Early Failure Detection and Intervention in Video Diffusion Models

Byungki Kwon[1] **Sohwi Lim**[2] **Nam Hyeon-Woo**[1] **Moon Ye-Bin**[1] **Tae-Hyun Oh**[2]

[1]Pohang University of Science and Technology (POSTECH)
[2]Korea Advanced Institute of Science and Technology (KAIST)

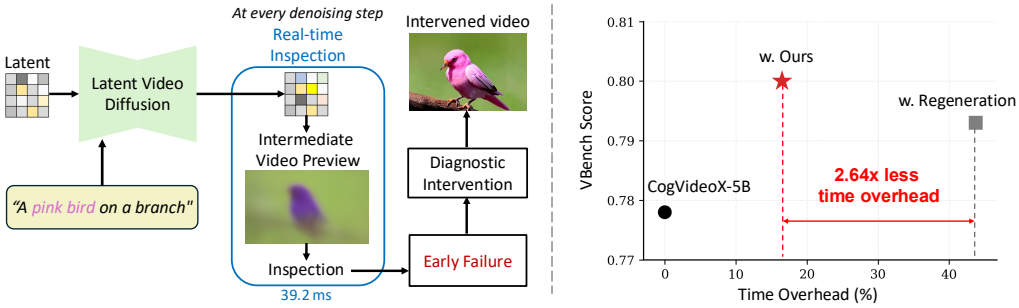


Figure 1: **Real-time inspection enables early failure detection in latent video diffusion.** By decoding intermediate latents at every denoising step, our method identifies failing trajectories early and selectively intervenes. Compared to post-hoc regeneration, we achieve comparable VBench scores with 2.64× lower time overhead.

Abstract

Text-to-video (T2V) diffusion models have rapidly advanced, yet generations still occasionally fail, forcing users into trial-and-error regeneration that wastes substantial compute and energy. While early failure detection has been explored for image diffusion, it remains unaddressed in on-the-fly, single-trajectory video diffusion due to the prohibitive cost of decoding spatiotemporal latents at every step. The VAE decoder of CogVideoX-5B takes approximately 4 s per call on an A100, comparable to a denoising step, rendering per-step inspection impractical. We propose an *early failure detection and intervention pipeline* for video diffusion that makes real-time, step-wise RGB inspection of sampling tractable. Our approach rests on a key empirical observation: *the final video quality is largely predictable within the first ∼10 of 50 denoising steps*, and is reliably detectable from intermediate latents once they are decoded into RGB and scored by standard vision–language evaluators. To exploit this on the fly, we introduce a *Real-time Inspection* (RI) module that scores intermediate previews in 39.2 ms, over 100× faster than a CogVideoX-5B denoising step. The key enabler is a *Latent-to-RGB* (L2R) converter (0.059 M parameters, 19.7 ms) showing that learned spatial upsampling alone without channel expansion suffices to produce plausible video previews 200× faster than the native VAE while preserving the semantic structure needed for scoring. When a failure is predicted, a diagnostic intervention framework hierarchically applies preview-driven corrections, escalating compute only when needed. Experiments on CogVideoX-5B and Wan2.1-1.3B show consistent improvements and a favorable cost-quality trade-off against full-video feedback regeneration, a strong post-hoc baseline that inspects complete videos, with ∼2.5× lower additional time overhead. Our method is also orthogonal to prompt refinement and training-free sampling guidance. Project page: `https://byungki-k.github.io/EarlyFailure/`

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Text-to-video (T2V) diffusion models [51, 43, 21, 8, 5, 25, 12, 2, 47] have rapidly advanced, yet generations still occasionally fail with weak text–video alignment, poor perceptual quality, or temporal inconsistency [20, 17, 24, 49, 40]. Such failures arise from missing semantic coverage or unlucky sampling trajectories [39, 36, 37, 50, 34], and are difficult to predict in advance, incurring substantial compute and energy cost as users iterate toward a satisfactory output [27].

A natural alternative is to monitor sampling and intervene early. For image diffusion, this has been explored via efficient inspection of intermediate latents in latent or model-internal space, either to evaluate alignment [48, 35, 46] or to steer generation [1, 27, 19, 34]. The same approach is not directly viable for video: the native VAE decoder of CogVideoX-5B takes approximately 4 s per intermediate latent on an A100, comparable to a single denoising step. Prior video-domain work either defers inspection until after full generation [10] or distributes it across parallel trajectories [22, 30, 3], leaving on-the-fly, single-trajectory failure detection largely impractical.

We address this gap by enabling on-the-fly failure detection during video diffusion sampling. Our approach is grounded in a key empirical observation: when intermediate latents are decoded into RGB and scored by an off-the-shelf vision–language evaluator, the resulting score reliably predicts final video quality within the first $\sim$10 of 50 denoising steps. While early structural convergence has been visualized in image diffusion as a generation property [23, 52], prior on-the-fly intervention systems rely on model-internal signals such as cross-attention maps [7, 16], denoiser activations [11], or small latent-space evaluators [31]; even systems that decode predicted latents to pixel space combine them with cross-attention features and a custom-trained hallucination predictor rather than off-the-shelf VLM scoring [4]. Concurrent video-domain work targets adjacent but distinct goals: inference-time scaling that ranks parallel candidates with fine-tuned verifiers [3], and human-in-the-loop interactive previews via lightweight multi-branch preview decoders [15], neither addressing on-the-fly, automated, single-trajectory failure detection within a per-step latency budget. We provide the first system that operates in this regime, and design it to make zero-shot, off-the-shelf VLM scoring on RGB previews actionable within that budget.

To realize this, we propose an *early failure detection and intervention pipeline* with two components: a *Real-time Inspection (RI)* module that scores intermediate previews in 39.2 ms ($100\times$ faster than a denoising step of CogVideoX-5B), enabled by a *Latent-to-RGB (L2R) converter* (0.059 M parameters, 19.7 ms; $200\times$ faster than the native VAE); and a *diagnostic intervention framework* that applies mode-specific corrections only when needed.

We validate on CogVideoX-5B [51] and Wan2.1-1.3B [43], achieving consistent VBench gains at $\sim 2.5\times$ lower time overhead than full-video feedback regeneration, a strong post-hoc baseline. The framework is orthogonal to prompt rewriting and sampling guidance, and remains effective on Wan2.1-14B at 720p, 81 frames. By substituting on-the-fly correction for post-hoc regeneration, our framework points toward reducing the compute and energy spent on unsatisfactory generations. Our contributions: (i) the first *single-trajectory, on-the-fly* failure detection and intervention system for video diffusion, operating within a per-step budget via zero-shot, off-the-shelf VLM scoring on RGB previews; (ii) a 0.059 M-parameter L2R converter showing that learned spatial upsampling without channel expansion suffices to produce plausible previews $200\times$ faster than the native VAE; and (iii) a favorable cost–quality trade-off against feedback regeneration, with orthogonal gains atop prompt rewriting and sampling guidance.

2 Related work

Evaluation and failure diagnosis in diffusion models. Prior work analyzes intermediate diffusion states for image generation through model-internal signals such as cross-attention maps [7, 16], denoiser activations [11], or small latent-space evaluators [31], used either to evaluate alignment with the prompt [35, 48] or to steer sampling toward better outcomes [1, 27, 19, 34]. HEaD [4] combines intermediate cross-attention maps with decoded predicted-final-image previews, scored by a custom-trained hallucination predictor, to support early stopping; DiffDoctor [45] trains artifact detectors for offline failure diagnosis. These approaches operate primarily in image latents or model-internal space, and their direct extension to video is hindered by the prohibitive cost of decoding spatiotemporal latents at every step (Sec. 1). In video diffusion, prior work treats failure handling as either a

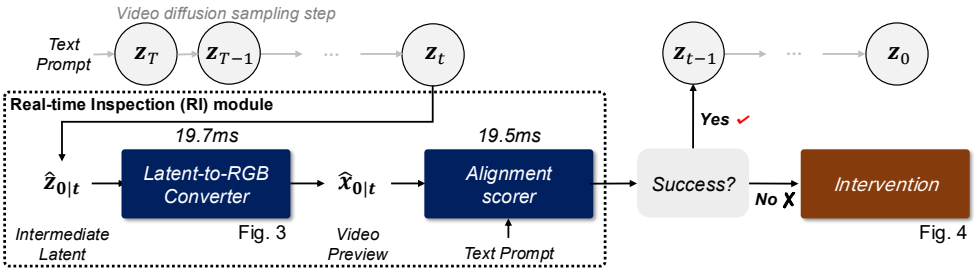


Figure 2: **Early failure detection and intervention pipeline.** The Real-time Inspection (RI) module converts predicted clean latents into RGB video previews and evaluates their semantic alignment score with the input prompt (19.7 ms + 19.5 ms). Intervention is triggered based on the score.

post-hoc problem [10] or a multi-candidate search problem that maintains parallel trajectories with intermediate evaluation [22, 30]. Extending this search paradigm, Baraldi et al. [3] characterize the temporal coherence of VLM-based verifiers on decoded intermediate videos and propose verifier fine-tuning for inference-time scaling, while Hong et al. [15] take a different direction with lightweight multi-branch preview decoders for human-interactive guidance. Concurrent works further pursue the search paradigm: LatSearch [53] learns a latent reward model for reward-guided resampling, and Tang et al. [41] score parallel candidates at intermediate checkpoints. In contrast, we operate on a single trajectory with zero-shot, off-the-shelf VLM scoring, requiring no trained reward model or verifier, enabling on-the-fly intervention with 20.3% additional overhead.

Sampling guidance for video diffusion. Following classifier-free guidance [14], several methods steer video diffusion sampling by exploiting temporal characteristics: modifying attention [1], skipping spatiotemporal layers [18], suppressing noise variance [26], or guiding flow coherence [38]. While effective on average, these methods do not guarantee success on every prompt or sampling trajectory. Our pipeline is orthogonal to them: it can be combined with any of these guidance methods to selectively intervene only on the residual failure cases.

3 Method

We address failure handling in latent text-to-video diffusion as a system-level problem under a strict per-step latency budget. The native VAE decoder of CogVideoX-5B, for instance, costs approximately 4 s per call on an A100, comparable to a single denoising step, so any inspection mechanism that operates inside the sampling loop must be substantially cheaper than this. Our framework (Fig. 2) is built around two components designed to fit this budget: a Real-time Inspection (RI) module (Sec. 3.1) that scores intermediate previews in 39.2 ms, and a diagnostic intervention framework (Sec. 3.2) that escalates compute only when a failure is predicted. Together they replace full-video feedback regeneration with cheap per-step inspection (39.2 ms per step) and selective intervention on flagged samples, keeping the dataset-averaged overhead within 20.3%.

3.1 Real-time Inspection (RI) module

The RI module operationalizes the empirical premise discussed above (validated in Sec. 4.5, Fig. 7): RGB-decoded previews enable plug-in use of pretrained VLMs in two roles, an off-the-shelf scorer for failure detection and a reasoner for Visual Feedback Intervention (Sec. 3.2); neither role is available in latent space without a learned reward model [53]. We realize this within the per-step budget through three components introduced below: a Latent-to-RGB (L2R) converter that produces previews 200× faster than the native VAE, an alignment scorer (19.5 ms), and a dynamic failure detector that adapts to per-sample semantic convergence.

Latent-to-RGB (L2R) converter. We design L2R as a purpose-built converter for the inspection role, rather than reusing any existing decoder. Its design rests on two principles from prior work: latent-to-RGB mapping can be largely linear [29], and learned upsampling restores spatial detail more faithfully than bilinear interpolation [42].

Building on these principles, our architecture (Fig. 3) makes two deliberate omissions justified by the inspection role. First, **no channel expansion**: unlike standard latent video decoders that

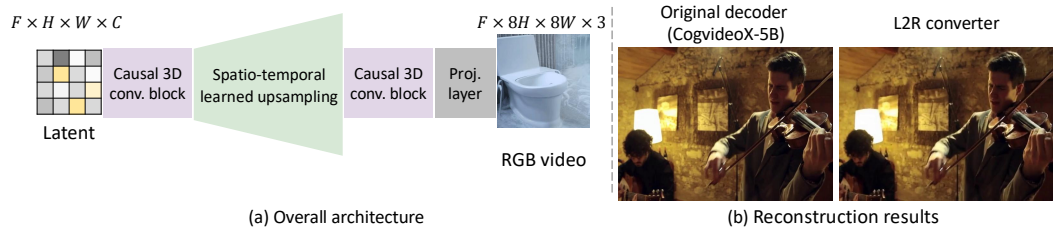


Figure 3: **Real-time Latent-to-RGB (L2R) converter.** (a) Our L2R architecture maps latents to an RGB video using causal 3D convolutional blocks, spatio-temporal learned upsampling, and a projection layer. It runs in 19.7 ms with only 0.059 M parameters. (b) The reconstruction result shows reasonable visual quality compared to the result produced by the original decoder.

expand intermediate features several-fold to recover RGB fidelity, all our intermediate features stay at the latent channel dimension and reduce to three channels only at the final projection layer. Second, **no temporal upsampling**: video quality metrics are more sensitive to spatial than temporal resolutions (Appendix C.4), so we leave the temporal axis at the latent's frame count. The remaining components are minimal: two causal 3D convolution blocks for temporal context and a spatio-temporal learned upsampling module that recovers the $8\times$ spatial compression, followed by the projection layer. The result is a 0.059 M-parameter module that produces video previews for 480p, 49-frame video generations in 19.7 ms, over $200\times$ faster than the native CogVideoX VAE, with $1700\times$ smaller parameter memory (0.138 MB vs. 242.5 MB; Table 5). L2R achieves reconstruction quality comparable to or better than TinyAE [6] while approaching Turbo-VAED [54], using $190\times$ and $690\times$ fewer parameters respectively (Appendix A.2). This confirms our design hypothesis: for preview-based inspection rather than pixel-perfect reconstruction, learned spatial upsampling alone, without channel expansion, suffices to produce plausible previews while preserving the semantic structure needed for VLM-based scoring (validated in Sec. 4.5). We provide the training details in Appendix A.2.

Alignment scorer. Within the per-step latency budget, the choice of scorer is otherwise flexible. We adopt ViCLIP [44] as our default for two reasons: (i) unlike image–text models, ViCLIP is purpose-built for video–text alignment and captures temporal semantics relevant to video previews, and (ii) it is sufficiently lightweight to fit our per-step budget. Our framework is not tied to ViCLIP: Sec. 4.3 validates the pipeline with OpenCLIP, indicating that the scorer is a swappable component rather than a hard dependency. To meet the budget, we move frame preprocessing and similarity computation onto the GPU and cache the text embedding once per sample, since the prompt is fixed throughout sampling. Together these optimizations reduce scoring to 19.5 ms per call, comparable to L2R itself. We denote by s_k the alignment score it produces at denoising step k.

Dynamic failure detector. Diffusion sampling is known to converge sample-dependently [17, 28], and we observe the same effect in semantic content (Appendix A.3): some samples crystallize their content within a few steps while others remain ambiguous longer. A fixed-step predictor would either miss late-converging samples or waste budget on early-converging ones. We instead aggregate the score sequence $\{s_k\}$ over a short inspection window and emit a single predicted final score $\tilde{s}_0$ via a sample-adaptive policy based on the score's level, recent trend, and local stability, so that confident trajectories exit early and ambiguous ones continue until their scores stabilize (full rules in Appendix A.3). A sample is classified as a failure if $\tilde{s}_0 < \tau$. We set $\tau = 0.22$ for ViCLIP based on the score distribution of *real* videos (Fig. 8), and validate this choice with a 26-participant user study at score levels 0.18, 0.22, and 0.26. The same procedure yields $\tau = 0.30$ when OpenCLIP is used as the scorer (Appendix A.5). Across both models, the detector reaches the dynamic decision at step 11/50 on average for CogVideoX and 10/50 for Wan2.1.

3.2 Diagnostic intervention framework

When the detector flags a sample, we apply a hierarchical sequence of interventions, escalating compute only when needed (Fig. 4). The order, from cheapest to most expensive, is: continue the base generation (Trial 0); inject a single-frame semantic anchor (Trial 1); refine the prompt from the failed preview with a VLM and restart (Visual Feedback Intervention, Trial 2); and apply Trial 1 on

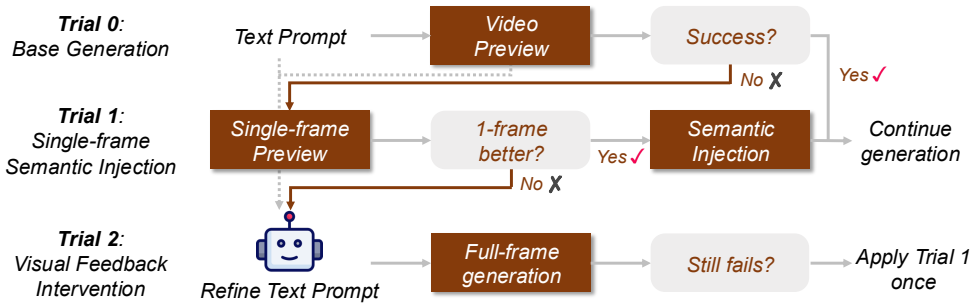


Figure 4: **Diagnostic intervention framework.** If the generation is predicted to succeed, Trial 0 continues to generate video without intervention. When failure is predicted, the next Trials are applied, escalating from inexpensive to more costly fixes. Trial 1 performs a single-frame probe and injects its latent as a semantic anchor; Trial 2 invokes a VLM as a visual feedback reasoner over the failed preview to refine the prompt. If Trial 2 also fails, Trial 1 is attempted once more.

top of the refined prompt if it still fails (Trial 2→1). Branching is observation-driven: each transition is decided from the preview signals already available at the inspection step, so most samples terminate before reaching the more expensive trials. Even in the worst case, the cumulative cost is about 56% of a full second generation (Fig. 6).

Trial 0. Base Generation. If the detector predicts $\tilde{s}_0 \geq \tau$, generation continues with no modification.

Trial 1. Single-frame Semantic Injection. Trial 1 targets a specific failure mode: the model captures what the prompt describes in a single frame, but fails to extend it into a coherent video. We diagnose this from the previews themselves. Using the same T2V model with the temporal dimension collapsed, we generate a single-frame preview of the prompt and score its predicted alignment $\tilde{s}_{\text{img}}$ (the single-frame analog of $\tilde{s}_0$) with the same dynamic detector. When $\tilde{s}_{\text{img}} > \tilde{s}_0 + \delta$ (with $\delta = 0.05$), the single frame is better aligned than the video, supporting the temporal-failure hypothesis. The margin δ guards against borderline cases where the single-frame and video scores differ only by detector noise, ensuring that Trial 1 is invoked only when single-frame generation is substantively stronger. We then reuse the single-frame intermediate prediction $\hat{\mathbf{z}}^{\text{img}}_{0|k_{\text{img}}}$ as a semantic anchor: it is broadcast along the temporal axis and blended into the video latent over the first one or two denoising steps using a soft mask matched to each step's noise level (Algorithm 1). Because single-frame generation is orders of magnitude cheaper than video, the entire trial completes in seconds, well below the cost of Trial 2. We therefore try Trial 1 first: if the single-frame test indicates a temporal failure, the injection corrects the sample; otherwise, we move on to Trial 2.

Trial 2. Visual Feedback Intervention. When the single-frame test fails ($\tilde{s}_{\text{img}} \leq \tilde{s}_0 + \delta$), the failure is unlikely to be purely temporal, and Trial 1 is bypassed. Instead, we directly leverage the second affordance unlocked by RGB-space inspection: a pretrained VLM as a reasoner over the failed preview itself. The VLM receives the original prompt and the decoded preview, identifies the visible mismatch, and returns an intent-preserving refined prompt with an optional negative prompt. We restart generation under the refined prompt and monitor it with the same RI module. If the new run still satisfies $\tilde{s}_0 < \tau$, the failure likely combines semantic and temporal causes, and we apply Trial 1 once on top (Trial 2→1) before terminating.

The VLM prompt (Fig. 25) and per-Trial examples (Appendix D) are provided in the Appendix.

4 Experiments

We evaluate the proposed early failure detection and intervention pipeline on widely used open-source video diffusion models, CogVideoX-5B and Wan2.1-1.3B. We further evaluate Wan2.1-14B at 720p with 81 frames in Section 4.3. Section 4.1 describes the setup. Section 4.2 reports the main cost–quality trade-off against full-video feedback regeneration. Section 4.3 shows the gains hold under an alternative internal scorer and an additional evaluation benchmark. Section 4.4 confirms orthogonality to prompt refinement and training-free sampling guidance. Section 4.5 empirically grounds the design choices behind the RI module, and analyzes what the detector and intervention do.

Table 1: **Comparison with regeneration across two prompt regimes.** We compare against full-video feedback regeneration using original user prompts (*Original prompt*) and LLM-rewritten prompts (*Prompt rewriting*). Ours achieves competitive or better VBench score with $\sim 2.5\times$ less additional time overhead. Shaded rows denote component variants, replacing the default VLM (GPT-5-mini) with Qwen3.8-27B or the alignment scorer (ViCLIP) with OpenCLIP. Bold indicates the better result between Regeneration and default Ours, excluding variants.

	CogVideoX-5B				**Wan2.1-1.3B**			
	VBench Score			**Time**	**VBench Score**			**Time**
Method	**Final**	**Quality**	**Semantic**	**Overhead (%)**	**Final**	**Quality**	**Semantic**	**Overhead (%)**
Original prompt	0.778	0.803	0.680	0.00	0.789	0.811	0.700	0.00
+ Regeneration	0.793	0.813	0.716	43.73	**0.815**	**0.823**	**0.783**	51.25
+ Ours	**0.800**	**0.816**	**0.735**	**16.55**	0.805	0.817	0.756	**20.30**
w/ Qwen3.8-27B (VLM)	0.791	0.812	0.706	16.49	0.802	0.820	0.732	20.08
w/ OpenCLIP (scorer)	0.799	0.813	0.741	15.75	0.810	0.826	0.749	20.44
Prompt rewriting	0.787	0.818	0.663	3.93	0.815	0.820	0.796	11.70
+ Regeneration	0.801	0.821	0.722	48.87	**0.822**	**0.824**	**0.815**	53.74
+ Ours	**0.813**	**0.822**	**0.776**	**18.70**	0.820	0.822	0.809	**25.12**

4.1 Experimental setup

Models and inference. We use CogVideoX-5B [51] and Wan2.1-1.3B [43] with their official codebases, default seeds, and default guidance scales. Unless otherwise stated, all generations use 50 denoising steps to produce 480p, 49-frame videos. For paired comparison, generations flagged as early failures are also run to full completion without any intervention, isolating the effect of our intervention from run-to-run stochasticity caused by GPU non-determinism.

Benchmarks and metrics. Unless noted otherwise, all experiments use VBench [17]. Crucially, the text–video alignment metric in VBench (*Overall Consistency*) is computed against the *original* prompts, regardless of any VLM-based prompt refinement applied during generation, ensuring that our prompt refinement does not artificially inflate alignment scores. We additionally evaluate with VideoScore2 [13], a learned video evaluator trained independently of VBench (Sec. 4.3).

VLM and scorer. We use GPT-5-mini as the default VLM for visual feedback intervention and ViCLIP as the default alignment scorer in the RI module. We additionally evaluate Qwen3.8-27B and OpenCLIP as replacements for the VLM and scorer, respectively (Table 1).

4.2 Main results: cost–quality trade-off

Comparison with full-video feedback regeneration. We compare against full-video feedback regeneration, a strong post-hoc baseline that, upon detecting a failure, observes the complete video, uses a VLM to refine the prompt, and regenerates from scratch. This baseline has strictly more visual information than our method, which intervenes based on early previews.

We evaluate at two operating points (Table 1): raw user prompts (*Original prompt*) and LLM-rewritten prompts (*Prompt rewriting*). Our pipeline establishes a Pareto-favorable operating point on the cost–quality frontier across both models and both prompt regimes, with $2.5\times$ lower time overhead than Regeneration. On CogVideoX-5B, this cheaper operating point further surpasses Regeneration on Final in both regimes (0.800 vs. 0.793; 0.813 vs. 0.801). On Wan2.1-1.3B, Ours closely matches Regeneration under Prompt rewriting (0.820 vs. 0.822); under raw prompts, Regeneration's $+0.010$ Final advantage requires $2.5\times$ more time overhead, leaving Ours strictly Pareto-optimal at the cheap end of the frontier. Because our system intervenes on only $\sim 40\%$ of samples by design, the dataset-averaged Final gains ($+0.022$ on CogVideoX-5B, $+0.016$ on Wan2.1-1.3B) understate the per-sample effect: on the intervened subset, the per-sample VBench Final gain reaches $\sim +0.054$ on CogVideoX-5B, all without model tuning. By construction, such recovery cannot be achieved by methods acting uniformly across all samples.

Where the savings come from. Figure 6 explains the source of efficiency. Per-sample overhead grows with intervention depth (Fig. 6b), but only a small fraction of samples reach the more expensive trials, while most either complete without intervention or are corrected early. As a result, the dataset-averaged overhead remains below 20.3% on both models (Fig. 6a). Including VLM cost, regeneration

Table 2: **Evaluation on VideoScore2.** Ours consistently improves all three dimensions over the baseline, with substantially larger gains on the predicted-failure subset. VideoScore2 scores *Visual Quality*, *Text–Video Alignment*, and *Physical Consistency* on a 1–5 scale.

	CogVideoX-5B			Wan2.1-1.3B		
Setting	Visual	T–V Align.	Phys.	Visual	T–V Align.	Phys.
Original prompt (baseline)	3.86	3.82	3.66	3.70	3.73	3.54
Original prompt + Ours	**3.93** (+0.07)	**3.96** (+0.14)	**3.74** (+0.08)	**3.82** (+0.12)	**3.86** (+0.13)	**3.62** (+0.08)
Predicted Failure (no intervene)	3.77	3.55	3.56	3.59	3.54	3.49
Predicted Failure + Ours	**3.94** (+0.17)	**3.91** (+0.35)	**3.74** (+0.18)	**3.84** (+0.25)	**3.82** (+0.27)	**3.67** (+0.18)

Table 3: **Orthogonality to existing sampling enhancements.** Our method consistently improves VBench scores across both CogVideoX-5B and Wan2.1-1.3B when applied on top of two training-free sampling enhancements: Enhance-A-Video [26] and Spatiotemporal Skip Guidance (STG) [18]. The gains come from early detection and selective intervention, which operate orthogonally to these enhancement techniques.

Method	**CogVideoX-5B**			**Wan2.1-1.3B**		
	Final	**Quality**	**Semantic**	**Final**	**Quality**	**Semantic**
Original prompt (baseline)	0.7781	0.8027	0.6796	0.7886	0.8109	0.6996
+ Ours	**0.7997**	**0.8157**	**0.7355**	**0.8050**	**0.8172**	**0.7563**
Original prompt + Enhance-A-Video	0.7664	0.7942	0.6555	0.7906	0.8119	0.7054
+ Ours	**0.7899**	**0.8035**	**0.7356**	**0.8047**	**0.8169**	**0.7557**
Original prompt + STG	0.7847	0.8112	0.6789	0.7942	0.8218	0.6834
+ Ours	**0.8021**	**0.8221**	**0.7220**	**0.8060**	**0.8254**	**0.7285**

incurs a per-sample overhead of 103.9% relative to base generation. The Wan2.1-1.3B breakdown is provided in Appendix B.1.

4.3 Generality of the gains

We examine the generality of our framework across evaluation metrics, component choices, and a more demanding setting with a larger model, higher resolution, and longer video sequences.

Additional evaluation with VideoScore2. We further evaluate with VideoScore2 [13], a learned video evaluator that, given a generated video and its text prompt, returns three human-aligned scores on a 1–5 scale: *Visual Quality* (perceptual fidelity), *Text–Video Alignment* (prompt faithfulness), and *Physical Consistency* (plausibility of motion and interactions). Importantly, VideoScore2 is trained independently of VBench, providing an external check on our gains. Table 2 reports results on the full VBench evaluation set. Our method consistently improves all three dimensions over the baseline on both models. The gains are substantially larger on the predicted-failure subset (*e.g.*, $+0.35$ in Text–Video Alignment for CogVideoX-5B), confirming that selective intervention is particularly effective on failure-prone samples.

Modularity of the alignment scorer. The scorer is a swappable component of the RI module rather than a hard dependency. We adopt ViCLIP as the default for two reasons: it is purpose-built for video–text alignment, and it is lightweight enough to fit the per-step latency budget. To demonstrate this modularity, we further evaluate the pipeline with OpenCLIP, an image–text model unrelated to VBench's evaluation and not designed for video. The *Ours (Scorer: OpenCLIP)* row in Table 1 shows comparable VBench gains at similar overhead on both models, indicating that future, more capable alignment scorers can be plugged in as drop-in replacements without re-engineering the pipeline.

Extension to a larger-model regime. We further evaluate our framework on Wan2.1-14B at 720p with 81 frames, using the full set of 946 VBench prompts. Since Wan2.1-1.3B and Wan2.1-14B share the same VAE, we reuse the same L2R converter trained on the Wan2.1 VAE latent space without retraining. We also retain the default ViCLIP scorer and failure threshold $\tau = 0.22$. As shown in Table 4, our method improves the VBench Final score from 0.812 to 0.819, primarily driven by improved semantic alignment (0.767 to 0.795), while maintaining comparable visual quality (0.823 to 0.825). At 720p, each denoising step takes 67.05 s for 81-frame video generation and 1.29 s for single-frame generation. Our method incurs only 8.17% estimated additional time overhead

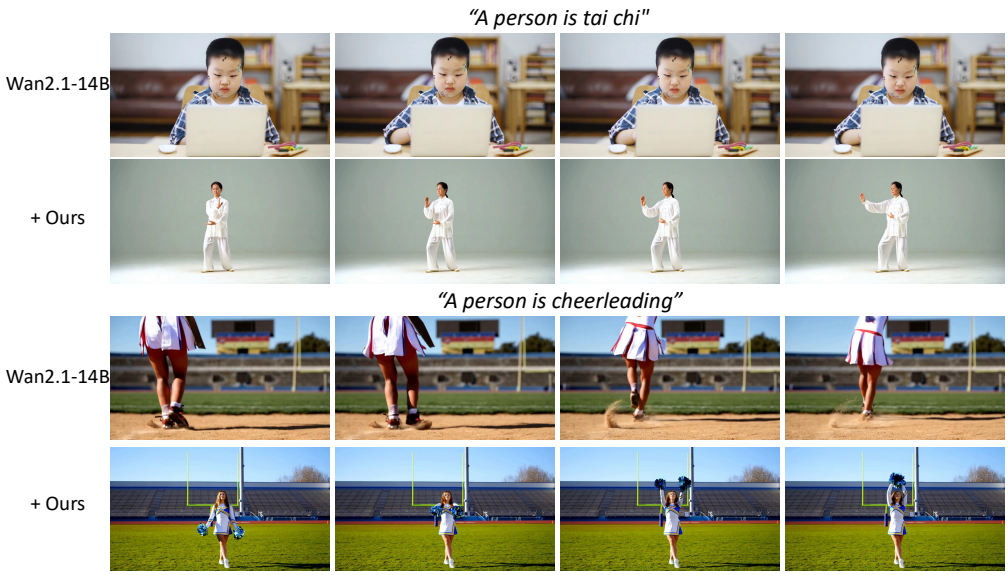


Figure 5: **Qualitative results in a larger-model regime.** On Wan2.1-14B at 720p with 81 frames, our method detects failures at an early stage and intervenes before full generation, avoiding the computation that would otherwise be wasted on completing failed trajectories.

Table 4: **VBench evaluation on Wan2.1-14B at 720p, 81 frames.** We evaluate on the full set of 946 VBench prompts. Our method improves semantic alignment with only 8.17% estimated additional time overhead. Bold indicates the better VBench score.

	VBench Score			Time
Method	**Final**	**Quality**	**Semantic**	**Overhead (%)**
Original prompt (baseline)	0.812	0.823	0.767	0.00
+ Ours	**0.819**	**0.825**	**0.795**	8.17

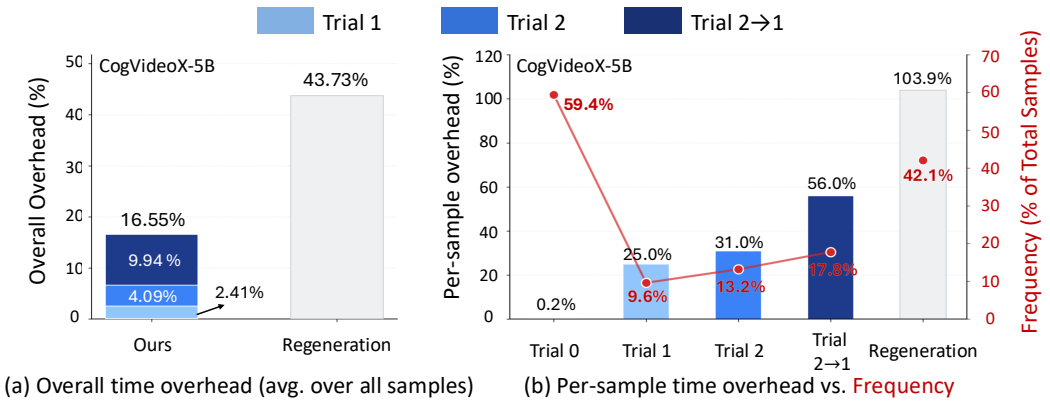
because of the increased cost of full-video denoising. Figure 5 shows qualitative examples where our method detects failures early and intervenes before full generation, reducing the resources spent on unsuccessful trajectories.

4.4 Orthogonality to existing enhancements

We examine whether our intervention remains beneficial when combined with existing enhancement techniques. Orthogonality with prompt rewriting is already evidenced in Table 1: applying our pipeline on top of LLM-rewritten prompts further improves Final score beyond the prompt-rewriting baseline alone (0.787→0.813 on CogVideoX-5B; 0.815→0.820 on Wan2.1-1.3B). Here we further integrate our pipeline with two recent training-free sampling enhancements, Enhance-A-Video [26] and Spatiotemporal Skip Guidance (STG) [18]. Table 3 summarizes the results. Across both models and both enhancements, our method consistently improves all scores, confirming that early detection and intervention operate orthogonally to existing sampling guidance techniques.

4.5 Empirical analysis of early predictability

We quantitatively analyze the conditions that make on-the-fly failure detection feasible. Three findings jointly ground our pipeline: (i) final video quality is largely predictable from RGB-decoded intermediate latents within the first $\sim$10 of 50 denoising steps, enabling intervention on incomplete trajectories; (ii) a single alignment signal, *Overall Consistency*, captures quality variations across other VBench metrics, keeping the per-step inspection budget tight; and (iii) the failure threshold $\tau = 0.22$ aligns with both the real-video score distribution and human perception.



(a) Overall time overhead (avg. over all samples) (b) Per-sample time overhead vs. Frequency

Figure 6: **Efficiency of selective early intervention.** (a) Overall time overhead relative to base generation. Ours incurs only +16.6% additional overhead, substantially lower than the regeneration of failure samples (+43.7%). (b) Per-sample cost increases with deeper trials, yet remains cheaper than regeneration. Since deeper trials are triggered for only a subset of samples (red curve), the overall time overhead stays modest.

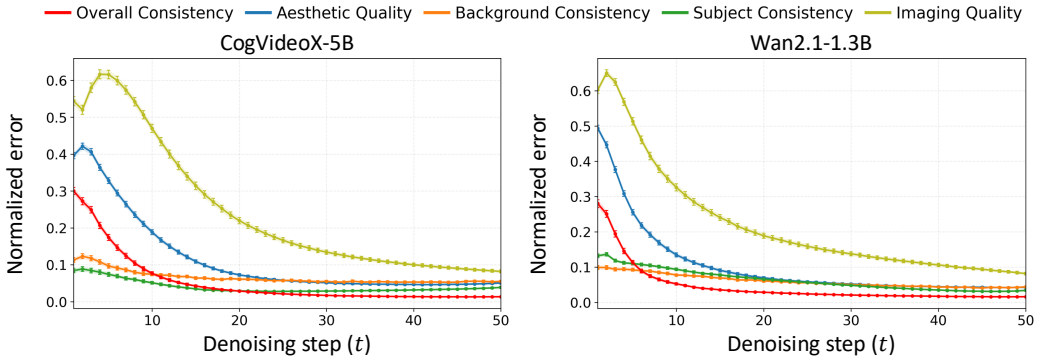


Figure 7: **Early predictability of final video quality.** For 1,800 prompts on CogVideoX and Wan2.1, we report MAE between VBench metrics computed on intermediate L2R previews and on final videos decoded by the original VAE. Errors drop sharply by $\sim$10 steps, notably for Overall Consistency.

Early predictability of final video quality. A core premise of our pipeline is that final video quality can be reliably estimated from early denoising steps. We validate this on 1,800 prompts (946 from VBench plus 854 from Panda70M [9]) across both models, computing at every denoising step the *normalized error*, defined as MAE divided by each metric's dynamic range, between L2R previews and final VAE-decoded videos. Fig. 7 shows that prediction error plateaus within the first $\sim$10 steps, indicating that intermediate scores are reasonable proxies for final quality.

Overall Consistency as a proxy for video quality. For real-time inspection, we use Overall Consistency as the sole alignment signal to fit the 19.5 ms budget. A balanced binary classification on top/bottom quantiles shows PR-AUC above 0.5 across all VBench metrics, confirming that this single signal captures diverse failure modes (Appendix C.5).

Threshold selection for failure detection. We set τ by combining distributional evidence with a human study (Fig. 8). Real videos (1,024 Pexels clips, captioned by GPT-5-mini and scored with the same alignment scorer [44]) concentrate at higher *Overall Consistency* than T2V outputs, which exhibit a pronounced low-score tail; $\tau = 0.22$ removes this tail. The framework remains effective across a range of τ (Appendix C.7), so τ acts as a practical quality–efficiency knob.

Behavior of detection and intervention. The dynamic failure detector reaches a decision at step 11/50 (CogVideoX-5B) and 10/50 (Wan2.1-1.3B) on average. Running flagged samples to completion *without intervention* yields cleanly separated final *Overall Consistency* ($\Delta\text{CDF} \approx 0.71$; Fig. 9), confirming early discrimination.

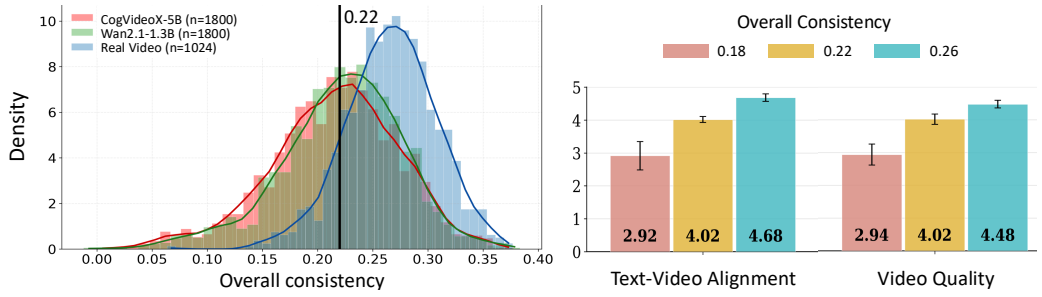


Figure 8: **Selecting the failure threshold τ for Overall Consistency. Left:** Distributions of Overall Consistency for generated and real videos. Real videos concentrate at higher scores; we set $\tau = 0.22$ so that the retained generated samples better match the real-video regime. **Right:** A human study with 26 participants over 30 samples (780 responses) validates the perceptual boundary: scores of 0.18 are often judged as failures, whereas scores$\geq$0.22 are considered reasonable.

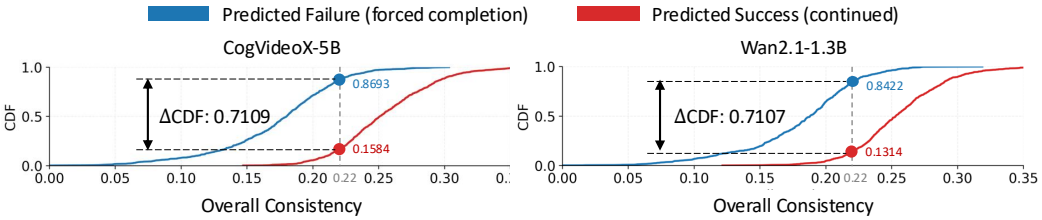


Figure 9: **Accuracy of early failure detection.** Final Overall Consistency distributions for samples predicted to pass or fail (failures forced to complete without intervention). Strong separation across models demonstrates reliable early discrimination.

5 Conclusion

We present the first early failure detection and intervention pipeline for latent text-to-video diffusion that operates on a single trajectory within a strict per-step latency budget. The key enabler is a Real-time Inspection (RI) module which decodes intermediate latents into RGB via a 0.059 M-parameter Latent-to-RGB converter and scores them with an off-the-shelf, zero-shot vision–language evaluator in 39.2 ms, over $100\times$ faster than a single denoising step. This makes step-wise RGB inspection practical for on-the-fly, single-trajectory video diffusion sampling, and corroborates our key empirical finding that final video quality is largely predictable from the first $\sim$10 of 50 denoising steps. When a failure is predicted, a diagnostic intervention framework escalates computation hierarchically, replacing post-hoc regeneration with on-the-fly correction at approximately $2.5\times$ lower additional time overhead under original prompts. Across CogVideoX-5B and Wan2.1-1.3B at 480p with 49 frames, as well as Wan2.1-14B at 720p with 81 frames, the framework delivers consistent quality gains while remaining orthogonal to prompt refinement and sampling guidance. Building on these results, detecting finer-grained failures at earlier stages, together with more efficient and effective intervention strategies, represents a promising direction toward reducing wasted generation compute for service providers and shortening the time to satisfactory outputs for users.

Broader impact. Text-to-video generation often requires multiple rounds of regeneration before users obtain a satisfactory result, with each failed attempt consuming substantial GPU compute and energy. By identifying and correcting failures within a single trajectory rather than running them to completion or restarting from scratch, our method directly reduces the compute and energy spent on unsatisfactory generations. As an efficiency contribution orthogonal to generative capability, it does not introduce new misuse vectors beyond those already present in the underlying T2V models.

Limitations and future work. While our framework supports interchangeable lightweight alignment scorers, validated with both ViCLIP (default) and OpenCLIP, subtle failures such as fine-grained motion artifacts or compositional errors fall below the discriminative resolution of current lightweight evaluators and remain a target for future work. More expressive yet efficient video evaluators would extend the failure modes our framework can detect and recover from.

Acknowledgments and Disclosure of Funding

Funding: This work was supported by the National Research Foundation of Korea (NRF) funded by the Korean government (MSIT) in 2026 (Grant No. RS-2024-00451947, Project Title: Interaction Framework for Connecting Users to Generative Experiences via Time-Space Expanded Multimedia: Centering on Social Communication Disorders (45%)), the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. RS-2026-25518317, Development of AI memory mechanism that reflects human cognitive principles (40%); No. 2022-0-00124, No. RS-2022-II220124, Development of Artificial Intelligence Technology for Self-Improving Competency-Aware Learning Capabilities (15%)), the InnoCORE program of the Ministry of Science and ICT (26-InnoCORE-01; N10250156), the Artificial Intelligence Industrial Convergence Cluster Development Project funded by the Ministry of Science and ICT (MSIT, Korea) & Gwangju Metropolitan City, and the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. RS-2019-II191906, Artificial Intelligence Graduate School Program (POSTECH)).

Competing interests: The authors declare no competing interests.

References

[1] Donghoon Ahn, Hyoungwon Cho, Jaewon Min, Wooseok Jang, Jungwoo Kim, SeonHwa Kim, Hyun Hee Park, Kyong Hwan Jin, and Seungryong Kim. Self-rectifying diffusion sampling with perturbed-attention guidance. In *ECCV*, 2024.

[2] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, et al. Lumiere: A space-time diffusion model for video generation. In *SIGGRAPH Asia 2024 Conference Papers*, 2024.

[3] Lorenzo Baraldi, Davide Bucciarelli, Zifan Zeng, Chongzhe Zhang, Qunli Zhang, Marcella Cornia, Feng Liu, Hu Zheng, Rita Cucchiara, et al. Verifier matters: Enhancing inference-time scaling for video diffusion models. In *Proceedings of the 36th British Machine Vision Conference*, 2025.

[4] Federico Betti, Lorenzo Baraldi, Lorenzo Baraldi, Rita Cucchiara, and Nicu Sebe. Optimizing resource consumption in diffusion models through hallucination early detection. In *European Conference on Computer Vision*, pages 312–326. Springer, 2024.

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

[6] Ollin Boer Bohan. Taehv: Tiny autoencoder for hunyuan video. `https://github.com/madebyollin/taehv`, 2025.

[7] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM transactions on Graphics (TOG)*, 42(4):1–10, 2023.

[8] Shoufa Chen, Chongjian Ge, Yuqi Zhang, Yida Zhang, Fengda Zhu, Hao Yang, Hongxiang Hao, Hui Wu, Zhichao Lai, Yifei Hu, et al. Goku: Flow based video generative foundation models. In *CVPR*, 2025.

[9] Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, et al. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *CVPR*, 2024.

[10] Minkyu Choi, SP Sharan, Harsh Goel, Sahil Shah, and Sandeep Chinchali. We'll fix it in post: Improving text-to-video generation with neuro-symbolic feedback. *arXiv preprint arXiv:2504.17180*, 2025.

[11] Huanlei Guo, Hongxin Wei, and Bingyi Jing. Toward early quality assessment of text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 38410–38419, June 2026.

[12] Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

[13] Xuan He, Dongfu Jiang, Ping Nie, Minghao Liu, Zhengxuan Jiang, Mingyi Su, Wentao Ma, Junru Lin, Chun Ye, Yi Lu, Keming Wu, Benjamin Schneider, Quy Duc Do, Zhuofeng Li, Yiming Jia, Yuxuan Zhang, Guo Cheng, Haozhe Wang, Wangchunshu Zhou, Qunshu Lin, Yuanxing Zhang, Ge Zhang, Wenhao Huang, and Wenhu Chen. Videoscore2: Think before you score in generated video evaluation. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=MpkVh4jH44.

[14] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[15] Susung Hong, Chongjian Ge, Zhifei Zhang, and Jui-Hsien Wang. Diffusionbrowser: Interactive diffusion previews via multi-branch decoders. *arXiv preprint arXiv:2512.13690*, 2025.

[16] Bukun Huang, Benlei Cui, Zhizeng Ye, Xuemei Dong, Tuo Chen, Hui Xue, Dingkang Yang, Longtao Huang, Haiwen Hong, and Jingqun Tang. Diffusion probe: Generated image result prediction using cnn probes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 35926–35935, June 2026.

[17] 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 *CVPR*, 2024.

[18] Junha Hyung, Kinam Kim, Susung Hong, Min-Jung Kim, and Jaegul Choo. Spatiotemporal skip guidance for enhanced video diffusion sampling. In *CVPR*, 2025.

[19] Vineet Jain, Kusha Sareen, Mohammad Pedramfar, and Siamak Ravanbakhsh. Diffusion tree sampling: Scalable inference-time alignment of diffusion models. In *NeurIPS*, 2025.

[20] Muhammad Tanveer Jan, Mohammed G Al-Jassani, Martinraj Nadar, Emmanuel Melchizedek Vunnava, Vangmai Chakrapani, Hayat Ullah, Abbas Khan, Sardar Ali Abbas, and Borko Furht. Text-to-video generators: a comprehensive survey. *Journal of Big Data*, 12(1):253, 2025.

[21] 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.

[22] Fangfu Liu, Hanyang Wang, Yimo Cai, Kaiyan Zhang, Xiaohang Zhan, and Yueqi Duan. Video-t1: Test-time scaling for video generation. In *ICCV*, 2025.

[23] Haozhe Liu, Wentian Zhang, Jinheng Xie, Francesco Faccio, Mengmeng Xu, Tao Xiang, Mike Zheng Shou, Juan-Manuel Perez-Rua, and Jürgen Schmidhuber. Faster diffusion through temporal attention decomposition. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=xXs2GKXPnH.

[24] Qihao Liu, Adam Kortylewski, Yutong Bai, Song Bai, and Alan Yuille. Discovering failure modes of text-guided diffusion models via adversarial search. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=TOWdQQgMJY.

[25] Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, et al. Sora: A review on background, technology, limitations, and opportunities of large vision models. *arXiv preprint arXiv:2402.17177*, 2024.

[26] 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. *arXiv preprint arXiv:2502.07508*, 2025.

[27] Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, et al. Scaling inference time compute for diffusion models. In *CVPR*, 2025.

[28] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15762–15772, 2024.

[29] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *CVPR*, 2023.

[30] Yuta Oshima, Masahiro Suzuki, Yutaka Matsuo, and Hiroki Furuta. Inference-time text-to-video alignment with diffusion latent beam search. In *NeurIPS*, 2025.

[31] Pinelopi Papalampidi, Olivia Wiles, Ira Ktena, Aleksandar Shtedritski, Emanuele Bugliarello, Ivana Kajic, Isabela Albuquerque, and Aida Nematzadeh. Dynamic classifier-free diffusion guidance via online feedback. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=z9YC9bvfUL`.

[32] Pexels. Pexels, royalty-free stock footage website. `https://www.pexels.com`. Accessed: 2026-03-15.

[33] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[34] Vignav Ramesh and Morteza Mardani. Test-time scaling of diffusion models via noise trajectory search. In *NeurIPS*, 2025.

[35] Vasco Ramos, Regev Cohen, Idan Szpektor, and Joao Magalhaes. Beyond the noise: Aligning prompts with latent representations in diffusion models. *arXiv preprint arXiv:2512.08505*, 2025.

[36] Esther Rodriguez, Monica Welfert, Samuel McDowell, Nathan Stromberg, Julian Antolin Camarena, and Lalitha Sankar. CORAL: Disentangling latent representations in long-tailed diffusion. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL `https://openreview.net/forum?id=VIf1Ygh1Pv`.

[37] Dvir Samuel, Rami Ben-Ari, Simon Raviv, Nir Darshan, and Gal Chechik. Generating images of rare concepts using pre-trained diffusion models. In *AAAI*, 2024.

[38] Ariel Shaulov, Itay Hazan, Lior Wolf, and Hila Chefer. Flowmo: Variance-based flow guidance for coherent motion in video generation. In *NeurIPS*, 2025.

[39] Yukai Shi, Jiarong Ou, Rui Chen, Haotian Yang, Jiahao Wang, Xin Tao, Pengfei Wan, Di Zhang, and Kun Gai. Imbalance in balance: Online concept balancing in generation models. In *ICCV*, 2025.

[40] 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 *CVPR*, 2025.

[41] Chujun Tang, Lei Zhong, and Fangqiang Ding. Seeking physics in diffusion noise. *arXiv preprint arXiv:2603.14294*, 2026.

[42] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020.

[43] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[44] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. In *ICLR*, 2024.

[45] Yiyang Wang, Xi Chen, Xiaogang Xu, Sihui Ji, Yu Liu, Yujun Shen, and Hengshuang Zhao. Diffdoctor: Diagnosing image diffusion models before treating. In *ICCV*, 2025.

[46] Zhaoyang Wang, Bo Hu, Mingyang Zhang, Jie Li, Leida Li, Maoguo Gong, and Xinbo Gao. Diffusion model-based visual compensation guidance and visual difference analysis for no-reference image quality assessment. *IEEE Transactions on Image Processing*, 34:263–278, 2025.

[47] Thaddäus Wiedemer, Yuxuan Li, Paul Vicol, Shixiang Shane Gu, Nick Matarese, Kevin Swersky, Been Kim, Priyank Jaini, and Robert Geirhos. Video models are zero-shot learners and reasoners. *arXiv preprint arXiv:2509.20328*, 2025.

[48] Xin Xie and Dong Gong. Dymo: Training-free diffusion model alignment with dynamic multi-objective scheduling. In *CVPR*, 2025.

[49] Zhen Xing, Qijun Feng, Haoran Chen, Qi Dai, Han Hu, Hang Xu, Zuxuan Wu, and Yu-Gang Jiang. A survey on video diffusion models. *ACM Computing Surveys*, 57(2):1–42, 2024.

[50] Katherine Xu, Lingzhi Zhang, and Jianbo Shi. Good seed makes a good crop: Discovering secret seeds in text-to-image diffusion models. In *WACV*, 2025.

[51] 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`.

[52] Mingyang Yi, Aoxue Li, Yi Xin, and Zhenguo Li. Towards understanding the working mechanism of text-to-image diffusion model. *Advances in Neural Information Processing Systems*, 37:55342–55369, 2024.

[53] Zengqun Zhao, Ziquan Liu, Yu Cao, Shaogang Gong, Zhensong Zhang, Jifei Song, Jiankang Deng, and Ioannis Patras. Latsearch: Latent reward-guided search for faster inference-time scaling in video diffusion. In *European Conference on Computer Vision (ECCV)*, 2026.

[54] Ya Zou, Jingfeng Yao, Siyuan Yu, Shuai Zhang, Wenyu Liu, and Xinggang Wang. Turbo-vaed: Fast and stable transfer of video-vaes to mobile devices. In *AAAI*, 2026.

Technical appendices and supplementary material

We provide implementation details of our early failure detection and intervention pipeline, and describe how the time overhead is computed in detail. In addition, we present a range of additional experiments, including the multi-seed evaluation, overhead comparison with test-time scaling methods, comparison with a Best-of-3 baseline, threshold sensitivity analysis, and disaggregated analysis of each intervention branch.

Contents

A Implementation details
- A.1 Experiment details
- A.2 Real-time Inspection (RI) module
- A.3 Dynamic failure detector
- A.4 Diagnostic intervention framework
- A.5 Threshold calibration for OpenCLIP

B Analysis of time overhead
- B.1 Efficiency of selective early intervention on Wan2.1-1.3B
- B.2 Computation of time overhead

C Additional experiments
- C.1 Qualitative results of our pipeline
- C.2 Multi-seed evaluation on VBench
- C.3 Overhead comparison with test-time scaling methods
- C.4 Spatiotemporal sensitivity of VBench metrics
- C.5 Overall Consistency as a proxy for video quality
- C.6 Comparison with a best-of-3 baseline
- C.7 Threshold sensitivity of early failure detection
- C.8 Effectiveness of intervention on failure cases
- C.9 Disaggregated analysis of intervention effects

D Additional qualitative results

A Implementation details

A.1 Experiment details

For VBench evaluation, we generate one video per text prompt using the same inference settings and seed across all methods. For CogVideoX-5B [51] and Wan2.1-1.3B [43], we use the default seed and guidance scale provided in the official codebases. To enable comparison with base generation, we force the sampling process to continue full completion without any intervention, even when early failures are flagged. This allows direct comparison against the base generation under the same evaluation setting. Following prior work [17], the text–video alignment metric in VBench, *Overall Consistency*, is computed using the original user prompts, regardless of any VLM-based prompt refinement applied during generation. We also report VBench results across all dimensions in Tab. 12.

Discussion of per-dimension VBench results. Table 12 reports our method's effect across all 16 VBench dimensions. The improvements are broad: our method improves the majority of dimensions on both models, with substantial gains on semantic dimensions such as Human Action ($+0.090$ on CogVideoX-5B; $+0.140$ on Wan2.1-1.3B), Multiple Objects ($+0.161$; $+0.067$), Object Class ($+0.051$; $+0.072$), and Scene ($+0.079$; $+0.109$), confirming that selective early intervention effectively corrects alignment failures. Beyond alignment, perceptual and temporal dimensions also improve consistently, including Subject Consistency, Background Consistency, Motion Smoothness, and Imaging Quality, indicating that the gains generalize beyond the alignment-driven failure detection signal. The one dimension that does not improve is Dynamic Degree, a binary classification of whether each video exceeds an optical-flow threshold estimated by RAFT [42]; the fraction of videos crossing this threshold is preserved on CogVideoX-5B ($0.542 \rightarrow 0.542$) but decreases on Wan2.1-1.3B ($0.542 \rightarrow 0.403$). Table 13 decomposes this trade-off across four settings and

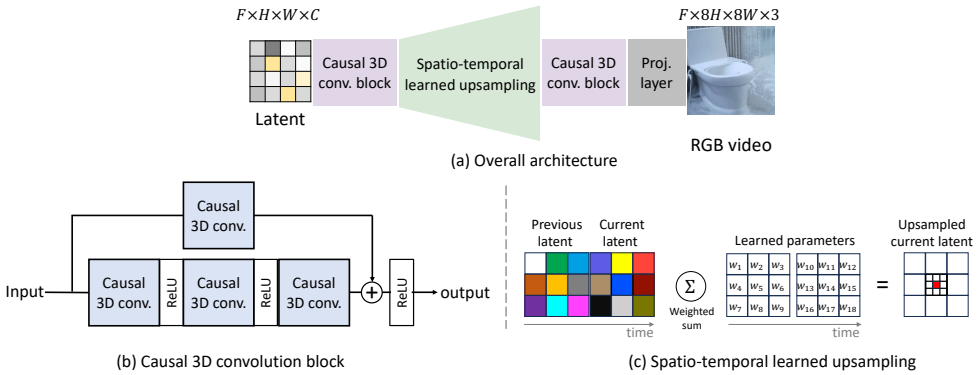


Figure 10: **Architecture of Latent-to-RGB (L2R) converter.** (a) Overall architecture. The converter takes a latent video tensor as input and produces an RGB video preview through two causal 3D convolution blocks, a spatio-temporal learned upsampling, and a projection layer. (b) Causal 3D convolution block with a main branch and a skip branch. (c) Illustration of spatio-temporal learned upsampling for a single **red target** location. Given the previous and current latents, learned parameters are used to predict the upsampled latent at the current step by weighted summation. In practice, this operation is implemented densely as a 3D convolution over the latent.

shows that the decrease is largely inherited from the VLM-based prompt refinement step rather than introduced by our intervention itself: applying VLM-refined prompts alone already reduces this fraction substantially on Wan2.1-1.3B ($0.542 \rightarrow 0.319$), and full-video feedback regeneration, which uses the feedback-based refinement step, exhibits a comparable drop ($0.542 \rightarrow 0.431$). Notably, on CogVideoX-5B our method preserves this fraction better than regeneration (0.542 vs. 0.458) while requiring $2.64\times$ lower time overhead.

A.2 Real-time Inspection (RI) module

The Real-time Inspection (RI) module is designed to monitor the denoising process of T2V diffusion models in real time. To do so, it first converts intermediate latent representations into RGB video previews, and then evaluates how well the current generation aligns with the input text using an alignment scorer. In this way, the RI module provides an online estimate of semantic alignment during sampling. In the following, we describe the implementation details of the Latent-to-RGB (L2R) converter and the alignment scorer used in the RI module.

Latent-to-RGB (L2R) converter. The design of our Latent-to-RGB (L2R) converter is primarily motivated by the need for an efficient preview module that balances low latency with moderate visual quality. Our architecture is informed by the observation in Latent-NeRF [29] that linear approximations are often sufficient for decoding latent representations into plausible RGB colors, as well as the findings in RAFT [42] regarding the superiority of learned upsampling over bilinear interpolation for recovering fine-grained spatial details.

Figure 10(a) shows the overall architecture of the L2R converter. The model consists of two causal 3D convolution blocks, one spatio-temporal learned upsampling module, and a final projection layer. To keep the model lightweight, all hidden features use the same channel dimension as the input latent, and the channel dimension is reduced to RGB only at the final layer. As shown in Fig. 10(b), each causal 3D convolution block is built with a main branch and a skip branch, followed by nonlinear activations. We use a kernel size of $2 \times 3 \times 3$ for time, height, and width dimensions, which allows the network to incorporate temporal context from previous frames. Figure 10(c) illustrates how the spatio-temporal learned upsampling works for a single target location. For clarity, we show a $3\times$ upsampling example. Given the previous and current latents, the module predicts the target latent at the current step by taking a weighted sum with learned parameters. In practice, this local operation is applied densely to all output locations and implemented efficiently as a 3D convolution. Finally, the projection layer converts the refined latent features into RGB using two 2D convolution layers with kernel sizes 9×9 and 3×3.

Table 5: **Efficiency and quality of latent-to-RGB converter.** We measure PSNR on DAVIS and Pexels. Then, we measure the decoding time/memory at 480×720. L2R offers moderate PSNR with the lowest latency and memory, providing previews suitable for inspection for both models.

	CogVideoX-5B					Wan2.1-1.3B				
Decoder	**Param. (M)**	**DAVIS PSNR ↑**	**Pexels PSNR ↑**	**Time (s)**	**Mem (MB)**	**Param. (M)**	**DAVIS PSNR ↑**	**Pexels PSNR ↑**	**Time (s)**	**Mem (MB)**
Original	123.4	**32.80**	**35.30**	3.990	242.5	73.30	**32.68**	**35.41**	1.702	140.4
TinyAE	11.32	26.56	27.35	0.116	21.60	11.32	30.50	33.46	0.116	21.60
Turbo-VAED	40.69	32.03	35.03	0.599	114.6	-	-	-	-	-
Ours	**0.059**	30.00	32.50	**0.0197**	**0.138**	**0.059**	28.98	31.40	**0.0197**	**0.138**

Training requires only the pretrained VAE encoder and is independent of the diffusion model. We collect 33k real-world video clips from Pexels [32] with sufficient temporal dynamics (mean optical-flow > 0.5 via RAFT), encode them into latents using the VAE encoder, and train on latent–RGB pairs. Training completes in 11 hours on a single NVIDIA A100 GPU.

Table 5 compares our L2R converter against the original decoders and efficient VAE baselines, including Turbo-VAED [54] and TinyAE [6]. We train separate converters for CogVideoX and Wan2.1 to match their respective VAE latent spaces, and evaluate reconstruction quality on the DAVIS-2017 [33] test split with videos truncated to 49 frames and resized to 480×720. We additionally construct a Pexels test set consisting of 500 videos and compare reconstruction performance as well. For L2R, PSNR is computed against ground-truth RGB frames under the same temporal subsampling setting. While the original decoders achieve higher PSNR, our L2R attains moderate PSNR with drastically reduced decoding time and memory usage. Compared to efficient VAE baselines, L2R uses substantially fewer parameters and achieves faster inference while maintaining moderate reconstruction quality across both models, making it well-suited for inspection alongside generation.

Alignment scorer. Within the per-step latency budget, the choice of scorer is otherwise flexible. We adopt ViCLIP [44] as our default for two reasons: (i) unlike image–text models, ViCLIP is purpose-built for video–text alignment and captures temporal semantics relevant to video previews, and (ii) it is sufficiently lightweight to fit our per-step budget. Our framework is not tied to ViCLIP: Sec. 4.3 of the main paper validates the pipeline with OpenCLIP as an alternative scorer, indicating that the alignment scorer is a swappable component rather than a hard dependency.

A natural concern is whether using ViCLIP at inspection time inflates VBench scores by construction. We argue this is not the case for two reasons. First, ViCLIP is used in only 2 of VBench's 16 dimensions (Overall Consistency and Temporal Style), yet our method improves the majority of dimensions on both models, with all but one dimension on Wan2.1-1.3B (Table 12) preserved or improved. Second, gains transfer to VideoScore2 [13] (Sec. 4.3 of the main paper), a learned video evaluator trained independently of VBench and ViCLIP, with consistent improvements across Visual Quality, Text–Video Alignment, and Physical Consistency on both models. Together, these results indicate that the gains reflect genuine quality improvements rather than artifacts of using ViCLIP as the inspection signal.

Built upon the VBench implementation, we apply several implementation-level optimizations to meet the per-step latency budget for online use during denoising. We move the frame preprocessing and similarity computation to the GPU, and cache the text feature once per sample since the prompt remains fixed throughout sampling. Together these optimizations reduce scoring to only 19.5 ms per call, comparable to L2R itself. Note that we use the official VBench codebase for the final VBench scores reported in our experiments.

A.3 Dynamic failure detector

In Fig. 7 of the main paper, we show that several final video quality metrics are already predictable from early denoising steps. Specifically, the error between the metric measured from an intermediate preview and that of the final video drops sharply within the first ~ 10 denoising steps.

However, the convergence behavior of diffusion models is known to be sample-dependent [17, 28]. We further observe that semantic convergence in text-to-video generation also differs substantially



Figure 11: **Semantic convergence varies across samples in video diffusion.** We visualize the center frame (24th of 49) decoded from intermediate denoising steps for two text-to-video generations. While the dog sample exhibits recognizable semantics at an early stage, the shark sample remains ambiguous until later steps. This illustrates that semantic convergence in video diffusion is highly sample-dependent, motivating the dynamic failure detector rather than relying on a fixed early step.

across prompts, as shown in Fig. 11. This suggests that monitoring only a fixed denoising step is less effective than using a sample-dependent monitoring strategy.

Since our RI module inspects a video preview in only 39.2 ms (L2R + alignment scorer), we use a dynamic failure detector that monitors the alignment score at every denoising step. Rather than making a decision from a single fixed step, the detector maintains a short history of intermediate scores and their step-wise changes, while also tracking the best score observed within a predefined inspection budget.

Table 6: **Overall Consistency between fixed-step vs. dynamic failure detector.** Despite nearly identical average prediction steps, the dynamic strategy achieves lower normalized error on both models.

Model	Method	Avg. Step	Normalized Error ($\times 10^{-2} \downarrow$)
CogVideoX-5B	Fixed	11	6.67 ± 0.45
	Dynamic	11.05	$\mathbf{5.73 \pm 0.41}$
Wan2.1-1.3B	Fixed	10	5.24 ± 0.34
	Dynamic	10.33	$\mathbf{4.70 \pm 0.30}$

Our dynamic failure detector tracks the alignment trajectory and issues a sample-adaptive stopping decision based on its level, trend, and stability. Let $\{s_k\}$ denote the alignment scores measured during denoising. For the video preview inspection, the detector begins making decisions from step 8 and inspects the trajectory up to step 14. Within this inspection window, it applies simple stopping rules based on the score level, recent trend, and local stability. Specifically, it triggers *early reject* when the score is low and remains nearly unchanged for several steps, *give up* when the score is low and continues to decrease, *early accept* when the score is already high and stable, and *stable stop* when the score has sufficiently converged. If none of these conditions are met within the inspection budget, the detector stops at the budget limit and returns the best score observed so far. We use the returned score as the predicted final alignment score, denoted by $\tilde{s}_0$. A sample is then classified as successful if $\tilde{s}_0 \geq \tau$, and as a failure otherwise.

Table 6 compares the dynamic failure detector with a fixed-step predictor under the same experimental setting as Fig. 7 in the main paper. For each model, we use the average prediction step of the dynamic detector as the corresponding fixed step. Even under this matched comparison, the dynamic detector achieves lower normalized error on CogVideoX-5B and Wan2.1-1.3B. This result suggests that a sample-dependent stopping strategy provides a more accurate estimate of the final alignment score than monitoring a single fixed denoising step. Notably, this improvement is achieved even though the detector includes failure-oriented policies such as *early reject*, rather than focusing purely on final score prediction. For single-frame preview inspection, we use a later inspection window, starting from step 20 and continuing up to step 30, since single-frame denoising progresses substantially faster than video denoising.

Algorithm 1: Trial 1: Single-frame Semantic Injection

Require: Single-frame preview latent $\mathbf{z}^{\text{img}}$, initial video latent $\mathbf{z}_T$, denoising timesteps $\{t_k\}_{k=2}^{N}$, text condition c, scheduler, T2V model, injection length $K = 2$

Expand $\mathbf{z}^{\text{img}}$ with a temporal dimension
Set the reference noise to the initial latent: $\boldsymbol{\epsilon} \leftarrow \mathbf{z}_T$
Initialize the latent $\mathbf{z}$ at t_2: $\mathbf{z} \leftarrow \text{AddNoise}(\mathbf{z}^{\text{img}}, \boldsymbol{\epsilon}, t_2)$
for $k = 2, \ldots, N$ **do**
  if $k \leq K + 1$ **then**
    $\hat{\mathbf{z}} \leftarrow \text{AddNoise}(\mathbf{z}^{\text{img}}, \boldsymbol{\epsilon}, t_k)$
    $w_k \leftarrow \text{SoftMask}(t_k)$
    $\mathbf{z} \leftarrow w_k \odot \hat{\mathbf{z}} + (1 - w_k) \odot \mathbf{z}$
  end if
  $\mathbf{z} \leftarrow \text{Scheduler}(\text{T2V}(\mathbf{z}, c, t_k))$
end for
return Decoder($\mathbf{z}$)

A.4 Diagnostic intervention framework

As shown in Sec. 3.2 of the main paper, our diagnostic intervention framework intervenes only on generations that are flagged as potential failures by the Real-time Inspection (RI) module. The intervention is preview-driven: rather than applying a fixed correction to every flagged sample, we use the intermediate preview signals to decide how to continue the generation, escalating compute hierarchically only when needed.

Trial 0. Base Generation. This trial corresponds to the default generation process without any modification. When the detector predicts $\tilde{s}_0 \geq \tau$, we simply continue standard denoising with the original prompt.

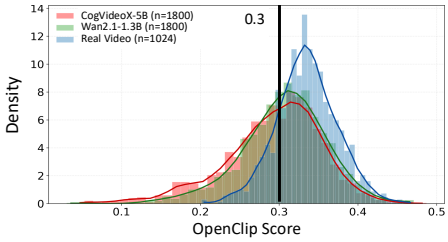


Figure 12: **OpenCLIP score distributions.** We set $\tau = 0.30$ (line) following the procedure of Fig. 8 (main).

Trial 1. Single-frame Semantic Injection. Trial 1 targets a specific failure mode in which the model captures the prompt in a single frame but fails to extend it into a coherent video. We diagnose this from the previews themselves: using the same T2V model with the temporal dimension collapsed, we generate a single-frame preview and score its predicted alignment $\tilde{s}_{\text{img}}$ with the same dynamic detector. When $\tilde{s}_{\text{img}} > \tilde{s}_0 + \delta$ (with $\delta = 0.05$), the single frame is better aligned than the video, supporting the temporal-failure hypothesis, and we apply single-frame semantic injection.

Algorithm 1 describes the detailed procedure. For simplicity, we denote the single-image intermediate prediction $\hat{\mathbf{z}}_{0|k_{\text{img}}}^{\text{img}}$ as $\mathbf{z}^{\text{img}}$. To make it compatible with video generation, we first expand $\mathbf{z}^{\text{img}}$ along the temporal dimension. We then add noise so that it matches the latent at the second timestep, where single-frame injection begins. During the following two denoising steps, we apply a soft temporal mask that assigns a weight of 1 to the starting frame and linearly decays to 0 over time, blending the injected latent with the model-updated latent. After these two steps, the remaining denoising proceeds in the same manner as the base generation. Since the latent at t_1 is essentially pure noise and carries no semantic signal, we start the injection at $k = 2$, the earliest step that retains usable information from $\mathbf{z}^{\text{img}}$.

Trial 2. Visual Feedback Intervention. When the single-frame test fails ($\tilde{s}_{\text{img}} \leq \tilde{s}_0 + \delta$), the failure is unlikely to be purely temporal, so Trial 1 is bypassed and we proceed directly to Trial 2. We provide the failed video preview and the original text prompt to a vision–language model (VLM), and ask it to identify the visible mismatch and return an intent-preserving refined prompt with an optional negative prompt. Figure 25 presents the instruction template used for this step. We then restart generation under the refined prompt and monitor it with the same RI module. This trial is

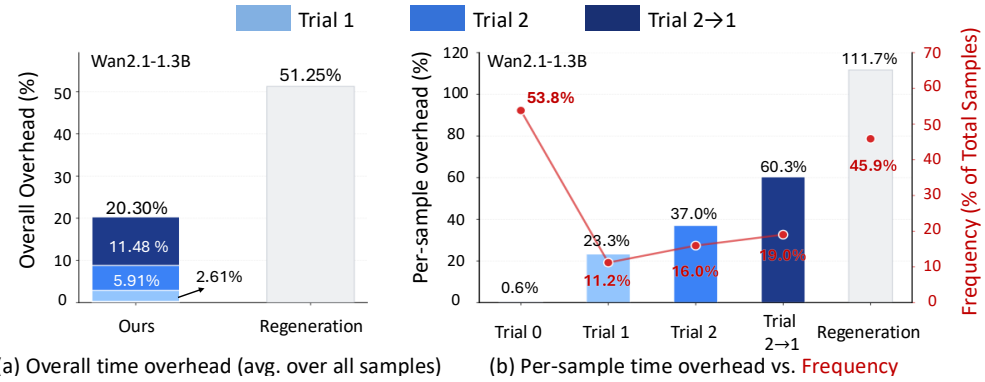


Figure 13: **Efficiency of selective early intervention on Wan2.1-1.3B.** (a) Overall time overhead relative to base generation. Ours incurs only +20.3% additional time overhead, substantially lower than the regeneration of failure samples (+51.25%). (b) Per-sample cost increases with deeper trials, yet remains cheaper than regeneration. Since deeper trials are triggered for only a subset of samples (red curve), the overall time overhead stays modest.

particularly effective when the failure is caused by missing semantic cues, incorrect actions, or weak style descriptions that are difficult to correct through latent injection alone.

Trial 2→1. If the regenerated run under the refined prompt still satisfies $\tilde{s}_0 < \tau$, the failure likely combines semantic and temporal causes. In this case, we apply Trial 1 once on top of the refined prompt before terminating, ensuring both the semantic and temporal failure modes are addressed.

A.5 Threshold calibration for OpenCLIP

We measure OpenCLIP scores on the same video set used for ViCLIP threshold calibration in the main paper. Real videos (1,024 Pexels clips) again concentrate at higher scores than generated videos from CogVideoX-5B and Wan2.1-1.3B (Fig. 12). Following the same procedure, we set $\tau = 0.30$, which removes the low-score tail of generated distributions and brings the retained samples into the real-video score range, mirroring the $\tau = 0.22$ choice for ViCLIP.

B Analysis of time overhead

We provided the efficiency results of selective early intervention for Wan2.1-1.3B, alongside those of CogVideoX-5B shown in Fig. 6 of the main paper. Now, we present a detailed breakdown of how the time overhead is computed.

B.1 Efficiency of selective early intervention on Wan2.1-1.3B

Figure 13 provides a detailed breakdown of the efficiency of selective early intervention on Wan2.1-1.3B. As shown in Fig. 13(b), the per-sample time overhead increases with intervention depth, rising from 23.3% for Trial 1 to 37.0% for Trial 2 and 60.3% for Trial 2→1. In contrast, naive regeneration incurs a much larger per-sample time overhead of 111.7% including VLM.

Despite this increasing per-sample time cost, only a relatively small fraction of samples reach the more expensive stages. Most samples either finish without intervention or are handled by earlier trials, which keeps the average cost low. As a result, the overall time overhead remains modest at 20.30% over all samples, substantially lower than regeneration at 51.25%, as shown in Fig. 13(a).

B.2 Computation of time overhead

We compute time overhead from executions over the full VBench evaluation set of 946 prompts, using two measured quantities: (i) the average runtime of each primitive operation in Table 7, and (ii) the average number of times each operation is executed in each branch in Table 8. For each branch, the total runtime is obtained by summing the costs of all executed operations, including video

Table 7: **Measured execution time of each primitive operation.** We report the average runtime of each primitive operation used in our framework, including video denoising, video preview inspection, single-frame denoising, single-frame preview inspection, and VLM calls. We also report the average detection step of the dynamic failure detector for video previews and single-frame previews. These measured runtimes are used together with the branch-wise execution counts in Table 8 to compute the per-sample and overall time overhead.

	Execution time (s)					**Dynamic failure detector average step**	
Models	Video denoising steps	Video preview inspections	Single-frame denoising steps	Single-frame preview inspections	VLM calls	Video preview	Single-frame preview
CogVideoX-5B	4.343	0.0392	0.405	0.0218	8.541	11.05	24.00
Wan2.1-1.3B	1.460	0.0392	0.103	0.0218	8.541	10.33	24.00

Table 8: **Detailed breakdown of execution statistics and time overhead.** We report the number of denoising steps, inspection operations, and wall-clock runtime for each execution branch. *Per-sample time overhead* measures the additional runtime relative to the original generation time when the branch is executed, while *Overall time overhead* represents the dataset-averaged contribution.

CogVideoX-5B: original generation time: 217.15 s (50 steps)										
		Execution statistics					**Runtime**		**Time overhead**	
Branch	Samples / Total	Video denoising steps	Video preview inspections	Single-frame denoising steps	Single-frame preview inspections	VLM calls	Average time (s)	Additional time (s)	Per-sample time overhead (%)	Overall time overhead (%)
Trial 0	562 / 946	50	11.05	0	0	0	217.58	0.43	0.20	0.12
Trial 1	91 / 946	60.05	11.05	24	24	0	271.47	54.32	25.02	2.41
Trial 2	125 / 946	61.05	11.05	24	24	1	284.36	67.21	30.95	4.09
Trial 2 → Trial 1	168 / 946	71.10	22.10	48	48	1	338.68	121.53	55.97	9.94
									Total overall time overhead: 16.55%	
Wan2.1-1.3B: original generation time: 73.00 s (50 steps)										
		Execution statistics					**Runtime**		**Time overhead**	
Branch	Samples / Total	Video denoising steps	Video preview inspections	Single-frame denoising steps	Single-frame preview inspections	VLM calls	Average time (s)	Additional time (s)	Per-sample time overhead (%)	Overall time overhead (%)
Trial 0	509 / 946	50	10.33	0	0	0	73.40	0.40	0.55	0.30
Trial 1	106 / 946	59.33	10.33	24	24	0	90.02	17.02	23.32	2.61
Trial 2	151 / 946	60.33	10.33	24	24	1	100.02	27.02	37.02	5.91
Trial 2 → Trial 1	180 / 946	69.66	20.66	48	48	1	117.04	44.04	60.34	11.48
									Total overall time overhead: 20.30%	

denoising, video preview inspection, single-frame denoising, single-frame preview inspection, and VLM calls.

We define the original generation time as the runtime of the default 50-step video generation without inspection or intervention. Even Trial 0 incurs a small non-zero overhead due to the preview inspection step. The per-sample time overhead is therefore computed as the additional runtime of each branch relative to the original generation time.

The overall time overhead is computed over the full evaluation set. Specifically, we measure the total runtime of applying our method to all 946 prompts, subtract the total runtime of original generation on the same 946 prompts, and divide the difference by the total original generation time. As a result, deeper branches can have high per-sample overhead, while their contribution to the final average time overhead remains limited because they are triggered less often. For example, on Wan2.1-1.3B, Trial 1 takes 90.02 s on average, compared with the original generation time of 73.00 s, resulting in 17.02 s additional runtime and 23.32% per-sample time overhead.

The numbers of video preview and single-frame preview inspections are determined by the average detection steps of the dynamic failure detector. For Trial 1 and Trial 2→1, single-frame semantic injection follows Algorithm 1 that starts from the second denoising step. Therefore, the subsequent video denoising process uses 49 steps instead of 50.

For Wan2.1-14B at 720p with 81 frames, each denoising step takes 67.052 s for video generation and 1.286 s for single-frame generation. Applying the same operation-based runtime accounting yields the additional time overhead of 8.17%. The lower relative overhead is due to the greater cost of full-video denoising in this larger-model regime.

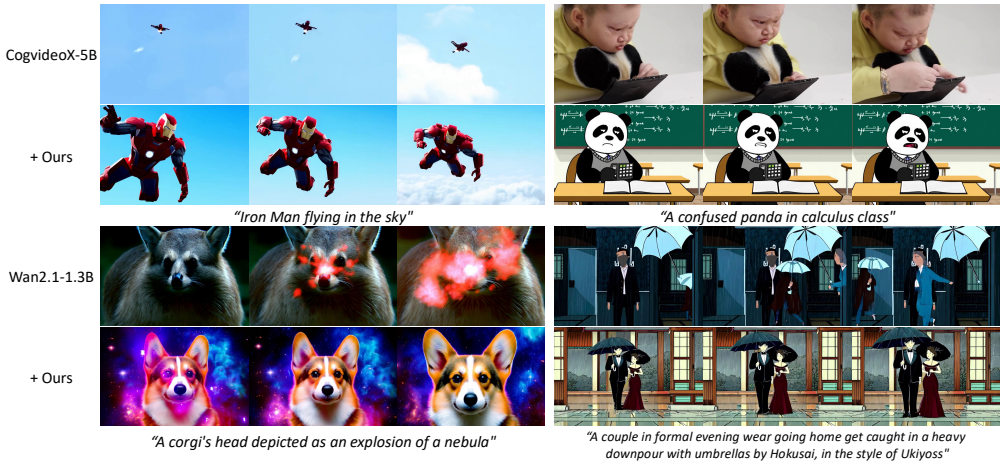


Figure 14: **Qualitative results.** Compared to baseline models, our method improves prompt adherence and visual fidelity by correcting early-failure generations. Top two rows: CogVideoX-5B. Bottom two rows: Wan2.1-1.3B.

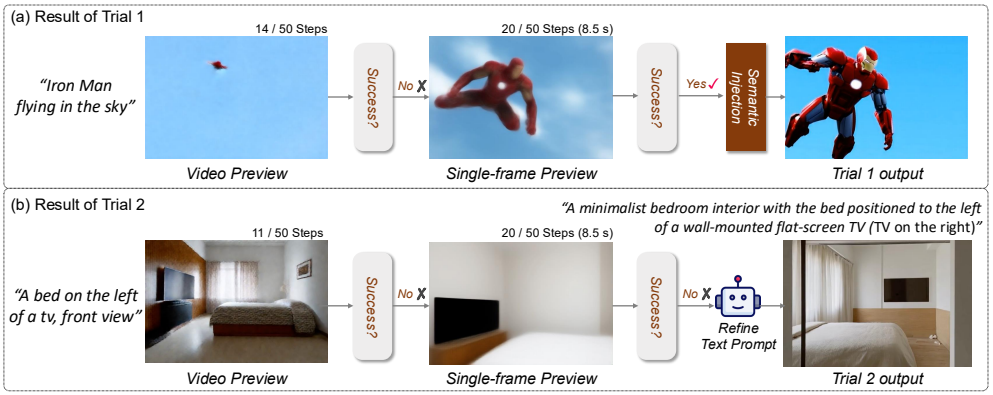


Figure 15: **Case visualization of diagnostic interventions. Top:** although the early video preview poorly reflects the prompt, the single-frame preview captures the semantics. The pipeline applies *single-frame semantic injection* (Trial 1), correcting the subject appearance without invoking a VLM. **Bottom:** both the video and single-frame previews place the TV on the left, violating the intended relation that the bed should be to the left of the TV. This demonstrates persistent prompt misalignment beyond temporal causes. The pipeline therefore invokes *Visual Feedback Intervention* (Trial 2), where a VLM refines the prompt based on the failed preview before regeneration.

C Additional experiments

C.1 Qualitative results of our pipeline

Figure 14 presents qualitative comparisons between baseline generations with and without our method on samples predicted as failures. Baseline outputs sometimes drift toward semantically misaligned content or exhibit weak subject depiction, whereas our pipeline mitigates these failure modes through early detection and selective intervention.

Figure 15 illustrates how the intervention framework performs hierarchical diagnostic trials. The Real-time Inspection (RI) module analyzes intermediate previews and predicts potential failures early in the denoising process, triggering targeted interventions with an early-exit strategy. In the top example, the early video preview poorly reflects the prompt while the single-frame preview captures the semantics, so the pipeline applies *single-frame semantic injection* (Trial 1), which corrects the subject appearance without invoking a VLM. In contrast, the bottom example shows a persistent prompt misalignment: both the video preview and Trial 1 fail to satisfy the intended spatial

Table 9: **Five-seed evaluation on VBench.** We re-run both base models in the Original prompt regime under VBench's official five-seed protocol. Results closely match the single-seed numbers in the main paper (Table 1), confirming that our reported gains are robust to seed choice.

Method	CogVideoX-5B			Wan2.1-1.3B		
	Final	**Quality**	**Semantic**	**Final**	**Quality**	**Semantic**
Original prompt (baseline)	0.776	0.803	0.670	0.787	0.813	0.686
+ Ours	**0.796**	**0.811**	**0.736**	**0.806**	**0.820**	**0.750**

relationship (the TV appearing on the left), indicating a failure beyond purely temporal causes. The pipeline therefore escalates to *Visual Feedback Intervention* (Trial 2), in which a VLM analyzes the failed preview and refines the prompt before regeneration. Additional qualitative examples of Trial 1 and Trial 2 are provided in Fig. 21 and Fig. 22, respectively.

C.2 Multi-seed evaluation on VBench

To complement the main paper results, we additionally evaluate under the VBench five-seed protocol on both base models in the Original prompt regime. The multi-seed numbers (Table 9) closely match the main paper: baseline Final scores differ by at most 0.003, and the gains from our intervention are nearly identical ($+0.020$ vs. $+0.022$ on CogVideoX-5B Final; $+0.019$ vs. $+0.016$ on Wan2.1-1.3B Final). The qualitative pattern is preserved across all dimensions. These results corroborate the main findings and confirm that the gains are not seed-specific.

C.3 Overhead comparison with test-time scaling methods

Recent test-time scaling methods for video generation [22, 30] share our goal of improving inference-time quality but take a fundamentally different approach: they search over multiple candidate trajectories at every step, whereas we monitor a single trajectory and intervene selectively.

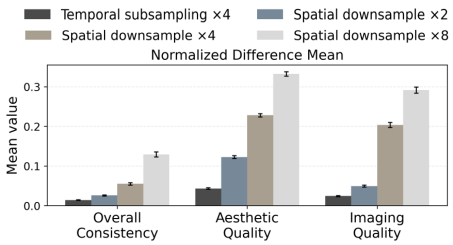


Figure 16: **Sensitivity of video quality metrics to temporal subsampling and spatial downsampling.** Compared to temporal subsampling, spatial downsampling causes larger deviations in VBench scores from the original videos.

Direct comparison is not straightforward: while Video-T1 [22] reports CogVideoX-5B results in its paper, the released code does not currently support CogVideoX-5B. DLBS [30], in turn, reports compute in NFE rather than wall-clock time, following the convention of inference-time scaling literature [27, 22].

We therefore report two complementary indicators of computational cost. First, on VBench, DLBS and its lookahead variant DLBS-LA report 1200 to 2400 NFE and 2500 to 5000 NFE, respectively, compared with 50 NFE for standard generation, corresponding to 24 to 48$\times$ and 50 to 100$\times$ more NFEs. Second, to obtain a concrete wall-clock reference, we ran the official DLBS `very_high` configuration (K=2, B=4, 50 DDIM steps, 49 frames, bf16) on CogVideoX-5B, on a single A100; this run takes 4 hours and 32 minutes per video.

These numbers are not intended as a strictly controlled benchmark, since NFE does not always translate linearly to wall-clock time under parallel batching or implementation-specific optimizations. Nevertheless, the order-of-magnitude gap relative to our measured 16.55% (CogVideoX-5B) and 20.30% (Wan2.1-1.3B) overhead suggests that multi-trajectory test-time scaling and our selective single-trajectory intervention operate in qualitatively different cost regimes. We therefore use this comparison only to contextualize the computational regime, not to claim a head-to-head runtime or quality advantage under matched implementations.

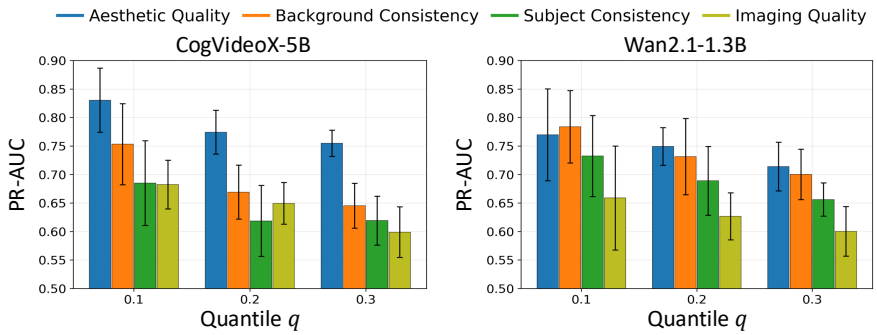


Figure 17: **Overall Consistency as a proxy for other VBench metrics.** Using all VBench prompts, we measure PR-AUC for predicting whether a sample falls into the top-q or bottom-q group of each target metric ($q \in \{0.1, 0.2, 0.3\}$) using Overall Consistency. All scores exceed the random baseline (0.5), indicating that Overall Consistency captures quality variations reflected by other metrics.

Table 10: **Comparison with a simple Best-of-3 baseline.** For *Best-of-3*, we generate three videos per prompt from the original prompt using different random seeds, score them with our alignment scorer (ViCLIP), and select the sample with the highest semantic alignment score. Since this requires three full generations, it incurs 200% additional time overhead relative to single-sample generation.

	CogVideoX-5B				Wan2.1-1.3B			
Method	Final	Quality	Semantic	Overhead (%)	Final	Quality	Semantic	Overhead (%)
Original prompt (baseline)	0.778	0.803	0.680	0.00	0.789	0.811	0.700	0.00
Best-of-3	0.787	0.802	0.728	200.00	0.796	0.812	0.730	200.00
Original prompt + Ours	**0.800**	**0.816**	**0.735**	**16.55**	**0.805**	**0.817**	**0.756**	**20.30**

C.4 Spatiotemporal sensitivity of VBench metrics

This section provides the empirical analysis behind the L2R converter's asymmetric spatiotemporal decoding design described in Sec. 3.1 of the main paper. Specifically, the L2R converter fully restores the $8\times$ spatial compression of the latent to preserve high-fidelity visual structures, while retaining the $4\times$ temporal compression inherent to the latent space without temporal upsampling. The rationale is that VBench metrics, which serve as our inspection signal, are substantially more sensitive to spatial than temporal resolution.

To verify this, we collect 1,024 real videos from Pexels [32], apply spatial downsampling and temporal subsampling at various rates, and measure the changes in Overall Consistency, Aesthetic Quality, and Imaging Quality [17] relative to the original videos. As shown in Fig. 16, the metrics are substantially more sensitive to spatial resolution: even mild spatial downsampling ($\times 2$) yields a larger normalized MAE than temporal subsampling ($\times 4$). This asymmetry justifies our design choice to spend the L2R converter's modeling capacity on spatial restoration while leaving the temporal axis at the latent's frame count, keeping the converter compact (0.059 M parameters) without sacrificing the inspection signal.

C.5 Overall Consistency as a proxy for video quality

This section provides the empirical analysis behind the choice of *Overall Consistency* as the single alignment signal used by the RI module, briefly summarized in Sec. 4.5 of the main paper. To fit the per-step latency budget of 19.5 ms, the RI module relies on a single alignment-based score; we examine here whether this single signal serves as a reliable proxy for broader video quality, that is, whether it can discriminate quality changes captured by other VBench metrics.

To this end, we generate videos using the full set of 946 VBench prompts and compute multiple VBench scores. For each target metric, we form a balanced binary classification task by selecting samples from the top-q and bottom-q quantiles ($q \in \{0.1, 0.2, 0.3\}$), and predict whether a sample belongs to the top or bottom group using Overall Consistency. We report PR-AUC with 10-fold cross-validation; since the subsets are balanced, a random classifier yields PR-AUC ≈ 0.5.

Table 11: **Effect of the success and failure threshold.** We compare the original prompt baseline with our selective early intervention under different failure thresholds τ. Higher thresholds make the detector more conservative, increasing the number of samples that receive intervention. As a result, VBench scores may improve further, but at the cost of substantially higher time overhead.

	CogVideoX-5B				Wan2.1-1.3B			
Method	Final	Quality	Semantic	Overhead (%)	Final	Quality	Semantic	Overhead (%)
Original prompt (baseline)	0.778	0.803	0.680	0.00	0.789	0.811	0.700	0.00
Original prompt + Ours ($\tau = 0.22$)	0.800	0.816	0.735	16.55	0.805	0.817	0.756	20.30
Original prompt + Ours ($\tau = 0.26$)	**0.807**	0.824	**0.740**	32.61	0.807	0.823	0.743	35.96
Original prompt + Ours ($\tau = 0.30$)	0.806	**0.827**	0.725	39.47	**0.812**	**0.826**	0.756	42.77

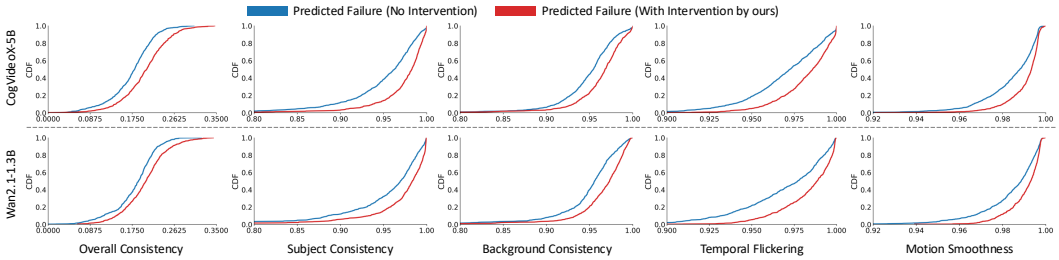


Figure 18: **Intervention improves detected failure cases.** CDFs of VBench metrics for samples predicted to fail and forced to complete without intervention, compared with our intervention. Despite using Overall Consistency for detection, our method shifts multiple VBench metrics rightward.

As shown in Fig. 17, Overall Consistency consistently achieves notable AUC scores across all evaluated dimensions. Performance is generally highest at $q = 0.1$, indicating stronger separability when focusing on more extreme cases. The signal is particularly predictive of Aesthetic Quality and Background Consistency, while remaining informative for Subject Consistency and Imaging Quality. These results justify our use of Overall Consistency as a single, low-cost inspection signal: it captures quality variations across multiple VBench dimensions, which is consistent with the broad per-dimension improvements reported in Table 12.

C.6 Comparison with a best-of-3 baseline

To further contextualize the efficiency–quality trade-off of our method, we compare it with a simple inference-time scaling baseline based on multi-seed sampling. For each prompt, we generate three videos from the original prompt using three different random seeds, compute the semantic alignment score of each sample using our alignment scorer (ViCLIP), and select the sample with the highest score to form the final evaluation set. We denote this baseline as *Best-of-3*. Since it requires three full generations per prompt, it incurs 200% additional time overhead relative to single-sample generation.

Table 10 compares *Best-of-3* with the original prompt baseline and our method on VBench. On CogVideoX-5B, *Best-of-3* improves the semantic score over the original baseline, but our method achieves a higher final score (0.800 vs. 0.787) and higher quality score (0.816 vs. 0.802) with only 16.55% overhead, compared with 200% for *Best-of-3*. On Wan2.1-1.3B, *Best-of-3* yields competitive final and semantic scores, but also requires substantially larger inference cost. These results suggest that, even compared with a simple multi-seed selection strategy using the same alignment scorer, our selective early intervention provides a substantially more favorable efficiency–quality trade-off.

C.7 Threshold sensitivity of early failure detection

We study the sensitivity of our method to the failure threshold τ used by the dynamic failure detector. As shown in Table 11, increasing τ causes more samples to be treated as likely failures and therefore receive intervention. This can further improve VBench scores, but it also leads to substantially larger time overhead on both CogVideoX-5B and Wan2.1-1.3B.

This trend is consistent with the observation in Fig. 8 of the main paper. The distribution of Overall Consistency for generated videos is already shifted to the left of the real-video distribution. As a result, increasing the threshold moves a larger portion of generated samples into the failure region, which

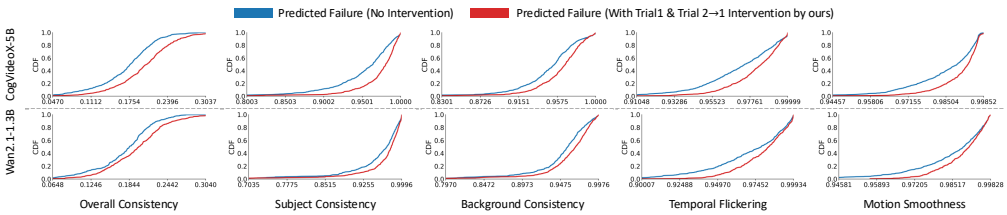


Figure 19: **Effect of the single-frame-based intervention family on detected failure cases.** We compare the CDFs of VBench metrics between predicted failure samples generated to completion without intervention and those corrected by the single-frame-based intervention family (Trial 1 and Trial 2→1). Across both CogVideoX-5B and Wan2.1-1.3B, the intervention shifts the distributions toward better scores over multiple VBench dimensions.

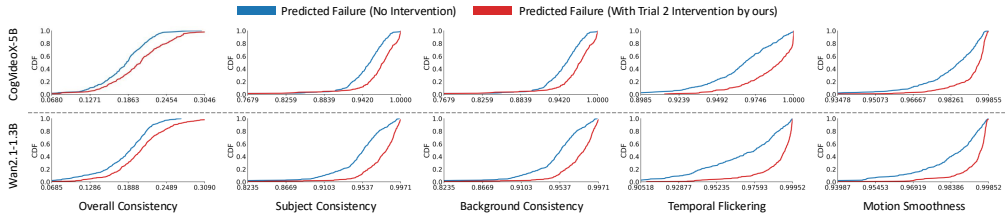


Figure 20: **Effect of Visual Feedback Intervention on detected failure cases.** We compare the CDFs of VBench metrics between predicted-failure samples generated to completion without intervention and those corrected by Visual Feedback Intervention (Trial 2). Across both CogVideoX-5B and Wan2.1-1.3B, this branch shifts the distributions toward better scores over multiple VBench dimensions.

raises the intervention trigger frequency. Consequently, higher thresholds can improve aggregate scores by intervening on more borderline samples, but they also increase inference cost significantly.

Consistent with this observation, increasing τ generally improves VBench scores up to a point, while substantially increasing time overhead. These results show that the failure threshold acts as a practical control knob for the quality–efficiency trade-off of selective early intervention.

C.8 Effectiveness of intervention on failure cases

We evaluate the impact of our intervention on samples flagged as early failures by the Real-time Inspection (RI) module. Because the dynamic failure detector operates per-model, the number of flagged samples differs slightly across generators. Out of the 1,800 prompts from VBench and Panda70M, the detector flags 872 samples as failures for CogVideoX-5B and 811 for Wan2.1-1.3B. Figure 18 compares the VBench metric CDFs of these flagged samples with and without our intervention pipeline. Although early failure detection is driven by a single signal (Overall Consistency), the intervention consistently improves multiple VBench dimensions, with clear rightward shifts in Overall Consistency, Subject Consistency, Background Consistency, Temporal Flickering, and Motion Smoothness across both CogVideoX-5B and Wan2.1-1.3B. This indicates that correcting failures detected through a single alignment-based signal yields broad quality gains beyond alignment alone.

C.9 Disaggregated analysis of intervention effects

The aggregate CDF shifts in Sec. C.8 combine the effects of all intervention branches. To isolate where the gains come from, we further decompose the results into two groups based on the final-stage correction applied: the *single-frame-based intervention family*, which includes Trial 1 and Trial 2→1, and *Visual Feedback Intervention* (Trial 2).

Single-frame-based intervention family. Figure 19 analyzes the family of branches that apply single-frame semantic injection as the final correction step. We group Trial 1 and Trial 2→1 together, since both branches share the same final-stage intervention. Compared with predicted-failure samples

generated to completion without intervention, the corrected samples show consistent rightward shifts in the CDFs across multiple VBench dimensions on both CogVideoX-5B and Wan2.1-1.3B. This experiment includes 629 samples on CogVideoX-5B and 505 samples on Wan2.1-1.3B, out of 1,800 prompts.

Visual Feedback Intervention. Figure 20 analyzes the effect of Trial 2, which invokes a VLM as a visual feedback reasoner over the failed preview to refine the prompt before regenerating. As in the main paper, we compare predicted-failure samples without intervention against those corrected by this branch. The resulting CDF shifts show that Visual Feedback Intervention also improves multiple VBench dimensions, complementing the single-frame-based family. This experiment includes 243 samples on CogVideoX-5B and 306 samples on Wan2.1-1.3B, out of 1,800 prompts.

D Additional qualitative results

In this section, we provide additional qualitative results for the proposed intervention framework. Specifically, we present representative examples of **Trial 1** (*single-frame semantic injection*) and **Trial 2** (*Visual Feedback Intervention*) to illustrate how the proposed method corrects early-failure trajectories during generation in Fig. 21 and Fig. 22, respectively. We also provide qualitative comparisons between the native decoders of CogVideoX-5B and Wan2.1-1.3B and the corresponding outputs of the proposed Latent-to-RGB (L2R) converter in Fig. 23 and Fig. 24, respectively, showing that L2R preserves the key semantic content and visual structure needed for real-time inspection.

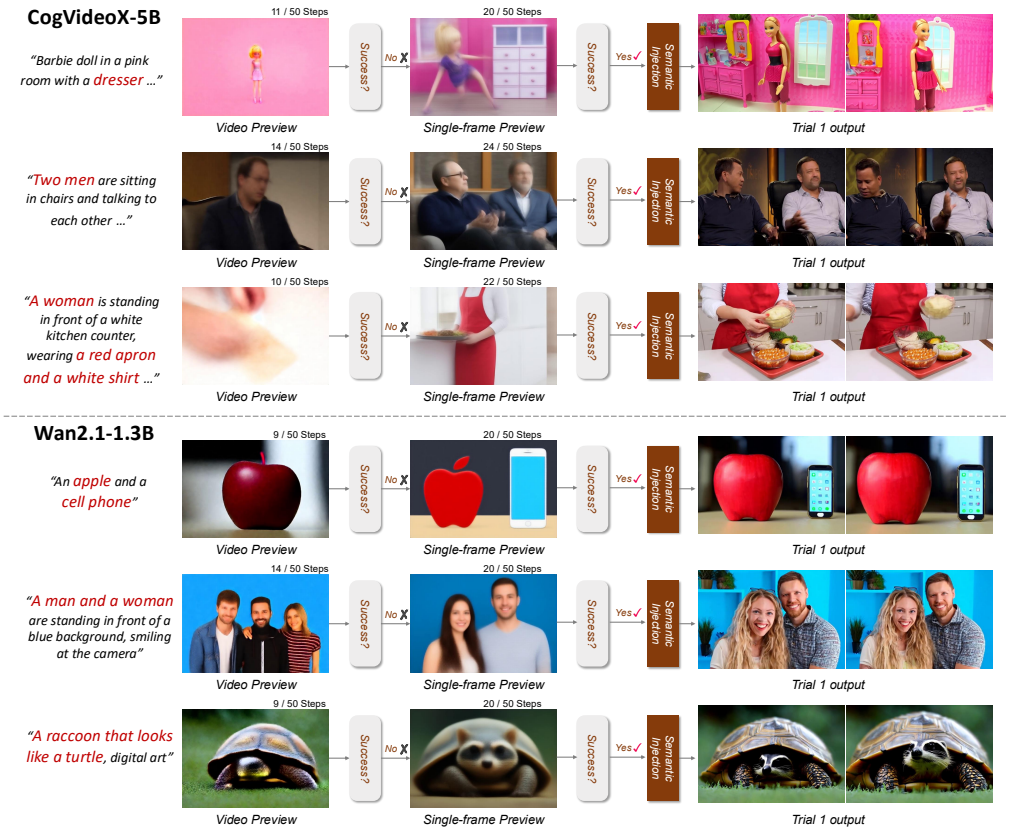


Figure 21: **Additional qualitative results of Trial 1 (single-frame semantic injection).** We present additional qualitative results from CogVideoX-5B (top) and Wan2.1-1.3B (bottom). For generations predicted as failures based on early video previews, Trial 1 generates a single-frame preview using the same T2V model with the temporal dimension collapsed. When the single-frame preview yields a higher alignment score than the video preview ($\tilde{s}_{\text{img}} > \tilde{s}_0 + \delta$), we reuse its intermediate prediction as a semantic anchor and inject it into the video latent over the first denoising steps. These examples show that Trial 1 improves prompt fidelity by recovering missing objects, attributes, and subject composition across diverse models and failure cases.



Figure 22: **Additional qualitative results of Trial 2 (Visual Feedback Intervention).** We present additional qualitative results from CogVideoX-5B (top) and Wan2.1-1.3B (bottom). For generations predicted as failures based on early video previews, Trial 2 invokes a vision–language model (VLM) as a visual feedback reasoner over the failed preview, returning a refined prompt and an optional negative prompt. The refined prompt strengthens the desired semantics, while the negative prompt suppresses undesired content or artifacts. These examples show that Trial 2 improves prompt fidelity by correcting subject identity, object count, composition, and visual attributes across diverse models and failure cases.

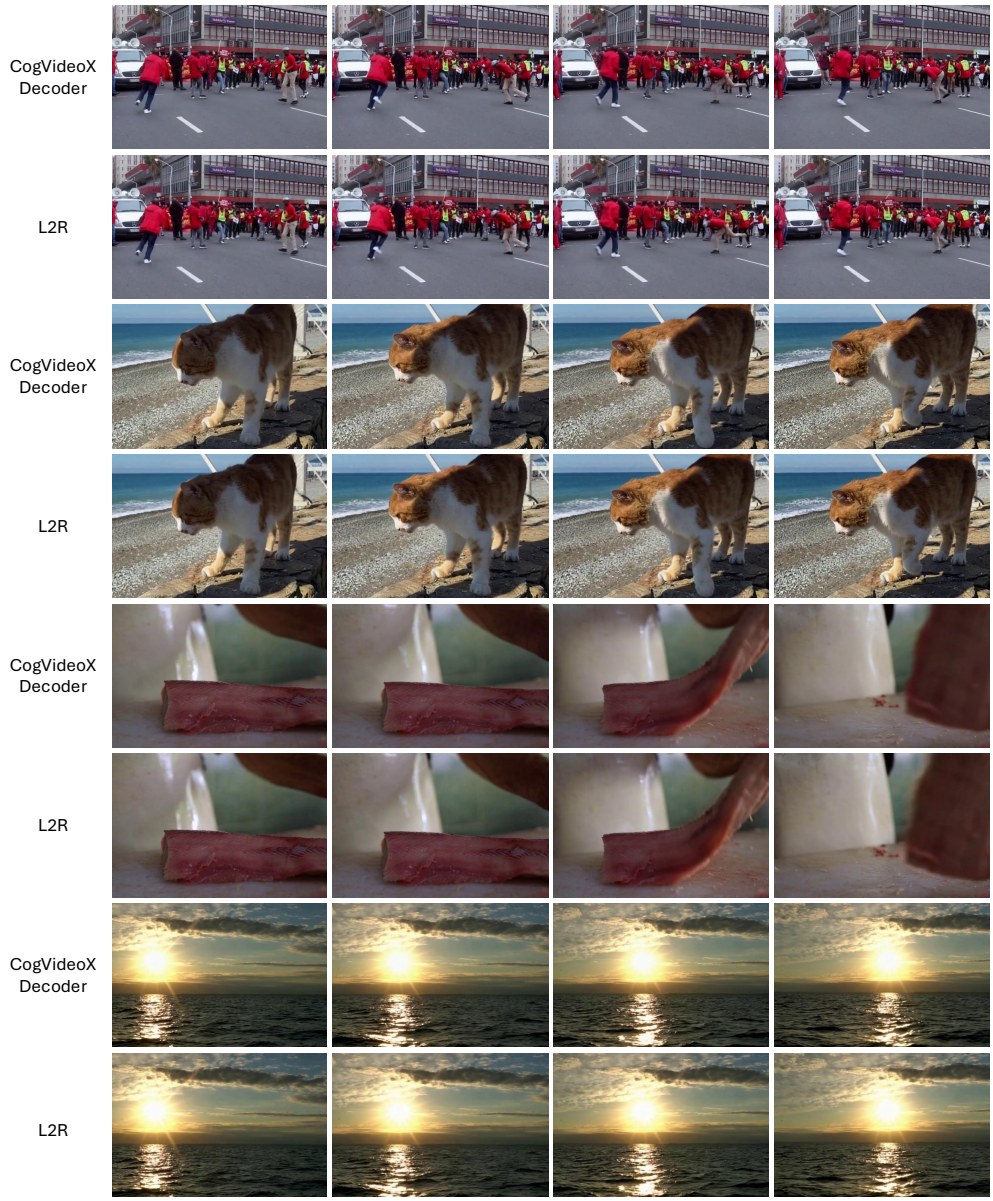


Figure 23: **Qualitative comparison between the native CogVideoX decoder and L2R converter.** We compare the reconstruction results between two methods. Across diverse scenes and multiple frames, L2R produces lightweight video previews that preserve the key semantic content and visual structure of the native CogVideoX decoder. These previews are sufficiently informative for real-time inspection and failure detection.

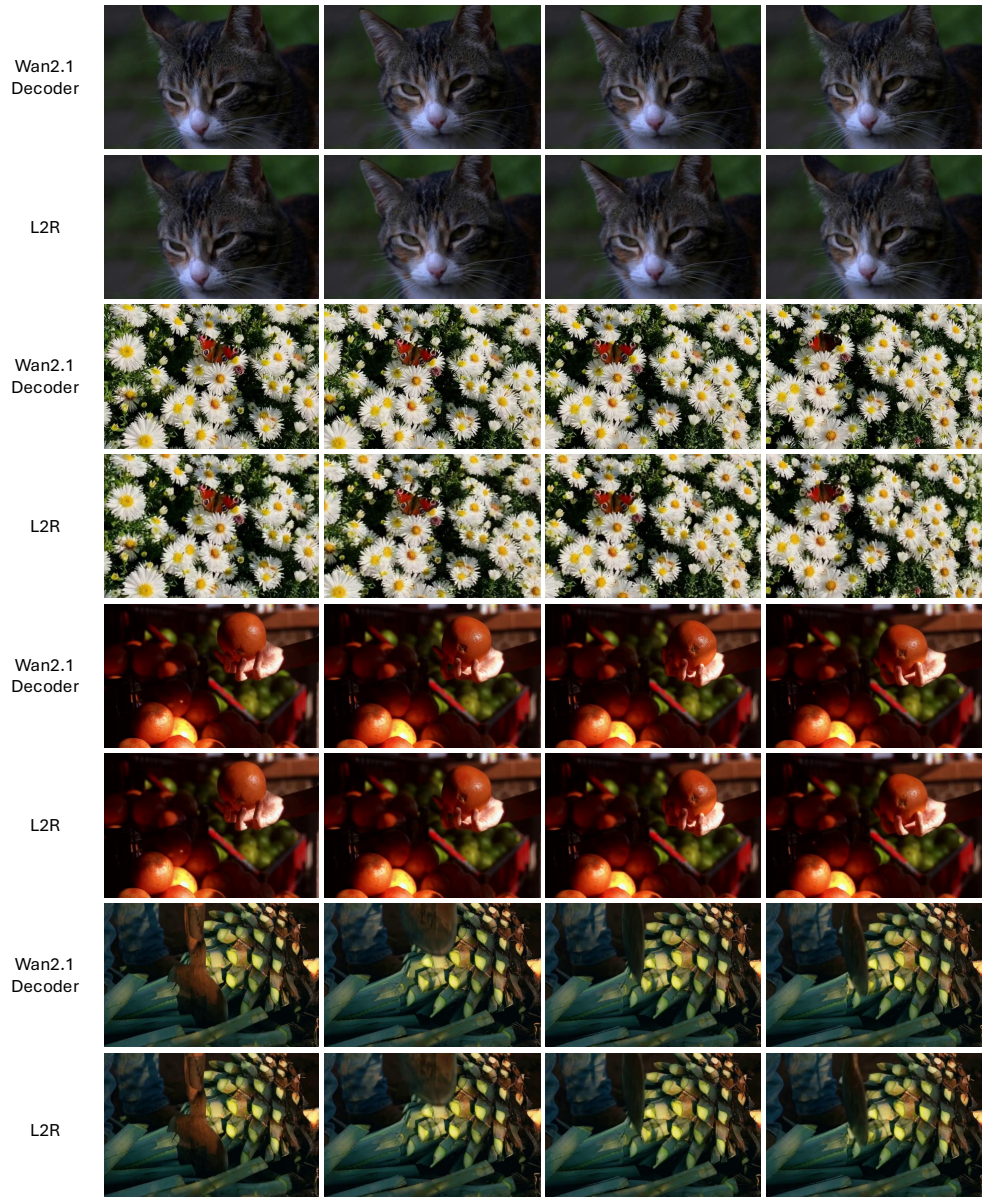


Figure 24: **Qualitative comparison between the native Wan2.1 decoder and L2R converter.** We compare the reconstruction results between two methods. Across diverse scenes and multiple frames, L2R produces lightweight video previews that preserve the key semantic content and visual structure of the native Wan2.1 decoder. These previews are sufficiently informative for real-time inspection and failure detection.

Table 12: **All VBench results across all dimensions.** We report all individual VBench dimension scores, along with the aggregated Semantic, Quality, and Final scores, for the original prompt baseline and our method on CogVideoX-5B and Wan2.1-1.3B. Overall, our method improves most VBench dimensions on both models, indicating that selective early intervention benefits not only alignment-related metrics but also broader perceptual and temporal aspects of video quality.

Dimension	**CogVideoX-5B**		**Wan2.1-1.3B**	
	Original Prompt	**+ Ours**	**Original Prompt**	**+ Ours**
Subject Consistency	0.957	**0.967**	0.949	**0.964**
Background Consistency	0.953	**0.964**	0.959	**0.972**
Temporal Flickering	0.969	**0.973**	0.991	**0.992**
Motion Smoothness	0.981	**0.985**	0.970	**0.979**
Dynamic Degree	**0.542**	**0.542**	**0.542**	0.403
Aesthetic Quality	0.592	**0.600**	0.590	**0.610**
Imaging Quality	0.610	**0.634**	0.644	**0.668**
Object Class	0.815	**0.866**	0.828	**0.900**
Multiple Objects	0.495	**0.656**	0.630	**0.697**
Human Action	0.870	**0.960**	0.770	**0.910**
Color	0.843	**0.859**	0.934	**0.943**
Spatial Relationship	0.450	**0.498**	0.687	**0.705**
Scene	0.393	**0.472**	0.289	**0.398**
Appearance Style	0.229	**0.231**	0.222	**0.223**
Temporal Style	0.238	**0.244**	0.238	**0.243**
Overall Consistency	0.258	**0.267**	0.243	**0.256**
Semantic Score	0.680	**0.735**	0.700	**0.756**
Quality Score	0.803	**0.816**	0.811	**0.817**
Final Score	0.778	**0.800**	0.789	**0.805**

Table 13: **Source of the Dynamic Degree trade-off.** We decompose Dynamic Degree across four settings on the full VBench evaluation set. On Wan2.1-1.3B, prompt rewriting alone accounts for most of the decrease, and full-video feedback regeneration, which uses the same VLM-based refinement step, exhibits a comparable drop. On CogVideoX-5B, our method preserves Dynamic Degree at the baseline level and outperforms regeneration.

Method	CogVideoX-5B	Wan2.1-1.3B
Original prompt (baseline)	0.542	0.542
Prompt rewriting	0.500	0.319
Full-video feedback regeneration	0.458	0.431
Original prompt + Ours	0.542	0.403

```
You are given sampled frames from an early-stage video generation preview.
This is not a final output.

Goal:
Improve overall consistency while preserving the original intent.

Inputs:
- original_prompt: "{prompt_en}"
- predicted_score_at_stop: {pred_score:.6f}
- score_range: 0.0 to 1.0 (higher is better)

Task:
- Evaluate the original prompt and preview frames together.
- Refine the prompt to better match the intended subject, action, and scene while fixing visible
      issues.

Guidelines:
- Base your revision on both the original prompt and what is visible in the preview.
- Make the prompt clearer, more concrete, and visually stable.
- Clarify subject visibility, action, framing, or scene when needed.
- Do not change the core meaning or add unrelated objects/actions.
- Keep the final prompt concise.

Negative prompt:
- Optional, single comma-separated string
- Short noun phrases only
- No instruction-style wording
- Maximum 5 items

Return ONLY valid JSON:
{
  "refined_prompt": "...",
  "negative_prompt": "..."
}
```

Figure 25: **Prompt template for Visual Feedback Intervention (Trial 2).** The VLM receives the original prompt, the decoded video preview, and the predicted preview score, and returns a refined prompt with an optional negative prompt in JSON format.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We state the main contributions in the abstract and Sec. 1, and they match the experimental scope: Sec. 4.2 reports the cost–quality trade-off against full-video feedback regeneration on CogVideoX-5B and Wan2.1-1.3B, Sec. 4.3 and 4.4 confirm generality across scorers/evaluators and orthogonality to existing enhancements, and Sec. 4.5 empirically grounds the design choices.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss limitations in Sec. 5.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: Our paper does not include theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: We describe the L2R converter (Sec. 3.1, Appendix A.2), the dynamic failure detector (Appendix A.3), and the diagnostic intervention framework (Sec. 3.2 with Algorithm 1). All hyperparameters (τ, δ) and inference settings (denoising steps, seeds, guidance scales) are specified in Sec. 3 and Sec. 4.1, and the VLM prompt template is in Fig. 25. Code and checkpoints are publicly available.

Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: Code and checkpoints are publicly available. All datasets used (VBench, Panda70M, DAVIS-2017, Pexels) are publicly available and cited.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We describe the experimental setup in Sec. 4.1 and Appendix A.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We report standard deviations in Table 6 and error bars in Fig. 8, Fig. 16, and Fig. 17.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: All experiments use a single NVIDIA A100. L2R training takes 11 hours, and per-operation and per-branch wall-clock times are reported in Tables 7 and 8.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Our research conforms to the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

Justification: Our method reduces GPU compute and the associated energy cost by detecting failed generations early rather than running them to completion or regenerating from scratch. As an efficiency improvement orthogonal to model capability, our contribution does not add new generative capacity or misuse vectors beyond those already present in the underlying T2V models.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: We do not release any new generative model or dataset that poses misuse risk. The L2R converter we release is a lightweight latent-to-RGB decoder with no generative capability.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: We cite all used assets (CogVideoX-5B, Wan2.1-1.3B, VBench, ViCLIP, OpenCLIP, VideoScore2, Panda70M, DAVIS-2017, Pexels) with their original sources.

Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: We publicly release the code and L2R checkpoints with documentation.

    Guidelines:

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

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

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

    Answer: [Yes]

    Justification: We describe the user study (26 participants, 30 samples, 780 responses) in Fig. 8.

    Guidelines:

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

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

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

    Answer: [Yes]

Justification: Participants were provided with a clear explanation of the study and its objectives before taking part, and participation was entirely voluntary. No personally identifiable information was collected

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: We use GPT-5-mini as the default VLM for Visual Feedback Intervention (Trial 2), and additionally evaluate Qwen3.8-27B as a replacement, as described in Sec. 4.1.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.