WE'LL FIX IT IN POST: IMPROVING TEXT-TO-VIDEO GENERATION WITH ZERO TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Current text-to-video (T2V) generation models are increasingly popular due to their ability to produce coherent videos from textual prompts. However, these models often struggle to generate semantically and temporally consistent videos when dealing with longer, more complex prompts involving multiple objects or sequential events. Additionally, the high computational costs associated with training or fine-tuning make direct improvements impractical. To overcome these limitations, we introduce *NeuS-E*, a novel zero-training video refinement pipeline that leverages neuro-symbolic feedback to automatically enhance video generation, achieving superior alignment with the prompts. Our approach first derives the neuro-symbolic feedback by analyzing a formal video representation and pinpoints semantically inconsistent events, objects, and their corresponding frames with respect to the prompt. This feedback then guides targeted edits to the original video. Extensive empirical evaluations on both open-source and proprietary T2V models demonstrate that *NeuS-E* significantly enhances temporal and logical alignment across diverse prompts by almost 40%.

1 INTRODUCTION

Imagine generating the following complex scenario from a text-to-video (T2V) model:

> *"An autonomous vehicle crosses an intersection* <u>*after*</u> *waiting for a pedestrian to cross."*

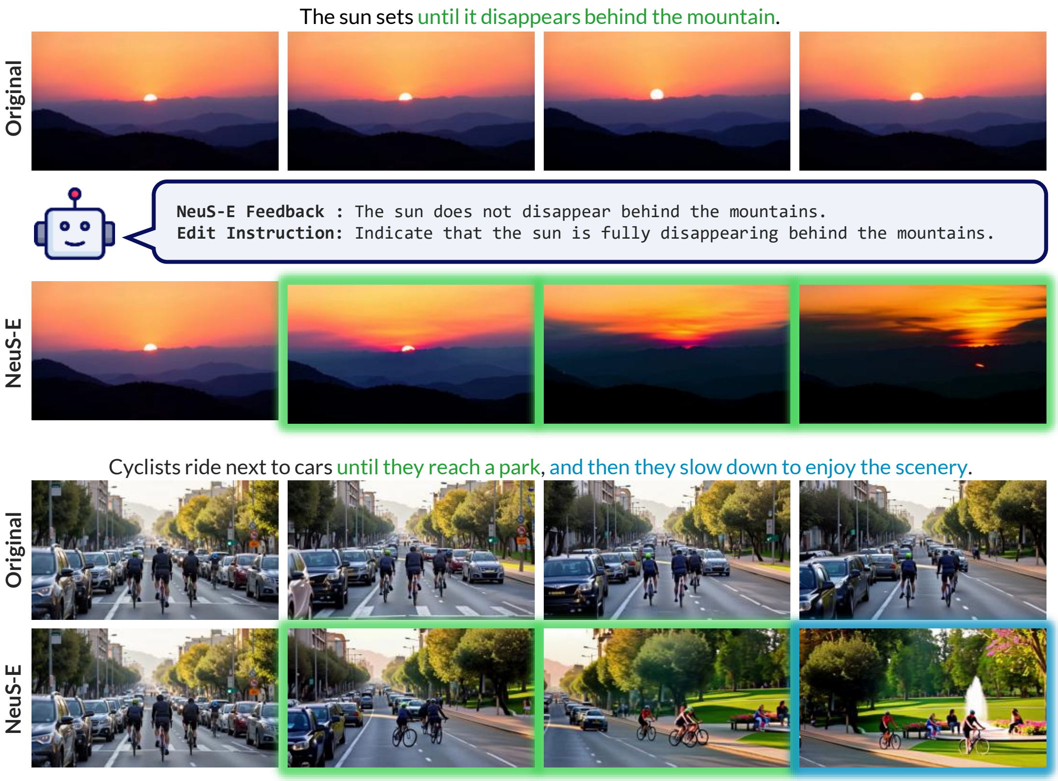
This involves three interdependent aspects: ❶ *semantic correctness* (the presence of objects such as the car and the pedestrian, and actions such as the pedestrian walking and the car stopping), ❷ *spatial coherence* (entities interact correctly in 3D space, for instance the car stops before the intersection when the pedestrian crosses), and ❸ *temporal consistency* (events occur in the correct order over time, for instance, the car stops when the pedestrian crosses and the car moves after the pedestrian is off the road).

Current research in T2V generation has primarily focused on the first two aspects. Existing methods improve visual quality and semantic accuracy by modifying rewards in diffusion models, tweaking attention maps, or making low-level architectural changes. Although these enhance visual fidelity, they fail to fix temporal misalignments and are often infeasible for models like Gen3 (Gen-3, 2024) and Pika (Pika Labs, 2023), which lack accessible model weights.

However, state-of-the-art T2V models fail catastrophically when prompted to generate events in a specific temporal order. Training-free methods that utilize excessive remarking to improve traditional T2V evaluation benchmarks, such as VBench (Huang et al., 2024), fail to improve temporal alignment since these metrics prioritize visual aesthetics. In contrast, *NeuS-V* (Sharan et al., 2025) introduces a neuro-symbolic method that rigorously quantifies a text-to-video model's temporal alignment with respect to the prompt. Taking inspiration from this work, we pose the following research question:

> *"Can we leverage neuro-symbolic feedback to surgically refine temporally misaligned video segments through targeted edits, thereby improving text-to-video temporal alignment?"*

We introduce *NeuS-E*, a zero-training framework for targeted video refinement. Inspired by *NeuS-V*, we devise a neuro-symbolic feedback loop that disentangles atomic events and objects (termed


The sun sets until it disappears behind the mountain.

NeuS-E Feedback : The sun does not disappear behind the mountains.
Edit Instruction: Indicate that the sun is fully disappearing behind the mountains.

Cyclists ride next to cars until they reach a park, and then they slow down to enjoy the scenery.

Figure 1: ***NeuS-E* improves the text-to-video (T2V) temporal alignment.** The border color of the frames corresponds to the identified events. Vanilla T2V models (top video) fail to generate the sunset behind the mountains. *NeuS-E* systematically identifies and surgically corrects this video segment to improve the temporal fidelity of the synthetic video with targeted feedback.

propositions) with their prompted temporal order (termed specifications). Therefore, instead of treating videos as static outputs, *NeuS-E* identifies video segments that weakly satisfy the objects and events in the prompt, edits the corresponding keyframes, and regenerates only these misaligned portions to improve temporal alignment with respect to the prompt. This iterative process produces a video that aligns with the prompt's temporal requirements without retraining the generative model. Our key contributions are summarized as follows:

- We develop a method to extract neuro-symbolic feedback, identifying weakly satisfied propositions and their corresponding problematic video segments.
- We introduce *NeuS-E*, which automatically corrects weak segments through neuro-symbolic feedback by identifying and editing the keyframe, then regenerating misaligned video segments to improve temporal alignment.
- We demonstrate that *NeuS-E* significantly boosts the temporal fidelity of both open-source (CogVideoX) and closed-source (Gen-3, Pika) models, all with zero additional training.

2 Related Work

Text-to-Video Generation. Text-to-video models such as SORA (OpenAI, 2024), GEN-3 Alpha (Research, 2024), Kling (Kuaishou, 2024), Veo (Sharma et al., 2024), Wan (Team, 2025), and PIKA (Labs, 2024) have seen significant advancements. The underlying research follows two main training strategies. The first involves end-to-end training of spatial and temporal modules using diffusion (Ho et al., 2020) or autoregressive (Hong et al., 2022; Yang et al., 2024) architectures, encompassing early latent-space works (Villegas et al., 2022; Zhang et al., 2023b; Esser et al., 2023) and numerous modern systems (Chen et al., 2023a; 2024a; Xing et al., 2024; Zhang et al., 2024b;c; Ho et al., 2022; He et al., 2023; Kong et al., 2025; Wang et al., 2024b; Team, 2025). The second strategy efficiently adapts pre-trained image models by training only a lightweight temporal module (Blattmann et al., 2023; Guo et al., 2023; Khachatryan et al., 2023). For a broader survey, see (Cho et al., 2024). In contrast to these methods, *NeuS-E* proposes a training-free approach to video generation, for temporally complex prompts.

Text-to-Video Refinement. Training-free frameworks improve text-to-video generation by leveraging text-to-image (T2I) models (Meng et al., 2021; Brooks et al., 2022; Zhang et al., 2023a) to edit video frames (Jeong & Ye, 2023; Zhang et al., 2024a; Khachatryan et al., 2023; Yang et al., 2023a; Wang et al., 2024a; Goel et al., 2024b;a). Most of these methods focus on improving the temporal coherence of individual objects by rectifying cross-frame attention maps and features (Geyer et al., 2023; Jeong & Ye, 2023; Qi et al., 2023; Yang et al., 2023a; Liu et al., 2023a), by refining text prompts (Kim et al., 2024; Luo et al., 2025), by editing keyframes (Ceylan et al., 2023; Zhang et al., 2024b), or employing video-to-video diffusion models (Molad et al., 2023). While effective for object-level consistency, these approaches fail to enforce the logical order of multiple, distinct events (e.g., ensuring a person enters a car before driving away). To bridge this critical gap, *NeuS-E* uses neuro-symbolic feedback to guide edits, ensuring the final video satisfies complex temporal relationships across the entire event sequence.

Evaluation. Current Text-to-Video (T2V) evaluation methods include distribution-based metrics (FID, FVD, CLIPSIM) (Shin et al., 2024; Liu et al., 2023b; Jain et al., 2024; Bugliarello et al., 2023; Hu et al., 2022; Yu et al., 2023), LLM-based VQA scoring (Kou et al., 2024; Liu et al., 2024b; Zhang et al., 2024d; Li et al., 2024), and metric ensembles for visual and spatial quality like VBench, EvalCrafter, FETV, and T2V-Bench (Huang et al., 2024; Liu et al., 2024a;c; Ji et al., 2024; Feng et al., 2024; Huang et al., 2023; Chu et al., 2024). While comprehensive, these approaches primarily assess aesthetic temporal quality, such as motion and flickering (Huang et al., 2024), rather than logical event sequencing. In contrast, *NeuS-E* leverages *NeuS-V* (Sharan et al., 2025) to address this gap, using Temporal Logic (TL) to formally verify complex temporal relationships.

Formal Verification. Formal verification methods are used to construct symbolic representations of events and tasks across various domains. These methods have been applied in video understanding (Feichtenhofer et al., 2019; Tran et al., 2019; Medioni et al., 2001; Xu et al., 2015; Li et al., 2022; Yi et al., 2018; Chen et al., 2022), robotics (Shoukry et al., 2017; Hasanbeig et al., 2019; Kress-Gazit et al., 2009), and autonomous driving (Jha et al., 2018; Mehdipour et al., 2023), using techniques like graph-based reasoning (Yu et al., 2022; Mavroudi et al., 2020; Xiong et al., 2019), latent-space abstractions (Sarkar et al., 2015; Bertasius et al., 2021; Kroshchanka et al., 2021), or formal languages (Baier & Katoen, 2008). In contrast, *NeuS-E* leverages formal verification to provide structured feedback—identifying text-based edit instructions and keyframes—to improve Text-to-Video generation without any training.

3 PRELIMINARIES

We begin with a comprehensive running example of generating a promotional video using the prompt: "A person meditates by the lake and, a few seconds later, stands up for a moment before leaving" to describe the following terminologies.

Temporal Logic. Temporal Logic (TL) is an expressive formal language that combines logical and temporal operators to express time-dependent statements (Emerson, 1991; Manna & Pnueli, 1992). A TL specification or formula is structured around three key components: ❶ a set of atomic propositions, ❷ first-order logic operators, and ❸ temporal operators. Atomic propositions are fundamental statements that evaluate to either `True` or `False` and serve as building blocks for more complex expressions. First-order logic operators include AND ($\wedge$), OR ($\vee$), NOT ($\neg$), IMPLY ($\Rightarrow$)., and the temporal operators consist of ALWAYS ($\Box$), EVENTUALLY ($\Diamond$), NEXT (X), UNTIL (U), etc. The set of atomic propositions $\mathcal{P}$, and TL specification Φ of our running example are:

$$\begin{aligned} \mathcal{P} &= \{\text{person is meditating, lake shore, person is standing, person is walking away}\}. \\ \Phi &= (\text{person is meditating} \wedge \text{lake shore})\,\mathsf{X}\,(\text{person is standing}\ \mathsf{X}\ \text{person is walking away}). \end{aligned} \tag{1}$$

Video Automaton. A video automaton formally represents a video sequence as a sequence of states and transitions (Choi et al., 2025; Sharan et al., 2025; Yang et al., 2023b). It is a form of

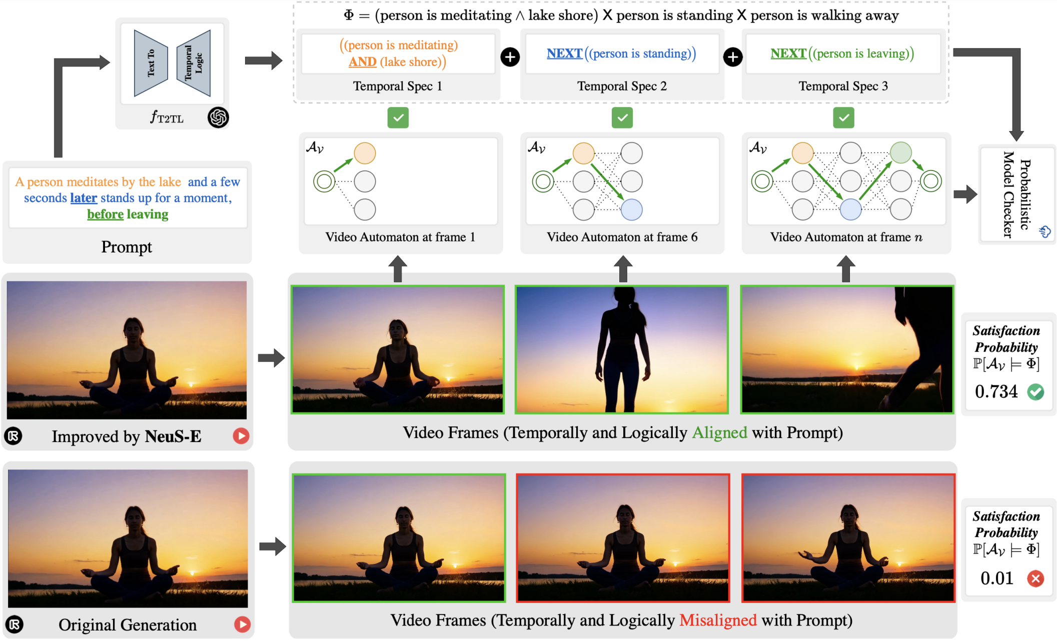

Φ = (person is meditating ∧ lake shore) X person is standing X person is walking away
((person is meditating) AND (lake shore))
Temporal Spec 1
NEXT((person is standing))
Temporal Spec 2
NEXT((person is leaving))
Temporal Spec 3
Text To
Temporal Logic
f_{T2TL}
A person meditates by the lake and a few seconds later stands up for a moment, before leaving
Prompt
$\mathcal{A}_\mathcal{V}$
Video Automaton at frame 1
Video Automaton at frame 6
Video Automaton at frame n
Probabilistic Model Checker
Improved by NeuS-E
Video Frames (Temporally and Logically Aligned with Prompt)
Satisfaction Probability
$\mathbb{P}[\mathcal{A}_\mathcal{V} \models \Phi]$
0.734
Original Generation
Video Frames (Temporally and Logically Misaligned with Prompt)
Satisfaction Probability
$\mathbb{P}[\mathcal{A}_\mathcal{V} \models \Phi]$
0.01

Figure 2: **Formally verify generated video with video Automaton**. The video automaton expands as new frames are added. Once fully constructed, we verify it against the TL specification. We have a very low probability of satisfaction from the initial generation, as the person neither stands nor walks away. To address this, we refine the video using *NeuS-E*, generating a better video that is temporally and logically aligned with the prompt and achieves a higher satisfaction probability.

Discrete-Time Markov Chain (DTMC) (Norris, 1998; Kemeny & Snell, 1960) used to model generated videos as each frame's transition depends only on the current frame. Since video sequences are inherently *discrete*, *finite*, and *temporal*, DTMCs can approximate frame transitions. To this end, we define the set of states in the video as Q and this frame transition function as $\delta : Q \times Q \to [0,1]$ where, given two states $q_1 \in Q$ and $q_2 \in Q$, $\delta(q_1, q_2) \in [0,1]$ gives the probability of transitioning from q_1 to q_2. Now, the video automaton, $\mathcal{A}_\mathcal{V}$, is the tuple $\mathcal{A}_\mathcal{V} = (Q, q_0, \delta, \lambda)$ where the initial state is $q_0 \in Q$ and the label function is $\lambda : Q \to 2^{|\mathcal{P}|}$.

Formal Verification. Formal verification ensures formal guarantees that the system satisfies the desired specification. (Clarke et al., 1999; Huth & Ryan, 2004). It necessitates a formal representation of the system, such as a finite-state automaton (FSA). Given the video automaton $\mathcal{A}_\mathcal{V}$, a *path* in a video is defined as a sequence of states $q_0 q_1 (q_2)^\omega$ starting from the initial state q_0 and ω denotes repetition. A trace corresponds to the sequence of labels $\lambda(q_0)\lambda(q_1)\lambda(q_2)\cdots \in (2^{|\mathcal{P}|})^\omega$ associated with the states along a path, where $\lambda(q)$ represents the labeling function that maps each state q to a subset of atomic propositions from the set $\mathcal{P}$. The trace captures the progression of events or properties over time as observed along the path. Next, we apply probabilistic model checking (Baier & Katoen, 2008) to determine how much the *trace* starting from the initial state satisfies the temporal logic (TL) specification Φ. Using these formal representations, we evaluate the videos.

Text to Temporal Logic. Text prompts are converted to temporal logic conversion using LLMs (Chen et al., 2023b; Cosler et al., 2023; Mendoza et al., 2024; Yang et al., 2023b; Sharan et al., 2025) for subsequent analysis. We denote the LLM-based text-to-temporal logic (T2TL) function as $f_{\text{T2TL}} : T \to (\mathcal{P}, \Phi)$ to decompose a text prompt T into a TL specification Φ and a set of propositions $\mathcal{P}$. A prompt example is presented in the Appendix.

Neuro-symbolic Video Verification. We uniquely adapt a VLM to construct the video automaton and formally verify it as a symbolic method. We calibrate the VLM model to map its naive confidence to the accuracy. We present a detailed explanation of the VLM calibration process and usage in the Appendix.

Definition 1 (Video Automaton Construction)**.** *Given a generated video $\mathcal{V}$ and a set of atomic propositions $\mathcal{P}$, we construct a video automaton. For each proposition $p_i \in \mathcal{P}$ and frame $\mathcal{F}_n \in \mathcal{V}$, a VLM computes a semantic confidence score: $\mathcal{M}_{VLM} : p_i \times \mathcal{F}_n \to c_{i,n}$, where $c_{i,n} \in [0,1]$ represents the confidence of p_i in $\mathcal{F}_n$. For each frame n, we define the confidence set as: $\mathbb{C} = \{c_{i,n} \mid p_i \in \mathcal{P}, \mathcal{F}_n \in \mathcal{V}\}$. The video automaton $\mathcal{A}_{\mathcal{V}}$ is then generated by applying a function ξ that processes the set of propositions $\mathcal{P}$ along with the confidence scores across all frames:*

$$\xi : \mathcal{P} \times \mathbb{C} \to \mathcal{A}_{\mathcal{V}}. \tag{2}$$

Definition 2 (Satisfaction Probability)**.** *Given a video automaton $\mathcal{A}_{\mathcal{V}}$ and a temporal logic specification Φ, the satisfaction probability $\mathbb{P}[\mathcal{A}_{\mathcal{V}} \models \Phi]$ is computed by verifying $\mathcal{A}_{\mathcal{V}}$ against Φ using STORM (Hensel et al., 2020; Junges & Volk, 2021), which uses probabilistic computation tree logic (PCTL). This verification process is formalized as a probabilistic model checking function:*

$$\Psi : \mathcal{A}_{\mathcal{V}} \times \Phi \to \mathbb{P}[\mathcal{A}_{\mathcal{V}} \models \Phi]. \tag{3}$$

We provide details on the video automaton generation function in the Appendix.

4 Methodology

Given a synthetic video $\mathcal{V}$ generated by T2V models $\mathcal{M}_{\text{T2V}} : T, I \to \mathcal{V}$, where T represents the input text prompt, and I is the *optional* input image, our objective is to refine $\mathcal{V}$ to improve its temporal and logical consistency with the intended semantics of T. We introduce *NeuS-E*, a neuro-symbolic framework that operates through the following iterative steps:

- **Step 1: Decompose & Represent.** The prompt T is decomposed into a TL specification, and a video automaton is constructed from $\mathcal{V}$ using a VLM.
- **Step 2: Identify Errors.** By analyzing the video automaton, the framework identifies weak propositions and their corresponding frames, pinpointing temporal misalignments.
- **Step 3: Refine & Iterate.** The identified errors are converted into keyframe editing instructions to guide a video editing pipeline. This process is repeated until the refined video meets a predefined coherence threshold.

4.1 Neuro-symbolic Video Verification.

First, we generate the video with the prompt without a input image: $\mathcal{V} = \mathcal{M}_{\text{T2V}}(T, \text{None})$. Then, we decompose T into the TL specification Φ and the set of propositions $\mathcal{P}$. Given Φ and $\mathcal{P}$, we construct the video automaton as described in Definition 1 and verify it according to Definition 2. Finally, we perform a comprehensive neuro-symbolic verification of the generated video. Further details are provided in Figure 2.

4.2 Identifying the Weakest Proposition.

In our running example (see Figure 2), the generated video fails to satisfy the full Temporal Logic (TL) specification—(A ∧ B) X C X D—because it omits the events corresponding to propositions C and D. To pinpoint which proposition is the weakest link, we systematically measure the impact of each one on the video's overall satisfaction probability, $\mathbb{P}[\mathcal{A}_{\mathcal{V}} \models \Phi]$.

Our method creates a hypothetical scenario for each proposition $\tilde{p}_i \in \mathcal{P}$ where we assume it is perfectly satisfied. Intuitively, we assess the video against partial specifications, such as $(B) \mathsf{X}\, C \,\mathsf{X}\, D$, $(A) \mathsf{X}\, C \,\mathsf{X}\, D$, $(A \wedge B) \mathsf{X}\, D$, and $(A \wedge B) \mathsf{X}\, C$, where each proposition's confidence score is systematically adjusted to find the weakest proposition. We achieve this by generating an adjusted confidence score set $\tilde{\mathbb{C}}$ that forces the confidence of the proposition under evaluation, $\tilde{p}_i$, to 1.0, while all other scores remain unchanged:

$$\tilde{\mathbb{C}} = \{\tilde{c}_{j,n} : \tilde{c}_{j,n} = 1.0 \text{ if } p_j = \tilde{p}_i; \quad \tilde{c}_{j,n} = c_{j,n} \text{ if } p_j \neq \tilde{p}_i,\ \forall p_j \in \mathcal{P}\}. \tag{4}$$

Using this adjusted set, we construct a new evaluating video automaton $\tilde{\mathcal{A}}_{\mathcal{V},i} = \xi(\mathcal{P}, \tilde{\mathbb{C}})$ and compute a new satisfaction probability $\mathbb{P}[\tilde{\mathcal{A}}_{\mathcal{V},i} \models \Phi] = \Psi(\tilde{\mathcal{A}}_{\mathcal{V},i}, \Phi)$. The proposition whose forced satisfaction causes the largest increase in this probability is identified as the weakest. We formalize this by finding the proposition p_i^* that maximizes the difference, δ_i:

$$p_i^* = \arg\max_{i \in |\mathcal{P}|} \{\delta_i \ : \ \delta_i = \mathbb{P}[\tilde{\mathcal{A}}_{\mathcal{V},i} \models \Phi] - \mathbb{P}[\mathcal{A}_{\mathcal{V}} \models \Phi]\}. \tag{5}$$

The proposition p_i^* with the highest δ_i value is the component most responsible for the video's failure to align with the prompt, allowing us to target it for refinement.

4.3 Localizing the Influence of the Weakest Proposition.

Given the weakest proposition p_i^*, we now localize its influence across $\mathcal{F}_n \in \mathcal{V}$ with respect to p_i^*. In our running example, after we identify the weakest proposition (*e.g.,* "person is standing") from the original generation, we determine the most impacted frames by the proposition, specifically the middle frame in the Figure 2.

For each frame $\mathcal{F}_n$, we construct the evaluating automaton $\tilde{\mathcal{A}}_{\mathcal{V},n}$ for the video segment from the initial frame upto $\mathcal{F}_n$. We construct a new confidence measure $\hat{c}_{i,n} = 1.0$ for the weakest proposition p_i^* at $\mathcal{F}_n$. For all other frames $\{\mathcal{F}_m \ : \ m \in [0, n-1]\}$ and propositions $p_i \in \mathcal{P}$ including p_i^*, we set $\hat{c}_{i,n} = c_{i,n} + \gamma$, where γ introduces controlled noise to enhance numerical stability in the model-checking process. Subsequently, we define the set of per-frame satisfaction probabilities $\mathbb{Z} = \{\mathbb{P}[\tilde{\mathcal{A}}_{\mathcal{V},n} \models \Phi] = \Psi(\tilde{\mathcal{A}}_{\mathcal{V},n}, \Phi) \ \mid \mathcal{F}_n \in \mathcal{V}\}$. Finally, we obtain the most impacted frame $\mathcal{F}_n^*$ by $\tilde{p}_i$ as follows:

$$\mathcal{F}_n^* = \arg\max_{n \in \{1,2,\ldots,N\}} \{z_n \in \mathbb{Z} \mid z_n = \mathbb{P}[\tilde{\mathcal{A}}_{\mathcal{V},n} \models \Phi]\}, \tag{6}$$

where N is the total number of frames in $\mathcal{V}$, and $\mathcal{F}_n^*$ denotes the frame at which enforcing p_i^*'s certainty yields the greatest satisfaction probability, thus identifying the critical time-step most sensitive to p_i^*'s variability.

4.4 Video Refinement Using Neuro-Symbolic Feedback

We refine the video iteratively to enhance the temporal consistency of the generated video $\mathcal{V}$ through the above-mentioned neuro-symbolic feedback, which includes the baseline satisfaction probability $\mathbb{P}[\mathcal{A}_{\mathcal{V}} \models \Phi]$, the weakest proposition p_i^*, and the most impacted frame $\mathcal{F}_n^*$,

Video trimming. We index the frame $\mathcal{F}_n^*$ in video $\mathcal{V}$ and trim the video to this index to obtain a segment to $\mathcal{F}_n^*$ as: $\mathcal{V}_{\text{trimmed}} = \mathcal{V}[0 : n^*]$, where n^* is the index of $\mathcal{F}_n^*$. This isolates the portion of the video most affected by p_i^*, preparing it for targeted regeneration.

New Video Segment Generation. We use an LLM to generate a new prompt T_{new} to generate the next video segment after $\mathcal{V}_{\text{trimmed}}$ by providing the weak proposition p_i^* along with the original prompt T_{video}. We present all prompts in the Appendix.

Iterative Video Generation and Edition. First, we merge the new video segment $\mathcal{V}_{\text{new}}$ with $\mathcal{V}_{\text{trimmed}}$, and repeat the neuro-symbolic feedback process as above to obtain the weakest proposition and the most impacted frame. This process is iterated until $\mathbb{P}[\mathcal{A}_{\mathcal{V}} \models \Phi]$ surpasses a predefined threshold or the maximum iteration limit is reached. We present the algorithm in the Appendix.

5 Experimental Setup

Following *NeuS-V*, we use GPT-4o (OpenAI, 2023) to decompose prompts into constituent propositions and generate temporal logic specifications. InternVL2.5-8B (Chen et al., 2024b) serves as the VLM for obtaining per-frame confidence scores during automata construction. For further details on this process, we refer readers to *NeuS-V* (Sharan et al., 2025).

Once our method identifies the weakest proposition and the most impacted frame, we use GPT-4o to generate the video continuation prompt. Then, the video is regenerated using the continuation prompt. This process iterates until either (1) the *NeuS-V* score surpasses 0.7, or (2) the number of iterations matches the number of propositions—ensuring that each weak proposition undergoes at

Prompts		Gen-3		Pika-2.2		CogVideoX-5B	
		Original	Edited	Original	Edited	Original	Edited
By Theme	Nature	0.581	0.677 (+0.096)	0.579	0.856 (+0.277)	0.481	0.623 (+0.142)
	Human & Animal Activities	0.613	0.767 (+0.153)	0.638	0.872 (+0.235)	0.493	0.596 (+0.103)
	Object Interactions	0.610	0.721 (+0.111)	0.420	0.707 (+0.287)	0.454	0.610 (+0.156)
	Driving Data	0.546	0.611 (+0.065)	0.676	0.810 (+0.134)	0.565	0.681 (+0.116)
By Complexity	Basic (1 TL op.)	0.723	0.781 (+0.059)	0.694	0.840 (+0.146)	0.621	0.738 (+0.117)
	Intermediate (2 TL ops.)	0.480	0.633 (+0.153)	0.480	0.795 (+0.315)	0.387	0.540 (+0.153)
	Advanced (3 TL ops.)	0.370	0.527 (+0.157)	0.373	0.729 (+0.356)	0.344	0.449 (+0.105)
Overall Score		0.587	0.694 (+0.107)	0.577	0.811 (+0.233)	0.499	0.628 (+0.129)

Table 1: ***NeuS-V* Score Improvements.** Comparison of original and edited *NeuS-V* scores across different themes and complexity levels for Gen3, Pika, and CogVideoX models.

least one round of refinement. Finally, we use MoviePy (Zulko et al., 2015) for video trimming and stitching to integrate the edited video clips into a final MP4 (see Figure 1).

T2V Models. We evaluate our method on both closed-source and open-source T2V models to demonstrate its broad applicability. Specifically, we conduct experiments on Gen-3 (Research, 2024) by RunwayML and Pika-2.2 (Labs, 2024) by PikaArt as representatives of closed-source models, and CogVideoX-5B (Yang et al., 2024) as an open-source model. While our approach is designed to be *model-agnostic* and can be applied to *any T2V system*, we select these models due to their widespread use in the community and their ability to accept image inputs.

Dataset and Metrics. We use the *NeuS-V*(Sharan et al., 2025) prompt suite, which contains temporally extended prompts that challenge T2V models. It spans four themes (Nature, Human & Animal Activities, Object Interactions, Driving Data) and three complexity levels (Basic, Intermediate, Advanced). We evaluate both original and edited videos using the *NeuS-V* score to measure temporal fidelity, and VBench(Huang et al., 2024), a widely used metric for synthetic video quality.

6 Results

Building on our experimental setup, we now evaluate *NeuS-E* through empirical analysis. Our experiments aim to answer three key questions that highlight the necessity of our approach:

1. Does *NeuS-E* boost *alignment* without sacrificing *aesthetics* on temporally extended prompts?
2. How does iterative refinement through neuro-symbolic feedback compare to a sequential generation, where complex prompts are carefully broken down and generated sequentially?
3. Does each iteration meaningfully refine the video, progressively improving temporal fidelity?

6.1 Improving Text-to-Video Generation with Neuro-Symbolic Feedback

As outlined earlier, we benchmark state-of-the-art T2V models, including Gen-3 and Pika as closed-source models, and CogVideoX-5B (Yang et al., 2024) as an open-source alternative.

Benchmarking *NeuS-E* Refined Videos. We evaluate the before-and-after performance on the *NeuS-V* prompt suite in Table 1, measuring improvements across different themes and complexity levels. By structuring results this way, we gain deeper insights into where and how *NeuS-E* enhances temporal alignment. At a glance, *NeuS-E* consistently improves text-to-video alignment across all themes and complexities. Notably, the improvement is more pronounced for higher-complexity prompts, where regular generation with T2V models struggle the most. This suggests that while baseline models perform poorly on complex temporal relationships, our refinement method helps retain high temporal fidelity even as prompt difficulty increases. Additionally, we observe that Pika-2.2 excels at following video editing instructions, achieving an over 40% improvement in *NeuS-V* scores—outperforming other models in leveraging keyframe-based refinements.

Do humans agree? To further validate the effectiveness of our approach, we conducted a fully-blind, randomized A/B human evaluation. For each prompt, annotators were shown two videos

Strategy	*NeuS-V*		VBench		Length
	Original	Edited	Original	Edited	
Neuro-Symbolic	0.577	0.811 (+0.233)	0.789	0.772 (−0.017)	14.7 sec
Step-by-Step	0.577	0.612 (+0.035)	0.789	0.784 (+0.005)	11.2 sec

Table 2: **Ablation Study on Refinement Strategies.** Comparison of *NeuS-V* and VBench scores for Pika-2.2 using neuro-symbolic feedback versus step-by-step prompting. Neuro-symbolic feedback is key for video refinement.

(original vs. edited) in shuffled order and asked to judge which one better aligned with the caption on a five-point scale: Strongly Agree, Agree, Neutral, Disagree, Strongly Disagree. As shown in Figure 3, our edits are preferred in 52% of trials, closely mirroring the trend observed in our benchmark evaluations. Notably, Pika-2.2 shows the largest improvement, with nearly half of its videos rated as better after refinement. Other models also exhibit consistent gains, and importantly, the number of videos rated as worse remains low across all models. A substantial portion of neutral responses typically arises from (1) videos that were already well-aligned, making edits unnecessary, or (2) cases where T2V models failed to reliably follow regeneration instructions despite multiple iterations. The latter further highlights an inherent limitation of current T2V models in generating specific complex instructions.

6.2 Ablation on Refinement Strategy – Is Neuro-Symbolic Feedback Key?

One could argue that our method is simply re-prompting a T2V model until the desired output is achieved. However, existing T2V models struggle to generate temporally coherent videos, even with repeated prompting. To determine whether our approach provides a meaningful advantage, we compare it against an alternative refinement strategy. Beyond standard re-prompting, we introduce a step-by-step generation baseline. Here, we leverage our proposition decomposition algorithm but *without* neuro-symbolic feedback. Instead of refining weak segments, this approach extends a video iteratively, generating each event sequentially by using the last frame of one generation as the first frame for the next.

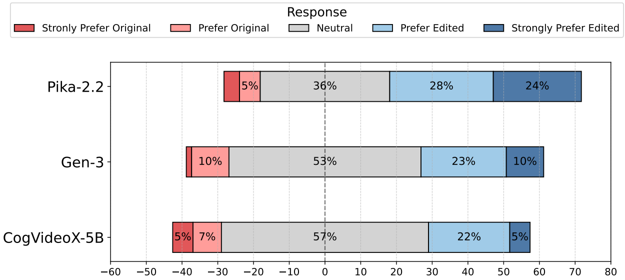
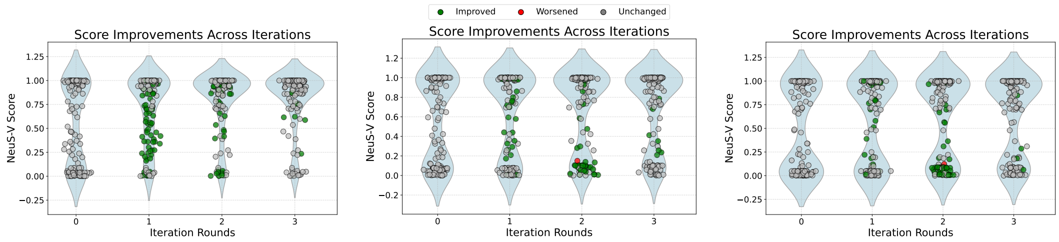


Figure 3: **Human Evaluation on Video Editing.** Diverging bar chart of human preference labels on the dataset shows that our editing pipeline improves temporal fidelity.

Our results (Table 2) show that while step-by-step generation does improve temporal coherence, it falls significantly short of what neuro-symbolic feedback achieves. The key advantage of *NeuS-E* lies in its ability to extract two critical insights: (1) identifying weak propositions in the video and (2) determining the optimal time for those propositions to occur to best satisfy the temporal logic (TL) specification. Formal verification adds rigor to our approach. Additionally, we observe that step-by-step generation leads to longer videos, suggesting that without targeted neuro-symbolic feedback, the model produces redundant content rather than precisely correcting temporal inconsistencies.

6.3 Effect of Iterative Refinement

In our experiments, we perform three rounds of refinement, analyzing how each iteration contributes to improving temporal alignment. Figure 4 visualizes these improvements. We observe that the violin plot widens progressively with each iteration, indicating increased variation in *NeuS-V* scores after editing. Our results show that each round of refinement provides measurable improvements, but gains plateau around the third iteration. This suggests that while early rounds effectively correct temporal inconsistencies, further refinements yield diminishing returns. Additionally, we find that CogVideoX-5B responds poorly to *NeuS-E*'s edit instructions, leading to minimal improvement compared to other models, whereas Pika-2.2 responds well.



(a) Pika-2.2

(b) Gen-3

(c) CogVideoX-5B

Figure 4: **Improvements from Iterative Rounds of Refinement.** Distribution of *NeuS-V* score changes with a violin plot overlay. Green/red points indicate improvements/degradation per sample.

7 Discussion

Key Frame Editing	*NeuS-V*		VBench	
	Original	Edited	Original	Edited
✗	0.577	0.811 (+0.233)	0.789	0.772 (−0.017)
✓	0.577	0.835 (+0.258)	0.789	0.683 (−0.106)

Table 3: **Ablation Study on Key Frame Editing.** Comparison of *NeuS-V* and VBench scores for Pika-2.2 with and without key frame editing. Since VBench scores are highly affected by key frame modifications, this is not our default architectural choice.

On Scope and Positioning. Our work does not aim to "solve" T2V; instead, we pursue a narrow but high-impact goal: correcting the under-addressed failure mode of misordered or missing events in today's T2V models, while tolerating a small, measured trade-off in aesthetic quality. Empirically, *NeuS-E* improves temporal-logic recall by $+23.3$ points with only a -1.7 point drop on VBench. Our contribution goes beyond existing methods that assess only holistic temporal coherence (Sharan et al., 2025): *NeuS-E* can pinpoint the weakest proposition in a temporal specification and localize its failure in time, revealing precisely *where* and *why* a video misaligns with the prompt. Moreover, because NeuS-E is a training-free, model-agnostic pipeline, it can be applied to *any* black-box T2V generator through targeted segment-level edits, making it broadly practical in today's landscape dominated by closed-source and proprietary video models.

Limitations. While *NeuS-E* consistently improves temporal fidelity, its effectiveness is constrained by the maturity of today's generation backbones. In particular, subject inconsistencies, flickering artifacts, implausible physics, and failures on abstract or highly stylized prompts expose the limits of current image and video generation models rather than our pipeline. Our ablation on keyframe editing with OmniGen (Table 3) illustrates this trade-off: while incorporating keyframe edits further boosts *NeuS-V* scores ($+0.258$), it also introduces a larger drop in VBench (-0.106), highlighting how *aggressive edits can destabilize visual quality*. Hence, we default to the non-keyframe-editing variant, which strikes a better balance between temporal alignment and aesthetics.

The key takeaway is that zero-training, neuro-symbolic feedback provides a principled and general mechanism for improving temporal alignment. As the underlying generation models advance, our approach will remain compatible and continue to enhance them, ensuring that temporal fidelity scales alongside improvements in visual quality.

8 Conclusion

To conclude, we propose *NeuS-E*, a training-free method to improve the generation of videos from the existing T2V model for temporally complex prompts. *NeuS-E* utilizes neuro-symbolic feedback to identify semantic discrepancies to effectively guide localized video editing. Our empirical evaluations of existing benchmarks and human evaluations demonstrate that *NeuS-E* significantly enhances temporal fidelity across diverse thematic and complexity-based categories. While *NeuS-E* attempts to bridge the gap between textual prompt complexity and video synthesis to enable temporally aligned video generation, *NeuS-E* is still limited by current T2V capabilities, and therefore, fails to meaningfully improve the video on the VBench score. We hope this work inspires further exploration into neuro-symbolic methods for advancing generative AI for long video generation.

REFERENCES

Meta AI. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models. Meta AI Blog, 2024. Available at https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/.

Christel Baier and Joost-Pieter Katoen. Principles of Model Checking. The MIT Press, 2008.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In Marina Meila and Tong Zhang (eds.), International Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research, pp. 813–824. PMLR, 2021.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 22563–22575, 2023.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 18392–18402, 2022. URL https://api.semanticscholar.org/CorpusID:253581213.

Emanuele Bugliarello, Hernan Moraldo, Ruben Villegas, Mohammad Babaeizadeh, Mohammad Taghi Saffar, Han Zhang, Dumitru Erhan, Vittorio Ferrari, Pieter-Jan Kindermans, and Paul Voigtlaender. Storybench: A multifaceted benchmark for continuous story visualization, 2023. URL https://arxiv.org/abs/2308.11606.

Duygu Ceylan, Chun-Hao Paul Huang, and Niloy Jyoti Mitra. Pix2video: Video editing using image diffusion. 2023 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 23149–23160, 2023. URL https://api.semanticscholar.org/CorpusID:257663916.

Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter1: Open diffusion models for high-quality video generation, 2023a. URL https://arxiv.org/abs/2310.19512.

Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7310–7320, 2024a.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. arXiv preprint arXiv:1504.00325, 2015.

Yongchao Chen, Rujul Gandhi, Yang Zhang, and Chuchu Fan. NL2TL: Transforming natural languages to temporal logics using large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 15880–15903, Singapore, December 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.985. URL https://aclanthology.org/2023.emnlp-main.985/.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. arXiv preprint arXiv:2412.05271, 2024b.

Zhenfang Chen, Kexin Yi, Yunzhu Li, Mingyu Ding, Antonio Torralba, Joshua B Tenenbaum, and Chuang Gan. Comphy: Compositional physical reasoning of objects and events from videos. arXiv preprint arXiv:2205.01089, 2022.

Joseph Cho, Fachrina Dewi Puspitasari, Sheng Zheng, Jingyao Zheng, Lik-Hang Lee, Tae-Ho Kim, Choong Seon Hong, and Chaoning Zhang. Sora as an agi world model? a complete survey on text-to-video generation. arXiv preprint arXiv:2403.05131, 2024.

Minkyu Choi, Harsh Goel, Mohammad Omama, Yunhao Yang, Sahil Shah, and Sandeep Chinchali. Towards neuro-symbolic video understanding. In European Conference on Computer Vision, pp. 220–236. Springer, 2025.

Zhixuan Chu, Lei Zhang, Yichen Sun, Siqiao Xue, Zhibo Wang, Zhan Qin, and Kui Ren. Sora detector: A unified hallucination detection for large text-to-video models. arXiv preprint arXiv:2405.04180, 2024.

Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. Model Checking. MIT Press, 1999.

Matthias Cosler, Christopher Hahn, Daniel Mendoza, Frederik Schmitt, and Caroline Trippel. nl2spec: Interactively translating unstructured natural language to temporal logics with large language models, 2023. URL https://arxiv.org/abs/2303.04864.

E. Allen Emerson. Temporal and modal logic. In Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics, 1991. URL https://api.semanticscholar.org/CorpusID:6062082.

Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 7346–7356, 2023.

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In IEEE/CVF International Conference on Computer Vision, pp. 6201–6210. IEEE, 2019.

Weixi Feng, Jiachen Li, Michael Saxon, Tsu-jui Fu, Wenhu Chen, and William Yang Wang. Tc-bench: Benchmarking temporal compositionality in text-to-video and image-to-video generation. arXiv preprint arXiv:2406.08656, 2024.

Gen-3. Gen-3. 2024. URL https://runwayml.com/blog/introducing-gen-3-alpha/.

Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. ArXiv, abs/2307.10373, 2023. URL https://api.semanticscholar.org/CorpusID:259991741.

Harsh Goel, Sai Shankar Narasimhan, Oguzhan Akcin, and Sandeep Chinchali. Syndiff-ad: Improving semantic segmentation and end-to-end autonomous driving with synthetic data from latent diffusion models. arXiv preprint arXiv:2411.16776, 2024a.

Harsh Goel, Sai Shankar Narasimhan, and Sandeep P. Chinchali. Improving end-to-end autonomous driving with synthetic data from latent diffusion models. In First Vision and Language for Autonomous Driving and Robotics Workshop, 2024b. URL https://openreview.net/forum?id=yaXYQinjOA.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Y. Qiao, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. ArXiv, abs/2307.04725, 2023. URL https://api.semanticscholar.org/CorpusID:259501509.

Mohammadhosein Hasanbeig, Yiannis Kantaros, Alessandro Abate, Daniel Kroening, George J Pappas, and Insup Lee. Reinforcement learning for temporal logic control synthesis with probabilistic satisfaction guarantees. In 2019 IEEE 58th conference on decision and control (CDC), pp. 5338–5343. IEEE, 2019.

Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation, 2023. URL https://arxiv.org/abs/2211.13221.

Christian Hensel, Sebastian Junges, Joost-Pieter Katoen, Tim Quatmann, and Matthias Volk. The probabilistic model checker storm. CoRR, abs/2002.07080, 2020. URL https://arxiv.org/abs/2002.07080.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. Advances in neural information processing systems, 33:6840–6851, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. ArXiv, abs/2204.03458, 2022. URL https://api.semanticscholar.org/CorpusID:248006185.

Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. arXiv preprint arXiv:2205.15868, 2022.

Yaosi Hu, Chong Luo, and Zhenzhong Chen. Make it move: Controllable image-to-video generation with text descriptions, 2022. URL https://arxiv.org/abs/2112.02815.

Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. Advances in Neural Information Processing Systems, 36:78723–78747, 2023.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 21807–21818, 2024.

Michael Huth and Mark Ryan. Logic in Computer Science: Modelling and Reasoning about Systems. Cambridge University Press, 2004.

Yash Jain, Anshul Nasery, Vibhav Vineet, and Harkirat Behl. Peekaboo: Interactive video generation via masked-diffusion, 2024. URL https://arxiv.org/abs/2312.07509.

Hyeonho Jeong and Jong Chul Ye. Ground-a-video: Zero-shot grounded video editing using text-to-image diffusion models. ArXiv, abs/2310.01107, 2023. URL https://api.semanticscholar.org/CorpusID:263605399.

Susmit Jha, Vasumathi Raman, Dorsa Sadigh, and Sanjit A Seshia. Safe autonomy under perception uncertainty using chance-constrained temporal logic. Journal of Automated Reasoning, 60:43–62, 2018.

Pengliang Ji, Chuyang Xiao, Huilin Tai, and Mingxiao Huo. T2vbench: Benchmarking temporal dynamics for text-to-video generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5325–5335, 2024.

Sebastian Junges and Matthias Volk. Stormpy - python bindings for storm, 2021. URL github.com/moves-rwth/stormpy.

J.G. Kemeny and J.L. Snell. Finite Markov Chains. Springer, 1960.

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, 2023. URL https://arxiv.org/abs/2303.13439.

Jaemin Kim, Bryan S Kim, and Jong Chul Ye. Free^{2}guide: Gradient-free path integral control for enhancing text-to-video generation with large vision-language models, 2024. URL https://arxiv.org/abs/2411.17041.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Dax Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. Hunyuanvideo: A systematic framework for large video generative models, 2025. URL https://arxiv.org/abs/2412.03603.

Tengchuan Kou, Xiaohong Liu, Zicheng Zhang, Chunyi Li, Haoning Wu, Xiongkuo Min, Guangtao Zhai, and Ning Liu. Subjective-aligned dataset and metric for text-to-video quality assessment, 2024. URL https://arxiv.org/abs/2403.11956.

Hadas Kress-Gazit, Georgios E Fainekos, and George J Pappas. Temporal-logic-based reactive mission and motion planning. IEEE transactions on robotics, 25(6):1370–1381, 2009.

Aliaksandr Kroshchanka, Vladimir Golovko, Egor Mikhno, Mikhail Kovalev, Vadim Zahariev, and Aleksandr Zagorskij. A neural-symbolic approach to computer vision. In International Conference on Open Semantic Technologies for Intelligent Systems, pp. 282–309. Springer, 2021.

Kuaishou. Kling. 2024. URL https://kling.kuaishou.com/en.

Pika Labs. Pika ai: Free video generator with scene ingredients, 2024. URL https://pikartai.com. Pika 2.1 documentation.

Baiqi Li, Zhiqiu Lin, Deepak Pathak, Jiayao Li, Yixin Fei, Kewen Wu, Xide Xia, Pengchuan Zhang, Graham Neubig, and Deva Ramanan. Evaluating and improving compositional text-to-visual generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 5290–5301, June 2024.

Nanjun Li, Faliang Chang, and Chunsheng Liu. Human-related anomalous event detection via spatial-temporal graph convolutional autoencoder with embedded long short-term memory network. Neurocomputing, 490:482–494, 2022.

Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-p2p: Video editing with cross-attention control. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8599–8608, 2023a. URL https://api.semanticscholar.org/CorpusID:257405406.

Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, and Ying Shan. Evalcrafter: Benchmarking and evaluating large video generation models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 22139–22149, 2024a.

Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, and Ying Shan. Evalcrafter: Benchmarking and evaluating large video generation models, 2024b. URL https://arxiv.org/abs/2310.11440.

Yuanxin Liu, Lei Li, Shuhuai Ren, Rundong Gao, Shicheng Li, Sishuo Chen, Xu Sun, and Lu Hou. Fetv: A benchmark for fine-grained evaluation of open-domain text-to-video generation, 2023b. URL https://arxiv.org/abs/2311.01813.

Yuanxin Liu, Lei Li, Shuhuai Ren, Rundong Gao, Shicheng Li, Sishuo Chen, Xu Sun, and Lu Hou. Fetv: A benchmark for fine-grained evaluation of open-domain text-to-video generation. Advances in Neural Information Processing Systems, 36, 2024c.

Yang Luo, Xuanlei Zhao, Mengzhao Chen, Kaipeng Zhang, Wenqi Shao, Kai Wang, Zhangyang Wang, and Yang You. Enhance-a-video: Better generated video for free, 2025. URL https://arxiv.org/abs/2502.07508.

Zohar Manna and Amir Pnueli. The Temporal Logic of Reactive and Concurrent Systems: Specification. Springer-Verlag, 1992.

Effrosyni Mavroudi, Benjamín Béjar Haro, and René Vidal. Representation learning on visual-symbolic graphs for video understanding. In European Conference on Computer Vision, volume 12374 of Lecture Notes in Computer Science, pp. 71–90. Springer, 2020.

Gérard G. Medioni, Isaac Cohen, François Brémond, Somboon Hongeng, and Ramakant Nevatia. Event detection and analysis from video streams. IEEE Trans. Pattern Anal. Mach. Intell., 23(8): 873–889, 2001.

Noushin Mehdipour, Matthias Althoff, Radboud Duintjer Tebbens, and Calin Belta. Formal methods to comply with rules of the road in autonomous driving: State of the art and grand challenges. Automatica, 152:110692, 2023.

Daniel Mendoza, Christopher Hahn, and Caroline Trippel. Translating natural language to temporal logics with large language models and model checkers. In Formal Methods in Computer-Aided Design (FMCAD), 2024. URL https://cs.stanford.edu/~trippel/pubs/mendoza_FMCAD24.pdf.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In International Conference on Learning Representations, 2021. URL https://api.semanticscholar.org/CorpusID:245704504.

Eyal Molad, Eliahu Horwitz, Dani Valevski, Alex Rav Acha, Yossi Matias, Yael Pritch, Yaniv Leviathan, and Yedid Hoshen. Dreamix: Video diffusion models are general video editors, 2023. URL https://arxiv.org/abs/2302.01329.

J.R. Norris. Markov Chains. Cambridge University Press, 1998.

OpenAI. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023.

OpenAI. Video generation models as world simulators, 2024. URL https://openai.com/sora/. Sora technical report.

OpenGVLab. Internvl 2.0: A suite of multimodal large language models for vision and language tasks. OpenGVLab Blog, 2024. Available at https://internvl.github.io/blog/2024-07-02-InternVL-2.0/.

Pika Labs. Accessed september 25, 2023, 2023. URL https://www.pika.art/.

Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. Fatezero: Fusing attentions for zero-shot text-based video editing. 2023 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 15886–15896, 2023. URL https://api.semanticscholar.org/CorpusID:257557738.

Runway Research. Introducing gen-3 alpha: A new frontier for video generation, 2024. URL https://runwayml.com/research/introducing-gen-3-alpha. Runway Gen-3 technical report.

Soumalya Sarkar, Kin Gwn Lore, and Soumik Sarkar. Early detection of combustion instability by neural-symbolic analysis on hi-speed video. In Proceedings of the NIPS Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches co-located with the 29th Annual Conference on Neural Information Processing Systems, volume 1583 of CEUR Workshop Proceedings, Montreal, Canada, 2015. CEUR-WS.org.

S P Sharan, Minkyu Choi, Sahil Shah, Harsh Goel, Mohammad Omama, and Sandeep P. Chinchali. Neuro-symbolic evaluation of text-to-video models using formal verification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2025.

Abhishek Sharma, Adams Yu, Ali Razavi, Andeep Toor, Andrew Pierson, Ankush Gupta, Austin Waters, Aäron van den Oord, Daniel Tanis, Dumitru Erhan, Eric Lau, Eleni Shaw, Gabe Barth-Maron, Greg Shaw, Han Zhang, Henna Nandwani, Hernan Moraldo, Hyunjik Kim, Irina Blok, Jakob Bauer, Jeff Donahue, Junyoung Chung, Kory Mathewson, Kurtis David, Lasse Espeholt, Marc van Zee, Matt McGill, Medhini Narasimhan, Miaosen Wang, Mikołaj Bińkowski, Mohammad Babaeizadeh, Mohammad Taghi Saffar, Nando de Freitas, Nick Pezzotti, Pieter-Jan Kindermans, Poorva Rane, Rachel Hornung, Robert Riachi, Ruben Villegas, Rui Qian, Sander Dieleman, Serena Zhang, Serkan Cabi, Shixin Luo, Shlomi Fruchter, Signe Nørly, Srivatsan Srinivasan, Tobias Pfaff, Tom Hume, Vikas Verma, Weizhe Hua, William Zhu, Xinchen Yan, Xinyu Wang, Yelin Kim, Yuqing Du, and Yutian Chen. Veo. Placeholder Journal, 2024. URL https://deepmind.google/technologies/veo/.

Andrew Shin, Yusuke Mori, and Kunitake Kaneko. The lost melody: Empirical observations on text-to-video generation from a storytelling perspective, 2024. URL https://arxiv.org/abs/2405.08720.

Yasser Shoukry, Pierluigi Nuzzo, Ayca Balkan, Indranil Saha, Alberto L Sangiovanni-Vincentelli, Sanjit A Seshia, George J Pappas, and Paulo Tabuada. Linear temporal logic motion planning for teams of underactuated robots using satisfiability modulo convex programming. In 2017 IEEE 56th annual conference on decision and control (CDC), pp. 1132–1137. IEEE, 2017.

Wan Team. Wan: Open and advanced large-scale video generative models. 2025.

Du Tran, Heng Wang, Matt Feiszli, and Lorenzo Torresani. Video classification with channel-separated convolutional networks. In IEEE/CVF International Conference on Computer Vision, pp. 5551–5560. IEEE, 2019.

Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. arXiv preprint arXiv:2210.02399, 2022.

Wen Wang, Yan Jiang, Kangyang Xie, Zide Liu, Hao Chen, Yue Cao, Xinlong Wang, and Chunhua Shen. Zero-shot video editing using off-the-shelf image diffusion models, 2024a. URL https://arxiv.org/abs/2303.17599.

Wenjing Wang, Huan Yang, Zixi Tuo, Huiguo He, Junchen Zhu, Jianlong Fu, and Jiaying Liu. Videofactory: Swap attention in spatiotemporal diffusions for text-to-video generation, 2024b. URL https://openreview.net/forum?id=dUDwK38MVC.

Jinbo Xing, Menghan Xia, Yuxin Liu, Yuechen Zhang, Yong Zhang, Yingqing He, Hanyuan Liu, Haoxin Chen, Xiaodong Cun, Xintao Wang, et al. Make-your-video: Customized video generation using textual and structural guidance. IEEE Transactions on Visualization and Computer Graphics, 2024.

Yu Xiong, Qingqiu Huang, Lingfeng Guo, Hang Zhou, Bolei Zhou, and Dahua Lin. A graph-based framework to bridge movies and synopses. In IEEE/CVF International Conference on Computer Vision, pp. 4591–4600. IEEE, 2019.

Zhongwen Xu, Yi Yang, and Alexander G. Hauptmann. A discriminative CNN video representation for event detection. In IEEE Conference on Computer Vision and Pattern Recognition, pp. 1798–1807, Boston, MA, USA, 2015. IEEE Computer Society.

Shuai Yang, Yifan Zhou, Ziwei Liu, and Chen Change Loy. Rerender a video: Zero-shot text-guided video-to-video translation. SIGGRAPH Asia 2023 Conference Papers, 2023a. URL https://api.semanticscholar.org/CorpusID:259144797.

Yunhao Yang, Jean-Raphaël Gaglione, Sandeep Chinchali, and Ufuk Topcu. Specification-driven video search via foundation models and formal verification. arXiv preprint arXiv:2309.10171, 2023b.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. arXiv preprint arXiv:2408.06072, 2024.

Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. Neural-symbolic VQA: disentangling reasoning from vision and language understanding. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), Advances in Neural Information Processing Systems, pp. 1039–1050, 2018.

Dongran Yu, Bo Yang, Qianhao Wei, Anchen Li, and Shirui Pan. A probabilistic graphical model based on neural-symbolic reasoning for visual relationship detection. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10599–10608, New Orleans, LA, USA, 2022. IEEE.

Jianhui Yu, Hao Zhu, Liming Jiang, Chen Change Loy, Weidong Cai, and Wayne Wu. Celebv-text: A large-scale facial text-video dataset, 2023. URL `https://arxiv.org/abs/2303.14717`.

David Junhao Zhang, Dongxu Li, Hung Le, Mike Zheng Shou, Caiming Xiong, and Doyen Sahoo. Moonshot: Towards controllable video generation and editing with multimodal conditions. ArXiv, abs/2401.01827, 2024a. URL `https://api.semanticscholar.org/CorpusID:266741873`.

David Junhao Zhang, Dongxu Li, Hung Le, Mike Zheng Shou, Caiming Xiong, and Doyen Sahoo. Moonshot: Towards controllable video generation and editing with multimodal conditions. arXiv preprint arXiv:2401.01827, 2024b.

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, pp. 1–15, 2024c.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In Proceedings of the IEEE/CVF international conference on computer vision, pp. 3836–3847, 2023a.

Shiwei Zhang, Jiayu Wang, Yingya Zhang, Kang Zhao, Hangjie Yuan, Zhiwu Qin, Xiang Wang, Deli Zhao, and Jingren Zhou. I2vgen-xl: High-quality image-to-video synthesis via cascaded diffusion models. arXiv preprint arXiv:2311.04145, 2023b.

Zhichao Zhang, Xinyue Li, Wei Sun, Jun Jia, Xiongkuo Min, Zicheng Zhang, Chunyi Li, Zijian Chen, Puyi Wang, Zhongpeng Ji, Fengyu Sun, Shangling Jui, and Guangtao Zhai. Benchmarking aigc video quality assessment: A dataset and unified model, 2024d. URL `https://arxiv.org/abs/2407.21408`.

G. Zulko et al. Moviepy - video editing with python. GitHub repository, 2015. URL `https://github.com/Zulko/moviepy`.

A Appendix A: Methodology Clarifications

A.1 Temporal Logic Operation Example

Given a set of atomic propositions $\mathcal{P} = \{\text{Event A, Event B}\}$, the TL specification $\Phi = \Box$ Event A (read as "Always Event A") means that 'Event A' is `True` for every step in the sequence. Additionally, $\Phi = \Diamond$ Event B (read as "eventually event b") indicates that there exists at least one 'Event B' in the sequence. Lastly, $\Phi =$ Event A U Event B (read as "Event A Until Event B") means that 'Event A' exists until 'Event B' becomes `True`, and then 'Event B' remains `True` for all future steps.

A.2 Text-to-Temporal-Logic

We use the prompt below to decompose a text prompt into 1) a set of propositions and 2) a TL specification.

f_{T2TL} Prompt for Text to TL Specification (.md)

```
**System Message**:
Your input field is:
- 'input_prompt' (str): Input prompt summarizing what happened in a video.

Your output fields are:
- 'input_propositions' (str): A list of atomic propositions that correlate with
the inputted prompt formatted as [proposition_1, proposition_2, ...].
- 'output_specification' (str): The formal specification of the inputted prompt.
This is a temporal logic sequence made by combining the inputted propositions with
temporal logic symbols.
Your objective is:
- Convert the prompt into a list of propositions and a temporal logic
specification using the specified schema.
```

```
**User message**:
Input Prompt: A person meditates by the lake, and a few seconds later, stands up
for a moment before leaving.
Respond with the corresponding output fields.

**Assistant message**:

Output Propositions: ['person is meditating', 'lake shore', 'person is standing',
'person is walking away']
Output Specification: (person is meditating ∧ lake shore) X person is standing X
person is walking away
```

Prompt 1: **Text to Specification Prompt.** System prompt to map prompts and propositions to the specification.

A.3 Vision Language Model

In this section, we provide the implementation details to detect the existence of propositions to label each frame in the synthetic video. We use the VLM to interpret semantics and extract confidence scores from $\mathcal{F}$ based on the text query T . We pass each $p_i \in \mathcal{P}$ along with the prompt to the VLM and calculate the token probability for the output response, which is either `True` or `False`. To calculate the token probability, we retrieve logits for the response tokens and compute the probability of that token after applying a softmax. Finally, the semantic confidence score is the product of these probabilities as follows:

$$c_i = \mathcal{M}_{\text{VLM}}(p_i, \mathcal{F}) \prod_{j=1}^{k} P\left(t_j \mid p_i, \mathcal{F}, t_1, \ldots, t_{j-1}\right) \; \forall p_i \in \mathcal{P}, \tag{7}$$

where $(t_1, \ldots, t_k)$ is the sequence of tokens in the response. Each term $P(t_j|\cdot) = \frac{e^{l_{j,t_j}}}{\sum_z e^{l_{j,z}}}$ is over the logits l_{j,t_j} at position j, whereas $P(t_k|\cdot) = \frac{e^{l_{j,t_k}}}{\sum_z e^{l_{k,z}}}$ is over those at position k.

A.3.1 Inference Via Vision Language Models

We use a VLM as a semantic detector. We pass each atomic proposition $p_i \in \mathcal{P}$ such as “person”, “car”, “person in the car”, etc. Once the VLM outputs either ‘Yes’ or ‘No’, we compute the token probability of the response and use it as a confidence score for the detection.

A.3.2 False Positive Threshold Identification

Dataset for Calibration: We utilize the COCO Captions Chen et al. (2015) dataset to calibrate the following open-source vision language models – InternVL2 Series (1B, 2B, 8B) OpenGVLab (2024) and LLaMA-3.2 Vision Instruct AI (2024) – for *NeuS-V*. Given that each image-caption pair in the dataset is positive coupling, we construct a set of negative image-caption pairs by randomly pairing an image with any other caption corresponding to a different image in the dataset. Once we construct the calibration dataset, which comprises 40000 image-caption pairs, we utilize the VLM to output a ‘Yes’ or a ‘No’ for each pair.

Prompt for Semantic Detector (VLM)

```
Is there {atomic proposition (pi)} present in the sequence of frames?
[PARSING RULE] 1.  You must only return a Yes or No, and not both, to any question
asked.
2.  You must not include any other symbols, information, text, or justification in
your answer or repeat Yes or No multiple times.
3.  For example, if the question is 'Is there a cat present in the Image?', the
answer must only be 'Yes' or 'No'.
```

Prompt 2: **Semantic Detector VLM.** Used to identify the atomic proposition within the frame by initiating the VLM with a single frame or a series of frames.

Thresholding Methodology We can identify the optimal threshold for the VLM by treating the above problem as either a single-class or multi-class classification problem. We opt to do the latter. The process involves first compiling detections into a list of confidence scores and one-hot encoded ground truth labels. We then sweep through all available confidence scores to identify the optimal threshold. Here, we calculate the proportion of correct predictions by applying each threshold (see Figure 5). The optimal threshold is identified by maximizing accuracy, which is the ratio of the true positive and true negative predictions. Additionally, to comprehensively evaluate model behavior, we compute Receiver Operating Characteristics (ROC), as shown in Figure 5, by computing the true positive rate (TPR) and false positive rate (FPR) across all thresholds. Once we obtain the optimal threshold, we utilize it to calibrate the predictions from the VLM. We show the accuracy vs confidence plots before and after calibration in Figure 5.

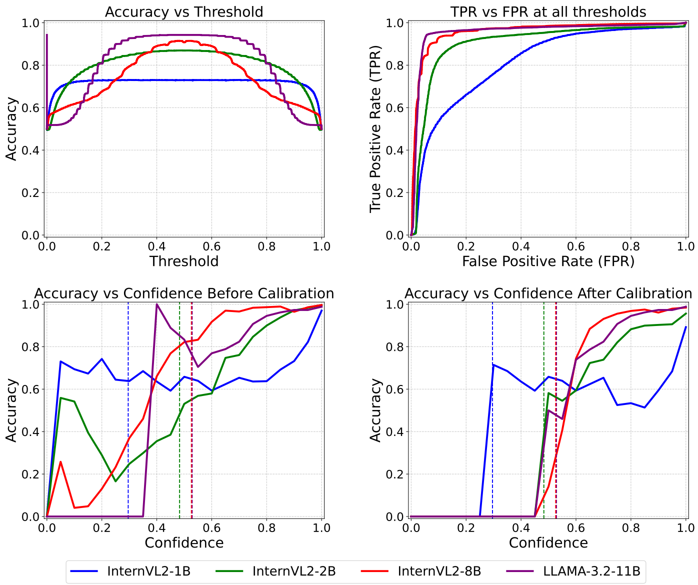


Figure 5: **Calibration Plots.** We plot the accuracy vs threshold for all VLMs on our calibration dataset constructed from the COCO Caption dataset (top left). We plot the True Positive Rate (TPR) vs False Positive Rate (FPR) across all thresholds on the top right. Finally, the bottom plots show the confidence vs accuracy of the model before and after calibration, respectively.

A.4 Video Automaton Generation Function

Given a calibrated score set across all frames $\mathcal{F}_n$ (where n is the frame index of the video) and propositions in $\mathcal{P}$, we construct the video automaton $\mathcal{A}_\mathcal{V}$ using the video automaton generation function (see Equation (2)).

$$\mathbb{C} = \{\mathbb{C}_{p_i,j} \mid p_i \in \mathcal{P}, j \in \{1, 2, \ldots, n\}\}. \tag{8}$$

As shown in Algorithm 2, we first initialize the components of the automaton, including the state set Q, the label set λ, and the transition probability set δ, all with the initial state q_0. Next, we iterate over $\mathbb{C}$, incrementally constructing the video automaton by adding states and transitions for each frame. This process incorporates the proposition set and associated probabilities of all atomic propositions. We compute possible labels for each frame as binary combinations of $\mathcal{P}$ and calculate their probabilities using the $\mathbb{C}$.

A.4.1 Edit Instruction Prompts

We present different prompts to edit the keyframe and generate the new video segment in this section.

Prompt for Image Editing (T_{img})

```
Add {Weak Proposition} to the image
```

Prompt 3: **Image (keyframe) Edit Instruction.** Used to edit the keyframe image to have missing semantics. The Weak Proposition will be given and passed along with the prompt

Prompt for New Video Generation (T_{video})

```
You are tasked with refining video narratives generated by text-to-video models
based on user feedback.  For each case, you will receive two inputs:
1.  Original Prompt:  A description of the intended video narrative.
2.  Feedback:  Textual guidance on what is missing or needs adjustment in the
video.
```

Prompt 4: **New Video Segment Generation Instruction.** Used to generate a new prompt to generate a new video segment.

B Additional Results

B.1 VBench Scores

We provide the VBench scores before and after editing. We see that the edited video shows very little degradation.

Model	Original	Edited
Gen-3	0.789	0.772 (−0.017)
Pika-2.2	0.799	0.784 (−0.015)
CogVideoX-5B (Yang et al., 2024)	0.672	0.660 (−0.012)

Table 4: **VBench Scores Before and After Editing.** Comparison of original and edited VBench scores across different models.

B.2 T2VCompBench Scores

On average, NeuS-E yields a +11% improvement across the seven T2VCompBench dimensions, driven primarily by substantial gains in `Action` and `Interaction`, which are most closely tied to correcting missing or misordered events. We include a table with category-wise improvements in Table 5.

T2VCompBench Category	**Original**	**Edited**
Consist-Attr	0.590	0.700 (+0.110)
Dynamic-Attr	0.055	0.155 (+0.100)
Spatial	0.510	0.600 (+0.090)
Motion	0.270	0.390 (+0.120)
Action	0.510	0.660 (+0.150)
Interaction	0.580	0.710 (+0.130)
Numeracy	0.240	0.320 (+0.080)
Avg	0.394	0.505 (+0.111)

Table 5: **T2VCompBench Scores Before and After Editing.** Category-wise improvement trends for Pika-2.2 under NeuS-E refinement.

B.3 Human Study Annotation

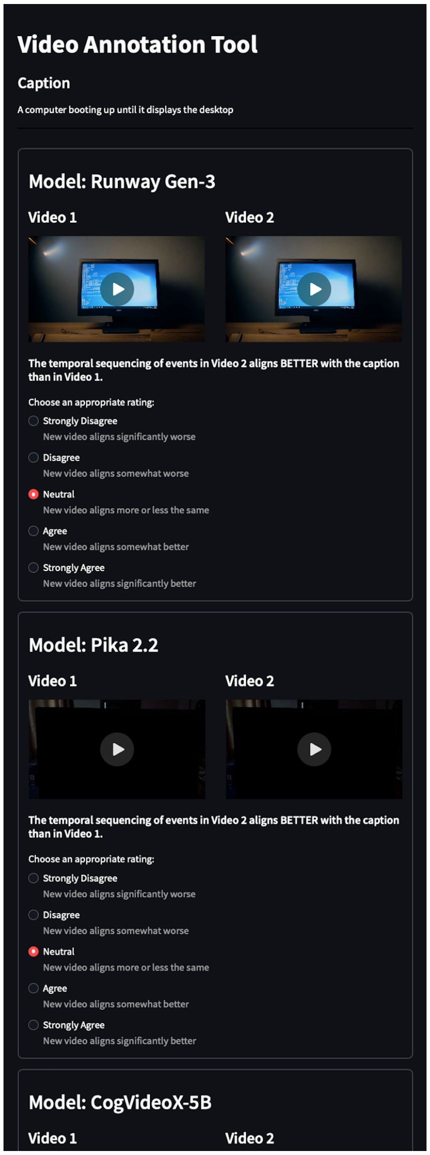
We provide the annotation tool for the randomized A/B testing of the temporal alignment of the generated videos with respect to the prompt. Here, Video 1 and Video 2 are randomly assigned as the original video and the edited videos for human judgment.

C Use of Large Language Models

The author would like to acknowledge the use of Google's Gemini large language model in the preparation of this paper. The model served as a writing assistant, which was only used for providing valuable improvements to the clarity, conciseness, and logical flow of isolated sections, more specifically, the introduction, the methodology, and the results. It was also utilized, for assistance, with specific LaTeX formatting challenges. While the AI provided helpful refinement, the core scientific contributions, conceptual framework, and all final editorial decisions were the author's own.

D Algorithms

We present the algorithms for both *NeuS-E* and the video automaton generation.



Video Annotation Tool

Caption

A computer booting up until it displays the desktop

Model: Runway Gen-3

Video 1 Video 2

The temporal sequencing of events in Video 2 aligns BETTER with the caption than in Video 1.

Choose an appropriate rating:

○ Strongly Disagree
New video aligns significantly worse

○ Disagree
New video aligns somewhat worse

● Neutral
New video aligns more or less the same

○ Agree
New video aligns somewhat better

○ Strongly Agree
New video aligns significantly better

Model: Pika 2.2

Video 1 Video 2

The temporal sequencing of events in Video 2 aligns BETTER with the caption than in Video 1.

Choose an appropriate rating:

○ Strongly Disagree
New video aligns significantly worse

○ Disagree
New video aligns somewhat worse

● Neutral
New video aligns more or less the same

○ Agree
New video aligns somewhat better

○ Strongly Agree
New video aligns significantly better

Model: CogVideoX-5B

Video 1 Video 2

Figure 6: **Tool for Annotating Videos.** Subjects evaluate the video edited by *NeuS-E* by comparing it to its original generation across five levels: strongly disagree, disagree, neutral, agree, and strongly agree.

Algorithm 1: *NeuS-E*

```
Require: Text-to-video model $\mathcal{M}_{\text{T2V}}$, prompt based image to image model omnigen Text-to-TL function
         $f_{\text{T2TL}}(T)$, video automaton generation function $\xi(\cdot)$, probabilistic model checking function
         $\Psi(\cdot)$, edit instruction generation function $\text{LLM}_{\text{EIG}}$, satisfaction probability threshold $\theta$,
         maximum iterations $\kappa$
Input  : Text prompt $T$, initial generated video $\mathcal{V} = \mathcal{M}_{\text{T2V}}(T)$
Output : Refined video $\mathcal{V}^*$
begin
    $\mathbb{P}[\mathcal{A}_{\mathcal{V}} \models \Phi] \leftarrow 0$
    $iter \leftarrow 0$
    $\mathcal{V} \leftarrow \mathcal{M}_{\text{T2V}}(T)$                                  // Generate a video
    $\mathcal{P}, \Phi \leftarrow f_{\text{T2TL}}(T)$                              // Decompose a prompt
    while $\mathbb{P}[\mathcal{A}_{\mathcal{V}} \models \Phi] < \theta \wedge$ iter $< \kappa$ do
        $\mathbb{C} \leftarrow \{\}$
        for $n = 0$ to length$(\mathcal{V})$ do
            for $p_i \in \mathcal{P}$ do
                $c_{i,n} \leftarrow \mathcal{M}_{\text{VLM}}(p_i, \mathcal{F}_n)$
            $\mathbb{C} \leftarrow \mathbb{C} \cup \{c_{i,n}\}$
        $\mathcal{A}_{\mathcal{V}} \leftarrow \xi(\mathcal{P}, \mathbb{C})$
        $\mathbb{P}[\mathcal{A}_{\mathcal{V}} \models \Phi] \leftarrow \Psi(\mathcal{A}_{\mathcal{V}}, \Phi)$
        if $\mathbb{P}[\mathcal{A}_{\mathcal{V}} \models \Phi] \geq \theta$ then
            Break
                                                  // Identify the weakest proposition
        max_delta $\leftarrow 0$
        $p_i^* \leftarrow$ None
        $i^* \leftarrow$ None
        for $p_i \in \mathcal{P}$ do
            $\tilde{\mathbb{C}} \leftarrow \mathbb{C}$
            for $n = 0$ to length$(\mathcal{V})$ do
                $\tilde{c}_{i,n} \leftarrow 1.0$
            $\tilde{\mathcal{A}}_{\mathcal{V},i} \leftarrow \xi(\mathcal{P}, \tilde{\mathbb{C}})$
            $\mathbb{P}[\tilde{\mathcal{A}}_{\mathcal{V},i} \models \Phi] \leftarrow \Psi(\tilde{\mathcal{A}}_{\mathcal{V},i}, \Phi)$
            $\delta_i \leftarrow \mathbb{P}[\tilde{\mathcal{A}}_{\mathcal{V},i} \models \Phi] - \mathbb{P}[\mathcal{A}_{\mathcal{V}} \models \Phi]$
            if $\delta_i >$ max_delta then
                max_delta $\leftarrow \delta_i$
                $p_i^* \leftarrow p_i$
                $i^* \leftarrow i$
                              // Localize the influence of the weakest proposition
        max_z $\leftarrow 0$
        $\mathcal{F}_n^* \leftarrow$ None
        $n^* \leftarrow$ None
        for $n = 0$ to length$(\mathcal{V})$ do
            $\tilde{\mathbb{C}} \leftarrow \mathbb{C}$
            $\tilde{c}_{p_i^*,n} \leftarrow 1.0$
            evan $\leftarrow \xi(\mathcal{P}, \tilde{\mathbb{C}})$
            $\mathbb{P}[\tilde{\mathcal{A}}_{\mathcal{V},n} \models \Phi] \leftarrow \Psi(\tilde{\mathcal{A}}_{\mathcal{V},n}, \Phi)$
            if $\mathbb{P}[\tilde{\mathcal{A}}_{\mathcal{V},n} \models \Phi] >$ max_z then
                max_z $\leftarrow \mathbb{P}[\tilde{\mathcal{A}}_{\mathcal{V},n} \models \Phi]$
                $\mathcal{F}_n^* \leftarrow \mathcal{F}_n$
                $n^* \leftarrow n$
                                                              // Video refinement
        $\mathcal{V}_{\text{trimmed}} \leftarrow \mathcal{V}[0 : n^*]$              // Trim video up to impacted segment
        $T_{\text{new}} \leftarrow \text{LLM}_{\text{EIG}}(p_i^*), T_{\text{video}}$      // Generate edit instruction for $p_i^*$
        $\hat{\mathcal{F}}_n^* \leftarrow \text{omnigen}(\mathcal{F}_n^*, T_{\text{img}})$        // Edit keyframe based on the prompt
        $\mathcal{V}_{\text{new}} \leftarrow \mathcal{M}_{\text{T2V}}(\hat{\mathcal{F}}_n^*, T_{\text{new}})$                // Generate new video segment
        $\mathcal{V} \leftarrow \mathcal{V}_{\text{trimmed}} + \mathcal{V}_{\text{new}}$   // Merge trimmed video with new segment with text
            prompt and keyframe
        $iter \leftarrow iter + 1$
```

Algorithm 2: Video Automaton Generation

```
Input: Set of semantic score across all frames given all atomic propositions
       ${\mathbb{C} = \mathbb{C}_{p_i,j} \mid p_i \in \mathcal{P}, j \in \{1, 2, \ldots, n\}}$, set of atomic propositions $\mathcal{P}$
Output: Video automaton $\mathcal{A}_\mathcal{V}$
begin
    $Q \leftarrow \{q_0\}$        // Initialize the set of states with the initial state
    $\lambda \leftarrow \{(q_0, \text{initial})\}$        // Initialize the set of labels with the initial
     label
    $\delta \leftarrow \{\}$                         // Initialize the set of state transitions
    $Q_p \leftarrow \{q_0\}$                       // Track the set of previously visited states
    $n \leftarrow \frac{|\mathbb{C}|}{|\mathcal{P}|}$                          // Calculate the total number of frames $n$
    for $j \leftarrow 1$ to $n$ do
        $Q_c \leftarrow \{\}$                              // Track the set of current states
        for $e_j^k \in 2^{|\mathcal{P}|}$ do
            // $e_j^k$: unique combination of 0s and 1s for atomic
               propositions in $\mathcal{P}$
            $\lambda(q_j^k) = \{v_1, v_2, \ldots, v_i \mid v_i \in \{1, 0\}, \forall i \in \{1, 2, \ldots, |\mathcal{P}|\}\}$
            $pr(j, k) \leftarrow 1$                 // Initialize probability for the label
            for $v_i \in \lambda(q_j^k)$ do
                // Calculate probability for $e_j^k$
                if $v_i = 1$ then
                    $pr(j, k) \leftarrow pr(j, k) \cdot \mathbb{C}_{p_i,j}$
                else
                    $pr(j, k) \leftarrow pr(j, k) \cdot (1 - \mathbb{C}_{p_i,j})$
                // Add state and define transitions if the probability is
                   positive
                if $pr(j, k) > 0$ then
                    $Q \leftarrow Q \cup \{q_j^k\}$
                    $Q_c \leftarrow Q_c \cup \{q_j^k\}$
                    $\lambda \leftarrow \lambda \cup \{(q_j^k, \lambda(q_j^k))\}$
                    for $q_{j-1} \in Q_p$ do
                        $\delta(q_{j-1}, q_j^k) \leftarrow pr(j, k)$
                        $\delta \leftarrow \delta \cup \{\delta(q_{j-1}, q_j^k)\}$
                    end for
            end for
            $Q_p \leftarrow Q_c$                                   // Update previous state
        end for
    end for
    // Add terminal state
    $Q \leftarrow Q \cup \{q_{j+1}^0\}$
    $\lambda \leftarrow \lambda \cup \{(q_{j+1}^0, \text{terminal})\}$
    for $q_{j-1} \in Q_p$ do
        $\delta(q_{j-1}, q_{j+1}^0) \leftarrow 1$
        $\delta \leftarrow \delta \cup \{\delta(q_{j-1}, q_{j+1}^0)\}$
    end for // Return video automaton
    $\mathcal{A}_\mathcal{V} \leftarrow (Q, q_0, \delta, \lambda)$
    return $\mathcal{A}_\mathcal{V}$
```