T-LESSA: TEMPORAL-LOGIC EVENT SPECIFICATION STEERING OF VIDEO DIFFUSION ATTENTION

Anonymous authors
Paper under double-blind review

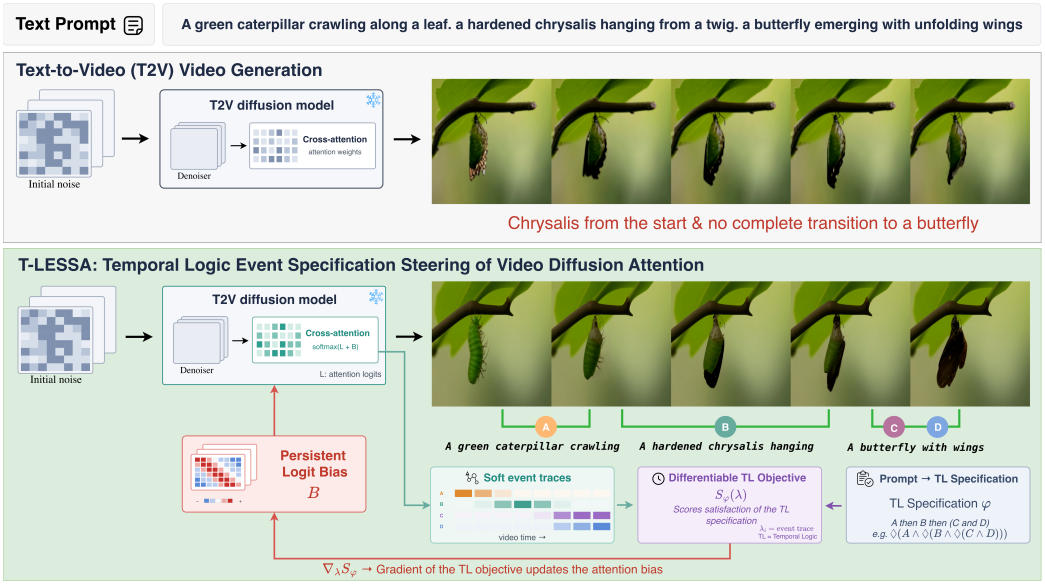


Figure 1: **T-LESSA steers video generation toward satisfying temporal-logic specifications.** We ask a caterpillar to become a chrysalis, then a butterfly shows up with its unfolded wings. The unsteered CogVideoX-5B model starts with a chrysalis and misses the first and last events. *T-LESSA* first decomposes the prompt into a temporal logic (TL) specification (temporal-logic representation of the prompt) as $\Diamond(A \wedge \Diamond(B \wedge \Diamond(C \wedge D)))$ and generates all four events in the correct order. During denoising, it guides video generation to be aligned with the TL specification.

ABSTRACT

Recent advancements in generative video promise to transform challenging scenarios into training data: rare driving situations for autonomous vehicles, synthetic scenes for robot simulation, and realistic deepfakes for stress testing detection models. However, in practice, we still prompt a model and pray to get faithful videos. While a large body of research exists on generating good video, there is a lack of methods to enforce temporal and logical requirements (e.g., ordering, negation, co-occurrence) during video generation. Prior work has focused on fine-tuning with preference feedback or verifiable rewards. However, these methods require additional training and reduce a temporal-logic requirement to a scalar reward. This limitation has motivated a recent line of work that steers video generation at inference time without any training. Although these approaches control when each required event appears, they do not ensure that the generated videos align with their temporal and logical requirements. Therefore, we introduce *T-LESSA*, a training-free method that steers video generation toward satisfying a temporal-logic requirement. We show that *T-LESSA* improves temporal completion by **14.3%** and, across four backbones, temporal-logic satisfaction by **21.8%** over the unsteered baseline, at 1.7% to 13.5% additional generation time. Furthermore, *T-LESSA* improves compositional alignment in 14 of 16 backbone-category pairs on T2V-CompBench, and largely preserves video quality.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Imagine asking a video generator to generate an urban driving scene in which "a pedestrian steps onto a crosswalk, the vehicle brakes, and a deer runs across the road after the vehicle has stopped". A good video generator must do more than render the correct pedestrian, vehicle, deer, and road. It must generate each event and arrange them in the prescribed order: the pedestrian enter the crosswalk first, the vehicle then brake and stop, then the deer crosses. Despite rapid advances, current text-to-video (T2V) models remain unreliable on prompts containing multiple events, state changes, and temporal logic (TL) relations. They may omit an event, blend distinct events together, or depict them in the wrong order. For example, in Figure 1, a vanilla T2V model fails to generate the life cycle of a butterfly: the chrysalis appears from the beginning and never turns into a butterfly.

Existing approaches address these limitations through either training-time alignment or inference-time guidance. Training and post-training methods use learned rewards, preference data, or temporal supervision (Prabhudesai et al., 2024; Liu et al., 2025; Wang et al., 2024; Wu et al., 2025). More recently, inference-time methods (Shaulov et al., 2026; Schiber et al., 2025) guide frozen generators. Their guidance objectives, however, require trained classifiers for T2V temporal fidelity or user-specified temporal masks to steer internal model representations. They therefore do not directly express the events, state changes, and temporal relationships in a prompt as an inference-time objective. This raises a central question: How can a prompt's events, states, and their TL relationships be expressed directly as an inference-time objective for steering a T2V model?

We answer this question by proposing **T**emporal **L**ogic **E**vent **S**pecification **S**teering of video-diffusion **A**ctivations (*T-LESSA*). Our key insight is to **use temporal logic (TL) as an intermediate representation** of a prompt's events and their relationships, from which we derive a differentiable inference-time objective. First, a Large Language Model (LLM) breaks a complex prompt into atomic events and states and encodes their temporal and logical relationships in a TL specification as shown in Fig. 1. Subsequently, during denoising, *T-LESSA* reads the cross-attention associated with each identified atomic event or object as a soft proxy for when it appears or occurs in the video. The soft scores are combined with the temporal order implied in the TL specification to compute a differentiable objective for temporal and logical T2V alignment. The gradient of this objective updates a persistent bias on the corresponding cross-attention logits, steering attention at the next denoising iteration toward better satisfaction of the specification. Thus, ***T-LESSA* progressively guides the video toward the TL specification derived from its text prompt** without retraining the T2V model or directly optimizing its diffusion latents. Our work makes two main contributions:

1. We encode text prompts into TL specifications via an LLM, formulate a differentiable TL score from cross-attention values associated with prompted events and objects, and prove that the score matches finite-trace TL semantics on binary traces.
2. We steer a frozen T2V model at inference time by updating a persistent bias on its cross-attention logits using the gradient of the TL objective. On TLVBENCH, this improves TL satisfaction by 21.8% over the unsteered backbone across four T2V models, without fine-tuning, direct latent optimization, or an external temporal classifier.

2 RELATED WORK

Text-to-Video Generation. Text-to-video generative models (OpenAI, 2025; Kling, 2026; Wan et al., 2025; Google DeepMind, 2026; Pika Labs, 2026) have seen massive strides in improving prompt fidelity, motion generation, and video quality. These models either rely on autoregressive architectures (Hong et al., 2022; Villegas et al., 2022) or diffusion-based models (Ho et al., 2022; Chen et al., 2024; Yang et al., 2025; Kong et al., 2024; Wan et al., 2025). We refer readers to Puspitasari et al. (2026) for a broader survey. *T-LESSA* is independent of the backbone architecture and instead better aligns the video to the text prompt during test-time generation.

Text-to-Video Editing. T2V video editing modifies either real or generated videos from T2V models, to better capture motion, structure, and identity. Popular approaches adapt text-to-image diffusion to edit videos (Meng et al., 2021; Brooks et al., 2023; Zhang et al., 2023; Goel et al., 2024). Additionally, temporal consistency has been enforced by cross-frame attention, and feature or key-frame propagation (Geyer et al., 2024; Jeong & Ye, 2023; QI et al., 2023; Yang et al., 2023; Liu et al.,

2024). Similarly, some methods edit selected keyframes and propagate the changes throughout the sequence (Ceylan et al., 2023; Zhang et al., 2025), refine the conditioning prompt (Luo et al., 2025; Choi et al., 2025), or use video-to-video diffusion models (Molad et al., 2023). These methods preserve visual consistency under an edit, but they do not explicitly represent logical dependencies among multiple events. *T-LESSA* takes a different approach. It steers generation from noise so that the resulting event sequence follows the prompted temporal and logical constraints.

Text-to-Video Alignment. T2V alignment methods are applied either during training or at inference time. Training-based approaches optimize generators using differentiable rewards, preference pairs, or human feedback (Prabhudesai et al., 2024; Liu et al., 2025; Wang et al., 2024), while MinT uses temporally grounded supervision to associate events with video intervals (Wu et al., 2025). Inference-time methods instead keep the generator frozen. Compositional Video Generation steers diffusion latents using a classifier trained on cross-attention features (Shaulov et al., 2026), while TempoControl uses attention objectives and an external temporal signal (Schiber et al., 2025). Free^{2}Guide used VLM-based rewards to guide sampling, whereas Video-MSG uses a generated spatiotemporal sketch to guide video structure (Kim et al., 2025; Li et al., 2025). In contrast, *T-LESSA* converts the prompt's temporal relations into a differentiable temporal-logic objective and uses it to bias cross-attention logits during test-time without retraining.

3 Preliminaries and Problem Formulation

Finite-trace temporal-logic (TL) specifications. We define a finite set of event propositions as AP and a length-T trace as a binary matrix $\sigma \in \{0,1\}^{T\times|\mathsf{AP}|}$, where $\sigma_{t,\rho} = 1$ means that proposition ρ holds at temporal position t. A finite-trace TL specification φ combines propositions using temporal operators, such as next X, which is false at the final temporal position, and until U, and logical operators, such as conjunction $\wedge$ and disjunction $\vee$. We use the standard abbreviations $\Diamond\varphi \equiv \top\,\mathsf{U}\,\varphi$ (eventually) and $\Box\varphi \equiv \neg\Diamond\neg\varphi$ (always). We define $\sigma \models \varphi$ when the trace σ satisfies φ.

TL satisfaction via model checking. We can represent a video as a video automaton: a probabilistic trace model whose states carry time-indexed event valuations $Y_t \subseteq \mathsf{AP}$ and whose transition probabilities define a distribution over sequences of these valuations. This model is constructed from the T2V model's attention mass over T latent temporal positions and combines it with an automaton that tracks satisfaction of the TL specification (Choi et al., 2024; 2025). On the resulting discrete-time Markov chain (DTMC), we evaluate the probabilistic computation tree logic (PCTL) bounded-reachability query $\mathbb{P}_{=?}[\Diamond^{\leq T}\mathsf{Sat}_T]$ (Baier & Katoen, 2008). Here, Sat_T labels accepting states after all T temporal positions. The probability that a trace satisfies the TL specification is differentiable quantity which we use as a surrogate satisfaction score.

Problem Description. Let $\mathcal{G}_\theta$ be a frozen T2V model that generates a video $v \in \mathbb{R}^{T_v\times H_v\times W_v\times 3}$ given a prompt p and initial Gaussian noise $z \in \mathbb{R}^{C_z\times T\times H_z\times W_z}$. Here, C_z is the number of latent channels, T is the number of latent temporal positions, H_z and W_z are the latent spatial dimensions, and T_v, H_v, and W_v are the output video dimensions. Let φ be a finite-trace TL specification whose propositions $\mathsf{AP}_\varphi \subseteq \mathsf{AP}$ correspond to events described in p. At denoising iteration i, we sum the cross-attention assigned to the prompt tokens associated with each proposition and average it across all heads of the selected layers and spatial positions. This produces the event scores $\lambda_i \in [0,1]^{T\times|\mathsf{AP}_\varphi|}$ over T temporal positions in the latent representation. Section 4.2 gives the full construction. Given λ_i, our first goal is to define a differentiable TL satisfaction score $S_\varphi(\lambda_i)$ that agrees with classical model checking when λ_i is a binary trace:

$$S_\varphi(\sigma) = \mathbf{1}\{\sigma \models \varphi\}, \qquad \sigma \in \{0,1\}^{T\times|\mathsf{AP}_\varphi|}. \tag{1}$$

Our second goal is to use this score to guide video generation without updating θ or placing an external video model in the denoising loop. Let $v_0 = \mathcal{G}_\theta(p,z)$ be the unsteered generation and v_φ the paired, steered generation. Both use the same prompt, initial noise, sampler, and number of denoising steps. For a video evaluation metric s_φ, we aim to have a positive gain:

$$\mathbb{E}[s_\varphi(v_\varphi) - s_\varphi(v_0)] > 0. \tag{2}$$

4 T-LESSA: TEMPORAL-LOGIC-GUIDED VIDEO GENERATION

Given a prompt p, *T-LESSA* steers a frozen T2V model to satisfy a finite-trace TL specification φ, in four steps: ❶ An LLM-based agent translates the prompt into φ and identifies its event phrases. ❷ At each denoising iteration i, cross-attention provides a sequence of event scores λ_i over the T temporal positions of the latent representation. ❸ We compute the attention-based satisfaction score $S_\varphi(\lambda_i) \in [0, 1]$ as a differentiable function of λ_i. ❹ We use its gradient to update a bias on the corresponding attention logits and add this bias to the next denoising iteration.

4.1 TEXT-TO-TEMPORAL-LOGIC AGENT

We introduce the *T-LESSA* agent to translate a text prompt p into a finite-trace TL specification φ using the operators defined in Section 3. The agent's two primary tasks are translation and validation. During translation, it identifies the atomic events described in p, assigns a proposition ρ to each event, and records the corresponding words in p as its event phrase. It then maps the temporal and logical relations in p to TL operators. During validation, it checks that every proposition is declared and grounded in an explicit phrase in the prompt. The agent returns φ and the event phrase associated with each atomic proposition. For example, consider the prompt A dog enters the room, and afterward the dog sits down. No cat appears. The agent returns the atomic propositions a, b, and c, where a denotes "a dog enters the room," b denotes "the dog sits down," and c denotes "cat appears," along with φ:

$$\varphi = \Diamond(a \wedge \mathsf{X}\Diamond b) \wedge \Box\neg c.$$

Note that we cover the eventually, globally and first-onset forms with their conjunctions and negation. We provide the agent's model, prompt rules, and automatic validation results in Appendix B.1.

4.2 CROSS-ATTENTION LABELING

Along with the TL specification, the agent returns an event phrase for each proposition. We use cross-attention to turn these phrases into a temporal sequence of event scores. At denoising step $i \in \{0, \dots, K-1\}$, let $t \in \{1, \dots, T\}$ index temporal positions in the latent representation. The T2V tokenizer maps the phrase for proposition ρ to prompt-token indices I_ρ which are text-token indices, not video positions. Let $\mathcal{C}$ contain every head of each selected cross-attention layer, as specified in Section B.2, and let each temporal position contain N_s spatial tokens. For pair $r \in \mathcal{C}$, spatial token s, and prompt token j, write $\ell_{i,r,t,s,j}$ for the attention logit. We normalize across all prompt tokens, sum the mass assigned to the event phrase, and average across the pairs in $\mathcal{C}$ and spatial tokens:

$$\alpha_{i,r,t,s,j} = \frac{\exp(\ell_{i,r,t,s,j})}{\sum_{j'} \exp(\ell_{i,r,t,s,j'})}, \qquad \lambda_{i,t,\rho} = \frac{1}{|\mathcal{C}|N_s} \sum_{r\in\mathcal{C}} \sum_{s=1}^{N_s} \sum_{j\in I_\rho} \alpha_{i,r,t,s,j}. \tag{3}$$

Here, j' ranges over all prompt tokens. Thus $\lambda_{i,t,\rho} \in [0, 1]$ measures the average attention mass assigned to ρ's phrase at temporal position t. In the calculation below, we use this mass as a proposition probability parameter.

4.3 DIFFERENTIABLE TL SATISFACTION VIA PCTL

We next ask how likely the sequence of attention-based event labels is to satisfy the TL specification φ. Let $\mathcal{A}_\varphi = (Q_\varphi, q^0_\varphi, \delta_\varphi, F_\varphi)$ be a complete deterministic finite-state automaton: Q_φ is its state set, q^0_φ its initial state, F_φ its accepting states, and $\delta_\varphi : Q_\varphi \times 2^{\mathsf{AP}_\varphi} \to Q_\varphi$ its transition function. We choose $\mathcal{A}_\varphi$ so that it accepts exactly the finite traces satisfying φ. At temporal position t, an event label $Y \subseteq \mathsf{AP}_\varphi$ lists the propositions that hold. For example, $Y = \{a, c\}$ means that a and c hold together. The automaton then moves from q to $\delta_\varphi(q, Y)$.

To obtain a distribution over labels from the continuous attention scores, we model each proposition ρ at each position t as an independent Bernoulli variable with parameter $\lambda_{i,t,\rho}$. This defines an attention-based probabilistic model of event traces, rather than a probability model of events in

rendered videos. The probability of a label Y and the resulting automaton transition are

$$\mu_{i,t}(Y) = \prod_{\rho \in Y} \lambda_{i,t,\rho} \prod_{\rho \in \mathsf{AP}_\varphi \setminus Y} (1 - \lambda_{i,t,\rho}),$$
$$P^{\varphi}_{i,t}(q, q') = \sum_{Y \subseteq \mathsf{AP}_\varphi} \mu_{i,t}(Y)\, \mathbf{1}\{\delta_\varphi(q, Y) = q'\}\,. \tag{4}$$

In words, $P^{\varphi}_{i,t}(q, q')$ adds the probabilities of every label that takes q to q'. Its rows sum to one. Independence here is an assumption for computing transitions from attention scores, and it makes no claim that events in generated videos are independent.

The transition matrix changes with t, so we pair each automaton state with its temporal index. The resulting time-unrolled DTMC moves from $(t-1, q)$ to (t, q') according to $P^{\varphi}_{i,t}(q, q')$ and has terminal self-loops at $t = T$. We label only terminal states (T, q) with $q \in F_\varphi$ as Sat_T. Waiting until T matters: a prefix can satisfy a requirement that the full sequence later violates, as when $\Box\neg c$ holds until c occurs.

The bounded-reachability PCTL query introduced in the preliminaries asks for the probability of reaching Sat_T in this DTMC. We evaluate that query by backward dynamic programming. Let $V_{i,t}(q)$ be the probability of terminal satisfaction when the automaton is in q after position t. Then

$$V_{i,T}(q) = \mathbf{1}\{q \in F_\varphi\}, \qquad V_{i,t-1}(q) = \sum_{q' \in Q_\varphi} P^{\varphi}_{i,t}(q, q') V_{i,t}(q'), \qquad S_\varphi(\lambda_i) = V_{i,0}(q^0_\varphi). \tag{5}$$

Here $t = 0$ means before the first temporal position. Thus $S_\varphi(\lambda_i)$ is the exact probability of satisfying φ under the attention-based probabilistic model, and it is not a calibrated probability that the rendered video satisfies φ. In the dog–cat example, attention favoring "dog enters" early, "dog sits" later, and little "cat appears" attention favors a satisfying trace. Because the transition probabilities are differentiable in the event scores, S_φ supplies a differentiable steering objective.

The following result relates this continuous score to ordinary TL evaluation on binary labels.

Theorem 1 (Agreement on binary traces)**.** *Let $\mathcal{A}_\varphi$ be a complete deterministic finite-state automaton. Then S_φ is multi-affine in λ: it is affine in each event score when all other scores are fixed. Write $\delta^*_\varphi(q, \sigma)$ for the state reached by applying δ_φ successively to the T event sets in σ. For every length-T binary trace σ,*

$$S_\varphi(\sigma) = \mathbf{1}\{\delta^*_\varphi(q^0_\varphi, \sigma) \in F_\varphi\}. \tag{6}$$

Since $\mathcal{A}_\varphi$ accepts exactly the traces satisfying φ, the theorem shows that our score agrees with classical TL evaluation on binary traces. Appendix A.1 gives the proof.

4.4 Steering Attention

Finally, we use the satisfaction gradient to update a persistent event bias $B_i \in \mathbb{R}^{T \times |\mathsf{AP}_\varphi|}$. At denoising iteration i, we add $B_{i,t,\rho}$ to the cross-attention logits associated with tokens in I_ρ. We stop gradients through the T2V model and differentiate the satisfaction score only with respect to the resulting event scores:

$$g_i = \nabla_{\lambda_i} S_\varphi(\lambda_i), \qquad \widehat{g}_i = \begin{cases} g_i / \|g_i\|_\infty, & \|g_i\|_\infty > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Starting from $B_0 = 0$, the bias update is

$$B_{i+1} = \Pi_{\mathcal{B}}\left(B_i + \eta \widehat{g}_i\right), \qquad \mathcal{B} = \{B : \|B\|_\infty \le b_{\max}\}\,, \tag{8}$$

where $\eta > 0$ is the step size, $b_{\max} > 0$ bounds the magnitude of each bias entry, and $\Pi_{\mathcal{B}}$ denotes projection onto $\mathcal{B}$. This forward-only controller updates only B_i, without changing model parameters or optimizing diffusion latents. It uses the label-space gradient $\nabla_{\lambda_i} S_\varphi$ rather than the exact gradient with respect to B_i. Appendix A.2 analyzes its local first-order behavior.

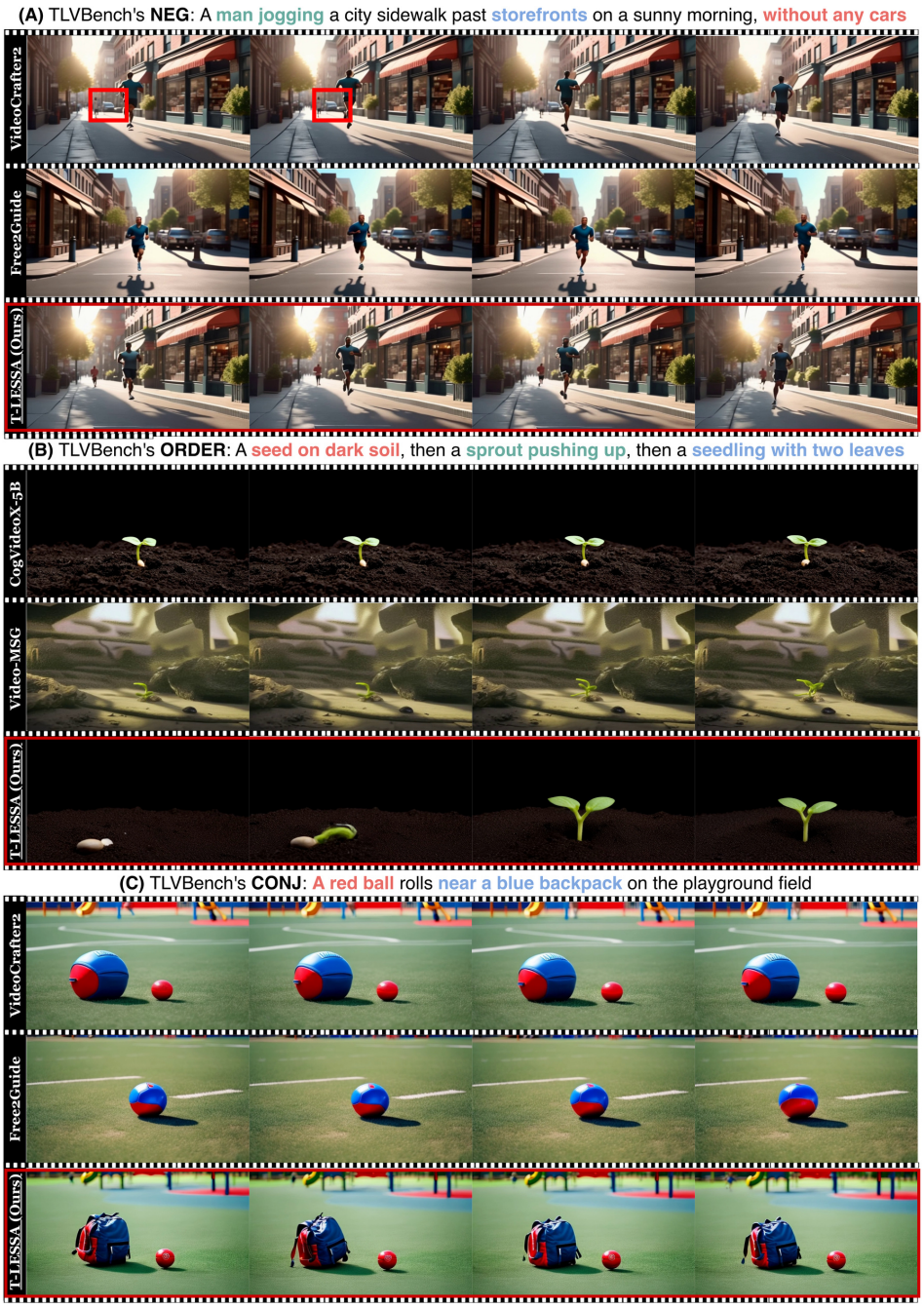


Figure 2: ***T-LESSA* generates videos satisfying its corresponding TL specification.** We compare *T-LESSA* with two unsteered T2V models and prior inference-time guidance methods on representative temporal-logic prompts. In Case A (top), VideoCrafter2 and Free2Guide generate cars despite the prompt requiring no cars (see the red boxes), whereas *T-LESSA* does not. In Case B (middle), the prompt asks for a seed that sprouts and becomes a seedling with two leaves. CogVideoX-5B starts with the seedling and misses the first and second events. Video-MSG fails even to begin with a seed. Yet, *T-LESSA* successfully generates three events in the correct sequence. In Case C (bottom), we look for a conjunctive scenario in which a red ball and a blue backpack appear in the same scene. VideoCrafter2 generates a red ball, but it generates a bluish ball instead of a blue backpack. Free2Guide fails to generate both objects correctly. However, *T-LESSA* correctly generates both target objects. We provide more qualitative examples in the supplementary.

Table 1: ***T-LESSA* improves every class on every unsteered T2V backbone**. We showcase *T-LESSA*'s performance on TLVBENCH using two VLM judges, Qwen2.5-VL-7B (Q) and InternVL3-8B (I3), along with NeuS-V (NV). We compare *T-LESSA* against unsteered T2V backbones and other inference-time guidance methods. Note that SwitchCraft and Video-MSG are not proper guiding methods for *TLVB-NEG* and *TLVB-TEMPLOG*: ❶ SwitchCraft requires a list of events in order to steer events to be generated in order, and due to this nature it does not guide *TLVB-CONJ* as well, and ❷ Video-MSG's LLM planner writes what each frame contains, so it has no way to state that an object must be absent. $*$ marks $p < .05$ against the unsteered backbones under a Wilcoxon signed-rank test on prompt means over three seeds. We provide how each baseline was run in Section C.3.

		TLVB-NEG			*TLVB-CONJ*			*TLVB-ORDER*			*TLVB-TEMPLOG*		
Backbone	Method	Q	I3	NV	Q	I3	NV	Q	I3	NV	Q	I3	NV
Wan2.1-14B	Unsteered	.559	.472	.516	.698	.796	.654	.569	.701	.204	.307	.296	.233
	SwitchCraft	–	–	–	–	–	–	**.670***	**.759***	.212	–	–	–
	T-LESSA (Ours)	**.652***	**.603***	**.630***	**.760***	**.874***	**.743***	.622*	.747*	**.214**	**.421***	**.430***	**.313***
CogVideoX-5B	Unsteered	.542	.433	.498	.244	.411	.216	.256	.478	.137	.197	.319	.235
	Video-MSG	–	–	–	.287	.516	.302	.223	.417	.135	–	–	–
	T-LESSA (Ours)	**.655***	**.578***	**.636***	**.350***	**.544***	**.320***	**.365***	**.566***	**.138**	**.297***	**.440***	**.303***
VideoCrafter2	Unsteered	.452	.410	.312	.370	.498	.230	.190	.346	.133	.139	.286	.167
	Free2Guide	.436	.405	.322	.364	.492	.229	.171	.303	.134	.137	.290	.157
	T-LESSA (Ours)	**.488***	**.457***	**.334**	**.514***	**.672***	**.270***	**.266***	**.430***	**.139**	**.188***	**.347***	**.175**
AnimateDiff	Unsteered	.405	.345	.297	.231	.414	.184	.186	.378	.133	.140	.293	.163
	VideoGuide	.412	**.391**	**.304**	.259	.399	.182	.180	.367	.135	.133	.279	.166
	T-LESSA (Ours)	**.435***	.381*	.303	**.297***	**.533***	**.203***	**.221***	**.440***	**.137**	**.187**	**.324**	**.177**

5 EXPERIMENTAL SETUP

Baselines. We use unsteered T2V models as our primary baselines: Wan2.1-T2V-14B, CogVideoX-5B, VideoCrafter2, and AnimateDiff (Wan et al., 2025; Yang et al., 2025; Chen et al., 2024; Guo et al., 2023). In addition, we compare our method with inference-time guidance methods: Video-MSG, Free2Guide, VideoGuide, SwitchCraft, and TempoControl (Li et al., 2025; Kim et al., 2025; Lee et al., 2025; Xu et al., 2026; Schiber et al., 2025). By default, we use the same sampler and 50 denoising steps for each unsteered baseline and steer the first 40 steps. We use $\eta = .15$ and $b_{\max} = 3$ for *TLVB-ORDER* and *TLVB-CONJ*, and category- and backbone-specific settings for *TLVB-NEG* and *TLVB-TEMPLOG*. We provide full generation settings in Appendix B.2.

Benchmarks. We introduce TLVBENCH (Temporal-Logic Video Benchmark), which evaluates whether generated videos satisfy finite-trace TL specifications. It contains 400 prompts, each paired with a TL specification and the atomic propositions of its events. For the logical assessment, we use negation (*TLVB-NEG*) and conjunction (*TLVB-CONJ*). *TLVB-CONJ* requires both events to appear in the video. For the temporal assessment, we provide *TLVB-ORDER*, which requires events A, B, and C to appear in that order. Finally, for the temporal-logic assessment, we provide *TLVB-TEMPLOG* which combines both by requiring A to occur before B and keeping C absent throughout the video. We note more detail on TLVBENCH in Appendix C.1. We also evaluate temporal completion on StoryEval (Wang et al., 2025) and compositional alignment on T2V-CompBench (Sun et al., 2025). Lastly, we use VBench (Huang et al., 2024) for video quality. For those benchmarks that do not come with a TL specification alongside the prompt, we use the *T-LESSA* agent to decompose their prompt into a corresponding TL specification (Appendix B.1).

Metrics. We evaluate TL satisfaction with VLM judges and NeuS-V (Sharan et al., 2025). Qwen2.5-VL-7B (Qwen Team, 2025) and InternVL3-8B (Zhu et al., 2025) score eight uniformly sampled frames using specification-specific yes–no questions. We provide the prompt for VLM judges in Appendix B.3. NeuS-V computes the TL satisfaction probability using neuro-symbolic formal verification. To further evaluate temporal completion and consistency, we report StoryEval's story-completion rate on all 423 prompts which describes two to four consecutive events. An event is complete only when all three judges accept it. In addition, we report official TCBench (Feng et al., 2025) metrics: ❶ TC-Score averages the pass rates of four assertions about the transition, whereas ❷ TCR requires all four to pass, penalizing videos that generate the correct events but in the wrong or-

Table 2: ***T-LESSA* improves compositional alignment.** T2V-CompBench scores increase in 14 of 16 backbone-category pairs. Each category contains the official 200 prompts. Note that SwitchCraft is not a proper guiding method for *Interaction* because all 200 prompts describe a single event.

	Wan2.1-14B			CogVideoX-5B			AnimateDiff			VideoCrafter2		
Category	Unsteered	SwitchCraft	Ours	Unsteered	Video-MSG	Ours	Unsteered	VideoGuide	Ours	Unsteered	Free2Guide	Ours
Dynamic	.0823	**.1072**	.0939	.0146	.0101	**.0335**	.0127	.0094	**.0363**	.0122	.0083	**.0303**
Consistent	**.8220**	.8057	.8190	.7740	.6332	**.7954**	.6740	.6681	**.6769**	.7526	**.7908**	.7748
Action	.7213	.7241	**.7441**	.5317	.3722	**.5654**	.4198	.4126	**.4250**	.4822	.5148	**.5380**
Interaction	.6950	–	**.7113**	.5878	.1969	**.6109**	**.5574**	.5373	.5067	.6411	.6794	**.6831**

Table 3: ***T-LESSA* improves dynamic degree and multiple-object scores while largely preserving overall video quality.** Each score uses its official prompt subset from VBench.

	VBench score										
Method	Subject	Backgr.	Aesthetic	Imaging	Motion	Dynamic	Flicker	Temp. style	Objects	Spatial	Overall
Wan2.1-14B (Unsteered)	**.9305**	.9725	**.6423**	**.6671**	**.9668**	.7917	**.9822**	.2486	.6471	.8013	**.2665**
T-LESSA (Ours)	.9255	**.9726**	.6367	.6583	.9657	**.8056**	.9819	**.2489**	**.7096**	**.8377**	.2638
VideoCrafter2 (Unsteered)	**.9523**	**.9819**	**.6231**	.6548	**.9708**	.7500	**.9790**	.2631	.5038	.3554	**.2867**
T-LESSA (Ours)	.9332	.9736	.6092	**.6629**	.9538	**.8194**	.9719	**.2640**	**.5213**	**.4232**	.2861
CogVideoX-5B (Unsteered)	.9487	.9725	**.6028**	**.6229**	**.9774**	.4306	.9724	**.2465**	.5488	**.5920**	**.2573**
T-LESSA (Ours)	**.9490**	**.9734**	.5752	.5725	.9753	**.5278**	**.9752**	.2395	**.6052**	.5087	.2522

der. We provide more details in Appendix B.4. Finally, we evaluate compositional alignment using the official category-specific scorers from T2V-CompBench and use VBench for video quality.

6 RESULTS

Can T-LESSA Improve Overall Finite-Trace TL Satisfaction? We first evaluate whether *T-LESSA* improves finite-trace TL satisfaction across four TLVBENCH classes. We show that *T-LESSA* visually improves TL satisfaction across the three T2V backbones and their available inference-time guidance baselines in Figure 2. Quantitatively, in Table 1, *T-LESSA* scores above all unsteered backbones. Our average relative TL satisfaction score is 19.7% on Wan2.1-14B, 32.1% on CogVideoX-5B, 20.7% on VideoCrafter2, and 14.9% on AnimateDiff. However, SwitchCraft scores higher under both VLM judges on *TLVB-ORDER* while *T-LESSA* is ahead on NeuS-V. Similarly, VideoGuide is ahead on *TLVB-NEG* under InternVL3-8B and NeuS-V. Though certain inference-guidance baselines show better results on certain categories, *T-LESSA* is the only method that improves every class on every backbone. Still, we observe that we only get marginal NeuS-V improvements on *TLVB-ORDER* compared to other categories. Therefore, to evaluate temporal ordering more precisely, we test *T-LESSA* on other well-known temporal sequence metrics.

Is T-LESSA Really Improving Temporal Ordering? We evaluate *T-LESSA* on two event sequence metrics, StoryEval and TCBench. In Table 4a, *T-LESSA* improves all three metrics and both backbones. TC-Score improves by 7.9% on Wan and 23.2% on CogVideoX. StoryEval improves by 9.7% and 26.6%, while TCR improves by 24.1% and 133.3%. We observe a larger relative TCR gain on CogVideoX due to its low unsteered score of .060, compared with .290 on Wan. These results address the concern raised by the small NeuS-V improvement and show that ***T-LESSA* improves temporal alignment and event ordering**.

Does TL Guidance Improve General Compositional Alignment? While our results show that *T-LESSA* improves TL satisfaction and temporal alignment, we wonder whether *T-LESSA* can improve general compositional alignment. Hence, we evaluate our method on four T2V-CompBench categories. Table 2 shows gains in 14 of the 16 reported backbone-category pairs. *T-LESSA* outperforms CogVideoX-5B in all four categories and Wan2.1-14B, AnimateDiff, and Free2Guide in three of four. These results show that **TL guidance also improves general compositional alignment**.

Are We Sacrificing Video Quality? Although *T-LESSA* meaningfully improves TL satisfaction and temporal and compositional alignment, these increments should not come at the cost of video quality. Therefore, we evaluate generated videos for each VBench dimension. Table 3 shows that

Table 4: ***T-LESSA* improves event sequencing, and its TL objective also steers latents.** (a) Sequence-completion metrics on *TLVB-ORDER*: TC-Score and TCR from TCBench and StoryEval. (b) TempoControl's latent-steering setup with only the objective replaced: temporal accuracy (Acc.), absence (Abs.), and presence (Pres.); for two objects the scorer reports per-video static- and temporal-object success rates (Static, Temp.) instead. TC reg. adds TempoControl's visibility and entropy terms.

(a) Sequence completion

	Wan2.1-14B			CogVideoX-5B		
Method	TC-Score	TCR	StoryEval	TC-Score	TCR	StoryEval
Unsteered	.675	.290	.336	.396	.060	.190
T-LESSA	**.728**	**.360**	**.368**	**.489**	**.140**	**.240**

(b) Latent steering (TempoControl setup)

	One object			Two objects		
Objective	Acc.	Abs.	Pres.	Acc.	Static	Temp.
None (unsteered)	69.94	59.75	80.13	38.68	48.05	30.16
TempoControl	83.31	82.25	**84.38**	55.52	62.80	**48.89**
T-LESSA	**87.25**	**93.50**	81.00	48.08	62.93	34.59
T-LESSA + TC reg.	82.75	81.88	83.63	**55.92**	**68.54**	44.46

T-LESSA improves Dynamic Degree and Multiple Objects on all three backbones. Overall VBench decreases by less than .006 in all cases. Individual dimensions vary by backbone. **Hence, *T-LESSA* improves TL satisfaction, temporal alignment, and general compositional alignment while largely preserving overall video quality.**

What Does Steering Cost? The video generation time with *T-LESSA* only rises by 1.7% on Wan2.1-14B and 13.5% on CogVideoX-5B, and peak allocated memory by 2.96 and 0.19 GiB, respectively. According to our ablation study in the next paragraph, TempoControl takes $4.5\times$ longer and uses $2.2\times$ the peak memory. We provide more detail of both comparisons in Appendix D.

Ablation: Is the TL Objective Just Tied to Attention Steering? We apply the TL objective through cross-attention, so the results may depend on this steering mechanism. Hence, we ask whether the same TL objective can steer diffusion latents. Our study uses TempoControl which steers noisy latents, and we only replace its objective with ours while keeping the rest of its released setup. Table 4b shows that the TL objective raises one-object temporal accuracy from 83.31 to 87.25, mainly by improving absence accuracy. For two objects, adding TempoControl's regularizers yields 55.92, compared with 55.52 for TempoControl. Thus, our TL-based objective can guide latent steering, but does not by itself ensure that every required object appears.

Limitations. First, we do not measure the reliability of the Text-to-TL agent or verify whether humans agree with its translations. We rely on an agentic framework and prompt engineering to translate text into TL specifications, which we can further improve with more rigorous neuro-symbolic methods. Second, our satisfaction score accepts any finite-trace formula that compiles to a deterministic automaton, though both that automaton and the $2^{|\mathsf{AP}_\varphi|}$ label sum grow exponentially. In this paper, we evaluate four specification classes with $|\mathsf{AP}_\varphi| \leq 4$. In future work, we will further expand the set of TL specifications to include more complex ones. Finally, although some metrics (e.g., TC-Score, TCR, and StoryEval) show that *T-LESSA* improves event sequencing and temporal alignment, its NeuS-V scores on the *TLVB-ORDER* category barely move. The limit is in the metric. NeuS-V evaluates order with nested eventualities, which a single frame can satisfy, so a correctly ordered video and one that shows all three events at once score the same. TC-Score, TCR, and StoryEval measure the same orderings on the same videos and all improve.

7 Conclusion

We introduced *T-LESSA*, a training-free, inference-time guidance method for video generation. It guides a video generation process to follow a critical requirement and constraints without fine-tuning, latent optimization, or user-specified temporal masks. *T-LESSA* improves TL satisfaction by 21.8% across four backbones and temporal completion by 14.3% compared to unsteered generation while preserving video quality. In addition, we show that the same TL objective improves latent steering. The results also identify a limit of inference-time guidance. A TL specification specifies what a video must or must not show. However, a frozen T2V model can satisfy the specification only when the required events are realizable through its control channel (e.g., cross-attention). Hence, TL-based guidance requires two components, a differentiable objective that encodes the specification, and a control channel through which its gradient guides the generation. We show that *T-LESSA* separates these two, so that each can be evaluated and improved independently.

ETHICS STATEMENT

In this study, we propose a method that guides video generation to adhere more closely to temporal and logical requirements, which also lowers the barrier to specifying fabricated footage and could enable impersonation or misleading content. We release specifications and evaluation code, and every prompt is a synthetic scene description containing no real individual. No human subjects took part in our research.

AI USE STATEMENT

An LLM is a component of the proposed method: it translates each prompt into a temporal-logic specification (Section B.1). We used generative AI tools to assist with coding and editing the text. We also used generative AI to create a teaser figure and to assist in verifying our proofs. We did not use it to develop our theoretical or conceptual framework or to formulate mathematical claims. The authors take responsibility for the final content of this work, including text, claims, and artifacts produced.

REPRODUCIBILITY STATEMENT

Generation and steering configurations for every backbone and specification class are given in Section B.2, and the verbatim agent and judge prompts in Section B.1 and Section B.3. All 400 TLVBENCH specifications and the qualitative video comparisons are attached in the supplementary material. Code will be released upon publication.

REFERENCES

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

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 18392–18402. IEEE, 2023.

Duygu Ceylan, Chun-Hao P. Huang, and Niloy J. Mitra. Pix2video: Video editing using image diffusion. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 23206–23217, October 2023.

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 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7310–7320. IEEE, 2024.

Minkyu Choi, Harsh Goel, Mohammad Omama, Yunhao Yang, Sahil Shah, and Sandeep Chinchali. Towards neuro-symbolic video understanding. In Proceedings of the European Conference on Computer Vision (ECCV), Milan, Italy, 10 2024. Springer. doi: 10.1007/978-3-031-73229-4_13.

Minkyu Choi, S P Sharan, Harsh Goel, Sahil Shah, and Sandeep Chinchali. We'll fix it in post: Improving text-to-video generation with neuro-symbolic feedback, 2025. URL https://arxiv.org/abs/2504.17180.

Weixi Feng, Jiachen Li, Michael Saxon, Tsu-Jui Fu, Wenhu Chen, and William Yang Wang. TC-bench: Benchmarking temporal compositionality in conditional video generation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), Findings of the Association for Computational Linguistics: ACL 2025, pp. 4638–4662, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.241. URL https://aclanthology.org/2025.findings-acl.241/.

Michal Geyer, Omer Bar Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. In International Conference on Learning Representations, volume 2024, pp. 1608–1620, 2024.

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, 2024.

Google DeepMind. Veo 3.1. https://deepmind.google/models/veo/, 2026. Accessed: 2026-09-12.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. arXiv preprint arXiv:2307.04725, 2023.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. Advances in neural information processing systems, 35:8633–8646, 2022.

Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers, 2022. URL https://arxiv.org/abs/2205.15868.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 21807–21818, June 2024.

Hyeonho Jeong and Jong Chul Ye. Ground-a-video: Zero-shot grounded video editing using text-to-image diffusion models. arXiv preprint arXiv:2310.01107, 2023.

Jaemin Kim, Bryan Sangwoo Kim, and Jong Chul Ye. Free2guide: Training-free text-to-video alignment using image lvlm. In 2025 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 17920–17929, 2025. doi: 10.1109/ICCV51701.2025.01665.

Kling. Kling 3.0. https://kling.ai/, 2026.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. arXiv preprint arXiv:2412.03603, 2024.

Dohun Lee, Bryan Sangwoo Kim, Geon Yeong Park, and Jong Chul Ye. Videoguide: Improving video diffusion models without training through a teacher's guide. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2599–2608, June 2025.

Jialu Li, Shoubin Yu, Han Lin, Jaemin Cho, Jaehong Yoon, and Mohit Bansal. Training-free guidance in text-to-video generation via multimodal planning and structured noise initialization. arXiv preprint arXiv:2504.08641, 2025.

Runtao Liu, Haoyu Wu, Ziqiang Zheng, Chen Wei, Yingqing He, Renjie Pi, and Qifeng Chen. VideoDPO: Omni-preference alignment for video diffusion generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8009–8019, 2025.

Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-p2p: Video editing with cross-attention control. In 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8599–8608. IEEE, 2024.

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.

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. arXiv preprint arXiv:2108.01073, 2021.

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. arXiv preprint arXiv:2302.01329, 2023.

OpenAI. Sora 2 system card. Technical report, OpenAI, September 2025. URL https://cdn.openai.com/pdf/50d5973c-c4ff-4c2d-986f-c72b5d0ff069/sora_2_system_card.pdf.

Pika Labs. Pika ai: Free video generator with scene ingredients, 2026. URL https://pika.art. Accessed: 2026-09-12.

Mihir Prabhudesai, Russell Mendonca, Zheyang Qin, Katerina Fragkiadaki, and Deepak Pathak. Video diffusion alignment via reward gradients. arXiv preprint arXiv:2407.08737, 2024. URL https://arxiv.org/abs/2407.08737.

Fachrina Dewi Puspitasari, Chaoning Zhang, Joseph Cho, Adnan Haider, Noor Ul Eman, Omer Amin, Alexis Mankowski, Muhammad Umair, Jingyao Zheng, Sheng Zheng, Lik-Hang Lee, Caiyan Qin, Tae-Ho Kim, Choong Seon Hong, Yang Yang, and Heng Tao Shen. Sora as a world model? a complete survey on text-to-video generation, 2026. URL https://arxiv.org/abs/2403.05131.

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. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 15932–15942, October 2023.

Qwen Team. Qwen2.5-vl, January 2025. URL https://qwenlm.github.io/blog/qwen2.5-vl/.

Shira Schiber, Ofir Lindenbaum, and Idan Schwartz. Tempocontrol: Temporal attention guidance for text-to-video models, 2025. URL https://arxiv.org/abs/2510.02226.

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), 6 2025. doi: 10.1109/cvpr52734.2025.00786.

Ariel Shaulov, Eitan Shaar, Amit Edenzon, Gal Chechik, and Lior Wolf. Compositional video generation via inference-time guidance. arXiv preprint arXiv:2605.14988, 2026. URL https://arxiv.org/abs/2605.14988.

Kaiyue Sun, Kaiyi Huang, Xian Liu, Yue Wu, Zihan Xu, Zhenguo Li, and Xihui Liu. T2v-compbench: A comprehensive benchmark for compositional text-to-video generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8406–8416, June 2025.

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.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314, 2025.

Yibin Wang, Zhiyu Tan, Junyan Wang, Xiaomeng Yang, Cheng Jin, and Hao Li. LiFT: Leveraging human feedback for text-to-video model alignment. arXiv preprint arXiv:2412.04814, 2024. URL https://arxiv.org/abs/2412.04814.

Yiping Wang, Xuehai He, Kuan Wang, Luyao Ma, Jianwei Yang, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Is your world simulator a good story presenter? a consecutive events-based benchmark for future long video generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 13629–13638, June 2025.

Ziyi Wu, Aliaksandr Siarohin, Willi Menapace, Ivan Skorokhodov, Yuwei Fang, Varnith Chordia, Igor Gilitschenski, and Sergey Tulyakov. Mind the time: Temporally-controlled multi-event video generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 23989–24000, 2025.

Qianxun Xu, Chenxi Song, Yujun Cai, and Chi Zhang. Switchcraft: Training-free multi-event video generation with attention controls. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 29136–29145, June 2026.

Shuai Yang, Yifan Zhou, Ziwei Liu, and Chen Change Loy. Rerender a video: Zero-shot text-guided video-to-video translation. In SIGGRAPH Asia 2023 Conference Papers, pp. 1–11, 2023.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer. In The Thirteenth International Conference on Learning Representations, 2025. URL https://openreview.net/forum?id=LQzN6TRFg9.

David Junhao Zhang, Dongxu Li, Hung Le, Mike Zheng Shou, Caiming Xiong, and Doyen Sahoo. Moonshot: Towards controllable video generation and editing with motion-aware multimodal conditions. International Journal of Computer Vision, 133:3629–3644, 2025. doi: 10.1007/s11263-025-02346-1.

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, 2023.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. arXiv preprint arXiv:2504.10479, 2025.

A Appendix

A.1 Proof of Agreement on Binary Traces

Proof of Theorem 1. Fix the denoising iteration i and regard the label probabilities $\lambda_{i,t,\rho}$ as independent coordinates. Each valuation probability is a product containing either $\lambda_{i,t,\rho}$ or $1-\lambda_{i,t,\rho}$ for each proposition ρ. Each entry of the time-indexed transition matrix is a sum of such products and is therefore multi-affine in the labels at that time. Since transition matrices at distinct times depend on disjoint sets of label variables and each time-indexed matrix appears once in the finite recursion, the resulting satisfaction score is multi-affine in the full collection of label variables.

At a binary input, exactly one valuation has probability one at each time. Consequently, the state distribution remains concentrated on the deterministic automaton trajectory induced by δ_φ. Its terminal state is $\delta^*_\varphi(q^0_\varphi, \sigma)$. The terminal acceptance condition in Equation (5) therefore yields Equation (6). □

A.2 First-Order Analysis of Attention Steering

We analyze the controller's update through the event scores of Equation (3), holding the attention logits fixed. This local analysis ignores how the bias changes those logits, within the same forward pass or when the transformer is rerun.

Let H be the token-event incidence matrix, with $H_{j,\rho} = 1$ when $j \in I_\rho$ and 0 otherwise. For a fixed token-logit vector z and an event-level bias b, define

$$\alpha(b) = \mathrm{softmax}(z + Hb), \qquad m(b) = H^\top \alpha(b).$$

The softmax is taken over the prompt tokens only, so $\mathbf{1}^\top \alpha(b) = 1$. On CogVideoX, z is the text-key block of the joint logits described in Section B.2. The Jacobian from event-level bias to event-attention mass is

$$J(\alpha) = H^\top \left(\mathrm{diag}(\alpha) - \alpha\alpha^\top\right) H. \tag{9}$$

The softmax Jacobian is symmetric positive semidefinite, so $J(\alpha) \succeq 0$ by congruence. Equivalently, for any event-space vector v,

$$v^\top J(\alpha) v = \mathrm{Var}_{j\sim\alpha}((Hv)_j) \geq 0.$$

This property holds even when event token spans overlap.

Equation (3) averages these Jacobians over layer-head pairs and spatial positions, which preserves positive semidefiniteness. With the logits fixed, the event scores at each latent time depend only on the bias at that time, so the Jacobian $\overline{J}_i$ of the event scores with respect to the bias is block diagonal over latent time, with symmetric positive semidefinite blocks. All Jacobians below are evaluated at the current bias B_i. Let

$$g_i = \nabla_\lambda S_\varphi(\lambda)|_{\lambda=\lambda(B_i)}, \qquad \widehat{g}_i = \frac{g_i}{\|g_i\|_\infty},$$

where $\|g_i\|_\infty > 0$. When the label clamp of Section B.2 is inactive, the exact gradient of S_φ with respect to the bias at fixed logits is $\nabla_{B_i} S_\varphi = \overline{J}_i^\top g_i = \overline{J}_i g_i$. The controller instead updates the bias in the direction $\widehat{g}_i$. The following proposition characterizes the first-order effect of this update.

Proposition 1 (First-order behavior at frozen logits)**.** *Hold fixed all attention logits that enter the event scores. Assume that $\|g_i\|_\infty > 0$, that the label clamp is inactive in a neighborhood of B_i, and that bias clipping does not modify the proposed update. Then, as $\eta \to 0^+$,*

$$S_\varphi(\lambda(B_i + \eta\widehat{g}_i)) - S_\varphi(\lambda(B_i)) = \frac{\eta}{\|g_i\|_\infty} g_i^\top \overline{J}_i g_i + O(\eta^2). \tag{10}$$

The directional derivative is nonnegative. If $g_i^\top \overline{J}_i g_i > 0$, the update strictly increases the frozen-logit satisfaction score for sufficiently small positive η.

Proof. With the logits fixed and the label clamp inactive, λ is a smooth function of the bias and satisfies

$$\lambda(B_i + \eta\widehat{g}_i) - \lambda(B_i) = \eta \overline{J}_i \widehat{g}_i + O(\eta^2).$$

Expanding S_φ at $\lambda(B_i)$ and substituting $\widehat{g}_i = g_i/\|g_i\|_\infty$ gives Equation (10). Since $\overline{J}_i \succeq 0$, the first-order coefficient in Equation (10) is nonnegative. When this coefficient is strictly positive, the linear term dominates the remainder for sufficiently small positive η. □

The proposition is a local statement about the event scores at fixed logits. It does not establish improvement for arbitrary finite step sizes or account for label clamping, bias clipping, or changes to downstream attention logits within the same forward pass or when the transformer is rerun. It also provides no guarantee of monotonic improvement across denoising iterations or of specification satisfaction in the generated video.

B Extended Methodology and Experimental Setup

In this section, we add detail on our methodology and experiments.

B.1 Text-to-TL Agent

The *T-LESSA* agent translates a natural-language prompt into a TL specification and a set of atomic propositions. It performs two primary tasks: ❶ *translation*, which decomposes the input prompt into the desired outputs, and ❷ *validation*, which validates the generated outputs against the original prompt. Both tasks use `gpt-5.6-terra` from OpenAI. The prompts used for both tasks:

Translation instruction

```
You are the translation component of the T-LESSA text-to-temporal-logic
agent. Translate each prompt into one executable finite-trace contract and
ground every atomic proposition in an exact, contiguous character span.

Rules:
1. Copy span_text verbatim from the prompt and return its zero-based Python
   slice [char_start:char_end]. Spans must be nonempty, pairwise disjoint,
   and ordered by char_start. Never invent an event or visual condition.
2. Use symbols a,b,c,d in proposition order. Prefer short content-bearing
   noun/action/relation phrases. Do not use articles, punctuation, or the
   entire prompt as a span when a shorter grounded phrase is available.
3. Select exactly one supported contract:
   - eventually_single: one salient visual event/condition occurs;
   - eventually_conj: two to four explicitly requested entities, attributes,
     actions, or relations occur, without an explicit temporal order;
   - globally_single: one explicitly persistent condition or temporal style;
   - globally_conj: two to four explicitly persistent conditions;
   - order: two to four events explicitly ordered by words such as then,
     afterward, before, or after. Do not infer an order from sentence order.
4. The canonical formula must be exactly one of:
   EVENTUALLY(a)
   EVENTUALLY(a) & EVENTUALLY(b) [& EVENTUALLY(c) ...]
   GLOBALLY(a)
   GLOBALLY(a) & GLOBALLY(b) [& GLOBALLY(c) ...]
   FIRST_ONSET(a < b [< c ...])
5. Dimension labels are context about the official VBench split. They do not
   license adding content absent from the prompt.
6. Return exactly one item for every uid supplied.
```

Validation instruction

```
You are the validation component of the T-LESSA text-to-temporal-logic
agent. Translation was performed by a separate model call. Accept a proposed
translation only if all of the following hold:
- every proposition describes explicit visual content in the prompt;
- every span is the correct exact substring for that proposition;
- no important temporal connective in the prompt is contradicted or omitted;
- the selected contract matches the prompt rather than the VBench dimension;
- the formula uses all and only the declared propositions;
- no proposition is redundant, hallucinated, or merely grammatical filler.
Be conservative. Return concise issues for every rejection. Return exactly
one validation item for every uid supplied.
```

We measure success rate on the 517 official VBench prompts. We observe that the first attempt passed 446 (86.3%) of the agent's own validation. All 517 passed after retries and repair. We will release all 400 TLVBench specifications.

Usage of the Agent. We use the agent to create TLVBench to have their own propositions, event phrases, and specifications. In addition, we use the agent on the external benchmarks that do not provide these required input for *T-LESSA*. *TLVB-NEG* and *TLVB-TEMPLOG* add $\Box\neg$ (Table 7).

B.2 Generation Configuration

Table 5 lists the generation configuration for all four backbones. For each prompt, the backbones and *T-LESSA* share the same prompt, seed, initial noise, sampler, and number of denoising steps. *T-LESSA* applies its attention bias during the first 40 of the 50 denoising steps and disables it thereafter. We use the same cutoff for every backbone and specification class, and did not tune or ablate it.

Table 5: **Generation configuration across backbones.** Here, T denotes the number of latent temporal positions indexed by the bias. For each backbone, we use the released default sampler. Note that AnimateDiff advances the latent with a deterministic update and therefore does not use η.

Backbone	Resolution	Frames	T	Steps	Sampler	CFG
Wan2.1-T2V-14B	480×832	81	21	50	UniPC, flow shift 3.0	5.0
CogVideoX-5B	480×720	49	13	50	DDIM, v-prediction	6.0
VideoCrafter2	320×512	16	16	50	DDIM, η=1.0	12.0
AnimateDiff	512×512	16	16	50	deterministic x_0 update	7.5

Bias parameterization and update. The bias $B \in \mathbb{R}^{T \times |\mathsf{AP}_\varphi|}$ stores one float32 value per latent temporal position and proposition, which we add to the attention logits of every prompt token in the corresponding proposition. At each guided step, we first clip λ to $[10^{-4}, 1 - 10^{-4}]$ and normalize ∇S_φ by its ℓ_∞ norm. We then update B with step size η and clip each entry to $[-b_{\max}, b_{\max}]$. The bias does not vary by layer or head. The specification says when an event should occur, not which layer should attend to it.

Layer subsampling. Importantly, the controller does not need to read every cross-attention layer. On Wan2.1-14B *TLVB-ORDER*, it observes every fourth layer (ten of forty), whereas all remaining runs use the default of observing every layer. Subsampling applies to reading only. On that run we still add the bias at all forty layers.

Joint attention on CogVideoX. CogVideoX runs one joint attention over the concatenated text and video tokens instead of a separate cross-attention. We take the video-query, text-key block of its logits as ℓ and add B to that block only, on the conditional branch of classifier-free guidance. Equation (3) then normalizes over text keys, so α is the share of a video token's prompt attention that falls on the span. The true attention weight is α times the total mass on text keys. The bias raises that mass as well, and the readout does not track it. On the other three backbones the keys are text only, so α is the attention weight itself.

Steering hyperparameters. Table 6 lists the steering hyperparameters. Note that η here is the step size on B, not the sampler's η in Table 5. Finally, negation cases use a wider bound, since suppressing attention to an event named in the prompt requires a larger negative bias.

Table 6: **Steering hyperparameters.** η is the step size on B and $b_{\max}$ the bound each entry of B is clipped to. On Wan2.1-14B, *TLVB-TEMPLOG* bounds its excluded event separately from its two ordered events; on CogVideoX-5B all three events share one bound.

Class	Backbone	η	$b_{\max}$
TLVB-ORDER, *TLVB-CONJ*	Wan2.1-14B, CogVideoX-5B	.15	3
all four	VideoCrafter2, AnimateDiff	.15	3
TLVB-NEG	Wan2.1-14B	.30	6
	CogVideoX-5B	.15	3
TLVB-TEMPLOG	Wan2.1-14B	.30	3 ordered, 6 excluded
	CogVideoX-5B	.15	3

Each guidance baseline runs on the backbone above that it was released for, and Section C.3 states what each one changes on top of that configuration. TempoControl is the exception: it appears only in the latent-steering transfer experiment of Table 4b and uses Wan1.3B.

B.3 VLM Evaluation Prompts

Qwen2.5-VL-7B and InternVL3-8B receive the same eight uniformly sampled frames, each bicubically resized to 448×448, and the same yes–no question, so a score difference between the two judges reflects the model. We use one prompt for event presence and one for three-event ordering. The ordering prompt is asked twice per video, once with the events in the specified order and once with them reversed, and we report the difference between the two (Equation (11)).

Event presence prompt

```
The 8 images are frames from one video. Is "[event]" clearly visible in the
video? Answer yes or no.
```

Three-event ordering prompt

```
The 8 images are frames sampled in chronological order from one video (frame
1 = earliest, frame 8 = latest). In the video, do these events occur IN THIS
TEMPORAL ORDER: first "[A]", then later "[B]", then finally "[C]"?
Answer yes or no.
```

Writing $P_{\rightarrow}$ and $P_{\leftarrow}$ for the two resulting $P(\text{Yes})$ values, the *TLVB-ORDER* score is their difference:

$$S_{\text{order}} = P_{\rightarrow} - P_{\leftarrow}. \tag{11}$$

A video that merely contains all three events, with no clear progression, tends to draw the same answer to both questions and therefore scores near zero. Subtracting the reverse reading removes the credit a video would otherwise receive for presence alone, which matters because on Wan2.1-14B every generation whose three events are all present is also judged correctly ordered. The reversed question can also be answered in part from the prompt alone, since some event sequences are implausible in reverse, so an unsteered video's S_{order} is not zero.

B.4 TCBench Transition Completion

TCBench scores a transition $A \rightarrow B \rightarrow C$ with four assertions over sixteen uniformly sampled frames. The first asks whether A is present in frame 1. The second asks whether frame 16 shows C, relative to frame 1. The third asks whether frame 9 shows B as a sign of the transition in progress. The fourth shows frames 1, 5, 9, 13 and 16 together and asks whether they form a consistent transition from A to C. The released benchmark judges these with GPT-4-Turbo, which is deprecated, so we use GPT-4o. It answers each assertion at temperature 0. We map yes to 1, no to 0, and any other reply to 0.5. TC-Score is the mean of the four values. TCR is 1 when all four are at least 0.5 and 0 otherwise, so one no fails a video while an ambiguous reply does not.

C Additional detail on TLVBench and objective functions of TL formula

In this section, we describe the specification classes of TLVBench, and their objective functions that we use for steering.

C.1 TLVBench Specification Classes

TLVBench contains 400 prompts across four specification classes. Table 7 lists the differentiable objective used for inference-time steering and the finite-trace formula used for symbolic evaluation in each class. These objectives and evaluation formulas are related but do not always impose the same requirements.

For *TLVB-NEG*, steering combines the presence of the retained event A_{keep} with the absence of the prohibited event C, whereas symbolic evaluation checks only the absence of C. For *TLVB-CONJ*, steering uses the presence product $S_{\Diamond A} S_{\Diamond B}$, which rewards both events appearing somewhere in the trace. NeuS-V instead evaluates $\Diamond(A \wedge B)$, which requires both events to occur in the same frame. Thus, a trace containing A and B only in separate frames satisfies the presence requirement but fails the co-occurrence requirement. The VLM judges multiply the two event-presence scores, providing a proxy for the presence requirement. Section C.2 compares steering with the presence product against steering directly for co-occurrence.

Table 7: **TLVBENCH tests logic, time, and their composition.** A_{keep} is the event retained by the *TLVB-NEG* counter-prompt, and $\varphi_{\prec}$ is the priority-phase formula of Equation (13).

Family	Class	Steering objective $S(\lambda)$	Symbolic evaluation φ
Logical	*TLVB-NEG*	$S_{\Diamond A_{\text{keep}}}\, S_{\Box\neg C}$	$\Box\neg C$
Logical	*TLVB-CONJ*	$S_{\Diamond A}\, S_{\Diamond B}$	$\Diamond(A \wedge B)$
Temporal	*TLVB-ORDER*	$S_{\varphi_{\prec}(A,B,C)}$	$\Diamond(A \wedge \Diamond(B \wedge \Diamond C))$
Temporal-logical	*TLVB-TEMPLOG*	$S_{\varphi_{\prec}(A,B)}\, S_{\Box\neg C}$	$\Diamond(A \wedge \Diamond B) \wedge \Box\neg C$

In Table 7, $S_{\psi}(\lambda)$ denotes the satisfaction score defined in Equation (5). Under a model in which Bernoulli proposition labels are independent across frames and propositions, the single-event scores have the closed forms

$$S_{\Diamond A}(\lambda) = 1 - \prod_{t=1}^{T}(1-\lambda_{t,A}), \qquad S_{\Box\neg C}(\lambda) = \prod_{t=1}^{T}(1-\lambda_{t,C}).$$

Under this independence assumption, products of satisfaction scores for formulas involving disjoint propositions equal the satisfaction probabilities of their conjunctions.

For *TLVB-ORDER* and *TLVB-TEMPLOG*, steering uses the priority-phase formula $\varphi_{\prec}$ defined in Equation (13). This encoding requires the occurrences that advance successive phases to occupy strictly increasing frame indices and restricts which events may occur before each phase transition. It does not require strictly ordered first appearances: simultaneous events can be admitted under the priority rule described in Section C.4. Symbolic evaluation instead uses nested eventualities, which allow successive requirements to be satisfied in the same frame and do not impose the priority-phase restrictions on earlier events.

C.2 STEERING THE EXACT CO-OCCURRENCE OBJECTIVE

For *TLVB-CONJ*, the default steering objective is the presence product $S_{\Diamond A}S_{\Diamond B}$, which rewards both events appearing somewhere in the trace without requiring them to co-occur. Under the independence assumption described in Section C.1, this product equals the satisfaction probability of $\Diamond A \wedge \Diamond B$. The VLM judges likewise use a product of event-presence scores, whereas NeuS-V evaluates the stricter formula $\Diamond(A \wedge B)$.

To assess whether matching the steering objective to the stricter symbolic evaluation improves performance, we also steer directly with

$$S_{\Diamond(A\wedge B)}(\lambda) = 1 - \prod_{t=1}^{T}\big(1-\lambda_{t,A}\lambda_{t,B}\big). \tag{12}$$

Under the same independence assumption, this expression is the exact satisfaction probability of the co-occurrence formula. On binary traces, it equals $\mathbf{1}\{\sigma \models \Diamond(A \wedge B)\}$.

We compare unsteered generation, presence-product steering, and co-occurrence steering on all 100 *TLVB-CONJ* prompts, using three seeds per prompt on each backbone. The two steered conditions differ only in their steering objective.

Table 8 reports results under both VLM judges and NeuS-V. Co-occurrence steering yields higher mean scores than unsteered generation in all six backbone–judge combinations. However, its mean scores are consistently lower than those obtained with presence-product steering, including under NeuS-V. All six confidence intervals for the exact-minus-presence difference include zero, and none of the six paired comparisons is statistically significant after Holm adjustment. Thus, this experiment provides no evidence that directly steering for co-occurrence outperforms the presence product, but it does not establish equivalence between the two objectives.

Table 8: **Direct co-occurrence steering does not show a statistically significant advantage over presence-product steering.** Results use all 100 *TLVB-CONJ* prompts with three seeds per prompt on each backbone. Δ denotes the mean score difference between co-occurrence and presence-product steering. Confidence intervals are reported at the 95% level; p-values are from paired t-tests with Holm adjustment across the six backbone–judge comparisons.

		Mean score			Exact − presence	
Backbone	Judge	Unsteered	Presence	Exact	Δ [95% CI]	p
Wan2.1-14B	Qwen2.5-VL-7B	.7667	.8376	.8203	$-.0173$ $[-.0367, +.0007]$	.42
	InternVL3-8B	.8492	.9316	.9205	$-.0111$ $[-.0298, +.0051]$	.71
	NeuS-V	.4730	.5202	.5116	$-.0086$ $[-.0204, +.0030]$	.61
CogVideoX-5B	Qwen2.5-VL-7B	.3813	.5119	.5026	$-.0093$ $[-.0284, +.0096]$	.71
	InternVL3-8B	.5582	.6932	.6795	$-.0138$ $[-.0286, +.0008]$	.42
	NeuS-V	.2687	.3282	.3224	$-.0057$ $[-.0201, +.0090]$	.71

Table 9: ***T-LESSA* is the only method whose interface states all four TLVBENCH categories directly.** ✓ marks a requirement the method's interface states directly, $\sim$ one it states only approximately, and $\times$ one it cannot state.

Method	*TLVB-NEG*	*TLVB-CONJ*	*TLVB-ORDER*	*TLVB-TEMPLOG*
T-LESSA	✓	✓	✓	✓
SwitchCraft	$\times$	$\times$	✓	$\times$
TempoControl	$\sim$	✓	$\sim$	$\times$
Video-MSG	$\times$	$\sim$	$\sim$	$\times$
Free2Guide	$\times$	$\sim$	$\sim$	$\times$
VideoGuide	$\sim$	$\sim$	$\sim$	$\sim$

C.3 RUNNING THE INFERENCE-TIME GUIDANCE BASELINES

For each baseline in Table 1, we use the released code with the default settings without changing any hyperparameter. Note that Table 9 shows what each interface-time guidance method can state, which differs from what we ran. A method without an interface can still be run on that class's prompts, as it receives the text prompt. However, if we do that, the result measures generic guidance rather than scoring for the specification. We use VideoGuide on all four classes, because it is the only baseline whose backbone we also control in the unsteered and *T-LESSA* pairs, For the remaining baselines, we run only the classes that their interfaces address and leave the other cells empty.

SwitchCraft. SwitchCraft takes a single prompt, an ordered list of verbatim event substrings, and a per-event length ratio, and edits cross-attention queries in the early blocks during the early denoising steps. Consequently, it can directly state *TLVB-ORDER* but none of the other three classes. First, every Wan2.1-14B *TLVB-ORDER* number in Table 1, was generated and scored in a single run, so the three rows share a generator, a seed set, and a frozen evaluator. Second, SwitchCraft runs the original Wan repository rather than the Diffusers pipeline that we use, and the two differ in scheduler shift and negative prompt. To this end, we additionally generate the same prompts and seeds through SwitchCraft's own pipeline with steering disabled (`--EDIT_UNTIL_STEP 0`). This control scores 0.520 and 0.659 on the two VLM judges, which is below the shared unsteered arm at 0.569 and 0.701. Hence, SwitchCraft's advantage is not an artifact of its pipeline defaults. Overall, SwitchCraft leads *T-LESSA* by $+0.048$ under Qwen ($p = 0.02$) and by $+0.012$ under InternVL3 ($p = 0.52$). That is, it is ahead under one judge and on par under the other, and we report this result as is.

VideoGuide. VideoGuide samples with AnimateDiff, while a second model provides denoised latents during the early steps. It accepts a plain prompt and exposes no interface for stating a specification beyond that text. Therefore, it consumes exactly the same input as our unsteered backbone. We use default setting, mode 1 against mode 0, where the base and guided modes use CFG 7.5 and 0.8, respectively, with `cfg_plus` enabled for the guided mode. Note that our unsteered and *T-LESSA* pairs on this backbone use the same mode-0 substrate at CFG 7.5.

Free2Guide. Free2Guide scores candidate continuations with an image-conditioned LLM and keeps the best, using 6 guidance steps and 10 candidates per step, or 60 judge calls per video. Its reward is a scalar preference over whole candidates, so it cannot require an event to be absent. We still run it on VideoCrafter2 for all four classes, and on T2V-CompBench. It matches the unsteered backbone on the two absence classes, scoring .436 and .405 against .452 and .410 on *TLVB-NEG*, and .137 and .290 against .139 and .286 on *TLVB-TEMPLOG* under the two VLM judges.

Video-MSG. Video-MSG plans first before sampling. Specifically, a language model first drafts a background and a per-frame object plan, converted into a spatiotemporal sketch that conditions CogVideoX-5B. The plan is generated from the prompt alone, without access to the specification.

C.4 Priority-Phase Encoding

We steer each *TLVB-ORDER* prompt with a priority-phase encoding. For a three-event sequence $A \to B \to C$, the automaton has waiting states q_A, q_B, q_C, an accepting state q_{sat}, and a rejecting state q_{fail}. It begins in q_A and advances by at most one state per frame: an occurrence of A advances q_A to q_B, a subsequent occurrence of B advances q_B to q_C, and a subsequent occurrence of C reaches q_{sat}.

At each waiting state, the awaited event takes priority over other events in the same frame. If the awaited event is absent but an event from a later phase occurs, the automaton enters q_{fail}. Events from completed phases may recur without causing rejection. If neither the awaited event nor a later-phase event occurs, the automaton remains in its current state.

For example, observing A and B together in q_A advances the automaton only to q_B; another occurrence of B in a later frame is required to reach q_C. Thus, the encoding requires strictly ordered phase transitions, but does not require strictly ordered first appearances of the events.

Under finite-trace semantics with strong next, this encoding is equivalent to

$$(\neg A \wedge \neg B \wedge \neg C)\,\mathsf{U}\,(A \wedge \mathsf{X}\,((\neg B \wedge \neg C)\,\mathsf{U}\,(B \wedge \mathsf{X}\Diamond C)))\,. \tag{13}$$

The initial until clause excludes all three events before the first A. After that transition, the second until clause permits additional occurrences of A but excludes B and C until the transition triggered by B. The automaton then waits for C. The X operators require consecutive phase transitions to occur in different frames. Because the exclusion conditions apply before each transition, later-phase events may co-occur with the awaited event on a transition frame.

The construction extends to the two-to-four-event sequences in StoryEval by using one waiting state per event, with the same priority and rejection rules.

Relation to nested eventualities. The formula $\Diamond(A \wedge \Diamond(B \wedge \Diamond C))$ expresses non-strict temporal order: it requires positions $t_A \le t_B \le t_C$ at which A, B, and C hold, respectively. Unlike the priority-phase encoding, it permits the same frame to satisfy multiple successive requirements and does not reject earlier occurrences of later events.

A faithful evaluation of this formula preserves the position dependence of each nested subformula. Replacing these position-dependent values with whole-video presence scores discards temporal information and yields an order-insensitive surrogate. This loss of order arises from the pooling operation, rather than from the semantics of nested eventualities. Steering with the pooled surrogate on the same 100 prompts and three seeds changes nothing: .563 and .691 on the two VLM judges against an unsteered .569 and .701.

D Extended Results

In this section, we report the runtime and memory cost of steering and a runtime comparison against latent-optimization guidance. We also state how to calculate the quantitative claims in the paper.

Table 10: ***T-LESSA* only adds a small, backbone-dependent overhead.** This table shows generation time and peak allocated memory for *T-LESSA* and unsteered backbones. It averages 300 *TLVB-CONJ* videos. We run Wan2.1-14B on an A100 80 GB , while CogVideoX-5B ran on an RTX A5000 24 GB. We measure Peak memory (`max_memory_allocated`) after model load *T-LESSA* does not require any additional model forward passes, and each denoising step adds one softmax per cross-attention layer and one gradient step.

	Time / video (s)			Peak memory (GiB)		
Backbone	Unsteered	T-LESSA	Δ Time	Unsteered	T-LESSA	Δ Mem
Wan2.1-14B	1150.8	1170.8	+20.0 (+1.7%)	41.94	44.90	+2.96
CogVideoX-5B	548.1	622.0	+73.9 (+13.5%)	11.86	12.05	+0.19

D.1 Comparison with TempoControl

TempoControl's ascent backpropagates its objective through the full backbone at every guided step. However, *T-LESSA* reads attention during forward pass and updates a small per-event bias. Specifically, *T-LESSA* requires 150 model forward passes to generate video using Wan1.3B while TempoControl needs 370 forward passes. Table 11 shows the runtime cost between two methods: *T-LESSA* adds 0.9% to generation time and no peak memory, while TempoControl takes $4.5\times$ longer and uses $2.2\times$ the peak memory. This comparison uses three prompts.

Table 11: **Latent optimization costs $4.5\times$ the time and $2.2\times$ the memory of attention steering.**

Method	Time / video (s)	Peak memory (GiB)
Unsteered	761.6	8.71
TempoControl	3461.6	19.02
T-LESSA (Ours)	768.4	8.71

D.2 How to Calculate the Quantitative Claim

The two headline numbers use different aggregations, so we state both.

The 21.8% TL satisfaction gain is the mean of the per-cell relative gains $(\text{ours} - \text{unsteered})/\text{unsteered}$ over all 48 cells of Table 1, that is four backbones by four classes by three judges. By backbone it is 19.7% on Wan2.1-14B, 32.1% on CogVideoX-5B, 20.7% on VideoCrafter2 and 14.9% on AnimateDiff.

The 14.3% temporal completion gain instead pools TC-Score and StoryEval over both backbones in Table 4a as $\sum(\text{ours} - \text{unsteered})/\sum \text{unsteered} = .228/1.597$. It leaves out TCR, whose unsteered CogVideoX score of .060 turns a .080 gain into a 133% relative improvement that would dominate an average over metrics.