\text{clamp}(x_i+\delta_{x,i},0,1)). \quad (6)$$

where $\text{clamp}(\cdot,0,1)$ clips each pixel value to $[0,1]$ to keep the perturbed frame within the valid pixel range.

Step 2: Latent-Space Defense The pre-computed TDD perturbation $\delta_\mathbf{z}=\{\delta_{\mathbf{z},i}\}_{i=1}^T$, optimized on clean data, is then applied to the base latent sequence. This further corrupts the representation by introducing temporal inconsistency:

$$\mathbf{z}_{\text{final},i} = \mathbf{z}_{\text{base},i} + \delta_{\mathbf{z},i}. \quad (7)$$

Step 3: Low-Strength Denoising Refinement Instead of directly decoding the final latent sequence $\mathbf{z}_{\text{final}}$, we use it as a starting point for a low-strength DDIM [27] denoising process. This step leverages the U-Net's generative capability to refine the perturbed latent code in the latent space, seamlessly integrating the adversarial signal into the video's natural structure. The entire refinement and decoding process is formulated as below. Specifically, we run the DDIM sampling for only the last few timesteps, controlled by a strength parameter s, which determines the starting timestep t_{start} for the reverse diffusion process. This subtly alters the video based on the corrupted latent code without overwriting its content, ensuring high visual quality. Finally, this refined latent code is passed through the VAE decoder $\mathcal{D}(\cdot)$ to produce the protected video frame:

$$x_{\text{protected},i} = \mathcal{D}(\text{DDIM}(\mathbf{z}_{\text{final},i}, \text{prompt}, \text{strength}=s)). \quad (8)$$

This three-step pipeline corrupts both spatial and temporal representations before the generative model performs any edits, enabling layered protection through complementary perturbations.

IV. Experiments

To validate the effectiveness of our proposed framework, we conduct a series of experiments designed to answer three key questions:

(RQ1) Can our method effectively defend against unauthorized or malicious edits in representative LDM-based video editing frameworks?

(RQ2) Can our method preserve the perceptual quality of the protected videos while maintaining effective protection?

(RQ3) Do the individual components of our method each contribute to the overall protection effectiveness?

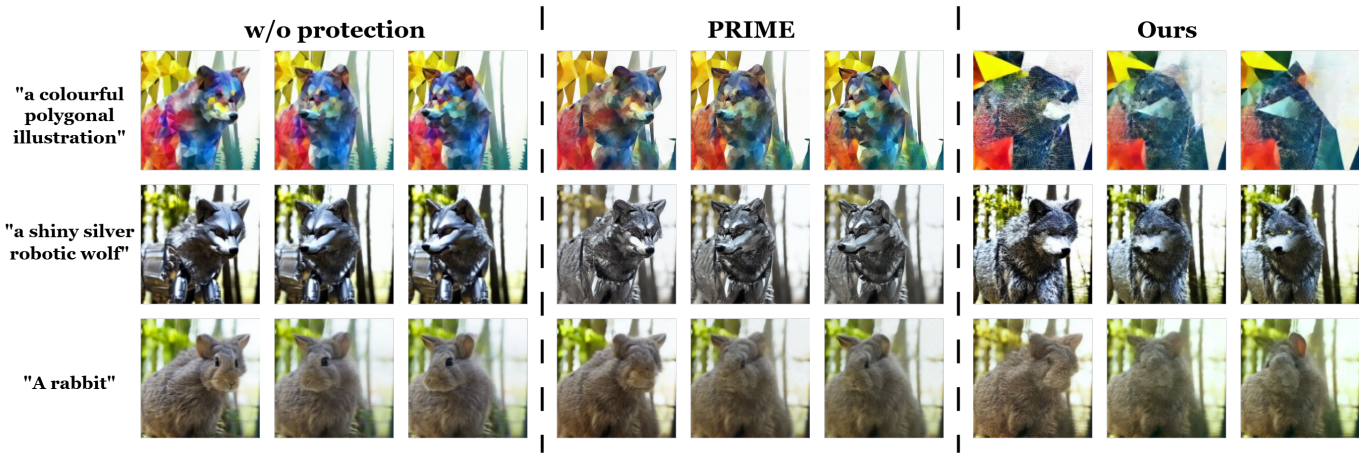

Fig. 3. Qualitative comparison of protection effectiveness on TokenFlow. Each row corresponds to a text prompt, and we show three representative frames from the edited video. Without protection (left), the edits succeed and produce coherent results. PRIME (middle) only partially disrupts the edits, as the target concepts remain recognizable. In contrast, JanusGuard (ours, right) causes the editing to break down, yielding distorted and visually incoherent outputs.

TABLE I
VBENCH EVALUATION RESULTS FOR TOKENFLOW. LOWER SCORES INDICATE BETTER PROTECTION PERFORMANCE.

Video Source	VBench↓			
	Subject Consistency	Motion Smoothness	Imaging Quality	Temporal Flickering
w/o protection	87.31	88.87	61.81	88.10
Random Noise	83.37	87.67	54.09	87.41
PRIME [18]	82.89	88.44	56.76	87.69
JanusGuard (Only SCD)	80.83	87.79	48.55	86.04
JanusGuard (Only TDD)	86.24	85.33	59.91	84.88
JanusGuard (Ours)	**79.77**	**85.02**	**46.46**	**84.58**

TABLE II
VBENCH EVALUATION RESULTS FOR FATEZERO. LOWER SCORES INDICATE BETTER PROTECTION PERFORMANCE.

Video Source	VBench↓			
	Subject Consistency	Motion Smoothness	Imaging Quality	Temporal Flickering
w/o protection	82.49	81.65	47.22	79.90
Random Noise	80.66	81.04	41.58	77.52
PRIME [18]	75.13	78.75	39.05	76.99
JanusGuard (Only SCD)	73.01	80.11	36.78	77.03
JanusGuard (Only TDD)	80.37	75.12	46.09	75.05
JanusGuard (Ours)	**72.48**	**74.05**	**36.03**	**74.54**

A. Experimental Setup

Datasets and Models We collected 100 videos from DAVIS [30] and additional real-world clips, covering diverse content with clear subjects and noticeable motion. Videos range from 8 to 40 frames at a 512×512 resolution. To evaluate real-world defense efficacy, we target two representative open-source video editing models: TokenFlow [3] and FateZero [4]. Our surrogate model consists of the pretrained VAE and U-Net from Stable Diffusion v1.5 [24], with the U-Net being augmented by AnimateDiff motion modules [31].

Baselines We conduct controlled comparisons with methods that are publicly available and reproducible, i.e., whose source code and key algorithmic details are sufficient for faithful implementation. We compare JanusGuard with three baselines: (1) w/o protection, the original unprotected video; (2) Random Noise, where Gaussian noise with the same L_∞ norm as our perturbation is added; and (3) PRIME [18], an open-source video defense method.

Evaluation Metrics We evaluate protection effectiveness and visual quality. For **protection effectiveness**, we assess the quality of the edited videos using four dimensions from VBench [32]: Subject Consistency, Motion Smoothness, Imaging Quality, and Temporal Flickering. For **visual quality**, we measure the imperceptibility of perturbations by comparing protected videos to originals using Peak Signal-to-Noise Ratio (**PSNR**) and Structural Similarity (**SSIM**).

Implementation Details Our framework is implemented in PyTorch and tested on a single NVIDIA A100 GPU. The segmentation masks for SCD are generated by Segment-

Anything Model (SAM) [33], and the VAE is from Stable Diffusion v1.5. For perturbation generation, we use an L_∞ budget of $\epsilon = 8/255$ to bound per-pixel changes. We run PGD for 50 iterations, after which the optimization typically shows diminishing returns in our pilot runs. For the TDD loss, the balancing hyperparameter is set to $\lambda_p = 0.1$. In the final protection pipeline, the DDIM refinement strength is set to $s = 0.2$.

B. Protection Effectiveness (RQ1)

To answer RQ1, we evaluate the protection effectiveness and cross-model transferability of JanusGuard. Table I and Table II report VBench scores (lower is better for protection because it indicates poorer edited results) on two representative LDM-based video editing models, TokenFlow [3] and FateZero [4]. Across all four dimensions, JanusGuard consistently achieves the lowest scores among all baselines, indicating that it degrades the editing outcome most effectively on both target models, even though the perturbation is generated using a generic surrogate model without access to the target architectures.

Compared with PRIME [18], JanusGuard yields a clear additional drop on TokenFlow, especially on Imaging Quality (56.76 $\rightarrow$ 46.46), while also reducing Subject Consistency, Motion Smoothness, and Temporal Flickering. A similar trend is observed on FateZero, where JanusGuard further lowers all four scores.

Fig. 3 provides qualitative evidence on TokenFlow. Without protection, the edited results closely follow the prompts and remain visually coherent across frames. PRIME only partially disrupts the edits, with the target concepts remaining recognizable and only mild degradation observed. In contrast, JanusGuard causes the editing process to break, producing severely distorted and unstable outputs with corrupted appearance and reduced subject coherence, consistent with the lowest VBench scores.

C. Quality of Protected Videos (RQ2)

To answer RQ2, we evaluate whether JanusGuard preserves the visual quality and main content of the original videos after protection. We report PSNR and SSIM between protected and original videos in Table III. JanusGuard achieves 33.56 PSNR and 0.913 SSIM, which is better than Random Noise (33.47 PSNR, 0.883 SSIM). This indicates that our perturbations are more imperceptible than directly injecting Gaussian noise. PRIME attains slightly higher PSNR/SSIM (34.43, 0.921), suggesting smaller pixel-wise deviations under these two pixel-aligned measures.

However, PSNR and SSIM do not always fully reflect human perception, especially when the changes are dominated by subtle high-frequency patterns that are visually benign but penalized by pixel-level comparisons. We therefore complement Table III with a qualitative comparison in Fig. 4. We show frames at the same frame index without applying any editing operation. JanusGuard preserves the overall structure and semantics of the scene without obvious geometric deformation, while Random Noise introduces noticeable high-frequency artifacts. PRIME shows grainy speckle-like patterns with chroma noise in the highlighted region. These observations support that JanusGuard maintains high perceptual fidelity while remaining effective for protection in RQ1.

TABLE III
VISUAL QUALITY COMPARISON OF PROTECTED VIDEOS.

Method	PSNR ↑	SSIM ↑
Random Noise	33.47	0.883
PRIME	34.43	0.921
JanusGuard (Ours)	33.56	0.913

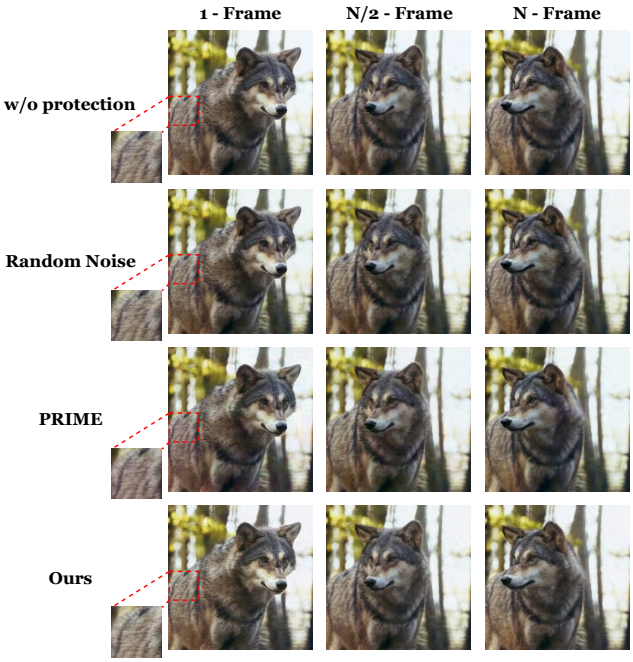

Fig. 4. Qualitative comparison of imperceptibility. We show the original frame and the protected frames from Random Noise, PRIME, and JanusGuard at the same frame index, without any editing operation. The red dashed box marks the zoomed-in region. JanusGuard preserves a cleaner and more coherent texture than the baselines.

D. Ablation Studies (RQ3)

To assess the contribution of each component, we conduct ablation studies on both TokenFlow and FateZero (Table I and Table II). Using only the spatial module (JanusGuard *Only SCD*) provides strong protection, yielding lower VBench scores than PRIME in Subject Consistency and Imaging Quality on both models. In contrast, using only the temporal module (JanusGuard *Only TDD*) more effectively degrades motion-related dimensions, leading to lower Motion Smoothness and Temporal Flickering.

Combining SCD and TDD consistently achieves the lowest scores across all four dimensions. The improvement over each single-component variant is most evident on motion-related metrics, which indicates that temporal disruption is necessary

in addition to spatial corruption. These results suggest that SCD and TDD provide complementary benefits, and both are important for comprehensive protection.

V. Conclusion

In this paper, we proposed JanusGuard, a target-agnostic defense against unauthorized video editing. By disrupting internal representations used by video editing models, JanusGuard avoids reliance on external targets. Through the joint effects of Spatial Confusion Defense (SCD) and Temporal Decoupling Defense (TDD), it interferes with spatial semantics and temporal dynamics while preserving visual fidelity. Experiments on mainstream video editing models demonstrate effective and transferable protection. Future work will study robustness under common post-processing operations such as compression, rotation, and resizing.

References

[1] J. Z. Wu, Y. Ge, X. Wang, S. W. Lei, Y. Gu, Y. Shi, W. Hsu, Y. Shan, X. Qie, and M. Z. Shou, "Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 7623–7633.

[2] S. Liu, Y. Zhang, W. Li, Z. Lin, and J. Jia, "Video-p2p: Video editing with cross-attention control," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 8599–8608.

[3] M. Geyer, O. Bar-Tal, S. Bagon, and T. Dekel, "Tokenflow: Consistent diffusion features for consistent video editing," in *The Twelfth International Conference on Learning Representations*, 2024, pp. 20 637–20 649.

[4] C. Qi, X. Cun, Y. Zhang, C. Lei, X. Wang, Y. Shan, and Q. Chen, "Fatezero: Fusing attentions for zero-shot text-based video editing," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 15 932–15 942.

[5] O. Kara, B. Kurtkaya, H. Yesiltepe, J. M. Rehg, and P. Yanardag, "Rave: Randomized noise shuffling for fast and consistent video editing with diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 6507–6516.

[6] Y. Cong, M. Xu, C. Simon, S. Chen, J. Ren, Y. Xie, J.-M. Pérez-Rúa, B. Rosenhahn, T. Xiang, and S. He, "Flatten: optical flow-guided attention for consistent text-to-video editing," in *The Twelfth International Conference on Learning Representations*, 2024, pp. 8826–8846.

[7] M. Ku, C. Wei, W. Ren, H. Yang, and W. Chen, "Anyv2v: A tuning-free framework for any video-to-video editing tasks," *Transactions on Machine Learning Research*, 2024.

[8] G. Li, Y. Yang, C. Song, and C. Zhang, "Flowdirector: Training-free flow steering for precise text-to-video editing," *arXiv preprint arXiv: 2506.05046*, 2025.

[9] J. Choi, T. Kim, Y. Jeong, S. Baek, and J. Choi, "Exploiting style latent flows for generalizing deepfake video detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 1133–1143.

[10] Y.-H. Han, T.-M. Huang, K.-L. Hua, and J.-C. Chen, "Towards more general video-based deepfake detection through facial component guided adaptation for foundation model," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 22 995–23 005.

[11] Z. Yu, S. Zhai, and N. Zhang, "Antifake: Using adversarial audio to prevent unauthorized speech synthesis," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023, pp. 460–474.

[12] S. I. A. Meerza, L. Sun, and J. Liu, "Harmonycloak: Making music unlearnable for generative ai," in *2025 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2025, pp. 430–448.

[13] C. Liang, X. Wu, Y. Hua, J. Zhang, Y. Xue, T. Song, Z. Xue, R. Ma, and H. Guan, "Adversarial example does good: preventing painting imitation from diffusion models via adversarial examples," in *40th International Conference on Machine Learning, ICML 2023*, 2023, pp. 20 763–20 786.

[14] H. Salman, A. Khaddaj, G. Leclerc, A. Ilyas, and A. Madry, "Raising the cost of malicious ai-powered image editing," in *International Conference on Machine Learning*, 2023, pp. 29 894–29 918.

[15] T. Van Le, H. Phung, T. H. Nguyen, Q. Dao, N. N. Tran, and A. Tran, "Anti-dreambooth: Protecting users from personalized text-to-image synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 2116–2127.

[16] Y. Liu, C. Fan, Y. Dai, X. Chen, P. Zhou, and L. Sun, "Metacloak: Preventing unauthorized subject-driven text-to-image diffusion-based synthesis via meta-learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 24 219–24 228.

[17] H. Yu, J. Chen, X. Ding, Y. Zhang, T. Tang, and H. Ma, "Step vulnerability guided mean fluctuation adversarial attack against conditional diffusion models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 7, 2024, pp. 6791–6799.

[18] G. Li, S. Yang, J. Zhang, and T. Zhang, "Prime: Protect your videos from malicious editing," *arXiv preprint arXiv:2402.01239*, 2024.

[19] K. Li, J. Gu, X. Yu, J. Cao, Y. Tang, and X.-P. Zhang, "Uvcg: Leveraging temporal consistency for universal video protection," *arXiv preprint arXiv:2411.17746*, 2024.

[20] J. Cao, K. Li, X. Yu, H. Li, and X. Zhang, "Videoguard: Protecting video content from unauthorized editing," *arXiv preprint arXiv:2508.03480*, 2025.

[21] D. Gui, X. Guo, W. Zhou, and Y. Lu, "I2vguard: Safeguarding images against misuse in diffusion-based image-to-video models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 12 595–12 604.

[22] Y. Pang, B. Chen, Y. Zhang, and T. Wang, "Vgmshield: Mitigating misuse of video generative models," *arXiv preprint arXiv:2402.13126*, 2024.

[23] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851.

[24] 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.

[25] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations*, 2014, pp. 1–14.

[26] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[27] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2021, pp. 14 205–14 224.

[28] J. Deng, C. Lin, Z. Zhao, S. Liu, Z. Peng, Q. Wang, and C. Shen, "A survey of defenses against ai-generated visual media: Detection, disruption, and authentication," *ACM Computing Surveys*, vol. 58, no. 5, pp. 1–35, 2025.

[29] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018, pp. 4138–4160.

[30] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, "The 2017 davis challenge on video object segmentation," *arXiv:1704.00675*, 2017.

[31] 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," *International Conference on Learning Representations*, pp. 3384–3402, 2024.

[32] Z. Huang, Y. He, J. Yu, F. Zhang, C. Si, Y. Jiang, Y. Zhang, T. Wu, Q. Jin, N. Chanpaisit, Y. Wang, X. Chen, L. Wang, D. Lin, Y. Qiao, and Z. Liu, "Vbench: Comprehensive benchmark suite for video generative models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 21 807–21 818.

[33] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollar, and R. Girshick, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 4015–4026.