Self-Correcting Text-to-Video Generation with Misalignment Detection and Localized Refinement

Daeun Lee[1] **Jaehong Yoon**[2] **Jaemin Cho**[3,4] **Mohit Bansal**[1]
[1]UNC Chapel Hill [2]NTU Singapore [3]Allen Institute for AI [4]Johns Hopkins University
https://video-repair.github.io/

Abstract

Recent text-to-video (T2V) diffusion models have made remarkable progress in generating high-quality videos. However, they often struggle to align with complex text prompts, particularly when multiple objects, attributes, or spatial relations are specified. We introduce **VIDEOREPAIR**, the first self-correcting, training-free, and model-agnostic video refinement framework that automatically detects fine-grained text–video misalignments and performs targeted, localized corrections. Our key insight is that even misaligned videos usually contain correctly generated regions that should be preserved rather than regenerated. Building on this observation, VIDEOREPAIR proposes a novel region-preserving refinement strategy with three stages: (i) *misalignment detection*, where MLLM-based evaluation with automatically generated evaluation questions identifies misaligned regions; (ii) *refinement planning*, which preserves correctly generated entities, segments their regions across frames, and constructs targeted prompts for misaligned areas; and (iii) *localized refinement*, which selectively regenerates problematic regions while preserving faithful content through joint optimization of preserved and newly generated areas. On two benchmarks, EvalCrafter and T2V-CompBench with four recent T2V backbones, VIDEOREPAIR achieves substantial improvements over recent baselines across diverse alignment metrics. Comprehensive ablations further demonstrate the efficiency, robustness, and interpretability of our framework.

1 Introduction

Recent text-to-video (T2V) diffusion models (Ho et al., 2022; Singer et al., 2022; Esser et al., 2023; Blattmann et al., 2023; Khachatryan et al., 2023; Wang et al., 2023a; Yang et al., 2024; Wan et al., 2025) have achieved impressive photorealism and versatility across diverse domains. Despite these advances, current models often struggle to faithfully follow input text prompts, especially when the prompt specifies multiple objects and attributes. Typical errors include generating the wrong number of objects, mismatched attribute bindings, or distorted regions.

To mitigate these issues, recent works (Yang and Wang, 2024; Tian et al., 2024; Qu et al., 2025) propose compositional T2V techniques that improve text–video alignment. While these methods enhance compositionality, they lack explicit feedback mechanisms to detect and correct misalignments, limiting their adaptability and interpretability in real-world scenarios. In parallel, several image-based studies (Mañas et al., 2024; Wu et al., 2024; Chen et al., 2025; Ji et al., 2025; Xiang et al., 2025) have introduced training-free frameworks that refine outputs using guidance from LLMs or MLLMs. However, as shown in Fig. 1, these approaches are computationally expensive, dependent on external generators, or prone to visual inconsistencies.

To address these challenges, we introduce VIDEOREPAIR, the first self-correcting framework for text-to-video generation that is compatible with any diffusion-based T2V backbone in a training-free manner. Our key insight is that even when generated videos contain misaligned or distorted objects, certain key elements are often accurately generated in specific regions. Similar to how humans revise creative work by fixing only the errors while keeping what is correct, VIDEOREPAIR preserves accurately generated regions and selectively refines only the problematic ones. This region-preserving strategy leverages diffusion models' natural ability to regenerate content from noise while avoiding unnecessary changes to faithful areas. Moreover, detecting correctly generated content via grounding and segmentation is substantially easier than exhaustively enumerating all possible distortions, making our approach both reliable and efficient.

Building on this intuition, VIDEOREPAIR implements region-preserving refinement through three

Findings of the Association for Computational Linguistics: ACL 2026, pages 36464–36489

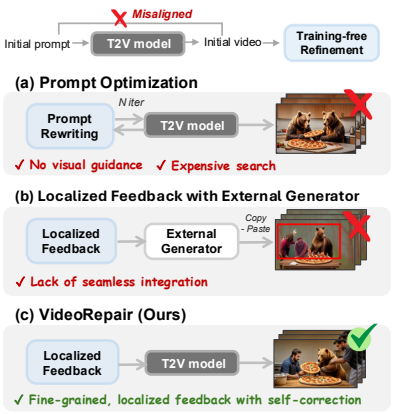

Figure 1: **Comparison with other baselines.** VIDEOREPAIR provides localized feedback with self-correcting.

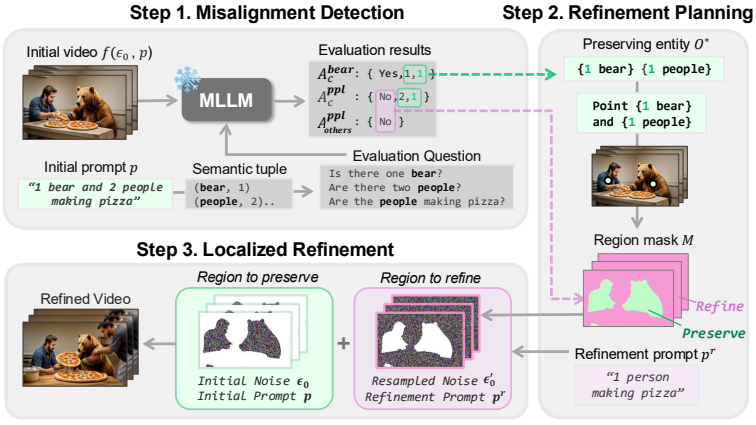

Figure 2: **Illustration of VIDEOREPAIR.** VIDEOREPAIR refines the generated video in three stages: (1) misalignment detection (Sec. 2.1), (2) refinement planning (Sec. 2.2) and (3) localized refinement (Sec. 2.3).

mutually reinforcing processes: *misalignment detection*, *refinement planning*, and *localized refinement* as illustrated in Fig. 2. Unlike prior refinement frameworks that operate on the entire video indiscriminately, VIDEOREPAIR follows a self-correcting, region-preserving paradigm: it distinguishes correctly generated regions from misaligned ones and regenerates only the latter. This transforms evaluation feedback into actionable generative guidance, allowing precise corrections without discarding high-quality content and establishing a new paradigm for efficient, interpretable video refinement. Specifically, spatio-temporal evaluation questions derived from the prompt expose fine-grained errors; these signals guide the selection of entities to preserve and the construction of a targeted refinement prompt; and localized regeneration is then harmonized with preserved regions to yield perceptually seamless videos.

We validate VIDEOREPAIR on two challenging benchmarks, EvalCrafter (Liu et al., 2024b) and T2V-CompBench (Sun et al., 2024a), which cover diverse prompt categories including object counts, spatial relations, and global scene attributes. Empirically, VIDEOREPAIR substantially outperforms existing refinement methods across a wide range of compositional prompts, while preserving global quality aspects such as visual fidelity, motion smoothness, and temporal consistency. We further provide detailed ablations on each component, error accumulation, inference latency, and robustness to different MLLM replacements, underscoring the generality and reliability of our framework.

Our key contributions are as follows:

- We present the first self-correcting, training-free framework for text-to-video generation, compatible with diffusion-based T2V backbones, that detects misalignments via MLLM-based evaluation and plans targeted refinements.
- We introduce a region-preserving refinement strategy that transforms evaluation feedback into actionable guidance, preserving correct regions while selectively regenerating misaligned ones, offering both effective correction and interpretable feedback.
- We show that VIDEOREPAIR consistently improves text–video alignment across diverse models and benchmarks, while maintaining fidelity, temporal coherence, and motion quality, outperforming all prior training-free refinement approaches.

2 VIDEOREPAIR

We introduce VIDEOREPAIR, the first *training-free, model-agnostic self-correcting* framework for text-to-video generation. Unlike prior refinement approaches that either use only prompt optimization (Mañas et al., 2024) or rely on external generative models (Wu et al., 2024), our approach follows a new principle: *preserve the correct region, selectively repair where it is wrong*. This principle distinguishes VIDEOREPAIR from generic mask-based inpainting or editing: the preserved regions are determined not by manual masks or heuristic rules, but by an automatic video evaluation and planning process that identifies semantically aligned objects directly from the input prompt and the generated video. By tightly coupling evaluation, planning,

and refinement within the same T2V backbone, VIDEOREPAIR enables localized regeneration that improves compositional fidelity while maintaining temporal consistency and visual realism.

Problem statement. Our goal is to improve text-video alignment using a pre-trained T2V diffusion model $f(\cdot)$, without requiring any additional fine-tuning. Given a text prompt p and initial noise $\epsilon_0 \sim \mathcal{N}(0, \boldsymbol{I})$, we generate an initial video $V_0 = f(p, \epsilon_0)$, where $V_0 \in \mathbb{R}^{3\times H\times W\times T}$ and H, W, and T denote the height, width, and number of frames, respectively. If V_0 exhibits misaligned or inaccurate content, we evaluate it using a set of questions derived from the prompt p and construct a refinement plan. Then, we perform localized self-refinement with the same T2V model $f(\cdot)$, producing a refined video V_1 that better aligns with the original prompt.

2.1 Misalignment Detection

Generate video evaluation questions. To diagnose the initial video V_0, we generate video evaluation questions from p. Unlike prior question-based evaluations in the image domain (Hu et al., 2023; Cho et al., 2024), these questions provide *spatio-temporal feedback signals* that directly guide refinement planning. It goes beyond simple object-existence checks by explicitly capturing *counts, attributes, spatio-temporal relations, actions, and scene-level global properties*, all of which are critical for faithful video-text alignment. Given a prompt p, we first extract a semantic tuple $\mathcal{T}$, a structured representation of entities, attributes, relationships, and actions relevant to the video. Using this as guidance, we employ in-context learning with an LLM to generate a set of evaluation questions Q. The resulting set Q is divided into two disjoint subsets: Q_c (questions focused on object counting) and Q_{others} (questions covering all other aspects, such as action, attributes, and scene-level global properties), reflecting the distinct nature of count-based reasoning versus semantic understanding. To better support fine-grained counting, we let the LLM generate count-specific questions for individual objects (e.g., "Is there *one* bear?") rather than merely verifying object existence (e.g., "Is there a bear?"). Our ablation study (see Tab. 3) demonstrates that these evaluation questions provide more effective refinement guidance compared to existing question-based evaluation methods. Additional details are provided in the Appendix Sec. A.1.

Answering to identify video errors. We now evaluate V_0 to determine which region requires refinement, as illustrated in Fig. 2 (top right). Given the entity set O (*i.e.*, object or scene element) from $\mathcal{T}$, we group $Q^o = \{Q^o_c, Q^o_{\text{others}}\} \subset Q$ as the subset of questions that contain the name of O Note that this entity captures not only localized object discrepancies but also global misalignments between p and V_0 To this end, we employ an MLLM to answer each predefined question set Q^o with binary judgments. For count-related questions Q^o_c, we prompt the model to output both a binary decision and an estimated object count, resulting in a triplet $A^o_c = \{b^o_c, n^o_p, n^o_v\}$, where n^o_p and n^o_v denote the number of instances of object o in the prompt p and the video V_0, respectively. The binary answer b^o_c is set to 1 if $n^o_p = n^o_v$, and 0 otherwise. For example, in Fig. 2, the question "Is there one bear?" results in $b^{\text{bear}}_c = 1$ when both the prompt and the video indicate a single bear (i.e., $n^{\text{bear}}_p = n^{\text{bear}}_v = 1$). For other type of questions Q^o_{others}, we prompt the model to return only a binary response $A^o_{\text{others}} = \{b^o_{\text{others}}\}$, where $b^o_{\text{others}} = 1$ indicates alignment between the element in V_0 and p, and $b^o_{\text{others}} = 0$ otherwise. If an entity disappears or becomes distorted across frames, we also regard it as a misalignment case. We aggregate binary evaluation results into a video-level accuracy score in the range $[0, 1]$. If the score is 1.0, we terminate the process early, as the initial video is already fully correct. If the score is 0.0, we instead re-generate the video with a new random seed to avoid uninformative outputs.

2.2 Refinement Planning

Identifying visual content to retain. As mentioned earlier, VIDEOREPAIR aims to retain accurately generated regions in the initial video while correcting only the mis-generated ones to ensure improved text-video alignment. To this end, we first identify the key entity O^* and determine the number N^* of its instances to be preserved. To select which entity should be retained, we prompt the MLLM with question–answer pairs and V_0 as input, allowing it to identify correctly generated entities to preserve. For countable entities, the number of preserved instances N^* is determined from the triplet $A^{o^*}_c = \{b^{o^*}_c, n^{o^*}_p, n^{o^*}_v\}$ as

$$N^* = \begin{cases} n^{o^*}_p & \text{if } n^{o^*}_p \leq n^{o^*}_v, \\ n^{o^*}_v & \text{otherwise,} \end{cases} \tag{1}$$

where $n_p^{o^*} < n_v^{o^*}$ indicates that excess instances should be removed, and $n_p^{o^*} > n_v^{o^*}$ suggests that additional instances are required. For example, in Fig. 2, if O^* represents people with $n_p^{o^*} = 2$ and $n_v^{o^*} = 1$, we set $N^* = 1$ to preserve one person. Note that multiple instances of O^* may exist if there are several plausible entities to retain. For *global* scene elements (e.g., background), which are not inherently countable, we treat N^* as a presence indicator, setting $N^* = 1$ if the element is preserved. This unified notation allows us to consistently handle both entity- and scene-level preservation within the same refinement planning framework. (see Appendix Sec. D.4 for global refinement performance).

Identifying regions to preserve. Based on the entity selection O^*, we localize the regions corresponding to correctly generated content within the video frames, as shown in Fig. 2 (top right). For countable entities, given the set of preserved instances O^* and their quantities N^*, we first construct a pointing prompt using the template: "Point the biggest $\{N^*\}$ $\{O^*\}$" (e.g., "Point the biggest 1 bear"). This prompt is used to obtain 2D coordinates indicating the spatial locations of O^* in each sampled frame. Using these coordinates as initialization, we apply a segmentation model to extract entity-specific regions, resulting in binary segmentation masks $\mathbf{M} \in \mathbb{R}^{H \times W \times T}$ that preserve the correctly generated entities. In practice, for global elements such as background, we simply preserve the entire frame region or assign a broad background mask if the property is rendered correctly. By combining these with the region-level masks, we obtain a dense, frame-aligned segmentation map $\mathbf{M}$ that preserves both entity- and scene-level regions.

Prompt regeneration for regions requiring refinement. We additionally generate a local prompt for refinement to enable distinct control over different regions during generation. To this end, we prompt an LLM to produce a refinement-oriented prompt, p^r, based on Q but excluding any questions related to O^*. As illustrated in Fig. 2, this regenerated local prompt will be used to guide the denoising process for specific areas to be refined during video generation in a later stage.

2.3 Localized Refinement

At this stage, we refine the video to improve alignment while preserving coherence with the original content. While video editing (Jiang et al., 2025; Yang et al., 2025) preserves masked regions and enforces visual consistency, it is limited in its ability to freely introduce or correct entities misaligned with the original prompts. Similarly, inpainting (Lugmayr et al., 2022; Bian et al., 2025) fills missing regions with locally consistent textures but lacks mechanisms for semantically guided object introduction or correction from textual input. (See Tab. 3) Instead of these approaches, we selectively re-initialize noise only in misaligned regions and apply distinct text prompts to preserved and refined areas, enabling targeted corrections while maintaining overall video consistency.

Localized noise re-initialization. We adopt a mask-based strategy in which only regions marked for refinement are re-initialized with newly sampled noise $\epsilon_0' \sim \mathcal{N}(0, \mathbf{I})$, while preserved regions retain their original noise ϵ_0. This selective resampling maintains consistency in faithful areas while allowing controlled updates in misaligned ones. To transform the pixel-level mask $\mathbf{M}$ into the latent space, we apply block averaging (pooling), yielding a hybrid noise map:

$$\epsilon_0^* = (\epsilon_0 \otimes \text{pool}(\mathbf{M}, d)) + \big(\epsilon_0' \otimes (1 - \text{pool}(\mathbf{M}, d))\big), \tag{2}$$

where $\text{pool}(\cdot, d)$ downsamples the mask and $\otimes$ denotes element-wise multiplication. This noise map ϵ_0^* is then used with localized prompts to guide the frozen diffusion model.

Localized text guidance. Afterward, we apply distinct text prompts to different spatial regions of the video based on their noise re-initialization status, using the binary segmentation mask $\mathbf{M}$ to separate preserved (M_{pres}) and re-initialized ($M_{\text{refine}} = 1 - M_{\text{pres}}$) areas. For the re-initialized regions, we guide generation in the latent space using regenerated prompts p^r (See Sec. 2.2) tailored to those areas. In parallel, motivated by recent findings on noise bias (Sun et al., 2024b; Ban et al., 2024; Qi et al., 2024), we reuse the original prompt p to preserve features related to O^* in the retained regions. This regionalized decomposition of the original prompt allows for the addition or modification of objects in re-initialized areas, while maintaining the integrity of correctly generated content in preserved regions.

Harmony with original elements. To further ensure global coherence between preserved and refined regions, we regenerate all pixels through two

Table 1: **Evaluation results on EvalCrafter with other baselines.** Note that we focus on these four splits, whereas the official website reports the average across all splits. We highlight the quality and consistency performance in red if it deteriorates by more than 1% from the original performance.

Method	Text-Video Alignment					Visual Quality	Motion Quality	Temporal Consistency
	Count	Color	Action	Others	Avg.			
VideoCrafter2	47.52	46.28	44.07	46.02	45.97	61.8	62.6	62.9
+ LLM paraphrasing	45.87	47.81	44.41	45.16	45.81	62.4	62.7	62.7
+ SLD	44.47	46.45	39.89	44.06	43.72	52.5	62.2	44.4
+ OPT2I	47.69	47.67	45.04	44.65	46.26	62.1	62.6	62.8
+ VIDEOREPAIR (Ours)	**49.84**	**51.57**	**45.78**	**48.12**	**48.83**	62.1	62.4	62.0
T2V-turbo	46.14	43.06	41.42	43.16	43.94	63.3	57.8	61.6
+ LLM paraphrasing	49.49	43.16	41.32	44.75	44.68	62.9	52.9	61.9
+ SLD	47.39	43.99	42.13	43.28	44.20	56.6	58.2	49.2
+ OPT2I	47.44	45.00	44.64	**45.54**	45.66	63.3	56.4	48.9
+ VIDEOREPAIR (Ours)	**51.27**	**46.66**	**45.81**	45.45	**47.30**	63.2	57.9	61.8
CogVideoX-5B	47.88	49.63	37.76	44.78	45.01	65.8	61.0	61.8
+ LLM paraphrasing	45.58	46.56	37.17	43.18	43.12	58.4	61.1	61.7
+ SLD	47.73	46.27	39.55	43.75	44.33	49.6	51.2	21.0
+ OPT2I	48.62	48.89	**41.39**	43.62	45.63	59.7	60.9	61.9
+ VIDEOREPAIR (Ours)	**49.63**	**49.94**	40.69	**45.36**	**46.41**	64.8	61.1	61.9
Wan 2.1-1.3B	45.06	48.18	41.06	45.00	44.83	63.2	61.0	62.1
+ LLM paraphrasing	44.38	47.19	42.35	44.03	44.49	63.5	61.2	62.2
+ SLD	48.24	49.77	43.64	46.78	47.11	49.1	58.3	32.5
+ OPT2I	49.10	51.86	45.88	47.92	48.69	64.3	61.0	62.0
+ VIDEOREPAIR (Ours)	**50.03**	**51.97**	**46.01**	**48.30**	**49.01**	65.1	61.6	62.0

separate diffusion paths and fuse them via joint optimization. Specifically, at each denoising step t, we run the diffusion model $f(\cdot)$ twice with different prompts and noises: $\hat{V}_{\text{pres}} = f(V_t, p, \epsilon_0)$ for the preserved regions, and $\hat{V}_{\text{refine}} = f(V_t, p^r, \epsilon_0')$ for the refined regions. The final fused output $\tilde{V}$ is obtained by solving:

$$V_1 = \arg\min_{\tilde{V}} \left\| M_{\text{pres}} \otimes (\tilde{V} - \hat{V}_{\text{pres}}) \right\|^2 + \left\| M_{\text{refine}} \otimes (\tilde{V} - \hat{V}_{\text{refine}}) \right\|^2 \quad (3)$$

This joint optimization allows $\tilde{V}$ to seamlessly blend preserved and refined regions, reducing mismatches at region boundaries and producing perceptually smooth, globally coherent videos.

Video ranking. Similar to generating multiple candidate prompts in (Mañas et al., 2024), we produce K refined videos using different random seeds and select the best one based on our video scores, as obtained in Sec. 2.1, thus avoiding additional computations or resource burdens. If multiple videos receive a tied video score, we select the video with the highest BLIP-BLEU score (Liu et al., 2024b) among them.

3 Experiments

3.1 Experiment Setups

Benchmarks and evaluation metrics. We evaluate our method on two T2V generation benchmarks: EvalCrafter (Liu et al., 2024b) and T2V-CompBench (Sun et al., 2024a).[1] Details are provided in the Appendix Sec. C.

(1) EvalCrafter. We follow the official metadata to split prompts by attributes into four sections: *count*, *color*, *action*, and *others*. The *others* category includes scenery-related prompts, such as Camera movement (e.g., *"Zoom in"*), Landscape (e.g., *"A bustling street in Paris"*), and Style (e.g., *"Polaroid style"*). For evaluation metrics, we report four groups: text-video alignment, video quality, motion quality, and temporal consistency.

(2) T2V-CompBench. We adopt three compositional reasoning categories from this benchmark: spatial relationships, generative numeracy, and consistent attribute binding. We use ImageGrid-LLaVA (Liu et al., 2024a) for consistent attribute binding evaluation and GroundingDINO (Liu et al., 2023) for the other two dimensions.

Implementation details. We apply VIDEOREPAIR on four recent T2V models: T2V-turbo (Li et al., 2024), VideoCrafter2 (Chen et al.,

[1] All reported results are based on our own experiments. We use ver.1 of T2V-CompBench.

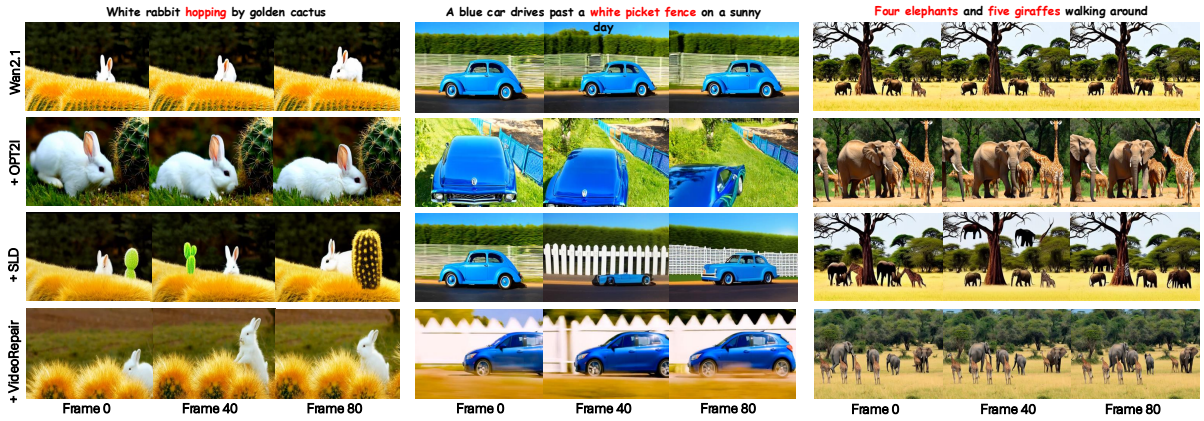

Figure 3: **Comparison of refinement framework with VIDEOREPAIR (backbone: Wan2.1-1.3B).** VIDEOREPAIR successfully refines misaligned motion, adds missing components, and corrects numeracy.

Table 2: **Evaluation results on T2V-CompBench.**

Method	Consist-Attr	Spatial	Numeracy	Avg.
ModelScope	0.5148	0.4118	0.1986	0.3750
ZeroScope	0.4011	0.4287	0.2408	0.3568
Latte	0.4713	0.4340	0.2320	0.3791
Show-1	0.5670	0.4544	0.3086	0.4433
Open-Sora-Plan	0.4246	0.4520	0.2331	0.3699
Vico	0.6470	0.5425	0.2762	0.4886
VideoTetris	0.6211	0.4832	0.3467	0.4836
VideoCrafter2	0.6812	0.5214	0.2906	0.4977
+ VIDEOREPAIR	**0.7275**	**0.5690**	**0.3278**	**0.5383**
T2V-turbo	0.7025	0.5492	0.2496	0.5004
+ VIDEOREPAIR	**0.7675**	**0.5807**	**0.2709**	**0.5439**
CogVideoX-5B	0.6220	0.4988	0.2228	0.4479
+ VIDEOREPAIR	**0.6725**	**0.5811**	**0.3034**	**0.5190**
Wan2.1-1.3B	0.6870	0.5690	0.3516	0.5358
+ VIDEOREPAIR	**0.7262**	**0.5841**	**0.3837**	**0.5646**

2024), CogVideoX-5B (Yang et al., 2024), and Wan2.1-1.3B (Wan et al., 2025). T2V-Turbo and VideoCrafter2 generate 16 frames, while CogVideoX-5B and Wan2.1 generate 81 frames. All experiments use K as 5 with a single iteration. For MLLM and LLM, we primarily use GPT-4o and for pointing and segmentation, we employ MolmoE-1B (Deitke et al., 2024) and Semantic-SAM (L) (Li et al., 2023a). Additional details are provided in the Appendix Sec. A.

Baselines. We compare VIDEOREPAIR against recent refinement frameworks, OPT2I (Mañas et al., 2024) and SLD (Wu et al., 2024), on the same three T2V models described above. Although these baselines were originally proposed for image refinement, we extend their implementations to the video setting. For OPT2I, we score the videos using the original DSG (Cho et al., 2024) and iteratively generate five prompt candidates. For SLD, since its refinement model is based on LMD+ (Lian et al., 2023), we apply SLD frame-by-frame to the initial outputs of T2V models. We also include LLM paraphrasing as a baseline, where GPT-4 generates diverse paraphrases of the initial prompt. To ensure fairness, we unify random seeds across all experiments so that all methods refine the same initial videos. Further details are provided in the Appendix Sec. B.

3.2 Quantitative Results

As shown in Tab. 1 and Tab. 2, VIDEOREPAIR consistently outperforms other refinement baselines as well as strong compositional T2V models (e.g., Vico, VideoTetris) across both benchmarks. On EvalCrafter, VIDEOREPAIR achieves relative alignment gains of **+6.22%**, **+7.65%**, **+3.11%** and **+9.32%** over VideoCrafter2, T2V-turbo, and CogVideoX-5B, Wan2.1 respectively. In addition, VIDEOREPAIR effectively corrects misalignments while preserving visual fidelity (std. deviation of visual quality scores: 0.55). By contrast, SLD underperforms particularly in the *action* and *count* categories because its frame-level latent fusion fails to maintain consistent object counts and spatial layouts over time. Although OPT2I yields only modest improvements, it operates solely in the textual domain without visual guidance, which limits its ability to correct fine-grained spatiotemporal localized misalignments. Moreover, its iterative search procedure, involving multiple LLM calls, makes the refinement process computationally expensive (will be discussed in Sec. 3.4).

3.3 Qualitative Results

Fig. 3 presents qualitative comparisons of refinement frameworks (OPT2I, SLD, and VIDEORE-

Table 3: **Ablations of VIDEOREPAIR components.** We replace each stage while keeping others fixed.

Question type (Sec. 2.1)	Planning (Sec. 2.2)	Ranking metric (Sec. 2.3)	Avg.
-	-	-	43.54
-	-	Ours	44.68
DSG	Random	Ours	45.18
Ours	Random	Ours	46.92
Ours	Ours	CLIP	45.85
Ours	Ours	BLIP-BLEU	47.77
Ours	Ours	Ours	**47.91**

Table 4: **Robustness under MLLM substitution.** We replace components in each stage with different MLLMs and report the average T2V alignment.

Misalign Det. (Sec. 2.1)	Planning (Sec. 2.2)	Ranking (Sec. 2.3)	Avg.
Human	Human	GPT-4o	49.05
Human	GPT-4o	GPT-4o	48.46
Human	GPT-4o	Gemini-2.5-Flash	**49.08**
GPT-4o	GPT-4o	GPT-4o	47.91
Qwen2.5VL-7B	GPT-4o	GPT-4o	48.61
Qwen2.5VL-7B	GPT-4o	Gemini-2.5-Flash	48.79

PAIR) applied to the Wan2.1 backbone. In the leftmost example, VIDEOREPAIR preserves the *golden cactus* from the initial video while refining the *white rabbit's motion* to a hopping action. In the middle example, VIDEOREPAIR maintains the *blue car* and more clearly introduces the *white picket fence* compared to the initial video. In the rightmost example, VIDEOREPAIR preserves the *two elephants* from the initial video while correcting object numeracy during refinement. Overall, these examples demonstrate that VIDEOREPAIR effectively refines misaligned motion, introduces missing components, and corrects numerical inconsistencies.

3.4 Additional Analysis

In this section, we present additional analyses of VIDEOREPAIR, including ablations of each component (Tabs. 3 to 5), error analysis (Tabs. 6 and 7), inference latency (Fig. 4), and iterative refinement results (Fig. 5). In each table, we report the average text–video alignment performance (Avg.) across the *Count*, *Color*, and *Action* splits of EvalCrafter using the T2V-turbo backbone. Our default setup is highlighted with a purple background.

Ablations of each step's components. In Tab. 3, we analyze the contributions of individual components, including the evaluation question type, planning strategy, and ranking metric. Since VIDEOREPAIR operates as a sequential pipeline, we ablate each component by replacing it with an alternative while keeping all other steps unchanged. For evaluation questions, we compare the original DSG questions with our proposed questions. For the object selection, we evaluate random planning against our strategy guided by previous evaluation results. For the ranking stage, we compare different metrics for final video selection, including CLIP, BLIP-BLEU, and our ranking method. Comparing with just applying ranking on T2V-turbo generations (44.64), VIDEOREPAIR achieves a notable improvement (47.91).

Table 5: **Ablations of localized refinement.**

Guidance M	*Guidance* p^r	Model	Avg.
✓		VideoGrain	40.52
✓		VACE	46.77
✓	✓	VACE	44.88
✓	✓	Ours	**47.91**

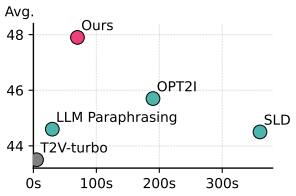

Figure 4: **Inference latency of refinement methods.**

Ablations of localized refinement models. In Tab. 5, we evaluate the effectiveness of our localized refinement by comparing it with state-of-the-art video-to-video (V2V) editing models, including VACE (Jiang et al., 2025) and VideoGrain (Yang et al., 2025), using our planning outputs (M and p^r) as editing guidance. Specifically, using M alone corresponds to masked V2V editing, while using both M and p^r incorporates p^r as an additional textual prompt during masked V2V editing. Although masked V2V editing can partially resemble our localized refinement, its editing capability is limited to modifying attributes or shapes of correctly generated objects. In contrast, our localized refinement generalizes to diverse failure cases by identifying and correcting various types of misaligned objects and errors. More examples are provided in Fig. 8.

Ablations of MLLM components. We analyze the impact of replacing VIDEOREPAIR's components across different stages (video evaluation, planning, and ranking) with human annotations and various MLLMs (GPT-4o, Qwen2.5VL-7B, and Gemini-2.5-Flash). As shown in Tab. 4, VIDEOREPAIR remains robust under diverse model substitutions, with only minimal performance variation, highlighting the flexibility and modularity of the framework. Using human evaluators for video

Figure 5: **Iterative refinement.** VIDEOREPAIR progressively improves text–video alignment, correcting object counts and attributes.

Table 6: **Categorized breakdown of failures.** We analyze failure modes across different stages of the pipeline and report their occurrence percentages.

Category	Error Type	Percentage (%)
Misalignment Detection	QA hallucination	35.3
Refinement Planning	Planning error	11.8
Localized Refinement	Mask drift	58.8
	Boundary artifacts	23.5
	Identity inconsistency	47.1

evaluation and planning provides a strong upper bound, while replacing them with GPT-4o results in only a modest degradation, indicating limited error accumulation when using strong proprietary models. Furthermore, mixing different models across stages can be beneficial: combining complementary strengths of models (e.g., Qwen2.5VL-GPT4o-Gemini) maintains or even slightly improves performance. This suggests that VIDEOREPAIR not only tolerates heterogeneous components but can also leverage them synergistically, enabling flexible deployment under varying resource constraints.

Inference latency. Fig. 4 presents the relationship between text–video alignment performance and inference latency across different refinement frameworks. OPT2I and SLD incur 3–5× higher inference latency compared to VIDEOREPAIR, without achieving superior alignment performance. OPT2I requires an iterative prompt search procedure, involving multiple rounds of video generation and DSG-based evaluation. Similarly, SLD depends on an external generator to synthesize missing objects and performs frame-wise copy-and-paste operations, resulting in additional computational overhead. In contrast, VIDEOREPAIR achieves the best trade-off between text–video alignment and efficiency, achieving strong performance with relatively modest inference cost. Step-wise inference latency and detailed cost comparisons are provided in Appendix Tabs. 8 and 9.

Table 7: **Error propagation analysis.** Conditional probabilities of downstream errors given upstream failures.

Upstream Error	Downstream Error	Prob. (%)
QA hallucination	Planning error	16.7
QA hallucination	Mask drift	21.3
QA hallucination	Identity inconsistency	9.8
Planning error	Mask drift	20.5
Planning error	Identity inconsistency	10.3
Mask drift	Identity inconsistency	**43.5**
No mask drift	Identity inconsistency	28.6

Iterative refinement. We further explore iterative refinement to progressively enhance text-video alignment, as a single refinement step of VIDEOREPAIR may not fully resolve all inconsistencies with the prompt. As illustrated in Fig. 5, the first refinement partially corrects the misalignment by generating a scene of *a night of camping under the stars*, but some family members disappear. In the second iteration, VIDEOREPAIR recovers all four family members while preserving the rest of the scene. Similarly, in the bottom example of Fig. 5, iterative refinement successfully produces the intended output of seven puppies. Additional qualitative examples are provided in the Appendix Sec. E.1.

Categorized failure analysis. To better understand failure modes, we randomly sample 50 failure cases and conducted a manual categorization analysis. We identify five representative error types: *QA hallucination* (incorrect misalignment detection), *planning error* (incorrect object selection), mask drift (frame-wise segmentation inaccuracies), *boundary artifacts* (unnatural blending), and *identity/motion inconsistencies* in preserved regions. Since multiple errors co-occur in a single case, percentages do not sum to 100%. As shown in Tab. 6, mask drift (58.8%) and identity/motion inconsistencies (47.1%) are the most frequent failure modes. Mask drift primarily arises from frame-level segmentation variability under occlusion. Identity/motion inconsistencies stem from the joint optimization formulation: although preserved regions retain original noise initialization and conditioning, they are not strictly frozen and may slightly adjust during least-squares fusion, leading to subtle perceptual shifts.

Error propagation analysis. To quantitatively assess potential error propagation, we analyze conditional probabilities $P(\text{Downstream Error}|\text{Upstream Error})$ based on the manually categorized failure cases described above. As shown in Tab. 7, *error propagation is neither deterministic nor systematically cascading across stages.* QA hallucinations rarely propagate downstream, as we conservatively preserve only confidently verified objects and filter out spurious detections. Planning errors are similarly weakly coupled with mask-level failures: even when an incorrect object is selected, refinement proceeds consistently with that selection rather than amplifying segmentation instability. In contrast, mask drift exhibits the strongest downstream effect, emerging as the primary driver of identity or motion inconsistencies during refinement.

4 Related Works

Text-to-video generation with diffusion models. Text-to-video (T2V) diffusion models (Esser et al., 2023; Wu et al., 2023b; Blattmann et al., 2023; Luo et al., 2023; Yang et al., 2024; Wan et al., 2025) aim to produce videos describing given text prompts. VideoCrafter2 (Chen et al., 2024) synthesizes low-quality videos with high-quality images through a joint training design of spatial and temporal modules, obtaining high-quality videos. T2V-turbo (Li et al., 2024) presents a distilled video consistency model (Wang et al., 2023c; Song et al., 2023) for improved and rapid video generation. However, even the recent T2V diffusion models suffer from misalignment problems. In the following, we discuss the research direction of refining the image/video diffusion models, including VIDEOREPAIR.

Training-free refinement for diffusion models. Recent works propose training-free refinement frameworks that automatically improve diffusion models' text alignment (Mañas et al., 2024; Wu et al., 2024; Chen et al., 2025; Ji et al., 2025; Xiang et al., 2025). In particular, OPT2I (Mañas et al., 2024) presents iterative prompt optimization, where an LLM provides various variations of text prompts, T2I diffusion models generate images from the prompts, and the images are ranked with a T2I alignment score such as DSG (Cho et al., 2024) to provide the final image. Since no explicit feedback is given to the backbone generation model, it usually takes long iterations (*e.g.*, 30 LLM calls) to find a prompt that provides improved alignment, making the framework expensive to use in practice. SLD (Wu et al., 2024) provides more explicit guidance by generating a bounding-box level plan with an LLM, followed by operations such as object addition, deletion, and repositioning. However, SLD depends on an external layout-guided object generator (*e.g.*, GLIGEN (Li et al., 2023b)) to insert objects, and the added content often fails to harmonize with the original image. In contrast, VIDEOREPAIR is the first training-free refinement framework that delivers fine-grained localized feedback and is compatible with any T2V diffusion model, without relying on additional generators.

5 Conclusion

We propose VIDEOREPAIR, a training-free, model-agnostic video refinement framework that improves text-to-video alignment, including three stages: misalignment detection, refinement planning with key-object preservation, and localized refinement. VIDEOREPAIR consistently outperforms recent baselines on two benchmarks, supported by extensive ablations and qualitative results.

6 Limitation

We note that the limitations of VIDEOREPAIR primarily stem from the capability of the underlying evaluator (MLLM), rather than the refinement framework itself. First, our method is less effective for ambiguous or subjective prompts. Since VIDEOREPAIR relies on explicit, semantically grounded feedback signals (e.g., discrepancies in object count, color, or action), cases without well-defined and objectively verifiable criteria make it inherently difficult to generate reliable corrective feedback. Second, VIDEOREPAIR is limited in handling prompts that require precise text rendering or fine-grained facial attributes (e.g., specific celebrity identity). This is because current MLLM-based evaluators are not sufficiently reliable for structured assessment of exact text fidelity (such as spelling accuracy) or subtle identity consistency in faces.

Acknowledgments

We thank the anonymous reviewers for their valuable feedback and constructive suggestions. This work was supported by DARPA ECOLE Program No. HR00112390060, NSF-AI Engage Institute DRL-2112635, DARPA Machine Commonsense

(MCS) Grant N66001-19-2-4031, ARO Award W911NF2110220, ONR Grant N00014-23-1-2356, Accelerate Foundation Models Research program, and a Bloomberg Data Science PhD Fellowship. The views contained in this article are those of the authors and not of the funding agency.

References

Yuanhao Ban, Ruochen Wang, Tianyi Zhou, Boqing Gong, Cho-Jui Hsieh, and Minhao Cheng. 2024. The crystal ball hypothesis in diffusion models: Anticipating object positions from initial noise. *arXiv preprint arXiv:2406.01970*.

Yuxuan Bian, Zhaoyang Zhang, Xuan Ju, Mingdeng Cao, Liangbin Xie, Ying Shan, and Qiang Xu. 2025. Videopainter: Any-length video inpainting and editing with plug-and-play context control. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–12.

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

Chieh-Yun Chen, Min Shi, Gong Zhang, and Humphrey Shi. 2025. T2i-copilot: A training-free multi-agent text-to-image system for enhanced prompt interpretation and interactive generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19396–19405.

Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. 2024. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. *arXiv preprint arXiv:2401.09047*.

Jaemin Cho, Yushi Hu, Roopal Garg, Peter Anderson, Ranjay Krishna, Jason Baldridge, Mohit Bansal, Jordi Pont-Tuset, and Su Wang. 2024. Davidsonian scene graph: Improving reliability in fine-grained evaluation for text-image generation. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, and 1 others. 2024. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146*.

Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. 2023. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. 2022. Video diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A Smith. 2023. Tifa: Accurate and interpretable text-to-image faithfulness evaluation with question answering. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Yatai Ji, Jiacheng Zhang, Jie Wu, Shilong Zhang, Shoufa Chen, Chongjian Ge, Peize Sun, Weifeng Chen, Wenqi Shao, Xuefeng Xiao, and 1 others. 2025. Prompt-a-video: Prompt your video diffusion model via preference-aligned llm. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18725–18735.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. 2025. Vace: All-in-one video creation and editing. *Preprint*, arXiv:2503.07598.

Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. 2023. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Jianwei Yang, Chunyuan Li, Lei Zhang, and Jianfeng Gao. 2023a. Semantic-sam: Segment and recognize anything at any granularity. *arXiv preprint arXiv:2307.04767*.

Jiachen Li, Weixi Feng, Tsu-Jui Fu, Xinyi Wang, Sugato Basu, Wenhu Chen, and William Yang Wang. 2024. T2v-turbo: Breaking the quality bottleneck of video consistency model with mixed reward feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. 2023b. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Long Lian, Boyi Li, Adam Yala, and Trevor Darrell. 2023. Llm-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. *arXiv preprint arXiv:2305.13655*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, and 1 others. 2023.

Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*.

Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, and Ying Shan. 2024b. Evalcrafter: Benchmarking and evaluating large video generation models. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. 2022. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11461–11471.

Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan. 2023. Videofusion: Decomposed diffusion models for high-quality video generation. *arXiv preprint arXiv:2303.08320*.

Oscar Mañas, Pietro Astolfi, Melissa Hall, Candace Ross, Jack Urbanek, Adina Williams, Aishwarya Agrawal, Adriana Romero-Soriano, and Michal Drozdzal. 2024. Improving text-to-image consistency via automatic prompt optimization. *arXiv preprint arXiv:2403.17804*.

OpenAI. 2024. GPT-4 technical report. *Preprint*, arXiv:2303.08774.

Zipeng Qi, Lichen Bai, Haoyi Xiong, and 1 others. 2024. Not all noises are created equally: Diffusion noise selection and optimization. *arXiv preprint arXiv:2407.14041*.

Leigang Qu, Ziyang Wang, Na Zheng, Wenjie Wang, Liqiang Nie, and Tat-Seng Chua. 2025. Ttom: Test-time optimization and memorization for compositional video generation. *arXiv preprint arXiv:2510.07940*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. *Advances in neural information processing systems*, 29.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, and 1 others. 2022. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. 2023. Consistency models. *arXiv preprint arXiv:2303.01469*.

Kaiyue Sun, Kaiyi Huang, Xian Liu, Yue Wu, Zihan Xu, Zhenguo Li, and Xihui Liu. 2024a. T2v-compbench: A comprehensive benchmark for compositional text-to-video generation. *arXiv preprint arXiv:2407.14505*.

Wenqiang Sun, Teng Li, Zehong Lin, and Jun Zhang. 2024b. Spatial-aware latent initialization for controllable image generation. *arXiv preprint arXiv:2401.16157*.

Zachary Teed and Jia Deng. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer.

Ye Tian, Ling Yang, Haotian Yang, Yuan Gao, Yufan Deng, Jingmin Chen, Xintao Wang, Zhaochen Yu, Xin Tao, Pengfei Wan, and 1 others. 2024. Videotetris: Towards compositional text-to-video generation. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, and 1 others. 2025. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*.

Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. 2023a. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*.

Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. 2023b. Videomae v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14549–14560.

Xiang Wang, Shiwei Zhang, Han Zhang, Yu Liu, Yingya Zhang, Changxin Gao, and Nong Sang. 2023c. Videolcm: Video latent consistency model. *Preprint*, arXiv:2312.09109.

Haoning Wu, Erli Zhang, Liang Liao, Chaofeng Chen, Jingwen Hou, Annan Wang, Wenxiu Sun, Qiong Yan, and Weisi Lin. 2023a. Exploring video quality assessment on user generated contents from aesthetic and technical perspectives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20144–20154.

Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. 2023b. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Tsung-Han Wu, Long Lian, Joseph E Gonzalez, Boyi Li, and Trevor Darrell. 2024. Self-correcting llm-controlled diffusion models. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Dawei Xiang, Wenyan Xu, Kexin Chu, Tianqi Ding, Zixu Shen, Yiming Zeng, Jianchang Su, and Wei Zhang. 2025. Promptsculptor: Multi-agent based text-to-image prompt optimization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 774–786.

Xiangpeng Yang, Linchao Zhu, Hehe Fan, and Yi Yang. 2025. Videograin: Modulating space-time attention for multi-grained video editing. In *The Thirteenth International Conference on Learning Representations*.

Xingyi Yang and Xinchao Wang. 2024. Compositional video generation as flow equalization. *arXiv preprint arXiv:2407.06182*.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, and 1 others. 2024. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*.

Appendix

A VIDEOREPAIR Implementation Details **13**
- A.1 Evaluation Question Generation . 13
- A.2 Video Object Evaluation 13
- A.3 Key Object Extraction 13
- A.4 Refinement Prompt Generation . . 13

B Additional Baseline Details **13**

C Additional Evaluation Details **14**

D Additional Quantitative Analysis **15**
- D.1 Human Evaluation Results 15
- D.2 Cost and Efficiency Analysis . . . 15
- D.3 Scaling Behavior 15
- D.4 Global Refinement Results 16
- D.5 Full Results on EvalCrafter 16

E Additional Qualitative Results **17**
- E.1 Iterative Refinement 17
- E.2 Object Selection 17
- E.3 Moving Key Objects. 18
- E.4 Step-by-Step Illustration 18
- E.5 Comparison with Baselines 18

F Future Work **18**

A VIDEOREPAIR Implementation Details

A.1 Evaluation Question Generation

Given the initial prompt, we construct a semantic tuple $\mathcal{T}$, similar to DSG (Cho et al., 2024) that contains *entities (subjects)*, *attributes*, and *relationships*. Here, attributes are expressed in 2-tuples (subjects, its attribute, (*e.g.*, `{bed, blue}`), and relationships are in 3-tuples (subject entity, object entity, and their relationship, (*e.g.*, `{people, pizza, make}`). Based on $\mathcal{T}$, which covers all scene-relevant information, we generate questions Q using *GPT-4-0125* (OpenAI, 2024). Note that although DSG does not account for object counts by design, we can incorporate assessments for whether the generated videos contain the correct number of target objects, thereby guiding automatic refinement with greater accuracy. For example, given a prompt 'there is a bear', DSG only generates an evaluation question "is there a bear?", which only checks the bear's existence, but does not penalize when more than one bear is generated.

A.2 Video Object Evaluation

To evaluate the generated videos, we utilize GPT-4o to answer both count-related (Q_c^o) and attribute-related (Q_a^o) questions, as illustrated in Fig. 24. For Q_c^o prompts, we guide GPT-4o through four steps: reasoning, answering, counting the predicted number of objects (n_p^o), and verifying the true count (n_v^o). These steps yield an answer triplet $A_c^o = \{b_c^o, n_p^o, n_v^o\}$. To ensure valid responses, we account for dependencies among questions, following the methodology of DSG (Cho et al., 2024). Each question is sequentially presented to GPT-4o, and the video score is computed as the proportion of correctly answered binary questions across all VQA tasks. If the video score reaches 1.0 (indicating a perfect score), the VIDEOREPAIR process is terminated.

A.3 Key Object Extraction

To extract the key concept O^* from the initial videos V_0, we sampled frames of V_0 and the list of question-answer pairs for each object to GPT4o as shown in Fig. 26. Here, we prioritize selecting objects with a higher number of 1.0 video scores. Moreover, we force GPT4o to select 'object' instead of 'background' elements to improve the accuracy of region decomposition by pointing.

A.4 Refinement Prompt Generation

To produce a refinement prompt p^r, we use GPT4 with instruction as shown in Fig. 25. After getting O^*, we can decompose the whole question set Q as Q^{o*} and others depending on whether the O^* keyword is included in the question. To generate p^r from specific question sets, we utilize five manually crafted in-context examples to ensure the accuracy of the generation process. If the video score is 0.0 (indicating a complete failure from VQA) and the key object O^* cannot be identified, we consider the T2V model to have failed in generating any object correctly. In such cases, we paraphrase Q directly into p^r using a large language model (LLM).

B Additional Baseline Details

LLM Paraphrasing. Following (Mañas et al., 2024), we compare VIDEOREPAIR with paraphrasing prompts from LLM. Here, we ask GPT4 to generate diverse paraphrases of each prompt, without any context about the consistency of the images generated from it. The prompt used to obtain paraphrases is provided in Fig. 27.

OPT2I. Since OPT2I (Mañas et al., 2024) aims to improve text-image consistency for T2I models, we reimplement OPT2I for T2V. Specifically, we replace the original T2I model part with T2V models (T2V-Turbo and VideoCrafter2) to generate outputs. Using GPT-4o, we then pose DSG questions to these outputs. For prompts, we directly adopt the ones provided in the original OPT2I paper. For LLM, we use GPT4 as VIDEOREPAIR. Finally, we perform iterative refinement, running 10 iterations for T2V-Turbo and 5 iterations for VideoCrafter2, with five video candidates per iteration.

SLD. To adapt SLD (Wu et al., 2024) to the T2V setup, we apply their official code to individual video frames and maintain their default setup. Note that SLD is a GLIGEN (Li et al., 2023b)-based T2I model, which poses challenges for direct extension to video generation. Since SLD operates using DDIM inversion, we use the initial videos generated by T2V-Turbo and VideoCrafter2 as inputs, enabling the implementation of their noise composition method. Here, we use one iteration for SLD and GPT4 for LLM.

C Additional Evaluation Details

EvalCrafter. To evaluate the effectiveness of VIDEOREPAIR across different prompt dimensions, we decompose EvalCrafter (Liu et al., 2024b) using the official `metadata.json`. Specifically, we utilize the `attributes` key for each prompt and categorize the dataset into 'count', 'color', 'action', 'text', 'face', and 'amp (camera motion)'. Prompts without explicit attributes are grouped into an 'others' category. Among these dimensions, we focus on 'count', 'color', 'action', and 'others', excluding 'text', 'face', and 'amp'. This decision is based on our observation that video errors related to text prompts (*e.g.*, *"the words 'KEEP OFF THE GRASS"*), face prompts (*e.g.*, *"Kanye West eating spaghetti"*), and amp prompts (*e.g.*, *"A Vietnam map, large motion"*) cannot be reliably detected through GPT-4o question-answering, therefore hard to proceed VIDEOREPAIR.

For evaluation metrics, we mainly adopt the average text-video alignment score they proposed. Among their all text-video alignment scores (CLIP-Score, SD-Score, BLIP-BLEU, Detection-Score, Count-Score, Color-Score, Celebrity ID Score, and OCR-Score) we exclude Celebrity ID Score and OCR-Score since they are related to 'face' and 'text' categories. Therefore, we calculate the text-video alignment score as Avg(CLIP-Score, SD-Score, BLIP-BLEU, DetectionScore, CountScore, ColorScore). For overall video quality, we directly adopt their metrics including Inception Score (Salimans et al., 2016) and Video Quality Assessment (VQA_A, VQA_T) (Wu et al., 2023a). For the motion quality score, we calculate the weighted average score of the Action Recognition score (from VideoMAE (Wang et al., 2023b)) and Average Flow score (Teed and Deng, 2020) from the official EvalCrafter code. For the temporal consistency score, we also calculate the weighted average score of Warping Error from optical flow (Wang et al., 2023b) and CLIP-Temp (Radford et al., 2021). For the *others* section of CogVideoX-5B, we report results on only 100 randomly sampled videos, as other baselines (e.g., OPT2I) require a significantly long refinement time (around 5h per one video refinement).

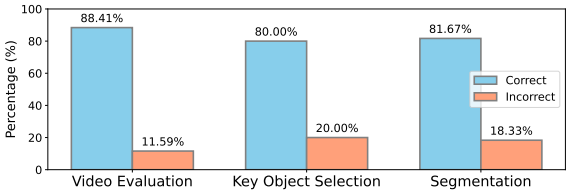

Figure 6: **Human error analysis results.** VIDEOREPAIR consistently achieves approximately 80% correctness across all components of the framework.

T2V-Compbench. Since VIDEOREPAIR has strength in compositional generation, we adopt T2V-Compbench (Sun et al., 2024a) and evaluate three dimensions: spatial relationships, generative numeracy, and consistent attribute binding. 'Spatial relationships' requires the model to generate at least two objects while maintaining accurate spatial relationships (*e.g.* 'to the left of', 'to the right of', 'above', 'below', 'in front of') throughout the dynamic video. 'Generative numeracy' specifies one or two object types, with quantities ranging from one to eight. 'Consistent attribute binding' contains color, shape, and texture attributes among two objects. Following (Sun et al., 2024a), we adopt Video LLM-based metrics for consistent attribute binding and detection-based metrics for spatial Relationships and numeracy.

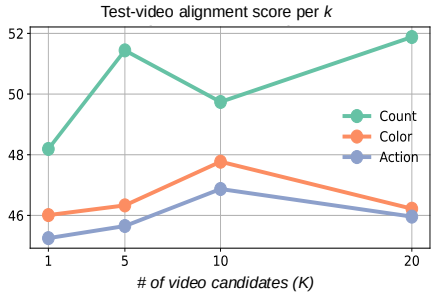

Figure 7: **Impact of the number of video candidates.** We vary the number of video candidates K as 1, 5, 10, and 20 for ranking.

D Additional Quantitative Analysis

D.1 Human Evaluation Results

To analyze errors, we enlist three AI experts to assess the correctness of each component in VIDEOREPAIR, including video evaluation, key object selection, and segmentation. For video evaluation, annotators are provided with the initial text prompt and the corresponding video generated by T2V-Turbo, presented as a sequence of frames, along with MLLM-generated question–answer pairs. They are asked to judge whether the answers are correct. In particular, a generation is marked as *incorrect* if the number of generated objects does not exactly match the count specified in the prompt. We include screenshots of the evaluation questionnaire and labeling instructions in Figs. 14 and 15. For key object selection, we present the objects selected by GPT-4o and ask annotators to verify whether the identified objects O^* and N^* are correct. For segmentation evaluation, annotators are given a pointing prompt (e.g., "Point to the largest umbrella and one picnic blanket") together with the corresponding segmentation map, and are asked to assess whether the segmentation is well-aligned with the prompt.

Overall, human evaluation shows that VIDEOREPAIR achieves approximately 80% correctness across all components (Fig. 6). While errors remain at each stage, we attribute these limitations primarily to the backbone model and expect improvements with more advanced architectures. The inter-annotator agreement is 93.18%, indicating strong consistency among annotators.

D.2 Cost and Efficiency Analysis

End-to-end inference latency. To provide a complete end-to-end cost analysis, we report full wall-clock latency, per-stage breakdown, and compute-matched baselines (seed generation) in Tab. 8. All experiments are conducted on two NVIDIA RTX 6000 40GB GPUs, and running results are averaged over the EvalCrafter subsets (count, color, and action). Compute-matched sampling baselines (e.g., 5-seed generation with ranking) improve only marginally (44.68), indicating that our gains are not attributable to increased sampling alone.

Runtime and model call. Tab. 9 provides a cost comparison with refinement baselines, including the # of LLM/MLLM calls, external model calls (e.g., segmentation and detection), and stage-wise runtime (s). For fair comparison, we match the candidate count across methods. The results show that while OPT2I and SLD incur substantial detection/planning and refinement overhead (288–545s total latency), VIDEOREPAIR achieves higher T2V alignment (47.77) with significantly lower latency (52.51s), yielding a **5–10× speedup** over these refinement-based baselines.

D.3 Scaling Behavior

To further characterize scaling behavior, we analyze the impact of (i) number of candidates (K), (ii) video length (16 v.s. 81 frames), and (iii) video resolution (480x832 v.s. 720x1280). All experiments are conducted using T2V-turbo and Wan2.1, evaluated on two NVIDIA H100 80GB GPUs.

Increasing # of Video Candidates. To evaluate the impact of video ranking, we vary the number of video candidates as $K = 1, 5, 10,$ and 20 during the ranking process. The variation among video candidates arises from different random seeds used to initialize ϵ_0'. For example, video ranking is not applied when $K = 1$, and only one refinement is produced using a single random seed noise ϵ_0'. For ranking metrics, we rely on the video score across all ablation studies. As depicted in Fig. 7, higher K values (5, 10, and 20) consistently yield higher scores across all categories than $K = 1$. This trend is particularly prominent in the 'count' category, where increasing K leads to noticeable performance improvements, highlighting the importance of considering multiple candidates for ranking.

Increasing Video Length and Resolution. As shown in Tabs. 10 and 11, increasing video length and resolution leads to longer refinement time. This is expected because localized refinement performs mask-based noise re-initialization and runs separate diffusion paths for preserved and refined regions, followed by fusion and candidate ranking, causing the cost to scale with video length, spatial

Table 8: **End-to-end inference latency (s).** We report end-to-end stage-wise runtime and total latency.

Model	Step 1 (Video Eval)	Step 2 (Planning)	Step 3 (Refinement)	Step 3 (Ranking)	Total (s)	T2V Alignment
T2V-turbo	-	-	-	-	3.55	43.54
T2V-turbo (5 seed + random select)	-	-	-	-	17.75	43.78
T2V-turbo (5 seed + ranking select)	-	-	17.75	24.2	41.95	44.68
VIDEOREPAIR (k=1)	5.33	22.1	5.35	-	32.78	46.53
VIDEOREPAIR (k=5) – w/ BLIP ranking	5.21	22.5	24.8	-	52.51	47.77
VIDEOREPAIR (k=5) – w/ DSG ranking	5.21	22.5	24.8	23.5	76.01	**47.91**

Table 9: **Cost comparison with other baselines.** We report the number of LLM/MLLM calls, external model calls (e.g., detection and segmentation), and stage-wise runtime.

Model	# LLM/MLLM Calls	# External Model Calls	Detection+Planning	Refinement+Ranking	Total Latency	T2V Alignment
T2V-turbo	-	-	-	-	3.55	43.54
LLM Paraphrasing (k=5)	5	-	1.3	17.7	19.0	44.6
OPT2I (k=5)	61	-	75.3	213	288.3	45.6
SLD	17	16	166.7	378.5	545.2	44.5
VIDEOREPAIR (k=5, BLIP ranking)	6	4	27.7	24.8	52.5	**47.7**

Table 10: **Scaling with respect to video length.** Increasing the number of frames incurs a higher computational cost due to longer diffusion trajectories and expanded spatio-temporal processing.

Model	Step1	Step2	Step3	Ranking	Total (s)
Wan2.1 (16f)	-	-	-	-	19.9
Wan2.1 (16f) + Ours (k=1)	6.21	22.3	37.2	-	65.7
Wan2.1 (16f) + Ours (k=5)	6.21	22.3	185	21.3	234.8
Wan2.1 (81f)	-	-	-	-	143.1
Wan2.1 (81f) + Ours (k=1)	6.3	50.2	279.6	-	336.1
Wan2.1 (81f) + Ours (k=5)	6.3	50.3	1300	20.5	1377.1

Table 11: **Scaling with respect to resolution.** Higher spatial resolution increases computational cost, as localized diffusion operates over larger spatial regions.

Model	Step1	Step2	Step3	Ranking	Total (s)
Wan2.1 (480×832)	-	-	-	-	143.1
Wan2.1 (480×832) + Ours (k=1)	6.3	50.2	279.6	-	336.1
Wan2.1 (480×832) + Ours (k=5)	6.3	50.3	1300	20.5	1377.1
Wan2.1 (720×1280)	-	-	-	-	596.9
Wan2.1 (720×1280) + Ours (k=1)	6.1	51.4	1148.2	-	1205.7
Wan2.1 (720×1280) + Ours (k=5)	6.2	51.6	5743.1	21.5	5822.4

refinement area, and the number of seeds. Importantly, this cost profile reflects an implementation characteristic rather than a fundamental limitation. The overhead can be mitigated through temporal sparsification (refining only key frames), spatial compression (operating on downsampled or region-focused representations), and acceleration strategies such as lightweight refinement backbones or feature caching.

D.4 Global Refinement Results

Tab. 12 reports results on the *Other* section of EvalCrafter, which includes camera movement, landscape, and style prompts. Applying VIDEOREPAIR yields consistent improvements across all three categories, with gains of +1.21 in camera movement, +1.77 in landscape, and +0.93 in style. These results highlight that VIDEOREPAIR not only enhances core compositional attributes (e.g., count, color, action) but also extends effectively to broader aspects of video quality such as dynamics, scenery, and artistic style.

Table 12: **VIDEOREPAIR performance on the EvalCrafter *'Other'* section.** VIDEOREPAIR consistently improves video quality in camera movement, landscape, and style categories over the initial generations.

	Camera movement	Landscape	Style
Initial video (T2V-turbo)	44.02	48.94	42.70
+ VIDEOREPAIR	**45.23**	**50.71**	**43.63**

D.5 Full Results on EvalCrafter

To enable direct comparison with prior work that reports the official average across all splits, we evaluated VIDEOREPAIR on the full EvalCrafter (Liu et al., 2024b) benchmark with T2V-turbo, including previously excluded categories (text, camera motion, and face). As shown in Tab. 13, VIDEOREPAIR consistently improves T2V alignment on the full benchmark while maintaining comparable visual quality, motion quality, and temporal consistency. The gains in motion quality and temporal consistency are intentionally modest, as VIDEOREPAIR is designed to correct semantic misalignment while preserving the original video dynamics rather than re-synthesizing motion. Overall, the results confirm that the improvements are not limited to selected splits but generalize to the complete

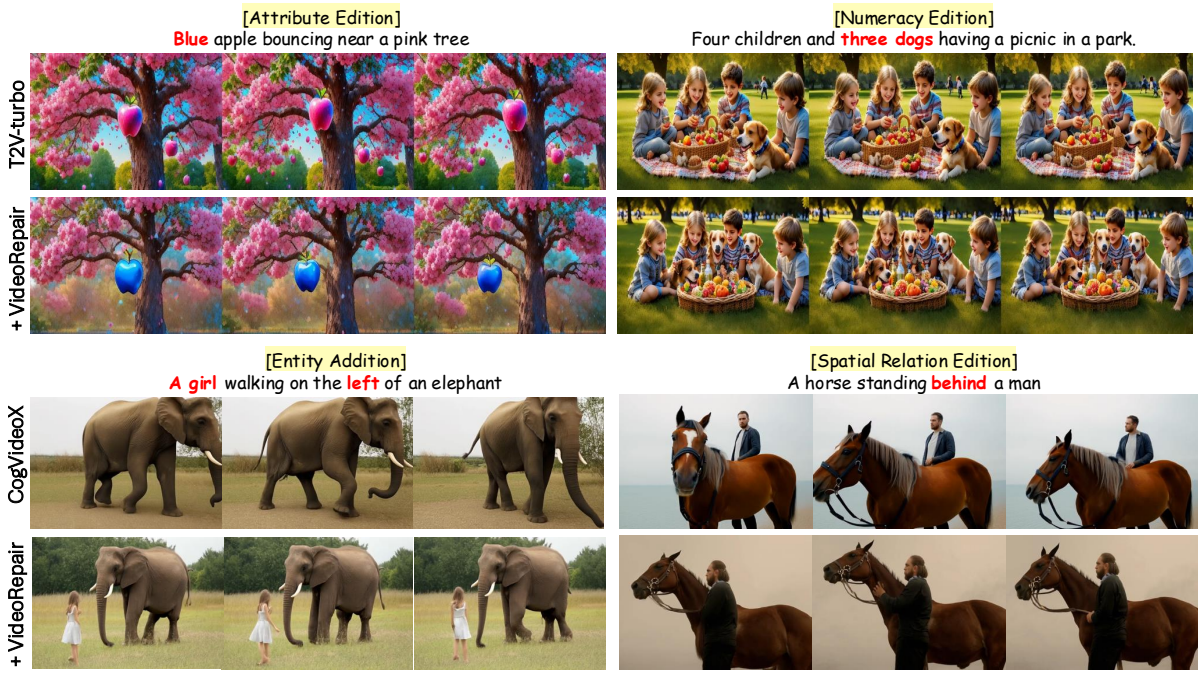

Figure 8: **Scalability of VIDEOREPAIR.** We illustrate the scalability of VIDEOREPAIR including attribute, numeracy, spatial relation edition, and entity addition from T2V-turbo and CogVideoX backbone.

Table 13: **Evaluation on full EvalCrafter benchmark.** For direct comparison with prior work, we additionally report the official average across all splits.

Model	Visual Quality	T2V Alignment	Motion Quality	Temporal Consistency	Final Score
Zeroscope	53.41	51.21	53.61	58.91	217.14
MoonValley	69.53	50.66	55.46	65.25	240.90
Show-1	52.19	62.07	53.74	60.83	228.83
T2V-turbo	61.23	59.87	55.94	62.78	239.82
+ VIDEOREPAIR	**61.26**	**62.30**	**55.90**	**62.88**	**242.34**

EvalCrafter evaluation protocol while maintaining stable video quality metrics.

E Additional Qualitative Results

E.1 Iterative Refinement

We observe the effect of applying VIDEOREPAIR iteratively to further improve text–video alignment. At each step, we monitor the video score and terminate the refinement process once it reaches the maximum value of 1.0. We use video ranking with $K = 5$. As shown in Fig. 9, iterative refinement consistently improves performance across all three splits (count, color, and action) in EvalCrafter.

Qualitative examples in Fig. 23 further illustrate this trend, where we compare the initial video with the first and second refinement results generated by T2V-Turbo. Overall, VIDEOREPAIR progressively enhances text–video alignment over successive refinement steps. In numeracy-related cases (*e.g.*, six dancers, five cows), VIDEOREPAIR iteratively adjusts object counts by adding or removing instances to match the prompt. When objects are missing (*e.g.*, biologists, ducks), the model successfully introduces additional instances while preserving the original scene context. For attribute-related prompts (*e.g.*, yellow umbrella, blue cup), VIDEOREPAIR refines object attributes, such as introducing a wooden handle to the umbrella and strengthening the blue color of the cup. These results demonstrate that VIDEOREPAIR effectively improves both object count and attribute alignment in a progressive and high-fidelity manner.

E.2 Object Selection

In refinement planning, we select the largest candidate among the correct objects. This approach can be seamlessly extended to select multiple correct objects when the number of objects in the initial video ($n_v^{o^*}$) meets or exceeds the prompt's specification ($n_p^{o^*}$). We implement this extension to enable the formulation of **object-wise pointing prompts** and the generation of multiple masks to preserve these objects. As shown in Fig. 10, this version can preserve *a bear* and *a man* while automatically refining the video to add *an additional person*.

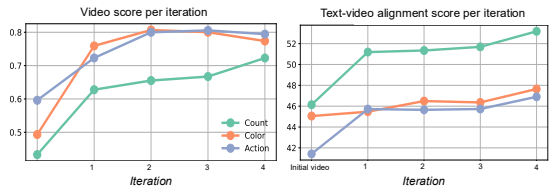

Figure 9: **Impact of iterative refinement.** Iterative refinement gradually improves the video score and text-video alignment score on all three prompt categories (count/color/action) of EvalCrafter.

Figure 10: **Single-object mask vs. Multi-object mask.**

E.3 Moving Key Objects.

In long videos (e.g., CogVideoX-generated videos with 81 frames), key objects may disappear or newly appear across different frames. As shown in Fig. 11, VIDEOREPAIR effectively captures moving key objects O^* using frame-wise masks M. This example illustrates how frame-wise masks help handle changes in object count and attributes - preserving disappearing objects (*car*) while incorporating previously missed objects (*house*).

E.4 Step-by-Step Illustration

In Figs. 12 and 13, we provide detailed illustrations of all three VIDEOREPAIR steps.

E.5 Comparison with Baselines

We present additional qualitative comparisons with baseline methods (OPT2I (Mañas et al., 2024), SLD (Wu et al., 2024), and Vico (Yang and Wang, 2024)) in Figs. 16 to 22. These examples address a variety of failure cases commonly observed in T2V models, including inaccuracies in object count and attribute depiction, as highlighted in our main paper. Figs. 16 to 19 correspond to results from T2V-Turbo, while Figs. 20 to 22 showcase examples from VideoCrafter2. Additionally, we provide binary segmentation masks that identify preserved areas (in black) and updated areas (in white).

Across these examples, VIDEOREPAIR effectively preserves the O^* areas while refining the remaining regions using p^r. For instance, in Fig. 16, the camel from the original T2V-Turbo video is preserved, and a snowman is successfully added. In contrast, while SLD also leverages DDIM inversion to preserve objects, it often fails to integrate new objects seamlessly. We also visualize the scalability of VIDEOREPAIR in Fig. 8.

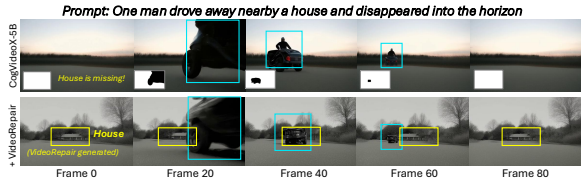

Figure 11: **Refining videos when the key object disappears.** VIDEOREPAIR successfully preserves disappearing objects (*car*) while incorporating previously missed objects (*house*).

F Future Work

To further address the identified failure modes on Tab. 6, we plan to incorporate targeted mitigation strategies tailored to each category. For instance, we can introduce confidence-based gating mechanisms to reduce QA hallucination by skipping low-confidence refinements and enforcing stricter acceptance criteria, with a fallback to global generation when the video score is unreliable. To alleviate mask drift, temporal mask smoothing can be applied to improve cross-frame consistency. Boundary artifacts can be mitigated by replacing hard binary masks with soft blending masks (e.g., Gaussian-blurred masks) for smoother transitions. Additionally, identity and motion inconsistencies can be reduced by increasing preservation weights in the joint optimization objective and adopting soft transition masks to better anchor preserved content. Importantly, as VIDEOREPAIR is a training-free and model-agnostic framework, these improvements can be seamlessly integrated without retraining, enabling flexible extensions and systematic evaluation in future work.

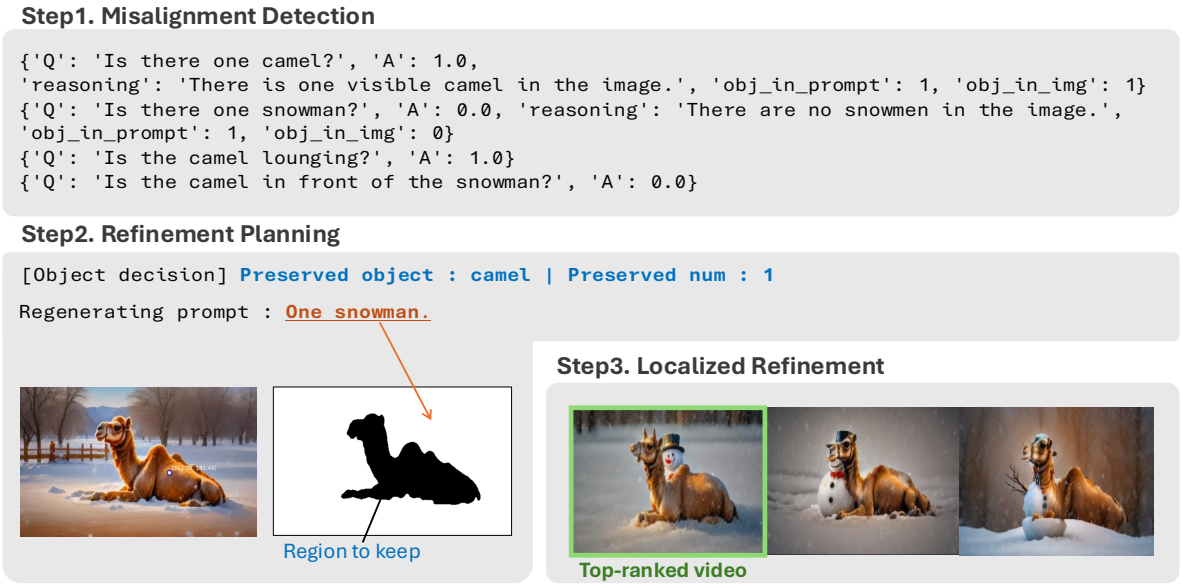

Figure 12: **Output from each step of VIDEOREPAIR.** We illustrate whole outputs from each step of VIDEOREPAIR.

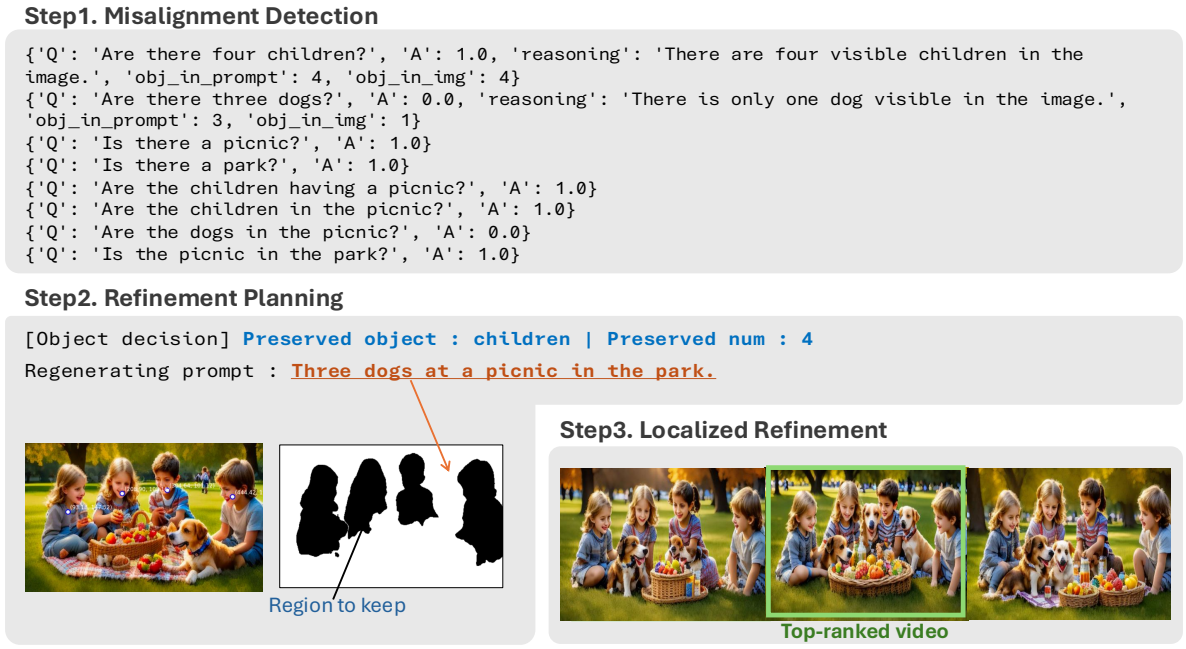

Figure 13: **Output from each step of VIDEOREPAIR.** We illustrate whole outputs from each step of VIDEOREPAIR.

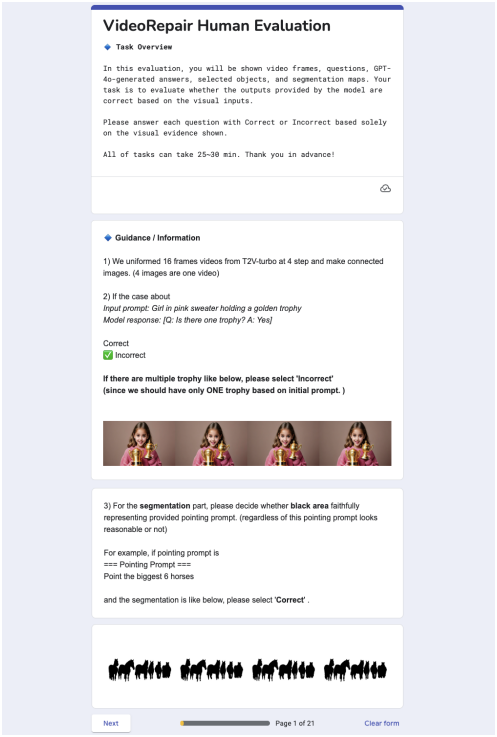

Figure 14: **A screenshot of questionnaires for error analysis.**

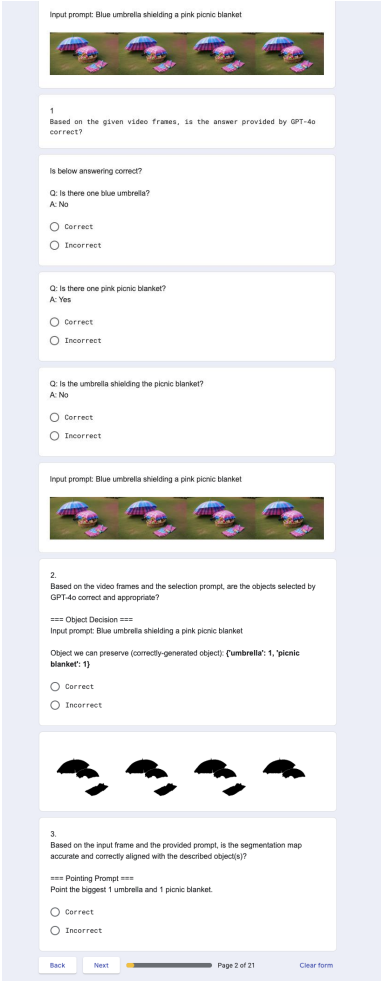

Figure 15: **A screenshot of questionnaires for error analysis.**

Figure 16: **Qualitative examples from T2V-turbo.**

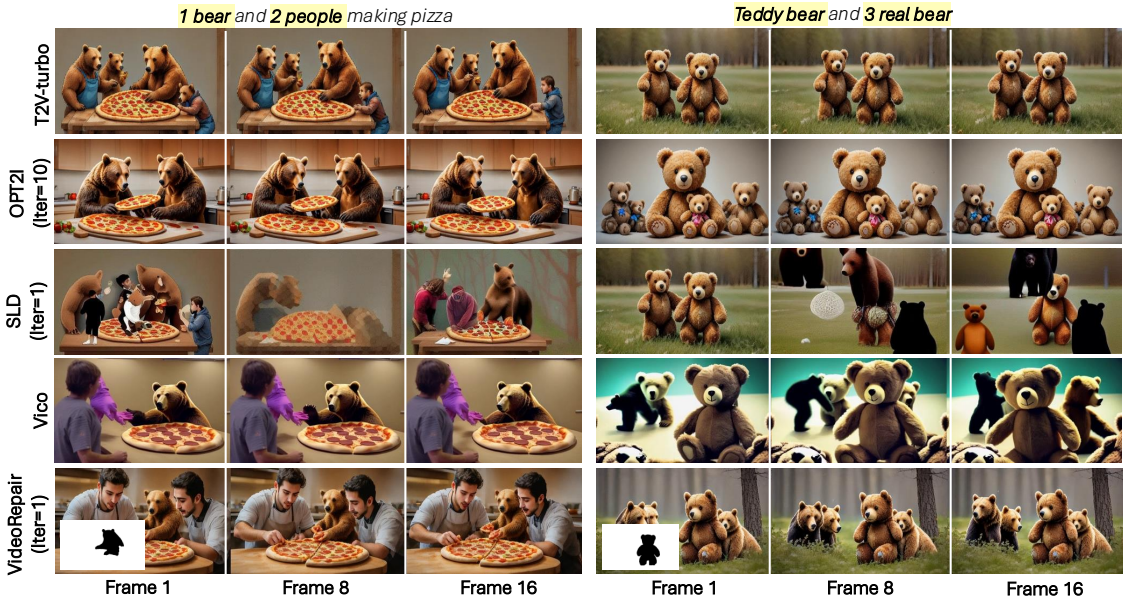

Figure 17: **Qualitative examples from T2V-turbo.**

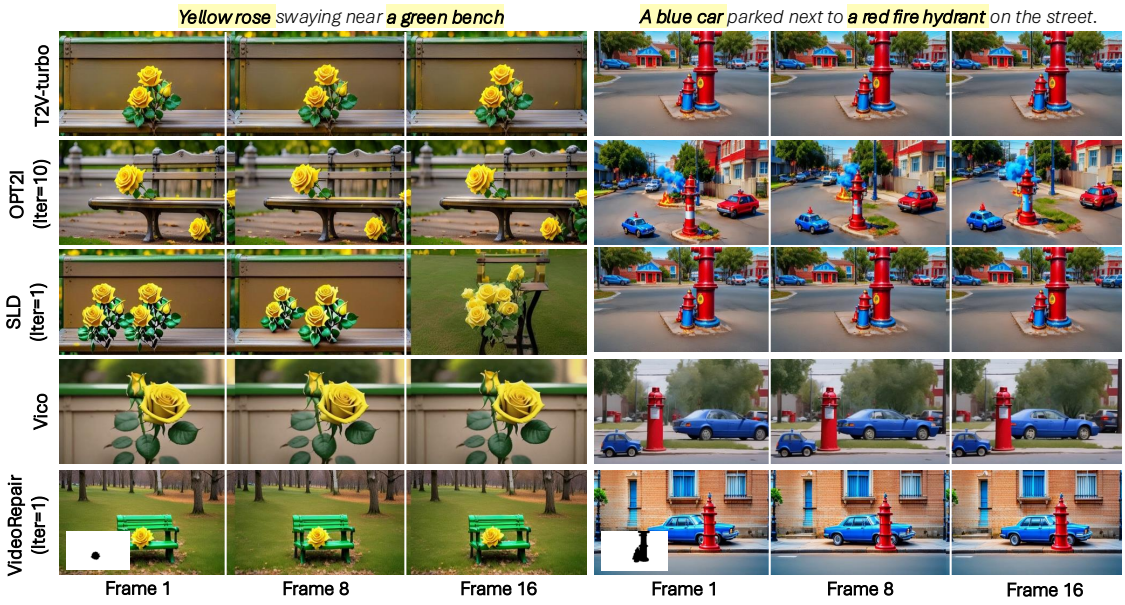

Figure 18: **Qualitative examples from T2V-turbo.**

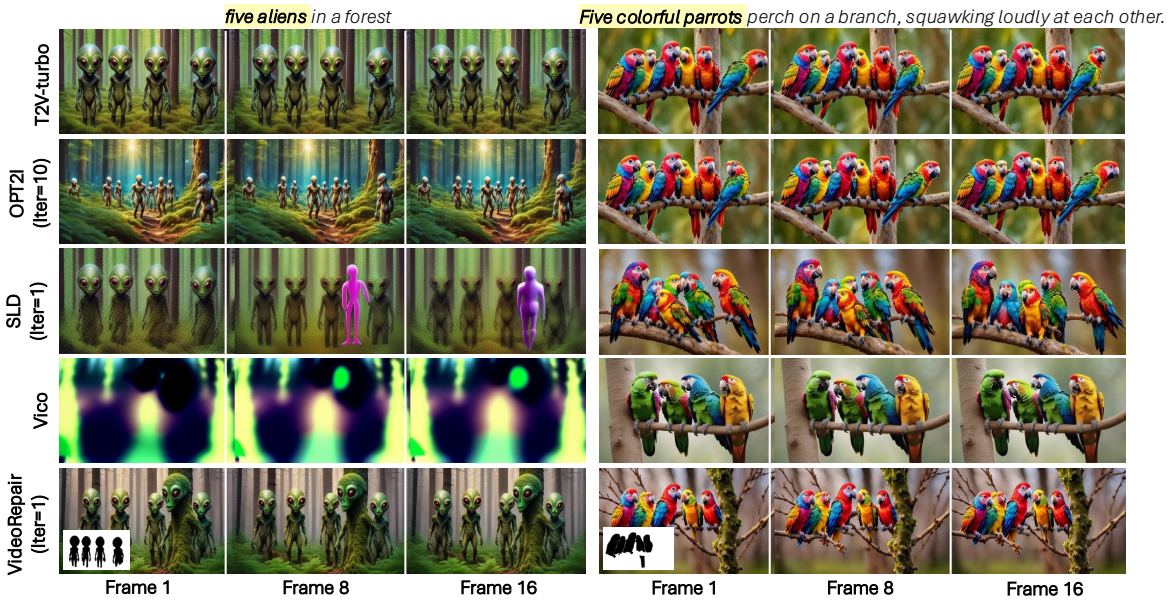

Figure 19: **Qualitative examples from T2V-turbo.**

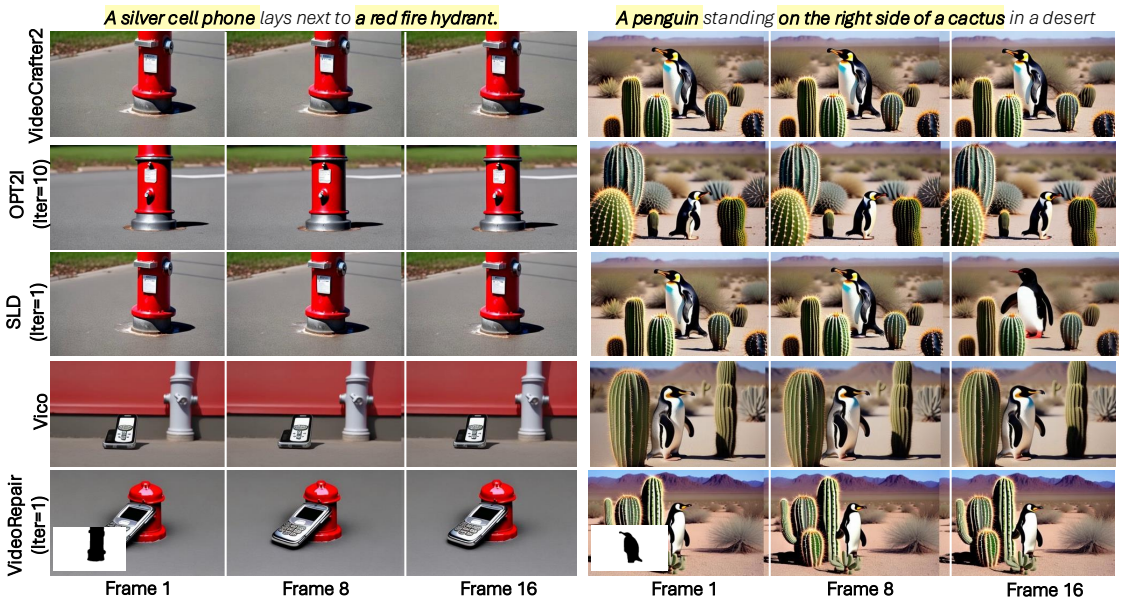

Figure 20: **Qualitative examples from VideoCrafter2.**

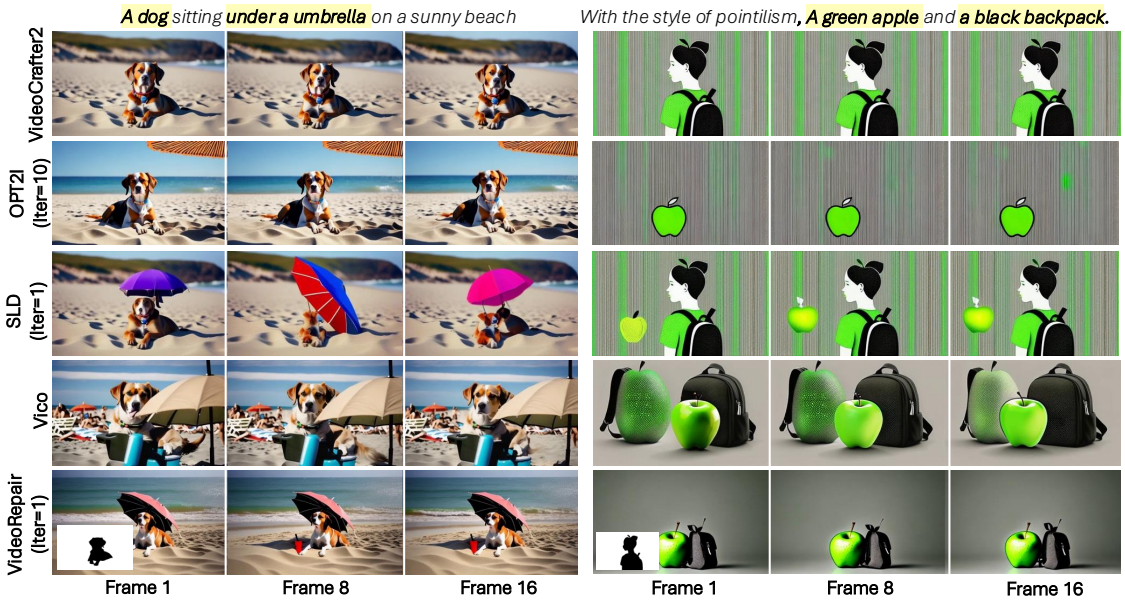

Figure 21: **Qualitative examples from VideoCrafter2.**

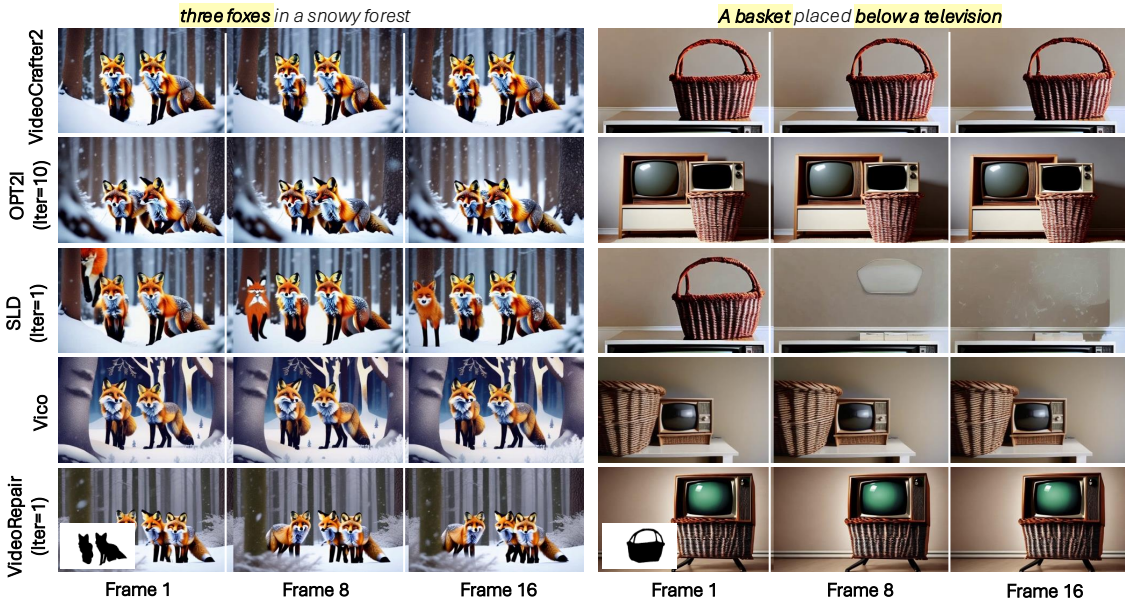

Figure 22: **Qualitative examples from VideoCrafter2.**

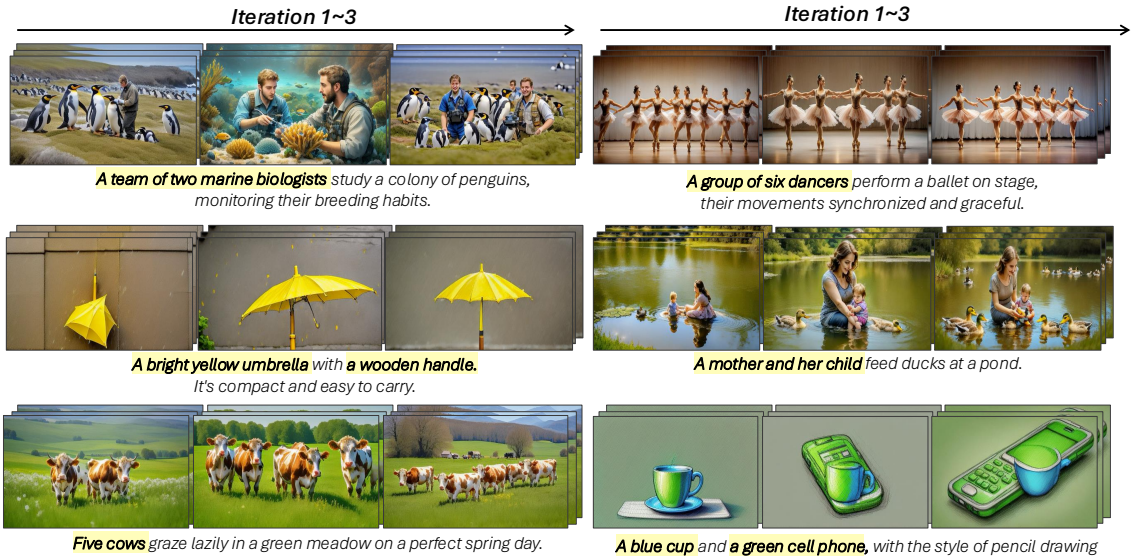

Figure 23: **Videos generated using iterative refinement with VIDEOREPAIR.** We depict iterative refinement results generated from T2V-Turbo. Overall, VIDEOREPAIR progressively enhances text-video alignment with each refinement step.

```
1. Given the question: "{cur_question}", provide a brief reasoning (up to two sentences) to determine the
accurate answer.
2. Respond to the question using binary values: 1.0 for "Yes" and 0.0 for "No". If the answer is uncertain
or unnatural due to image distortion or other issues, respond with 0.0 ("No").
3. Return the number of "{key_objects}" (as an integer) mentioned in the initial prompt "{cur_question}".
4. Return the number of "{key_objects}" (as an integer) in the provided image.

Return the result as a dictionary in the following format (not in JSON format):
{{"Q": "<question>",
"A": <binary answer>,
"reasoning": "<brief reasoning>",
"obj_in_prompt": <number of key object mentioned in the initial prompt>,
"obj_in_img": <number of key object in the image>}}

Example:
{{"Q": "Is there one robot?",
"A": 0.0,
"reasoning": "There are two visible robots in the image.",
"obj_in_prompt": 1,
"obj_in_img": 2}}

Please provide only the dictionary as the output without any additional text or explanation.
```

```
Respond to "{cur_question}" using binary values: 1.0 for Yes and 0.0 for No.
If the answer is uncertain due to image distortion or other issues, respond with 0.0 (No). \
Return the result as a dictionary in the following format (not in JSON format): \
{{"Q": "<question>", "A": <binary answer>}} \
(e.g., {{"Q": "Is there one robot?", "A": 0.0}}) \
Provide only the dictionary as the output, without any additional text or explanations.
```

Figure 24: **Prompts to perform visual question answering in video evaluation steps. Top:** The prompt for Q_c^o (count-related question), **Bottom:** prompt for Q_{others}^o (attribute-related question). `cur_question` means each video evaluation question and `key_objects` means entity word in each question.

```
Given the following list of questions {question_list}, create a single descriptive sentence that combines
the meaning of each question into a natural, affirmative statement that provides a full, concise summary.

Examples:
- Example 1
Question list: ['Is there a bed?', 'Is the bed blue?', 'Are the pillows beige?', 'Are the pillows with the
bed?']
Answer: "Blue bed with beige pillows."

- Example 2
Question list: [Are there three real bears?]
Answer: "Three real bears."

- Example 3
Question list: [Are there two people?, Are the people making pizza?]
Answer: "Two people making pizza.

- Example 4
Question list: [Is there a family?, Is there one cat?, Is there a park?, Is the family taking a walk?, Is
the cat walking?, Is the family enjoying?, Is the family breathing fresh air?, Is the family exercising?]
Answer: "A family and a cat are walking in the park."

- Example 5
Question list: [Is there a green bench?, Is there an orange tree?, Is the bench green?, Is the tree
orange?]
Answer: "Green bench and orange tree."

Your Current Task: Your response should be a concise 1 phrase, without additional explanation
(e.g., "a small bear")
```

Figure 25: **Prompt to plan how to refine the other regions.** We use five in-context examples to create the refinement prompt from the question related to other objects.

```
Given the image which compose of multiple concatenated frames from a video and the list of question-
answer pairs for each object, represented as {object_wise_dict}, choose all the accurately or visibly
generated objects from the list {objects_from_Question}. Prioritize selecting objects with a high
number of answers rated 1.0 for each question. Select the object that is both large and clearly
visible, prioritizing prominent objects (such as animals, humans, or specific items) over background
elements (like ocean or city). Return only the name of the best object to keep from the list, without
additional explanation (e.g., dog)
```

Figure 26: **Prompt to choose which object(s) to preserve.** We ask GPT4o to select objects to preserve in the scene.

```
Generate 1 paraphrase of the following image description
while keeping the semantic meaning: "{init_prompt}".
Provide your response as a single phrase without any explanation.
Format it as: <PROMPT> ... </PROMPT>.
(e.g., <PROMPT>Two dogs and a whale embark on a sea adventure.</PROMPT>)
```

Figure 27: **Prompt for LLM paraphrasing.** Following OPT2I (Mañas et al., 2024), we ask GPT4 to generate diverse paraphrases of each prompt for LLM paraphrasing baseline experiments.