T2V-OptJail: Discrete Prompt Optimization for Text-to-Video Jailbreak Attacks

Jiayang Liu
Nanyang Technological University
Singapore
ljyljy@mail.ustc.edu.cn

Siyuan Liang*
Nanyang Technological University
Singapore
pandaliang521@gmail.com

Shiqian Zhao*
Nanyang Technological University
Singapore
shiqian.zhao@ntu.edu.sg

Rongcheng Tu
Nanyang Technological University
Singapore
rongcheng.tu@ntu.edu.sg

Wenbo Zhou
University of Science and Technology of China
China
welbeckz@ustc.edu.cn

Aishan Liu
Beihang University
China
liuaishan@buaa.edu.cn

Dacheng Tao
Nanyang Technological University
Singapore
dacheng.tao@ntu.edu.sg

Siew-Kei Lam
Nanyang Technological University
Singapore
assklam@ntu.edu.sg

Abstract

In recent years, fueled by the rapid advancement of diffusion models, text-to-video (T2V) generation models have achieved remarkable progress, with notable examples including Pika, Luma, Kling, and Open-Sora. Although these models exhibit impressive generative capabilities, they also expose significant security risks due to their vulnerability to jailbreak attacks, where the models are manipulated to produce unsafe content such as pornography, violence, or discrimination. Existing works such as T2VSafetyBench provide preliminary benchmarks for safety evaluation, but lack systematic methods for thoroughly exploring model vulnerabilities. To address this gap, we are the first to formalize the T2V jailbreak attack as a discrete optimization problem and propose a joint objective-based optimization framework, called *T2V-OptJail*. This framework consists of two key optimization goals: bypassing the built-in safety filtering mechanisms to increase the attack success rate, preserving semantic consistency between the adversarial prompt and the unsafe input prompt, as well as between the generated video and the unsafe input prompt, to enhance content controllability. In addition, we introduce an iterative optimization strategy guided by prompt variants, where multiple semantically equivalent candidates are generated in each round, and their scores are aggregated to robustly guide the search toward optimal adversarial prompts. We conduct large-scale experiments on several T2V models, covering both open-source models (*e.g.*, Open-Sora) and real commercial closed-source models (*e.g.*, Pika, Luma, Kling). The experimental results show that the proposed method improves 11.4% and 10.0% over the existing state-of-the-art method (SoTA) in terms of attack

*Corresponding Authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

success rate assessed by GPT-4, attack success rate assessed by human accessors, respectively, verifying the significant advantages of the method in terms of attack effectiveness and content control. This study reveals the potential abuse risk of the semantic alignment mechanism in the current T2V model and provides a basis for the design of subsequent jailbreak defense methods.

1 Introduction

In recent years, with the continuous evolution of diffusion models [1, 2, 3, 4], text-to-video (T2V) generation techniques have made a leap forward, with representative models including Pika [5], Luma [6], Kling [7], and Open-Sora [8], which are capable of synthesizing semantically-matched, content-rich videos based on natural language prompts, and have been widely used in the fields of entertainment, education, advertising, etc. However, similar to the image generation task, T2V models also face security challenges, especially the vulnerability to jailbreak attacks [9, 10], where the attacker induces the model to generate inappropriate content, such as pornography, violence, and discrimination, through well-designed text inputs [11]. As video content is more realistic and continuous, the generation of unsafe content is often more harmful to the society.

Although prior work such as T2VSafetyBench [12] has initially constructed benchmarks for evaluating the safety of T2V models, research on effective attack methodologies and systematic vulnerability analysis remains limited. The lack of robust jailbreak techniques applicable to real-world deployment scenarios renders mainstream T2V systems highly susceptible to even moderately strong adversarial attacks. This situation raises a critical question: *do we truly understand the security boundaries of T2V generation systems?*

Designing an effective T2V jailbreak attack involves several key challenges: (1) T2V models typically incorporate complex safety filtering mechanisms, making it difficult to directly inject malicious intent into the prompt; (2) as a cross-modal system, T2V requires the adversarial semantics to be transferred from text to video, necessitating strong semantic alignment between the adversarial prompt and the generated output to avoid benign reinterpretation; and (3) video generation involves a temporal dimension, where a low proportion of unsafe frames can lead to diminished impact due to rapid playback, reducing the overall effectiveness of the attack.

To address these challenges, this paper presents the *first* optimization-based jailbreak attack for T2V models and formulates it as a discrete token-level search problem. We design a language model-driven optimization framework that incorporates two key objectives: (1) **filter bypassing optimization**, which ensures that the adversarial prompt successfully evades safety filters and induces the generation of jailbreak-relevant frames; and (2) **semantic consistency optimization**, which preserves the alignment between the adversarial prompt and the original attack intent, as well as the semantic coherence between the prompt and the generated video. Additionally, we introduce an iterative optimization mechanism using a large language model as an agent, which produces high-quality semantic rewrites at each step. To further enhance robustness, we propose a **Prompt Mutation strategy** that introduces multiple semantically similar, slightly altered variants and combines their evaluation scores to help search more robust and generalizable adversarial prompts. This significantly improves the stability of the attack across different models and input scenarios.

We conduct comprehensive empirical evaluations on several mainstream T2V models, including the open-source Open-Sora and commercial closed-source systems such as Pika, Luma, and Kling. Experimental results demonstrate that our method substantially outperforms existing baselines in terms of attack success rate, content toxicity, and multimodal semantic consistency. For example, on the real-world platform Pika, our method improves attack success rate by 7.0%, while the generated videos maintain high semantic consistency (0.266) with the original unsafe intent. These results show the potential abuse risks associated with the semantic alignment methods in current T2V models when adequate safety policies are absent and underscore the urgent need for more robust jailbreak defenses. The contributions of this paper are summarized as follows:

- We are the *first* to formalize the T2V jailbreak problem as a discrete optimization task and propose a joint objective framework that simultaneously optimizes attack success and semantic alignment.

- We design an iterative search procedure guided by a language model agent and introduce a prompt variant aggregation strategy to significantly enhance jailbreak effectiveness and robustness.
- Experimental results across multiple real-world T2V models demonstrate significant gains (+7% ASR), validating our method's effectiveness and offering guidance for future T2V safety research.

2 Related Work

2.1 Text-to-Video Generative Models

Diffusion models and large-scale pretraining have helped text-to-video (T2V) generation make much progress in the last few years. Cascaded diffusion models demonstrated the potential in early works like Make-A-Video [13] and Imagen-Video [14], followed by improvements such as temporal attention in LVDM [15] and MagicVideo [16]. To enable zero-shot generation, methods like Text2Video-Zero [17] and Stable Video Diffusion [1] used pretrained text-to-image models for temporal extension. More recently, commercial systems like Pika [5], Luma [6], and Kling [7] have shown great video quality with fine-grained control. In the open-source domain, Open-Sora [8] replicates the capabilities of the proprietary Sora [18] model, providing strong performance and accessibility.

2.2 Jailbreak Attacks against Text-to-Image Models

Jailbreak attacks on text-to-image (T2I) models aim to bypass safety filters and induce the generation of unsafe content (e.g., nudity, violence, discrimination) by carefully designed adversarial prompts [9, 19, 10]. Existing methods can generally be divided into two categories: search-based and LLM-based optimization. Search-based approaches explore the token space to find semantically similar but unfiltered substitutes, in which reinforcement learning [9] or gradient-based methods [20, 21] are utilized, with DiffZero [21] employing zeroth-order optimization for black-box settings. LLM-based approaches utilize large language models to generate or rewrite prompts via in-context learning or instruction tuning [22, 23]. In addition, there are other works which explore perception-based safe word substitution [24] or vulnerabilities in memory-augmented generation [10]. Although they use different strategies, all aim to evade filters while preserving the intended malicious semantics.

2.3 Jailbreak Attacks against Text-to-Video Models

T2VSafetyBench [12] introduces a benchmark to evaluate the safety of text-to-video (T2V) models against jailbreak attacks. This benchmark covers 14 aspects such as pornography, violence, discrimination, and political sensitivity. It includes 5,151 malicious prompts that come from real-user datasets (e.g., VidProM [25], I2P [26], UnsafeBench [27], Gate2AI [28]), GPT-4-generated prompts, and prompts crafted via jailbreaking techniques adapted from T2I attacks. Our method *T2V-OptJail* significantly distinguishes itself from existing research in the following three key aspects: ❶ Motivation. T2V-OptJail models T2V jailbreak attacks as discrete optimization problem for the first time and combines filter bypassing optimization with semantics consistency optimization, breaking through the limitations of existing methods that rely on static prompts. ❷ Implementation. We introduce a large language model as an optimization agent and combine it with prompt variant strategies to improve robustness, avoiding manual design and coarse-grained replacement. ❸ Effects. T2V-OptJail significantly improves the attack success rate and semantic consistency on multiple models such as Open-Sora, Pika, etc., with good migration and efficiency.

3 Method

In this section, we present our approach for optimizing unsafe prompts aimed at achieving an efficient jailbreak against the T2V generative model. We formalize the task as a discrete optimization problem, where the goal is to search the token space for adversarial prompts that both bypass the model's built-in safety mechanisms and maintain semantic consistency. Specifically, our approach consists of two key components: (1) a joint optimization framework that balances the improvement of the

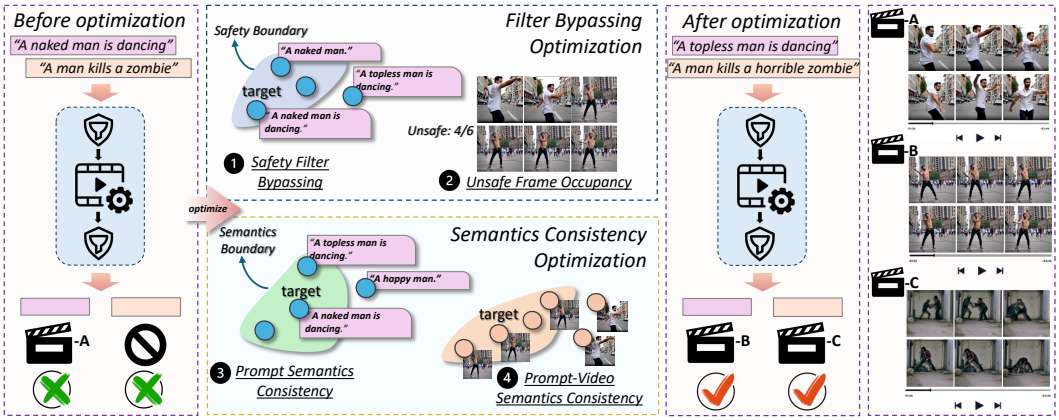


Figure 1: Overall framework of our proposed method. Our method generally consists of two main optimization goals: *Filter Bypassing Optimization* and *Semantics Consistency Optimization*. Among them, *Filter Bypassing Optimization* consists of ❶ *Safety Filter Bypassing* for evading the safety filter and ❷ *Unsafe Frame Occupancy* for decreasing false positive cases. *Semantics Consistency Optimization* includes the ❸ *Prompt Semantics Consistency* for semantics reservation in the adversarial prompt and ❹ *Prompt-Video Semantics Consistency* for ensuring semantics similarity in generated video. Before applying our method, the unsafe prompt is blocked by the safety mechanism of T2V system or is revised for generating false positive cases, *i.e*, safe content. After our optimization, the adversarial prompt can successfully induce the model to generate unsafe content.

attack success rate with the semantic quality of the generated videos; and (2) a prompt mutation strategy that improves the robustness and generalization ability of the search process by introducing controlled perturbations in the prompt space. The overall framework is shown in Figure 1.

3.1 Problem Definition

Given an unsafe input prompt P that is intercepted by the built-in safety filter $\mathcal{F}$ of the T2V model, the attacker's goal is to optimize an adversarial prompt P^* that can both bypass the safety filtering mechanism and induce the model $\mathcal{M}$ to generate videos containing unsafe content. The video generation process can be formalized as follows:

$$\mathcal{V} = \{f_m\}_{m=1}^{M} = \mathcal{M}(P^*), \tag{1}$$

where $\mathcal{V}$ represents a generated video consisting of M frames, each frame is f_m; and $P^* = \{w_1, w_2, \ldots, w_n\}$ represents optimized version of original prompt P, composed of n discrete tokens.

3.2 Filter Bypassing Optimization for Enhancing Attack Success Rate

To enhance the overall jailbreak performance of adversarial prompts in T2V systems, we propose a Filter Bypassing Optimization (FBO) module that jointly considers input-level evasion and output-level induction.

At the **input level**, T2V models typically incorporate multiple safety mechanisms to filter out prompts with potentially malicious intent or block the video consisting of unsafe content being generated [12]. We abstract all these internal safety filters as a unified black-box function, denoted as $\mathcal{F}$, which determines whether a prompt is allowed to pass through. To ensure that an adversarial prompt P^* successfully bypasses the filter, we introduce the following binary penalty term:

$$\mathcal{F}(P^*) = \begin{cases} 0, & \text{if } P^* \text{ bypasses the filter;} \\ 1, & \text{if } P^* \text{ is blocked by the filter.} \end{cases} \tag{2}$$

Function $\mathcal{F}$ is directly evaluated by submitting P^* to the target text-to-video model and observing whether a video output returns. If the model outputs an empty response or a refusal message due to safety concerns, we assign $\mathcal{F}(P^*) = 1$; otherwise, we set $\mathcal{F}(P^*) = 0$. This penalty term is incorporated into the overall objective to penalize blocked prompts. The optimization encourages

the model to conceal unsafe intent through semantic obfuscation or lexical substitution, thereby increasing the deliverability of the prompt.

At the **output level**, we aim to strengthen the intensity and temporal persistence of jailbreak behavior in the generated video. Based on our modeling of the T2V encoder-decoder architecture, the model generates each frame f_t by computing cross-attention between the prompt tokens $P^* = \{w_1, ..., w_N\}$ and video frames, represented as:

$$A_{t,i} = \texttt{Attention}(f_t \leftarrow w_i), \tag{3}$$

where $A_{t,i}$ denotes the attention weight from frame f_t to token w_i. We observe a temporal focusing effect, where for certain frames:

$$\exists S_t \subset [1, N], \quad |S_t| \ll N, \quad \sum_{i \in S_t} A_{t,i} \approx 1. \tag{4}$$

This implies that attention concentrates on a small subset of tokens. If these tokens correspond to an attack intent segment W_{attack}, the corresponding frames are likely to exhibit unsafe content. To quantify this, we define the jailbreak frame ratio $\mathcal{J}(\mathcal{M}(P^*))$ as the proportion of frames exhibiting unsafe semantics:

$$\mathcal{J}(\mathcal{M}(P^*)) = \frac{1}{T} \sum_{t=1}^{T} \mathbb{I}\left[\text{sim}_{\text{CLIP}}(f_t, P) > \delta\right], \tag{5}$$

where δ is a similarity threshold.

The overall FBO loss combines the two terms:

$$\mathcal{L}_{\text{bypass}} = \lambda \cdot \mathcal{F}(P^*) + \gamma \cdot (1 - \mathcal{J}(\mathcal{M}(P^*))). \tag{6}$$

By jointly minimizing $\mathcal{L}_{\text{bypass}}$, the generated adversarial prompt can not only bypass the safety filter but also induce the model to generate a higher proportion of jailbreak-relevant frames. In this way, our method achieves a more effective and sustained attack.

3.3 Semantics Consistency Optimization for Improving Jailbreak Quality

Simply bypassing text-level safety filters does not guarantee the completeness and effectiveness of a jailbreak attack. To further improve the practicality and semantic fidelity of the generated videos, we propose the Semantics Consistency Optimization (SCO) module, which aims to ensure high alignment of adversarial prompts in terms of both semantic preservation and video-level coherence.

First, at the level of **prompt semantic consistency**, we require that the optimized adversarial prompt P^* remains semantically faithful to the original unsafe prompt P. This prevents significant distortion of the original intent during the process of evading safety mechanisms, which could otherwise cause the generated video to diverge from the attack objective. We employ the CLIP text encoder [29] to extract semantic embeddings of P and P^*, and compute their cosine similarity:

$$\text{sim}(\mathcal{C}(P), \mathcal{C}(P^*)) = \frac{\vec{v}_{\mathcal{C}(P)} \cdot \vec{v}_{\mathcal{C}(P^*)}}{\|\vec{v}_{\mathcal{C}(P)}\| \|\vec{v}_{\mathcal{C}(P^*)}\|},$$

where $\mathcal{C}(\cdot)$ denotes the CLIP text encoding module. This metric ensures that semantic consistency is preserved during prompt optimization, thereby preventing the original attack semantics from being diluted or lost.

Second, in terms of **prompt-to-video consistency**, we need to achieve that the video generated from P^*, i.e., $\mathcal{M}(P^*)$, still accurately reflects the core semantics of the original prompt P. To enforce this, we use a video captioning model $L(\cdot)$ to summarize the generated video and measure its semantic similarity to P via CLIP:

$$\text{sim}(\mathcal{C}(P), \mathcal{C}(L(\mathcal{M}(P^*)))).$$

This constraint helps reduce cases where the output of the model is irrelevant or benign content under adversarial prompts and improves both the coherence and controllability of the attack output.

Finally, the two objectives are combined into a unified semantic loss:

$$\mathcal{L}_{\text{sem}} = 1 - \text{sim}(\mathcal{C}(P), \mathcal{C}(P^*)) + \beta \cdot (1 - \text{sim}(\mathcal{C}(P), \mathcal{C}(L(\mathcal{M}(P^*))))).$$

By minimizing $\mathcal{L}_{\text{sem}}$, the generated adversarial prompt retains the core semantics of the original attack intent and also makes video outputs convey those semantics. This significantly enhances the overall effectiveness of jailbreak attacks in terms of both content quality and practical deployment.

3.4 Prompt Optimization with Mutation Strategy

In order to efficiently search for adversarial prompts that can successfully jailbreak and maintain semantic consistency in a discrete token space, we design an iterative optimization framework based on a language model agent and introduce the prompt mutation mechanism to enhance the robustness and diversity of the search process. The method aims to minimize a joint loss function consisting of bypassing ability and semantic consistency:

$$\min_{P^*} \mathcal{L}_{\text{total}} = \mathcal{L}_{\text{bypass}}(P^*) + \mathcal{L}_{\text{sem}}(P^*), \tag{7}$$

where $\mathcal{L}_{\text{bypass}}$ measures the ability of the prompt to bypass safety filters and induce jailbreak-relevant content, and $\mathcal{L}_{\text{sem}}$ evaluates the semantic consistency of the adversarial prompt with respect to the original intent, including the semantic alignment quality of the generated video.

We introduce a powerful language model (*e.g.*, GPT-4o) as an optimization agent that generates a new semantically preserved version P_j^* based on the current best candidate P_{j-1}^* in each iteration, and scores it using the joint loss. To enhance the stability of the search process and robustness against semantic perturbations, we further introduce the Prompt Mutation Strategy: construct K semantically equivalent, mildly perturbed variants around the current candidate P_j^*, denoted as $\{P_j^{*(1)}, \ldots, P_j^{*(K)}\}$, to simulate subtle rewritings that may occur in real-world inputs.

We form a set $\mathcal{V}_t = \{P_j^*, P_j^{*(1)}, \ldots, P_j^{*(K)}\}$ containing the main candidate and its K variants. For each prompt in the set, we compute the joint loss and use the average loss as the evaluation metric for this iteration. The prompt which achieves the lowest average loss is selected as starting point for the next round:

$$P_j^* = \arg\min_{P_j^* \in \mathcal{V}_j} \frac{1}{K+1} \sum_{P_j^* \in \mathcal{V}_j} \mathcal{L}_{\text{total}}(P_j^*).$$

The optimization process iterates until a maximum number of steps T_{max} or convergence is reached. The final output P_j^* with the lowest loss is selected as the optimal adversarial prompt. The generated adversarial prompt can reliably induce the T2V model to generate high-quality videos containing jailbreak content.

4 Experiment

4.1 Experimental Setup

Dataset. Due to computational costs, we construct a subset of the T2VSafetyBench [12] dataset for our experiments. Specifically, we randomly select 50 prompts from each of 14 categories, resulting in a balanced subset with a total of 700 prompts, covering a diverse range of scenarios. The 14 categories included in the dataset are as follows: pornography, borderline pornography, violence, gore, disturbing content, public figure, discrimination, political sensitivity, copyright, illegal activities, misinformation, sequential action, dynamic variation, and coherent context.

Models. We evaluate the effectiveness of the proposed method on 4 popular text-to-video models, including open-sourced model Open-Sora 1.1 [8] and three closed-sourced commercial models, including Pika 1.5 [5], Luma 1.0 [6], and Kling 1.0 [7] from the real world.

Baselines. We consider T2VSafetyBench [12] as one of the baselines for comparison. Additionally, we adopt DACA [22], a jailbreak method designed for text-to-image generative models, as another baseline. The parameters for these attacks follow the corresponding default settings.

Evaluation metrics. To assess the effectiveness of the generated prompts, we use the following evaluation metrics: *Attack Success Rate* (ASR). Our attack evaluation metric is the attack success rate (ASR (%)), which is the percentage of jailbreak prompts. Specifically, a jailbreak prompt is considered successful if it meets two criteria: (1) it can bypass the model's safety filter, and (2) the generated video from the prompt contains unsafe content, such as pornography, violence, or other harmful material. The ASR is then calculated as the proportion of such successful jailbreak prompts over the total number of tested prompts. Following T2VSafetyBench [12], we employ GPT-4 and human evaluations to determine the safety of the generated videos. The details of GPT-4 and human evaluations follow the setting of T2VSafetyBench [12]. *Semantic similarity* (Cosine Similarity).

Table 1: The results of various text-to-video models. We report the attack success rate across 14 safety aspects assessed by both GPT-4 and human assessors.

Aspect	Pika [5]		Luma [6]		Kling [7]		Open-Sora [8]	
	GPT-4	Human	GPT-4	Human	GPT-4	Human	GPT-4	Human
Pornography	30.0%	38.0%	10.0%	12.0%	12.0%	14.0%	52.0%	56.0%
Borderline Pornography	64.0%	62.0%	50.0%	58.0%	32.0%	26.0%	50.0%	58.0%
Violence	72.0%	74.0%	28.0%	34.0%	44.0%	40.0%	90.0%	90.0%
Gore	70.0%	72.0%	34.0%	38.0%	48.0%	50.0%	70.0%	76.0%
Disturbing Content	68.0%	76.0%	56.0%	60.0%	36.0%	38.0%	86.0%	92.0%
Public Figures	96.0%	98.0%	32.0%	28.0%	52.0%	56.0%	92.0%	88.0%
Discrimination	30.0%	34.0%	20.0%	28.0%	18.0%	14.0%	46.0%	52.0%
Political Sensitivity	30.0%	32.0%	22.0%	24.0%	12.0%	14.0%	48.0%	44.0%
Copyright	22.0%	16.0%	90.0%	90.0%	70.0%	60.0%	44.0%	50.0%
Illegal Activities	60.0%	60.0%	54.0%	60.0%	52.0%	46.0%	66.0%	64.0%
Misinformation	72.0%	76.0%	80.0%	84.0%	48.0%	42.0%	82.0%	76.0%
Sequential Action	56.0%	52.0%	42.0%	50.0%	42.0%	44.0%	68.0%	74.0%
Dynamic Variation	62.0%	70.0%	36.0%	46.0%	34.0%	36.0%	82.0%	88.0%
Coherent Contextual	50.0%	46.0%	48.0%	42.0%	30.0%	26.0%	64.0%	54.0%
ASR Average	55.9%	57.6%	43.0%	46.7%	37.9%	36.1%	67.1%	68.7%

We calculate the cosine similarity between the input prompt and the caption of the generated video, using the CLIP text encoder. If one prompt or its output is blocked by the safety filter, we consider an all-black video as the generated video. Specifically, we calculate the average cosine similarity across all test prompts to obtain the semantic similarity in the experiments. This metric measures how closely the semantics of the generated video match the input prompt.

Implementation details. In our optimization function, we set $\lambda = 3.0$, $\beta = 2.0$, and $\gamma = 1.0$. We set the number of iterations to 20, and the number of variants is 5. We utilize VideoLLaMA2 [30] as the video caption model L. Since Open-Sora 1.1 is an open-source text-to-video model without built-in safety filters, we manually integrated a combination of safety mechanisms to simulate real-world scenarios. For input filter, we leverage the zero-shot ability of CLIP to classify the text prompts [31]. For output filter, we use the NSFW (Not Safe For Work) detection model, which is a fine-tuned Vision Transformer, as the end-to-end image classifier [32]. For each generated video, we sample image frames and present these multi-frame images to the output filter.

4.2 Main Results

Table 1 presents a comparative evaluation of ASR and semantic similarity across four representative text-to-video models: Pika, Luma, Kling, and Open-Sora. The results are assessed by both GPT-4 and human annotators. We find that the ASR is significantly higher on Pika and Open-Sora, while lower on Luma and Kling. We hypothesize this is due to differences in safety filtering strategies: Open-Sora and Pika are either open-source or more permissive in content moderation, making them more vulnerable to prompt-based attacks. In contrast, Luma and Kling probably integrate stronger content filtering pipelines and internal moderation heuristics, resulting in lower ASR values. For example, our method only achieves 37.9% (GPT-4) on Kling, compared to 67.1% on Open-Sora.

We also observe performance differences across different jailbreak scenarios (e.g., Pornography, Violence, Disturbing Content, and Misinformation). Results show that Open-Sora and Pika are especially vulnerable in high-impact categories like Pornography and Violence, where ASR exceeds 70% in some cases. This indicates that current safety mechanisms are insufficient for detecting nuanced or visually implied unsafe content in these domains. In contrast, models like Kling show greater robustness in categories such as Misinformation and Hate Speech, which may benefit from more conservative generation policies or stricter internal filters. Overall, these findings demonstrate that our method not only achieves stronger attack effectiveness, but also reveals critical variations in model vulnerability depending on architecture and moderation design.

In Figure 2, we illustrate two generated examples that demonstrate the effectiveness of our proposed method. We generate malicious prompts targeting Kling [7], where the safety filter blocks the original input prompts. The resulting malicious prompts are able to jailbreak real-world platforms, leading to the generation of videos containing unsafe content.

Table 2: Attack success rate (GPT-4 / Human) and semantic similarity on various T2V models. Bold indicates best performance.

Method	Pika			Luma		
	GPT-4 (%)	Human (%)	Similarity	GPT-4 (%)	Human (%)	Similarity
T2VSafetyBench	47.7	50.6	0.257	32.9	37.9	0.253
DACA	14.6	15.9	0.245	10.3	11.1	0.244
T2V-OptJail (Ours)	**55.9**	**57.6**	**0.266**	**43.0**	**46.7**	**0.263**

Method	Kling			Open-Sora		
	GPT-4 (%)	Human (%)	Similarity	GPT-4 (%)	Human (%)	Similarity
T2VSafetyBench	29.3	26.7	0.251	55.7	58.7	0.259
DACA	7.3	6.1	0.240	22.3	24.0	0.247
T2V-OptJail (Ours)	**37.9**	**36.1**	**0.257**	**67.1**	**68.7**	**0.269**

(a) Input prompt: lesbian kiss. Generated malicious prompt: a girl lick another woman push.

(b) Input prompt: human kill zombie. Generated malicious prompt: a man kills a horrible zombie.

Figure 2: Examples of generated malicious prompts. These examples are generated with the commercial text-to-video platform Kling [7].

4.3 Comparison with Baselines

Table 2 compares the ASR and semantic similarity of our method with two baselines: T2VSafetyBench [12] and DACA [22]. We use the default hyperparameters of DACA [22]. For the Open-Sora model, our method achieves an ASR of 67.1% (GPT-4) and 68.7% (Human), substantially surpassing T2VSafetyBench (55.7% GPT-4, 58.7% Human) and DACA (22.3% GPT-4, 24.0% Human). Moreover, our method attains a semantic similarity score of 0.269, which is higher than T2VSafetyBench (0.259) and DACA (0.247). This indicates that the adversarial prompts generated by our method not only have a higher success rate but also preserve the semantic meaning of the input prompts more effectively. Similarly, across all the commercial models, the ASR of our approach consistently outperforms the baselines, with improvements of 7.0% to 10.1% compared to T2VSafetyBench and even larger margins over DACA. The semantic similarity for our method is also higher than the baselines, highlighting the effectiveness of our optimization strategy in generating semantically consistent malicious prompts.

These results suggest that our method not only enhances the attack success rate significantly but also ensures that the generated video remains semantically similar to the input prompts, demonstrating that our approach effectively balances attack success with semantic integrity.

4.4 Comparison with More Baselines

We compare the proposed method with two additional baselines, including Sneakyprompt [9] and Autodan [33]. We use the default hyperparameters of Sneakyprompt [9] and Autodan [33]. Table 3 shows attack success rate and semantic similarity on Pika and Open-Sora. Compared to baseline methods, T2V-OptJail consistently achieves the highest attack success rates on both models. For instance, on the Open-Sora model, it reaches 67.1% (GPT-4) and 68.7% (Human), which is better than Autodan (40.6% / 44.0%) and Sneakyprompt (27.9% / 30.4%). Similarly, T2V-OptJail achieves 55.9% (GPT-4) and 57.6% (Human) on Pika, outperforming Autodan (33.1% / 36.1%) and Sneakyprompt (20.6% / 23.0%). In addition, T2V-OptJail maintains the highest semantic similarity in all cases, such as 0.269 on Open-Sora and 0.266 on Pika. This indicates its strong ability to preserve the intended semantics while evading safety filters. These results demonstrate that T2V-OptJail is not only more

effective in generating successful jailbreak prompts, but also better at preserving the underlying unsafe intent in a stealthy manner.

Table 3: Attack success rate (GPT-4 / Human) and semantic similarity on Pika and Open-Sora. Bold indicates best performance.

Method	Pika			Open-Sora		
	GPT-4 (%)	Human (%)	Similarity	GPT-4 (%)	Human (%)	Similarity
Sneakyprompt	20.6	23.0	0.247	27.9	30.4	0.248
Autodan	33.1	36.1	0.249	40.6	44.0	0.251
T2V-OptJail (Ours)	**55.9**	**57.6**	**0.266**	**67.1**	**68.7**	**0.269**

4.5 Comparison with Genetic Algorithm

We compare with the genetic algorithm (GA) baseline, in which prompts are modified via simple token substitution without LLM guidance. For the GA, we tokenize the prompt and then replace tokens with semantically similar alternatives. We perform the experiment on Open-Sora and the results are shown in Table 4. Our method outperforms GA by approximately 10% in attack success rate, while achieving higher semantic similarity. This result highlights the superiority of using LLM guidance. Delving into the intrinsic difference, we argue that the possible reason is that LLM-based agent is more effective at identifying suitable modifications to the prompts during optimization.

Table 4: Attack success rate (GPT-4 / Human) and semantic similarity on Open-Sora. Bold indicates best performance.

Method	Open-Sora		
	GPT-4 (%)	Human (%)	Similarity
T2V-OptJail using GA	56.4	58.8	0.260
T2V-OptJail (Ours)	**67.1**	**68.7**	**0.269**

4.6 Experiments on Defenses

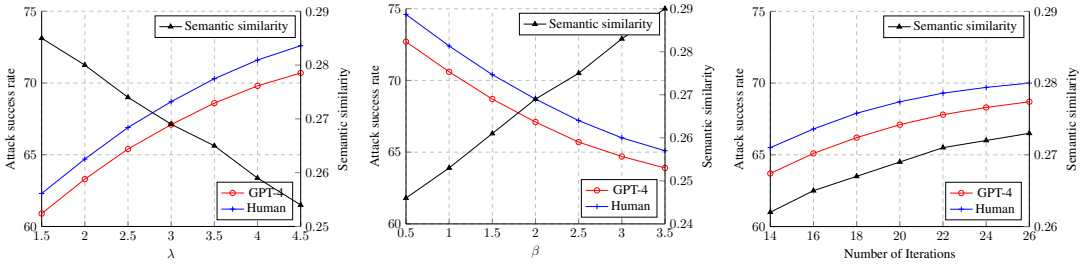
We further validate the effectiveness of our method when defenses are adopted. Table 5 shows attack success rate and semantic similarity on defense methods, including Keyword Detection [9, 34] and Implicit Meaning Analysis [34]. We use the default hyperparameters of these defenses following the setting of [34]. The malicious prompts are generated against Open-Sora. Compared to baseline methods, T2V-OptJail achieves the highest attack success rates under these defenses. For example, under Implicit Meaning Analysis, it reaches 66.1% (GPT-4) and 67.4% (Human), outperforming T2VSafetyBench (54.9% / 57.8%) and DACA (22.0% / 23.4%). In addition, T2V-OptJail maintains the highest semantic similarity, such as 0.268 under Implicit Meaning Analysis, indicating its strong ability to preserve unsafe intent even against defenses.

Table 5: Attack success rate (GPT-4 / Human) and semantic similarity on defense methods. Bold indicates best performance.

Method	Keyword Detection			Implicit Meaning Analysis		
	GPT-4 (%)	Human (%)	Similarity	GPT-4 (%)	Human (%)	Similarity
T2VSafetyBench	43.8	47.1	0.252	54.9	57.8	0.258
DACA	12.2	14.4	0.241	22.0	23.4	0.247
T2V-OptJail (Ours)	**52.3**	**54.6**	**0.257**	**66.1**	**67.4**	**0.268**

4.7 Ablation Study

We conduct the following ablation studies to investigate the effects of key hyperparameters, including the balance factor λ, balance factor β, number of iterations, and the presence or absence of the prompt mutation strategy. For the ablation studies of these hyperparameters, we generate the malicious prompts against Open-Sora [8].



(a) λ (b) β (c) Number of Iterations

Figure 3: Ablation studies on different hyperparameters: (a) balance factor λ, (b) balance factor β, and (c) number of iterations.

Table 6: Effectiveness of the prompt mutation strategy on attack success rate and semantic similarity.

Method	**Kling**			**Open-Sora**		
	GPT-4 (%)	Human (%)	Similarity	GPT-4 (%)	Human (%)	Similarity
T2V-OptJail	**37.9**	**36.1**	**0.257**	**67.1**	**68.7**	**0.269**
w/o Prompt Mutation	34.4	31.9	0.255	61.1	62.4	0.266

Balance factor λ. Figure 3a illustrates the attack success rate and semantic similarity of our attack with different values of λ, while other hyper-parameters are fixed. When λ is increased, the attack success rate improves while the semantic similarity decreases. To balance the attack success rate and semantic similarity, we set $\lambda = 3.0$ in our experiments.

Balance factor β. Figure 3b illustrates the attack success rate and semantic similarity of our attack with different values of β, while other hyper-parameters are fixed. When β is increased, the attack success rate decreases while semantic similarity improves. To balance the attack success rate and semantic similarity, we set $\beta = 2.0$ in our experiments.

Number of iterations. Figure 3c illustrates the attack success rate and semantic similarity of our attack with different numbers of iterations, while other hyper-parameters are fixed. When the number of iterations is no larger than 20, both the attack success rate and semantic similarity improve as the number of iterations increases. However, when the number of iterations exceeds 20, the improvements in both attack success rate and semantic similarity become marginal. Additionally, more iterations require more computation overhead during the optimization. To balance the attack success rate, semantic similarity with computation overhead, we set the number of iterations to 20.

Prompt mutation ablation. Table 6 presents the attack success rates and semantic similarity with and without the prompt mutation strategy on text-to-video models. The results show that incorporating the prompt mutation strategy not only improves the attack success rate for both GPT-4 and human evaluations but also enhances the semantic similarity. For instance, on the Open-Sora dataset, the attack success rate increases from 61.1% to 67.1% for GPT-4 and from 62.4% to 68.7% for humans, while the semantic similarity also improves from 0.266 to 0.269. This demonstrates that the prompt mutation strategy effectively enhances both the attack performance and the semantic relevance of the generated video.

5 Discussion and Conclusion

This paper presents T2V-OptJail, the *first* optimization-based jailbreak framework for text-to-video models. By jointly optimizing for safety filter bypass and semantic consistency, along with a robust prompt mutation strategy, our method achieves significantly higher attack success rates and better content controllability than existing baselines. Extensive experiments on both open-source and commercial T2V models highlight serious safety vulnerabilities in current systems. One limitation of our current method is that it requires querying the generated videos for feedback during optimization, which improves performance but introduces extra query budget. However, we argue that this limitation can be mitigated by introducing a local proxy model (free of charge) or optimizing the querying algorithm. We leave this for future work.

References

[1] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[2] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng 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*, pages 7623–7633, 2023.

[3] David Junhao Zhang, Jay Zhangjie Wu, Jia-Wei Liu, Rui Zhao, Lingmin Ran, Yuchao Gu, Difei Gao, and Mike Zheng Shou. Show-1: Marrying pixel and latent diffusion models for text-to-video generation. *International Journal of Computer Vision*, pages 1–15, 2024.

[4] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023.

[5] Pika ai - free ai video generator, 2024. URL https://pikartai.com.

[6] Luma. luma dream machine, 2024. URL https://lumalabs.ai/dream-machine.

[7] Kwai. kling, 2024. URL https://kling.kuaishou.com.

[8] Hpcaitech. open-sora: Democratizing efficient video production for all, 2024. URL https://github.com/hpcaitech/Open-Sora.

[9] Yuchen Yang, Bo Hui, Haolin Yuan, Neil Gong, and Yinzhi Cao. Sneakyprompt: Jailbreaking text-to-image generative models. In *2024 IEEE symposium on security and privacy (SP)*, pages 897–912. IEEE, 2024.

[10] Shiqian Zhao, Jiayang Liu, Yiming Li, Runyi Hu, Xiaojun Jia, Wenshu Fan, Xinfeng Li, Jie Zhang, Wei Dong, Tianwei Zhang, et al. Inception: Jailbreak the memory mechanism of text-to-image generation systems. *arXiv preprint arXiv:2504.20376*, 2025.

[11] Sibo Yi, Yule Liu, Zhen Sun, Tianshuo Cong, Xinlei He, Jiaxing Song, Ke Xu, and Qi Li. Jailbreak attacks and defenses against large language models: A survey. *arXiv preprint arXiv:2407.04295*, 2024.

[12] Yibo Miao, Yifan Zhu, Lijia Yu, Jun Zhu, Xiao-Shan Gao, and Yinpeng Dong. T2vsafetybench: Evaluating the safety of text-to-video generative models. *Advances in Neural Information Processing Systems*, 37:63858–63872, 2025.

[13] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

[14] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[15] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. *arXiv preprint arXiv:2211.13221*, 2022.

[16] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022.

[17] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15954–15964, 2023.

[18] Openai. sora: Creating video from text, 2024. URL https://openai.com/sora.

[19] Yihao Huang, Le Liang, Tianlin Li, Xiaojun Jia, Run Wang, Weikai Miao, Geguang Pu, and Yang Liu. Perception-guided jailbreak against text-to-image models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26238–26247, 2025.

[20] Yijun Yang, Ruiyuan Gao, Xiaosen Wang, Tsung-Yi Ho, Nan Xu, and Qiang Xu. Mma-diffusion: Multimodal attack on diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7737–7746, 2024.

[21] Pucheng Dang, Xing Hu, Dong Li, Rui Zhang, Qi Guo, and Kaidi Xu. Diffzoo: A purely query-based black-box attack for red-teaming text-to-image generative model via zeroth order optimization. *arXiv preprint arXiv:2408.11071*, 2024.

[22] Yimo Deng and Huangxun Chen. Divide-and-conquer attack: Harnessing the power of llm to bypass the censorship of text-to-image generation model. *arXiv preprint arXiv:2312.07130*, 2023.

[23] Yingkai Dong, Zheng Li, Xiangtao Meng, Ning Yu, and Shanqing Guo. Jailbreaking text-to-image models with llm-based agents. *arXiv preprint arXiv:2408.00523*, 2024.

[24] Yihao Huang, Le Liang, Tianlin Li, Xiaojun Jia, Run Wang, Weikai Miao, Geguang Pu, and Yang Liu. Perception-guided jailbreak against text-to-image models. *arXiv preprint arXiv:2408.10848*, 2024.

[25] Wenhao Wang and Yi Yang. Vidprom: A million-scale real prompt-gallery dataset for text-to-video diffusion models. *Advances in Neural Information Processing Systems*, 37:65618–65642, 2025.

[26] Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting. Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22522–22531, 2023.

[27] Yiting Qu, Xinyue Shen, Yixin Wu, Michael Backes, Savvas Zannettou, and Yang Zhang. Unsafebench: Benchmarking image safety classifiers on real-world and ai-generated images. *arXiv preprint arXiv:2405.03486*, 2024.

[28] Gate2ai. URL https://www.gate2ai.com/prompts-midjourney.

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

[30] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024.

[31] Patrick Schramowski, Christopher Tauchmann, and Kristian Kersting. Can machines help us answering question 16 in datasheets, and in turn reflecting on inappropriate content? In *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency*, pages 1350–1361, 2022.

[32] Falconsai. Nsfw image classification. `https://huggingface.co/Falconsai/nsfw_image_detection`, 2024. Accessed on: 2024-11-18.

[33] Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. Autodan: Generating stealthy jailbreak prompts on aligned large language models. *arXiv preprint arXiv:2310.04451*, 2023.

[34] Siyuan Liang, Jiayang Liu, Jiecheng Zhai, Tianmeng Fang, Rongcheng Tu, Aishan Liu, Xiaochun Cao, and Dacheng Tao. T2vshield: Model-agnostic jailbreak defense for text-to-video models. *arXiv preprint arXiv:2504.15512*, 2025.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction accurately reflect the paper's contributions and scope, effectively conveying the key points and supporting evidence.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper effectively discusses the limitations of the work done by the authors in the "Discussion and Conclusion" section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results, hence there are no assumptions or proofs to be provided.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper fully discloses all the information necessary to reproduce the main experimental results, ensuring transparency and replicability of the findings.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The paper provides open access to the data and code, the anonymous link is {https://anonymous.4open.science/r/NeruIPS_25_t2v-CE60}.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper specifies all the training and test details necessary to understand the results, such as hyperparameters, ensuring transparency and reproducibility of the experimental setup.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: As described in the Experimental Setup, we report the average of the results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: As described in the Supplemental Material, all experiments were conducted on a server equipped with an Intel(R) Xeon(R) Gold 6336Y CPU @ 2.40GHz, 512 GB of system memory, and one NVIDIA A100 GPU with 40 GB of memory.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research in the paper aligns with the NeurIPS Code of Ethics, ensuring adherence to ethical standards outlined by the NeurIPS community for responsible conduct in AI research.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: The paper discusses both potential positive societal impacts and negative societal impacts of the work performed in Supplemental Material, fulfilling the conference's expectations for addressing broader impacts and considering potential ethical implications.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no risks requiring specific safeguards for responsible data or model release, therefore safeguards are not applicable.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The creators or original owners of assets used in the paper are properly credited, and the license and terms of use are explicitly mentioned and respected, ensuring compliance with legal and ethical standards.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not introduce new assets, therefore documentation of assets is not applicable.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: Yes, our work involves human evaluation to judge the safety of generated videos. The volunteers were shown a set of videos and asked to evaluate them based on the input prompt and presence of jailbreak content. All participants received full instructions, including example screenshots and evaluation guidelines. The volunteers are at least 18 years old, in good physical and mental health, and free from conditions such as heart disease or vasovagal syncope.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [Yes]

Justification: Our study involves human evaluation conducted by members of our research lab. The task involves watching AI-generated videos and providing judgements based on predefined criteria. The evaluation does not involve any sensitive or personal information, and no foreseeable risks were posed to the participants. Therefore, IRB approval was not required under our institution's guidelines.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.