CAN VIDEO GENERATION AGENTS BEAT SIMPLY TRYING AGAIN?

Anonymous authors
Paper under double-blind review

ABSTRACT

A better video may come from a better prompt, or simply another try. Video generation agents revise prompts before generation or after inspecting results, yet they spend the same resource as best-of-N (BoN) sampling: generation attempts. Existing evaluations rarely compare agents against BoN with a simple prompt under the same budget, leaving unclear whether reported gains would survive such a comparison. We therefore ask: given the same number of attempts, can an agent beat simply trying again? We evaluate nine methods on a single reference-conditioned request across up to six video generators, giving each method 20 generations per generator and counting every video an agent produces, including those used only for feedback. Published agents are compared with Simple Prompt, which repeatedly samples a minimal prompt, and with a simple counterpart in their own family. The answer is largely no. No agent's best videos improve on Simple Prompt across all dimensions; gains in one dimension come with losses in another, especially in action fulfillment. More elaborate agents fare no better than Simple Prompt even when they improve on their simple counterparts, expert-written prompts perform on par, and the few gains rarely transfer across generators. Iterative agents also do not improve round by round: their final-round videos fall well below their best ones, and picking the best round turns them into BoN with a changing prompt. These results suggest that agents and simply trying again are far closer than published evaluations imply, and that agents should be evaluated against matched-budget BoN on the target generator.

1 INTRODUCTION

When a generated video falls short, a creator can revise the prompt or simply try again. Video generation agents take the first path, emulating an experienced creator: prompt enhancement rewrites the prompt before generation (Gao et al., 2025a; Cheng et al., 2025), while iterative agents inspect the results and revise the prompt in response (Gao et al., 2025b; Song et al., 2026). The second path is best-of-N (BoN) sampling: generate several candidates and keep the best. With a verifier to select among candidates, drawing more samples has been shown to substantially improve both image and video generation (Ma et al., 2025; Liu et al., 2025a): more attempts alone buy better videos. Agents spend this same budget: an iterative agent uses an attempt in every round, and a rewritten prompt can be sampled repeatedly just like the original. Agents and BoN thus differ only in whether the prompt changes. In this paper, we therefore ask a simple question: *given the same number of attempts, can an agent beat simply trying again?* Existing evaluations rarely answer this question. Prompt enhancement is usually evaluated by the average quality of single attempts, often on the generators it was built for (Gao et al., 2025a; Cheng et al., 2025; Yang et al., 2026); this does not show whether a rewrite also improves the best of several attempts, or transfers to another generator (Figure 1). Iterative agents use several attempts by design, yet some report gains without an equal-budget baseline (Gao et al., 2025b), leaving open whether the revisions or the extra attempts made the difference. The few works that compare against BoN at a matched budget (Song et al., 2026; Kim et al., 2026; Long et al., 2026) evaluate only their own method, on one or two model families. To our knowledge, no study compares multiple agents against BoN with a simple prompt at equal budget across generators.

To answer this question, we evaluate every method on a single reference-conditioned request, with two characters, one scene, and a fixed action sequence, across up to six generators, three open-

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

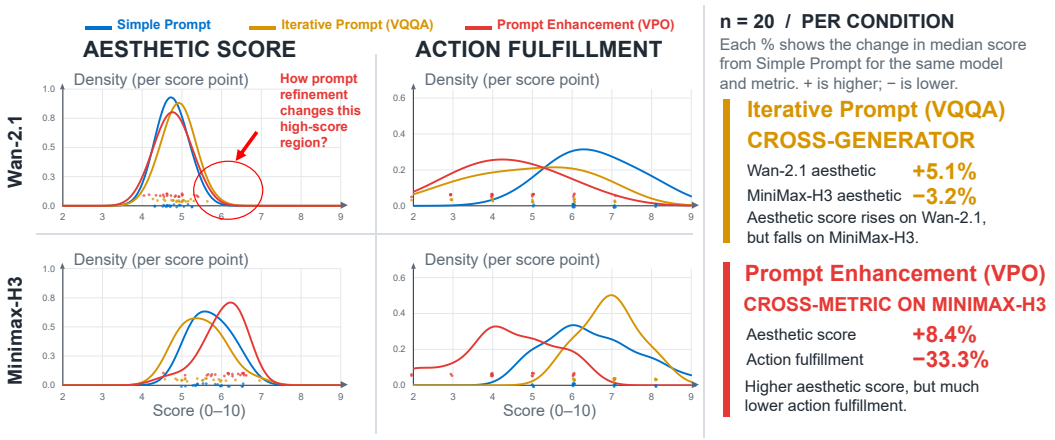

Figure 1: **Prompt rankings depend on generator and metric.** VQQA raises median aesthetics on Wan-2.1 but lowers them on MiniMax-H3; VPO improves aesthetics while weakening action fulfillment on MiniMax-H3.

source and three proprietary. Each method gets 20 generations per generator; for iterative agents, this covers every video in the loop, including those used only for feedback, while their extra LLM and VLM calls go uncounted, which favors the agents. The baseline, Simple Prompt, repeatedly samples a minimal conversion of the request. Within each agent family, a simple version that shares Simple Prompt's LLM is compared with published methods: Simple Rewrite with VPO, RAPO, and SCMAPR, and Simple Iterative with VQQA and RAPO++. Expert-written Human Prompt serves as a reference for human revision. Since a creator keeps the best result, we compare the best videos in each pool, with blinded human judgments as corroborating evidence. For iterative agents, we also compare final-round videos with the best ones to test whether refinement improves round by round.

The answer is largely no. Our findings are:

- **Agents do not beat Simple Prompt.** Under the same budget, no agent's best videos improve on Simple Prompt across all dimensions: gains in one dimension come with losses in another, especially in action fulfillment. Blinded human judgments show the same pattern.
- **More elaborate methods add little.** Published agents fare no better than Simple Prompt, even when they improve on their simple counterparts, Simple Rewrite and Simple Iterative. Expert-written Human Prompt also performs on par with Simple Prompt.
- **Gains rarely transfer.** The few gains we observe rarely carry over to other generators.
- **Iteration does not improve round by round.** Final-round videos fall well below the best ones; picking the best round helps, but only by turning the agent into BoN over a changing prompt.

Agents and Simple Prompt are thus far closer than published evaluations suggest, which calls current evaluation practice into question.

In summary, we contribute a matched-budget, cross-generator evaluation of video generation agents and the findings it yields; Section 6.1 further discusses why current agents may fall short.

2 Related Work

Video generation models. Video diffusion models have progressed from extensions of image diffusion (Ho et al., 2022b;a; Singer et al., 2022; Blattmann et al., 2023; Bar-Tal et al., 2024) to open diffusion transformers such as CogVideoX, Open-Sora 2.0, Wan2.1, and HunyuanVideo-1.5 (Yang et al., 2025; Zheng et al., 2025; Wan et al., 2025; Wu et al., 2025a). Recent systems, including MiniMax H3, Seedance 2.0, Wan 3.0, and Gemini Omni Flash (MiniMax, 2026; Team Seedance et al., 2026; Alibaba Cloud, 2026; Google DeepMind, 2026), accept multimodal references and support editing, giving creators more control. Whether prompting methods use this control well, and whether their gains hold across generators, remains open; we study six such generators.

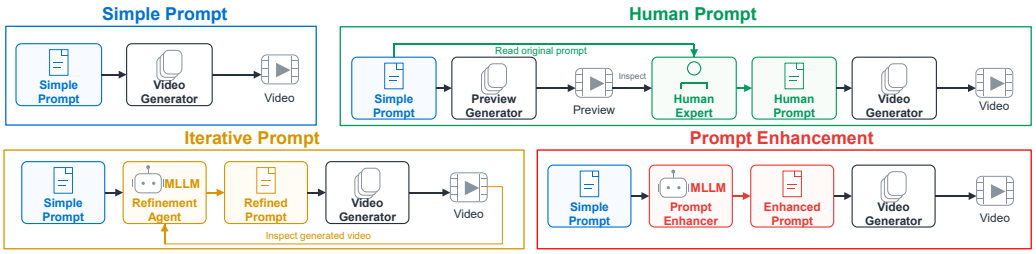

Figure 2: **The four prompting strategies.** Prompt Enhancement rewrites the prompt before generation; Human Prompt and Iterative Prompt revise it after inspecting generated videos.

Video generation agents. Agents use an LLM or MLLM to prepare or revise prompts. Prompt enhancement rewrites the prompt before generation, by expanding it into scene or shot descriptions (Lin et al., 2024; Wu et al., 2025b), retrieving modifiers (Gao et al., 2025a), applying scenario-specific policies with consistency checks (Yang et al., 2026), or training a rewriter (Ji et al., 2025; Cheng et al., 2025); the official pipelines of HunyuanVideo-1.5 and Wan2.1 also include rewriting (Wu et al., 2025a; Wan et al., 2025). Iterative agents instead follow a generate–inspect–revise loop, guided by physical knowledge (Xue et al., 2025), visual question answering (Song et al., 2026), multiple feedback signals (Gao et al., 2025b), pairwise video selection (Long et al., 2026), or failures recurring across samples (Kim et al., 2026). We evaluate representative methods from both families.

Evaluation and test-time scaling. Benchmarks such as VBench, EvalCrafter, and T2V-CompBench decompose generation quality into dimensions (Huang et al., 2024; Liu et al., 2024; Sun et al., 2025); Video-Bench, UVE-Bench, and VF-Eval study how well MLLMs judge videos (Han et al., 2025; Liu et al., 2025b; Song et al., 2025); and FilmBench and DirectorBench target creative production (Wang et al., 2026; Chen et al., 2026). These benchmarks measure the quality of generated videos, but not whether a method beats resampling under the same budget. Test-time scaling shows that selecting among more samples improves generation (Ma et al., 2025; Liu et al., 2025a), and PRIS, VQQA, and VISTA compare their own methods against such baselines within one or two model families (Kim et al., 2026; Song et al., 2026; Long et al., 2026). We instead compare multiple agents against BoN with a simple prompt at a matched budget across generators (see also Appendix A).

3 Task Formalization

3.1 Task Scope

We study agentic video generation in practical creative production, where a video must fulfill the user's intent while maintaining visual quality. The input consists of textual requirements and visual references. Figure 2 illustrates the four prompting strategies. Simple Prompt is included as a non-refined baseline to measure whether these interventions provide improvements beyond unchanged prompting. Human Prompt represents the commonly adopted human-in-the-loop process, where creators inspect generated results and manually revise instructions. Iterative Prompt (RAPO++, VQQA, Simple Iterative) and Prompt Enhancement (RAPO, SCMAPR, VPO, Simple Rewrite) represent agentic prompt refinement approaches. In the text, we refer to methods in these two families as prompt enhancement methods and iterative agents; Simple Rewrite and Simple Iterative are their simple counterparts, sharing the LLM used by Simple Prompt.

3.2 Sampling under a Fixed Budget

In practical video production, creators often generate multiple candidates and select the most suitable one, motivating a best-of-N comparison. For each method–generator condition, we fix the video-generation budget at $N = 20$ videos. For iterative methods, these videos are generated across successive refinement steps; videos inspected for feedback also count toward the same $N = 20$ BoN candidate pool. The additional LLM and VLM calls made by Prompt Enhancement and Iterative Prompt methods for rewriting and feedback are not counted, which favors these methods.

Table 1: **Prompting methods and video allocation.** Every method generates $N = 20$ videos per generator; for iterative agents, videos used only for feedback also count. Implementation details are given in Appendix Table 7.

Strategy	Method	Revision model	Feedback	Videos	Generators
Baseline	Simple Prompt	Gemini-3.7-Flash	No	1 prompt × 20	All six
Human Expert	Human Prompt	Two experts	Yes	1 SP + 9 + 10	All six
Prompt Enhancement	Simple Rewrite	Gemini-3.7-Flash	No	20 prompts × 1	All six
	VPO	VPO-5B	No		All six
	RAPO	Mistral-7B-Instruct-v0.3	No		Open-source
	SCMAPR	DeepSeek-V4-Flash, BGE-M3	No		Open-source
Iterative Prompt	Simple Iterative	Gemini-3.7-Flash	Yes	1 SP + 19 rounds	All six
	RAPO++	Qwen2.5-VL-7B, Qwen2.5-7B	Yes		All six
	VQQA	Gemini-3.1-Pro-Preview	Yes		All six

SP: one Simple Prompt video; Human Prompt adds 9 and 10 videos from two expert prompts.

3.3 Evaluation Dimensions

We evaluate every video on four dimensions. *Entity/object fulfillment* checks whether the requested characters, objects, and scene elements appear; *action fulfillment* checks whether the requested actions are carried out as described; and *identity/appearance persistence* checks whether the characters and scene match the reference images and remain consistent throughout the video. These three semantic dimensions are scored by an MLLM evaluator on a 1–10 scale (Section 4.3). *Visual aesthetics* is measured by the VADB Total Score (Qiao et al., 2025), a reference-free aesthetic model that sees only the video frames. Together, these dimensions cover the aspects that the compared methods target in their original papers, and all four are required by our request, so the evaluation reflects what the user asks for rather than any single method's objective.

4 Study Design

4.1 Prompting Strategies

We compare four prompting strategies in our controlled experiment. Simple Prompt, Simple Rewrite, and Simple Iterative all use Gemini-3.7-Flash, but differ in the prompt-revision procedure and feedback available to the model. In the Simple Prompt condition, Gemini-3.7-Flash only executes a fixed instruction that minimally converts the original script into a video-generation prompt. Simple Rewrite performs a constrained one-shot rewrite without inspecting the generated videos. Simple Iterative instead uses a revision instruction and generated-video feedback in an iterative refinement loop. Human Prompt uses two prompts written by two domain experts, who were allowed to inspect a generated preview from Simple Prompt before finalizing their prompts.

Prompt Enhancement comprises Simple Rewrite, RAPO (Gao et al., 2025a), SCMAPR (Yang et al., 2026), and VPO (Cheng et al., 2025). These methods revise the prompt without conditioning on videos generated in our evaluation loop. Beyond the Simple Iterative baseline, Iterative Prompt includes a reproduction of the SSPO stage of RAPO++ (Gao et al., 2025b) and our VQQA reproduction (Song et al., 2026); all three iterative methods use generated-video feedback to form generator-specific revised prompts. Table 1 summarizes the revision models, feedback access, video allocation, and generator coverage.

4.2 Generation Protocol

We evaluate six video generators spanning open-source and proprietary systems. HunyuanVideo-1.5 (Hunyuan-1.5) is deployed through Tencent's OmniWeaving inference pipeline, while MiniMax-H3 and Wan-2.1 are run locally through ComfyUI using their public model releases. The proprietary generators are Seedance-2.0-fast, Gemini Omni Flash, and Wan-3.0, accessed through their hosted interfaces (Table 2). The official prompt enhancement (PE) of Hunyuan-1.5 and Wan-3.0 is enabled and held fixed across methods; Appendix B compares results with it disabled.

Table 2: **Video generators.** Official PE: generator-native prompt enhancement, enabled ("apply") or available but disabled.

Generator	Model size	Official PE	Deployment
HunyuanVideo-1.5	8.3B	apply	OmniWeaving (local)
MiniMax-H3	33B	disabled	ComfyUI (local)
Wan-2.1	14B	disabled	ComfyUI (local)
Seedance-2.0-fast	Undisclosed	Undisclosed	Hosted interface
Gemini Omni Flash	Undisclosed	Undisclosed	Hosted interface
Wan-3.0	Undisclosed	apply	Hosted interface

All prompting strategies are evaluated on the same controlled request, consisting of one fixed Simple Prompt, two character reference images, one scene reference image, and the same target action sequence. Within each generator, all methods use the same generation interface and settings, including the reference inputs and available controls for duration, resolution, sampling, and prompt enhancement where exposed by the interface.

4.3 Automatic Evaluation

Each semantic dimension is scored independently using GPT-5.6 Sol with low reasoning effort: the evaluator receives the Simple Prompt text, which is the original request, a dimension-specific evaluation prompt, and 16 temporally ordered frames sampled from the video. The identity evaluation additionally receives the three reference images. Each dimension returns a score on the same 1–10 scale. Structured MLLM-based video evaluation has shown meaningful agreement with human judgments in prior video-generation benchmarks (Han et al., 2025). We treat automatic scores as scalable, dimension-specific evidence rather than a substitute for human judgment, and complement them with blinded human evaluation.

The VADB Total Score (Qiao et al., 2025) is computed from the video frames alone, without the prompt, reference images, or agent history.

4.4 Human Evaluation

We conducted a blinded A/B evaluation. Each comparator video was paired only with Simple Prompt videos from the same generator. Reviewers saw the Simple Prompt, the three reference images, and two videos labelled A and B. They did not see method identities, optimized prompts, or which video was the baseline. For each pair, reviewers made two forced choices: which video better matched the requested content and references, and which had better visual quality. Each comparator video faced two different Simple Prompt videos in two deterministic rounds.

The data contain 42 comparator groups and 1680 completed video pairs, yielding 3360 binary judgments. Forty-nine unique reviewer IDs, representing film-school students and faculty as well as film-industry professionals, contributed retained votes after deduplication. Each pair was rated by one reviewer on both criteria, and the experts who wrote Human Prompt did not serve as reviewers.

5 Empirical Results

5.1 Best Videos under a Matched Budget

Figure 3 compares each method's best video with that of Simple Prompt on every generator, and Table 3 summarizes them as normalized radar areas. The results follow the four findings of Section 1.

Agents do not beat Simple Prompt. No method's Maximum improves on Simple Prompt in all four dimensions (Table 3a). Every gain comes with a loss elsewhere, and the losses concentrate in action fulfillment: VPO loses 30.2 points, RAPO++ 20.0, Simple Rewrite 8.0, and Simple Iterative 5.5, while only VQQA stays level (+0.2). Visual quality barely moves; the largest VADB gain is 3.3 points. Top-5 Mean, available for the non-iterative methods, shows the same pattern (Table 3b), so the result does not hinge on a single video.

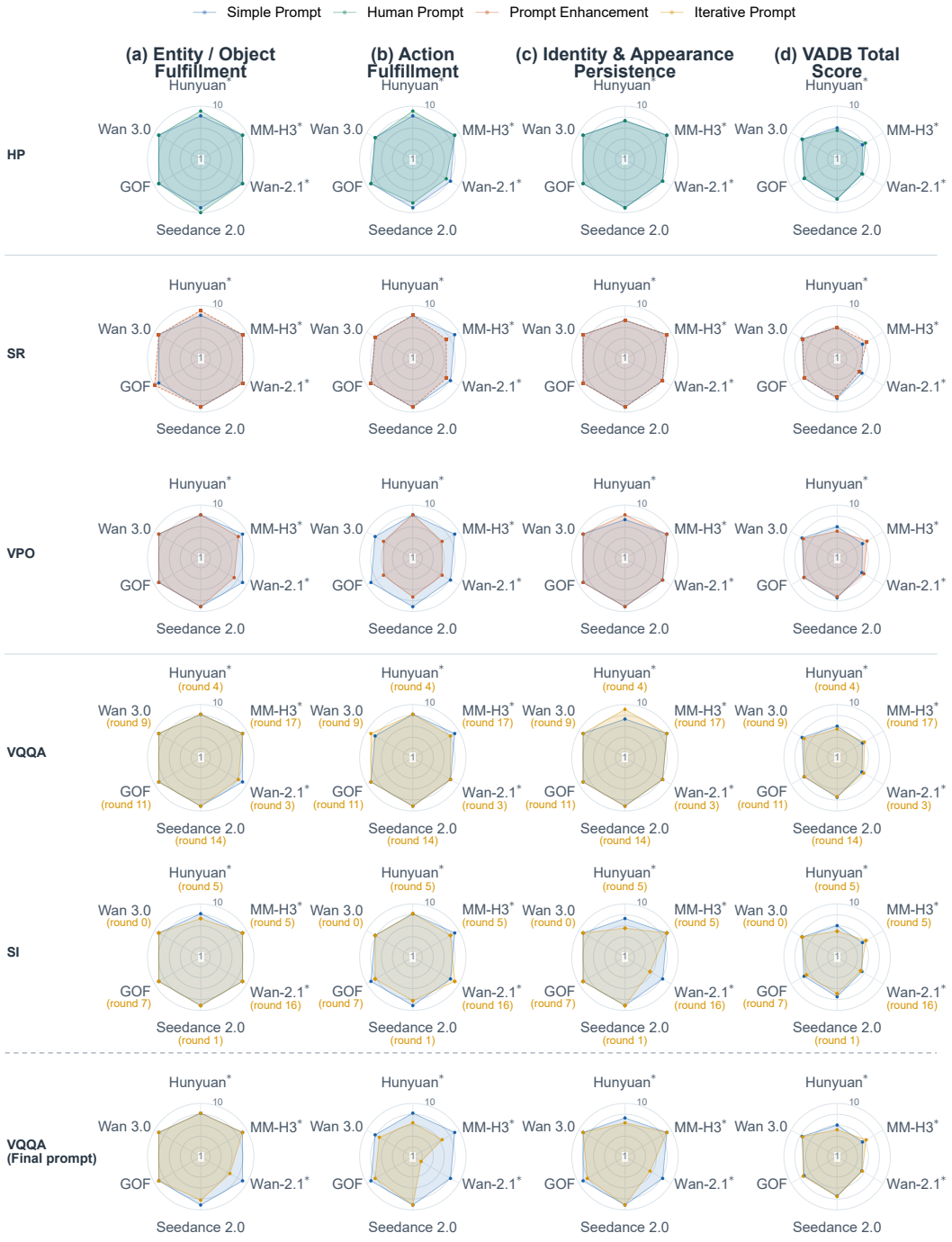

Figure 3: **Best and final-round videos compared with the best Simple Prompt video.** Rows show, from top, the Maximum of Human Prompt, Simple Rewrite, VPO, VQQA, and Simple Iterative, and the VQQA Final Round. Within a row, the four columns show the *same video*, selected by the sum of its four scores. Round labels mark the selected VQQA rounds; asterisks mark open-source generators. Further profiles appear in Appendix Figs. 19 and 20.

Table 3: **Normalized radar areas.** Areas are percentages of the full 10-point polygon over six generators. Blocks marked * cover only the three open-source generators (Hunyuan-1.5, MiniMax-H3, Wan-2.1), on which RAPO and SCMAPR were evaluated, and are compared with **SP** on the same three generators. **SP** gives absolute areas; other columns give differences from **SP** in the same block, with Final Round (round 19) compared to **SP** Maximum. Green and red mark values above and below **SP**; small numbers show the highest and lowest per-generator values. HP: Human Prompt; SR: Simple Rewrite; SI: Simple Iterative.

(a) Best video per pool

	Maximum							Final Round		
	SP	**HP**	**SR**	**VPO**	**SI**	**RAPO++**	**VQQA**	**SI**	**RAPO++**	**VQQA**
Entity/Obj.	78.0 (9.0, 8.0)	+6.0 (+1.0, 0.0)	+6.0 (+1.0, 0.0)	−8.5 (0.0, -2.0)	−3.0 (0.0, -1.0)	−3.0 (0.0, -1.0)	−3.0 (0.0, -1.0)	−36.0 (-1.0, -4.0)	−19.3 (0.0, -3.0)	−11.5 (0.0, -3.0)
Action	72.2 (9.0, 8.0)	−2.8 (+1.0, -1.0)	−8.0 (0.0, -2.0)	−30.2 (0.0, -3.0)	−5.5 (+1.0, -1.0)	−20.0 (0.0, -3.0)	+0.2 (+1.0, -1.0)	−41.5 (-1.0, -6.0)	−25.3 (-1.0, -3.0)	−35.3 (0.0, -7.0)
Ident./Appear.	72.0 (9.0, 7.0)	0.0 (0.0, 0.0)	0.0 (0.0, 0.0)	+3.0 (+1.0, 0.0)	−15.0 (0.0, -3.0)	0.0 (+1.0, -1.0)	+6.0 (+2.0, 0.0)	−40.2 (0.0, -5.0)	−32.2 (0.0, -7.0)	−15.0 (0.0, -3.0)
VADB Total	38.3 (7.4, 4.9)	−0.1 (+0.7, -0.5)	−0.6 (+1.0, -0.7)	−0.1 (+1.1, -0.9)	−4.3 (+0.8, -1.2)	+3.3 (+2.1, -0.4)	−0.9 (+0.4, -0.6)	−7.7 (-0.2, -1.7)	−1.6 (+0.3, -0.5)	−0.1 (+0.9, -1.0)
Area Sum	260.5	+3.1	−2.6	−35.8	−27.8	−19.7	+2.2	−125.4	−78.5	−61.9

(b) Top-5 Mean and open-source subset

	Top-5 Mean				Top-5 Mean*			Maximum*		
	SP	**HP**	**SR**	**VPO**	**SP**	**RAPO**	**SCMAPR**	**SP**	**RAPO**	**SCMAPR**
Entity/Obj.	76.2 (9.0, 8.2)	+2.9 (+0.8, -0.2)	+3.6 (+0.6, -0.2)	−9.5 (+0.2, -0.8)	71.7 (8.8, 8.2)	−11.1 (-0.2, -1.6)	+3.4 (+0.6, -0.2)	75.0 (9.0, 8.0)	−11.3 (+1.0, -2.0)	0.0 (+1.0, -1.0)
Action	64.6 (8.8, 7.2)	−2.3 (+0.4, -0.4)	−4.1 (+0.2, -0.6)	−30.9 (-1.8, -2.6)	56.7 (7.8, 7.2)	−13.2 (-0.6, -1.2)	−1.0 (0.0, -0.2)	69.3 (9.0, 8.0)	−16.0 (0.0, -2.0)	−5.3 (0.0, -1.0)
Ident./Appear.	65.5 (9.0, 5.2)	+1.2 (+1.4, -1.0)	+1.3 (+1.2, -0.4)	+2.9 (+2.0, -0.8)	52.5 (9.0, 5.2)	+2.7 (+2.2, -1.6)	+4.8 (+1.4, -0.4)	63.7 (9.0, 7.0)	−5.3 (0.0, -1.0)	+5.7 (+1.0, 0.0)
VADB Total	37.5 (7.3, 5.0)	−1.3 (+0.1, -0.4)	−0.8 (+0.1, -0.3)	+0.9 (+0.5, -0.3)	27.5 (5.6, 5.0)	−1.5 (+0.2, -0.4)	−1.2 (+0.1, -0.4)	26.8 (5.5, 4.9)	−0.4 (+0.4, -0.6)	−3.1 (+0.5, -0.7)
Area Sum	243.9	+0.5	−0.2	−36.6	208.4	−23.0	+6.0	234.8	−33.0	−2.7

More elaborate methods add little. Among prompt enhancement methods, Simple Rewrite stays near Simple Prompt (area sum −2.6 at Maximum and −0.2 at Top-5 Mean), VPO and RAPO fall far below it, and SCMAPR, evaluated only on open-source generators, is comparable (+6.0 and −2.7). Among iterative agents, VQQA (+2.2) and RAPO++ (−19.7) improve on Simple Iterative (−27.8), but neither clearly exceeds Simple Prompt. Human Prompt is likewise near parity (+3.1 and +0.5).

Gains rarely transfer. The largest per-generator gains concentrate on Hunyuan-1.5. There, the Maximum videos of Human Prompt, VPO, and VQQA meet or exceed Simple Prompt in the sum of four scores, and VPO raises the Top-5 Mean identity score from 5.2 to 7.2 (Appendix Figure 19). On the other five generators, VPO falls below Simple Prompt, and its identity gain does not recur.

Iteration does not improve round by round. For all three iterative agents, final-round videos fall far below Simple Prompt (area sums −125.4, −78.5, and −61.9) and well below each agent's own Maximum (Table 3; Appendix Figure 19). The best videos come from earlier rounds, marked in Figure 3 for VQQA and in Appendix Figure 19 for RAPO++, and per-round scores fluctuate rather than rise (Appendix B.2). Selecting the best round brings VQQA to the level of Simple Prompt, but this selection turns the agent into BoN over a changing prompt.

5.2 Human Evaluation and Expert Prompts

Blinded pairwise judgments against Simple Prompt (Table 4) show the same pattern as the automatic scores. VPO wins 55.4% of visual comparisons overall, with rates above 50% confined to the three open-source generators, but wins only 23.8% on semantic alignment. Four of the five automated methods evaluated on all six generators have semantic win rates below 50%. Human Prompt reaches

Table 4: **Human win rates (%) against Simple Prompt**, reported as semantic alignment / visual quality. Green: above 50%; red: below 50%.

Prompt Strategy	Method	Open-Source Generators			Proprietary Generators			Average
		Hunyuan-1.5	MiniMax-H3	Wan-2.1	Seedance-2.0-fast	Gemini Omni Flash	Wan-3.0	
Human Expert	Human Prompt	80.0 / 80.0	47.5 / 40.0	40.0 / 47.5	45.0 / 40.0	50.0 / 47.5	40.0 / 40.0	50.4 / 49.2
Prompt Enhancement	Simple Rewrite	70.0 / 67.5	50.0 / 55.0	47.5 / 52.5	45.0 / 50.0	37.5 / 42.5	50.0 / 25.0	50.0 / 48.8
	VPO	47.5 / 70.0	25.0 / 62.5	45.0 / 72.5	10.0 / 47.5	7.5 / 47.5	7.5 / 32.5	23.8 / 55.4
	RAPO	62.5 / 65.0	30.0 / 52.5	37.5 / 52.5	–	–	–	43.3 / 56.7
	SCMAPR	65.0 / 47.5	52.5 / 60.0	47.5 / 55.0	–	–	–	55.0 / 54.2
Iterative Prompt	Simple Iterative	30.0 / 70.0	12.5 / 25.0	12.5 / 32.5	15.0 / 52.5	25.0 / 25.0	47.5 / 55.0	23.8 / 43.3
	VQQA	67.5 / 62.5	22.5 / 15.0	42.5 / 42.5	47.5 / 60.0	55.0 / 60.0	35.0 / 45.0	45.0 / 47.5
	RAPO++	47.5 / 45.0	52.5 / 67.5	20.0 / 15.0	47.5 / 60.0	60.0 / 60.0	42.5 / 52.5	45.0 / 50.0

80.0% on both criteria on Hunyuan-1.5, yet averages 50.4% in semantic alignment and 49.2% in visual quality across all six generators.

5.3 Qualitative Experiment

On MiniMax-H3, the Simple Prompt video shows the attack, grappling, and wall breach, whereas VPO omits much of this action (Figure 4(a,b)). Their Action Fulfillment scores are 7 and 3.

The Human Prompt examples differ across generators: Hunyuan-1.5 shows the scanner, confrontation, and wall breach, while Wan-2.1 lacks a comparably clear ending (Figure 4(c,d)). The corresponding human win rates are 80.0% and 40.0% for semantic alignment (Table 4).

On Gemini Omni Flash, VQQA's round-4 video preserves the two protagonists and their action sequence, whereas round 12 adds an extra character and confuses their roles (Figure 4(e,f)). Their mean score across the three semantic dimensions falls from 9.00 to 4.33, illustrating a worse later output in this trajectory. Additional qualitative examples are shown in Appendix D.



Figure 4: **Qualitative comparisons across methods, generators, and refinement rounds.** (a,b) VPO sample 17 and Simple Prompt sample 6 on MiniMax-H3. (c,d) Human Prompt on Hunyuan-1.5 and Wan-2.1. (e,f) VQQA on Gemini Omni Flash at rounds 4 and 12. Frames within each panel progress from left to right in video time.

6 Discussion

6.1 Why Refinement Fails to Transfer

Both Prompt Enhancement and Iterative Prompt can be seen as attempts to rewrite the prompt into a form that a given generator handles well. They differ in how this target is found: Prompt Enhancement fixes it offline, whereas Iterative Prompt estimates it online from the videos it generates.

Prompt enhancement fixes the target offline. A prompt enhancer implicitly maps requests to prompts that a particular generator prefers. Since each generator is trained on its own data and captions, a mapping built for one can push prompts out of distribution for another. Accordingly, light rewrites such as Simple Rewrite stay close to Simple Prompt, whereas heavy rewrites such as VPO help on some generators and hurt on others (Table 3). The mapping also follows the enhancer's own objective: VPO's Top-5 Mean action score falls on every generator (Appendix Figure 19). For proprietary generators, this preferred distribution is not observable from outside, which may explain why providers couple their own enhancers to their generators (e.g., Wan-3.0). Simple Prompt is not optimal for any particular generator, but it imposes no other model's preferences, making it a low-risk input.

Iterative refinement estimates the target online. Iterative agents query the target generator directly, but this relies on two conditions: (1) the judge model must reliably tell better videos from worse ones, and (2) the agent must turn this judgment into a prompt edit that moves generation in the right direction. Unlike training with gradients, the agent has no clear signal for how to edit the prompt, so its edits are largely guesses; a wrong edit can make later videos worse, and the loop may not recover. Accordingly, scores fluctuate across iterations rather than improving steadily (Appendix B.2), and their final-round videos vary widely across generators, close to Simple Prompt on some and far below it on others (Table 3).

In short, prompt enhancement risks a mismatched target, while iterative refinement risks a noisy one. Under the same budget, the revised candidates rarely surpass what additional samples from Simple Prompt already provide.

6.2 Why Our Results Differ from Prior Reports

Several of the methods we evaluate report clear gains in their original papers. Three differences in setup may explain why these gains do not carry over. First, the question differs. RAPO, SCMAPR, and VPO report the average quality of videos generated from rewritten prompts, and RAPO++ reports scores across refinement rounds, without comparing against an equal number of Simple Prompt generations; our protocol counts every video, including those used for feedback, and compares the best candidates. Second, the generators differ. Most methods were developed and evaluated mainly on one or two open models, such as LaVie, Latte, or CogVideoX, whereas we evaluate each on up to six generators, including proprietary ones; VQQA, which does compare against BoN, reports its gains with at most five videos on CogVideoX and Veo. Third, the task differs. Prior benchmarks mostly use short text-only prompts, whereas our request combines reference images with a multi-character action sequence, in which a rewrite can easily drop required actions (Appendix C.3). Showing that an agent helps therefore requires evidence at two levels: that its prompts improve across repeated generations, and that its full budget beats simply trying again on the target generator.

7 Conclusion

We asked whether video generation agents can beat simply trying again when both receive the same number of generations. Across up to six generators, they largely cannot: published agents fare no better than Simple Prompt, the few gains rarely transfer across generators, and iterative refinement does not improve round by round. Our evidence comes from a single request, and final-round results rest on one video per round over more rounds than some agents were designed for, so testing more requests is an important next step. Agent gains should be tested against matched-budget BoN with a simple prompt on the target generator, alongside a simple counterpart from the same family.

AI USAGE STATEMENT

Large language models (LLMs) and multimodal large language models (MLLMs) are used as components of the methods evaluated in this work, including prompt rewriting, iterative refinement, and automatic evaluation, as described in the main text and appendix. All video samples analyzed in this study are generated by the evaluated video-generation models under the experimental protocols described in the paper.

Generative AI tools were also used to assist with research implementation and manuscript preparation. In particular, AI-assisted tools were used to help write and debug scripts for video generation, evaluation, and iterative refinement, and to support grammar polishing, wording revisions, LaTeX formatting, and consistency checking. All experimental procedures, results, analyses, and final manuscript content were reviewed and verified by the authors.

REPRODUCIBILITY STATEMENT

We will release the code, model weights, sampled data, and evaluation scripts.

REFERENCES

Alibaba Cloud. Wan3.0 Video Generation. Alibaba Cloud Model Studio documentation, 2026.

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*, pp. 1–11, 2024.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22563–22575, June 2023.

Jiamin Chen, Qianben Chen, Jiawen Zhang, Yidi Wu, Yuchen Li, Xiaokun Zhang, Wangchunshu Zhou, and Chen Ma. DirectorBench: Diagnosing long-form video generation with personalized multi-agent evaluation. *arXiv preprint arXiv:2605.30090*, 2026.

Jiale Cheng, Ruiliang Lyu, Xiaotao Gu, Xiao Liu, Jiazheng Xu, Yida Lu, Jiayan Teng, Zhuoyi Yang, Yuxiao Dong, Jie Tang, Hongning Wang, and Minlie Huang. VPO: Aligning Text-to-Video Generation Models with Prompt Optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15636–15645, October 2025.

Bingjie Gao, Xinyu Gao, Xiaoxue Wu, Yujie Zhou, Yu Qiao, Li Niu, Xinyuan Chen, and Yaohui Wang. The Devil is in the Prompts: Retrieval-Augmented Prompt Optimization for Text-to-Video Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3173–3183, June 2025a.

Bingjie Gao, Qianli Ma, Xiaoxue Wu, Shuai Yang, Guanzhou Lan, Haonan Zhao, Jiaxuan Chen, Qingyang Liu, Yu Qiao, Xinyuan Chen, Yaohui Wang, and Li Niu. RAPO++: Cross-Stage Prompt Optimization for Text-to-Video Generation via Data Alignment and Test-Time Scaling. *arXiv preprint arXiv:2510.20206*, 2025b.

Google DeepMind. Gemini Omni Flash Model Card. Official model card, 2026.

Hui Han, Siyuan Li, Jiaqi Chen, Yiwen Yuan, Yuling Wu, Yufan Deng, Chak Tou Leong, Hanwen Du, Junchen Fu, Youhua Li, Jie Zhang, Chi Zhang, Li-jia Li, and Yongxin Ni. Video-Bench: Human-Aligned Video Generation Benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18858–18868, June 2025.

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

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022b.

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

Yatai Ji, Jiacheng Zhang, Jie Wu, Shilong Zhang, Shoufa Chen, Chongjian Ge, Peize Sun, Weifeng Chen, Wenqi Shao, Xuefeng Xiao, Weilin Huang, and Ping Luo. Prompt-A-Video: Prompt Your Video Diffusion Model via Preference-Aligned LLM. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18725–18735, October 2025.

Subin Kim, Sangwoo Mo, Mamshad Nayeem Rizve, Yiran Xu, Difan Liu, Jinwoo Shin, and Tobias Hinz. Rethinking prompt design for inference-time scaling in text-to-visual generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22090–22099, 2026.

Han Lin, Abhay Zala, Jaemin Cho, and Mohit Bansal. Videodirectorgpt: Consistent multi-scene video generation via llm-guided planning. In *COLM*, 2024.

Fangfu Liu, Hanyang Wang, Yimo Cai, Kaiyan Zhang, Xiaohang Zhan, and Yueqi Duan. Video-T1: Test-time Scaling for Video Generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18671–18681, October 2025a.

Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, and Ying Shan. EvalCrafter: Benchmarking and Evaluating Large Video Generation Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22139–22149, June 2024.

Yuanxin Liu, Rui Zhu, Shuhuai Ren, Jiacong Wang, Haoyuan Guo, Xu Sun, and Lu Jiang. UVE: Are MLLMs Unified Evaluators for AI-Generated Videos? In *Advances in Neural Information Processing Systems*, volume 38, pp. 18866–18901, 2025b. doi: 10.52202/085713-0564. Datasets and Benchmarks Track.

Do Xuan Long, Xingchen Wan, Hootan Nakhost, Chen-Yu Lee, Tomas Pfister, and Sercan Ö. Arik. VISTA: A Test-Time Self-Improving Video Generation Agent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6021–6032, June 2026.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. Scaling Inference Time Compute for Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2523–2534, June 2025.

MiniMax. MiniMax H3. Official model card, 2026.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

Qianqian Qiao, DanDan Zheng, Yihang Bo, Bao Peng, Heng Huang, Longteng Jiang, Huaye Wang, Jingdong Chen, Jun Zhou, and Xin Jin. VADB: A Large-Scale Video Aesthetic Database with Professional and Multi-Dimensional Annotations. In *Advances in Neural Information Processing Systems*, volume 38, pp. 99743–99765, 2025. doi: 10.52202/085713-3003. Datasets and Benchmarks Track.

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

Tingyu Song, Tongyan Hu, Guo Gan, and Yilun Zhao. VF-Eval: Evaluating Multimodal LLMs for Generating Feedback on AIGC Videos. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21126–21146. Association for Computational Linguistics, July 2025. doi: 10.18653/v1/2025.acl-long.1027.

Yiwen Song, Tomas Pfister, and Yale Song. VQQA: An Agentic Approach for Video Evaluation and Quality Improvement. *arXiv preprint arXiv:2603.12310*, 2026.

Kaiyue Sun, Kaiyi Huang, Xian Liu, Yue Wu, Zihan Xu, Zhenguo Li, and Xihui Liu. T2V-CompBench: A Comprehensive Benchmark for Compositional Text-to-Video Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8406–8416, June 2025.

Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, et al. Seedance 2.0: Advancing video generation for world complexity, 2026.

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.

Shengyi Wang, Niantong Li, Guangzheng Hu, Hong Qi, Fei Ding, Weixu Qiao, Jinlin Wang, Xiaotong Lv, Peng Han, Zimeng Li, Fanshu Ding, Yushu Wang, Han Wu, Jingjing Chen, Chongxiao Wang, Yanhao Wu, Chenglong Huang, Xiaoqian Zhu, Jie Tian, Hua Li, Jingjing Fan, Mingshuang Tang, Zhong Li, Hengxia Qiang, Weibin Chen, Jinyang Zhen, Bing Zhao, Lin Qu, Jing Li, and Hu Wei. FilmBench: A film-grade benchmark for cinematic video generation. *arXiv preprint arXiv:2607.24241*, 2026.

Bing Wu, Chang Zou, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Jack Peng, Jianbing Wu, Jiangfeng Xiong, Jie Jiang, et al. Hunyuanvideo 1.5 technical report. *arXiv preprint arXiv:2511.18870*, 2025a.

Weijia Wu, Zeyu Zhu, and Mike Zheng Shou. Automated movie generation via multi-agent cot planning. *arXiv preprint arXiv:2503.07314*, 2025b.

Qiyao Xue, Xiangyu Yin, Boyuan Yang, and Wei Gao. Phyt2v: Llm-guided iterative self-refinement for physics-grounded text-to-video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18826–18836, June 2025.

Chengyi Yang, Pengzhen Li, Jiayin Qi, Aimin Zhou, Ji Wu, and Ji Liu. SCMAPR: Self-Correcting Multi-Agent Prompt Refinement for Complex-Scenario Text-to-Video Generation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 6942–6973. Association for Computational Linguistics, July 2026. doi: 10.18653/v1/2026.findings-acl.345.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, volume 2025, pp. 83048–83077, 2025.

Zangwei Zheng, Xiangyu Peng, Yuxuan Lou, Chenhui Shen, Tom Young, Xinying Guo, Binluo Wang, Hang Xu, Hongxin Liu, Mingyan Jiang, et al. Open-sora 2.0: Training a commercial-level video generation model in $200k. *arXiv preprint arXiv:2503.09642*, 2025.

A Detailed Related Work

Video Generation Models. Early video diffusion work extended image diffusion architectures to model temporal coherence (Ho et al., 2022b). Imagen Video (Ho et al., 2022a) combined a base video diffusion model with spatial and temporal super-resolution, while Make-A-Video (Singer et al., 2022) built on text-to-image models and learned motion from unpaired videos. Video LDM (Blattmann et al., 2023) adapted pretrained image latent diffusion models by adding temporal layers. Lumiere (Bar-Tal et al., 2024) later used a space-time U-Net to generate an entire clip in one pass. Openly released models such as CogVideoX (Yang et al., 2025) and Open-Sora

2.0 (Zheng et al., 2025) made video generation systems available for further research. Wan2.1 (Wan et al., 2025) and HunyuanVideo-1.5 (Wu et al., 2025a) use diffusion transformers to support both text-to-video and image-to-video generation. Movie Gen (Polyak et al., 2024) demonstrated video personalization, instruction-guided editing, and synchronized audio. More recent models broaden the available inputs and outputs: MiniMax H3 (MiniMax, 2026) and Seedance 2.0 (Team Seedance et al., 2026) accept multimodal references and generate video with audio, while Wan 3.0 (Alibaba Cloud, 2026) supports generation and editing within a unified interface. Gemini Omni Flash (Google DeepMind, 2026) also accepts text, images, audio, and video, with conversational instructions supporting successive edits. These developments give creators greater control over the content supplied to a generator, but leave open how effectively different prompting methods use that control. We examine this question across six generators by comparing prompt refinement strategies under a shared creative request.

Agentic Video Generation. Video generation agents use an LLM or MLLM to prepare generation instructions and coordinate video models, with some also inspecting generated videos to guide revisions. We group methods that refine generation instructions into *prompt enhancement (PE)* and *iterative refinement*, according to whether they use feedback from videos generated for the current request. PE rewrites, expands, or structures the request before generation. For example, VideoDirectorGPT (Lin et al., 2024) uses GPT-4 to expand a request into scene descriptions and entity layouts, while MovieAgent (Wu et al., 2025b) uses GPT-4o-based agents to develop scripts into scene and shot descriptions. At the level of individual prompts, the official HunyuanVideo-1.5 (Wu et al., 2025a) and Wan2.1 (Wan et al., 2025) pipelines already provide rewriting and extension. Dedicated methods use retrieval, verification, or learned rewriting: RAPO (Gao et al., 2025a) combines retrieved modifiers with instruction-based rewriting; SCMAPR (Yang et al., 2026) applies scenario-specific policies and checks revisions for semantic consistency; Prompt-A-Video (Ji et al., 2025) and VPO (Cheng et al., 2025) train prompt rewriters through supervised learning and preference optimization. Iterative refinement instead follows a generate–inspect–revise loop. PhyT2V (Xue et al., 2025) uses physical knowledge and discrepancies between requested and generated content to guide revisions, while VQQA (Song et al., 2026) obtains actionable critiques through visual question answering. The sample-specific optimization stage of RAPO++ (Gao et al., 2025b) combines multiple feedback signals to update prompts. VISTA (Long et al., 2026) selects a video through pairwise comparisons before critique and rewriting, whereas PRIS (Kim et al., 2026) revises prompts to address failures recurring across multiple samples. We evaluate representative methods from both families.

Video Generation Benchmarks. Existing benchmarks examine video generation from several perspectives: VBench (Huang et al., 2024) decomposes generation quality into dimensions such as subject consistency, motion smoothness, and spatial relationships. EvalCrafter (Liu et al., 2024) combines objective metrics with human feedback to assess visual, content, and motion quality. T2V-CompBench (Sun et al., 2025) focuses on compositional requirements, including attribute binding, action binding, and object interactions. Human-centered evaluation provides another perspective: Video-Bench (Han et al., 2025) evaluates video quality and condition alignment using MLLM judgments designed to agree with human preferences, while UVE-Bench (Liu et al., 2025b) tests automatic evaluators against human pairwise judgments across multiple dimensions. VF-Eval (Song et al., 2025) examines whether MLLMs can detect and explain errors in generated videos, assessing a capability needed by feedback-driven agents. Benchmarks closer to creative production introduce further criteria: FilmBench (Wang et al., 2026) evaluates cinematic qualities using prompts grounded in professional film references, while DirectorBench (Chen et al., 2026) diagnoses long-form generation workflows against structured criteria and user profiles. PRIS (Kim et al., 2026) compares prompt revision with BoN sampling, while VISTA (Long et al., 2026) combines iterative refinement with tournament-based candidate selection. However, their video-generation evaluations center on the proposed optimizers within individual model families. We instead compare multiple agents against BoN with a simple prompt at a matched budget across generators.

B Additional Quantitative Results

B.1 Candidate Score Distributions

Figures 5–10 show the score distributions of the candidate pools for the prompting methods evaluated on each of the six video generators. For each method–generator condition, the candidate pool contains the $N = 20$ videos used in our matched video-generation budget. We report distributions for Entity/Object Fulfillment, Action Fulfillment, Identity/Appearance Persistence, and VADB Total Score. For prompt-enhancement methods, the 20 candidates are generated from different rewritten prompts; therefore, these plots characterize strategy-level candidate pools rather than repeated samples from a single fixed prompt.

For Hunyuan-1.5 and Wan-3.0, the official prompt-enhancement mechanism is enabled in the main experiments and held fixed across prompting methods. To isolate the effect of this generator-native prompt enhancement, Figures 12 and 11 additionally compare score distributions with the official mechanism disabled (Original) and enabled (Expand), while keeping the external prompting method unchanged.

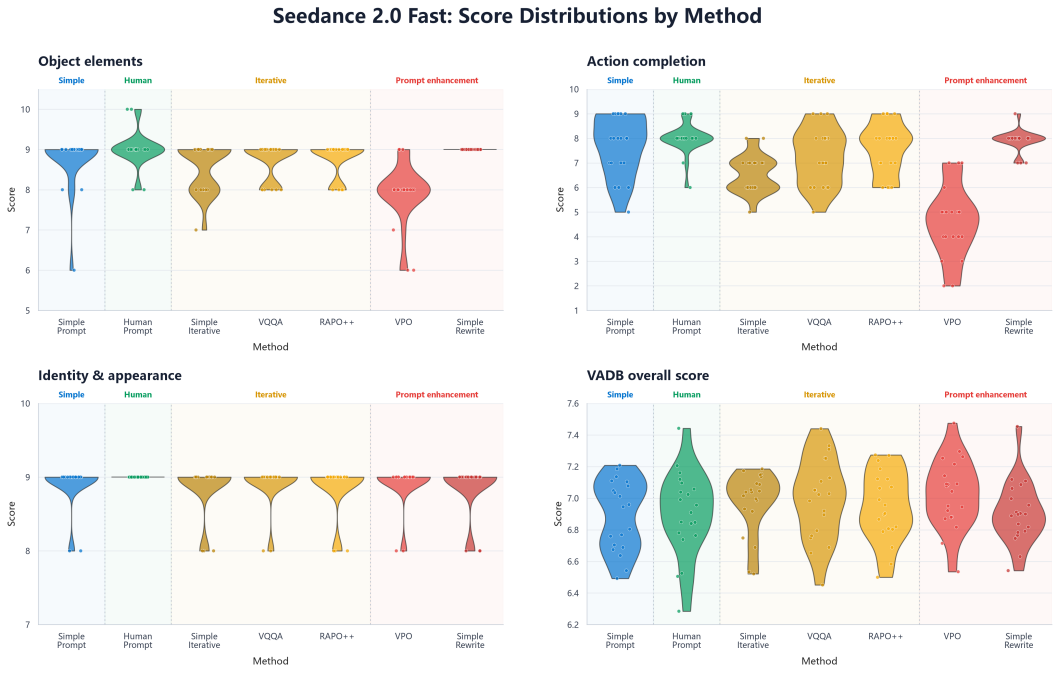

Figure 5: **Score distributions across prompting methods on Seedance-2.0-fast.**

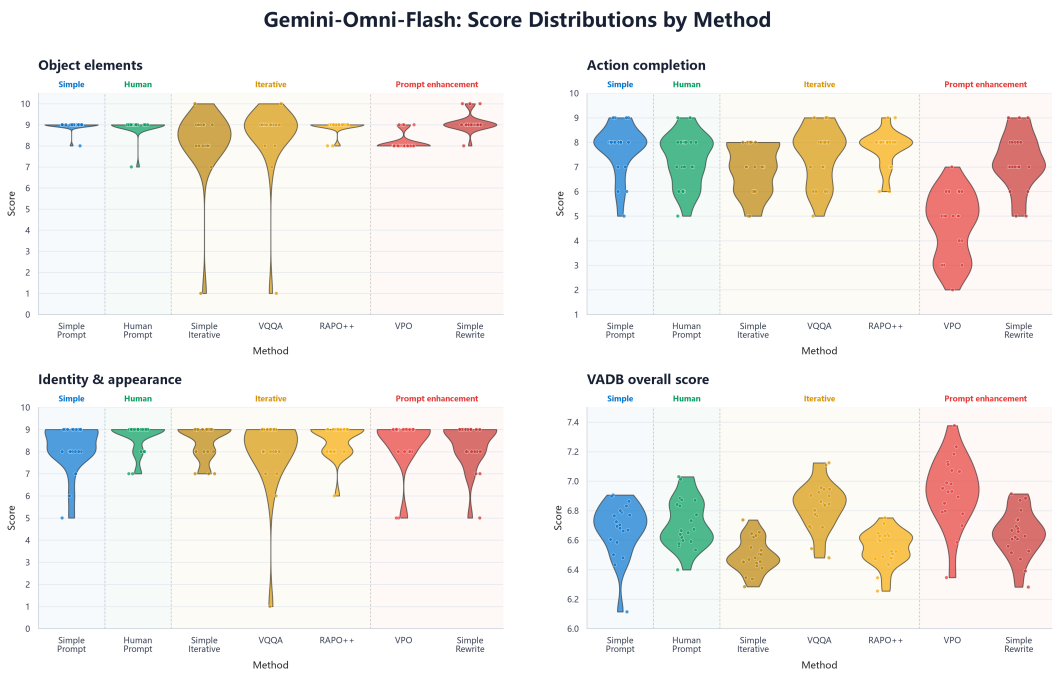

Figure 6: **Score distributions across prompting methods on Gemini Omni Flash.**

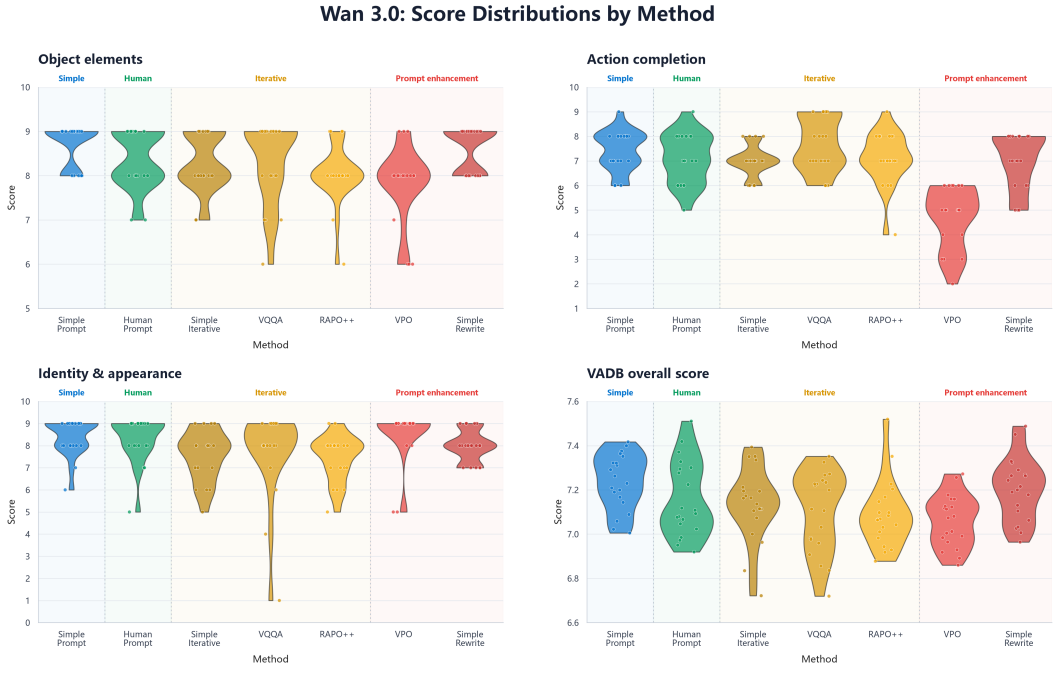

Figure 7: **Score distributions across prompting methods on Wan-3.0.**

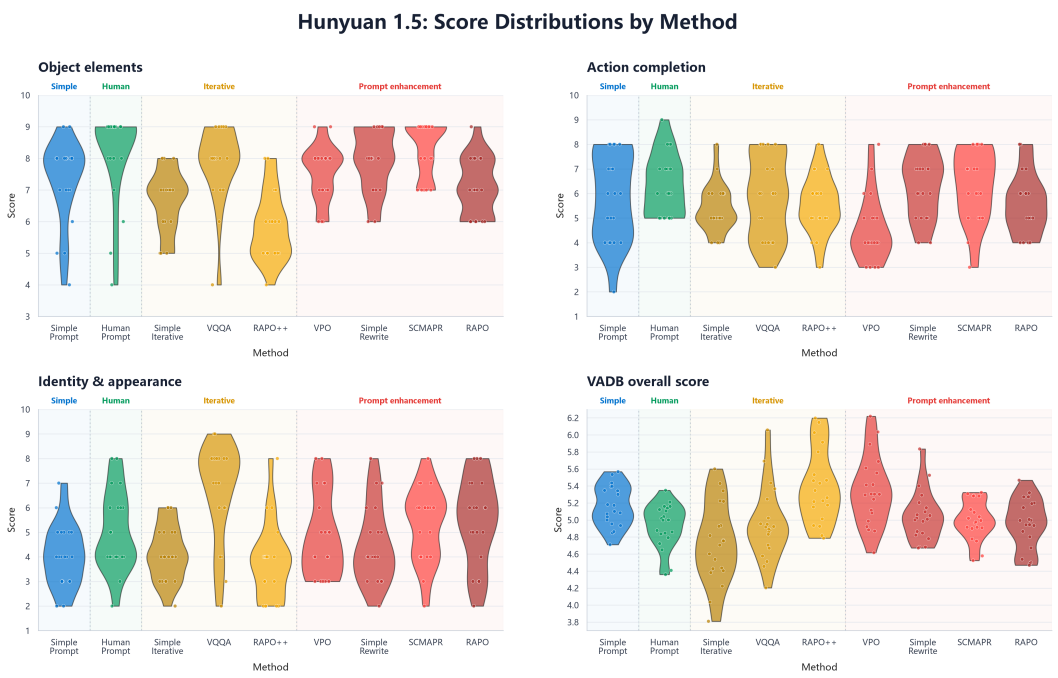

Figure 8: **Score distributions across prompting methods on Hunyuan-1.5.**

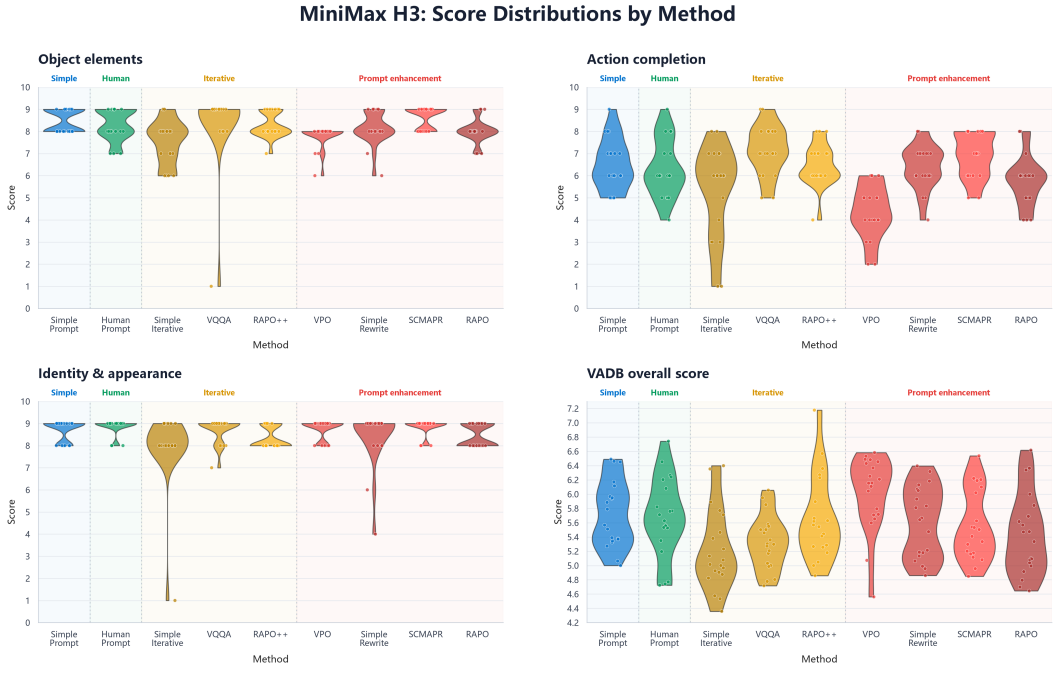

Figure 9: **Score distributions across prompting methods on MiniMax-H3.**

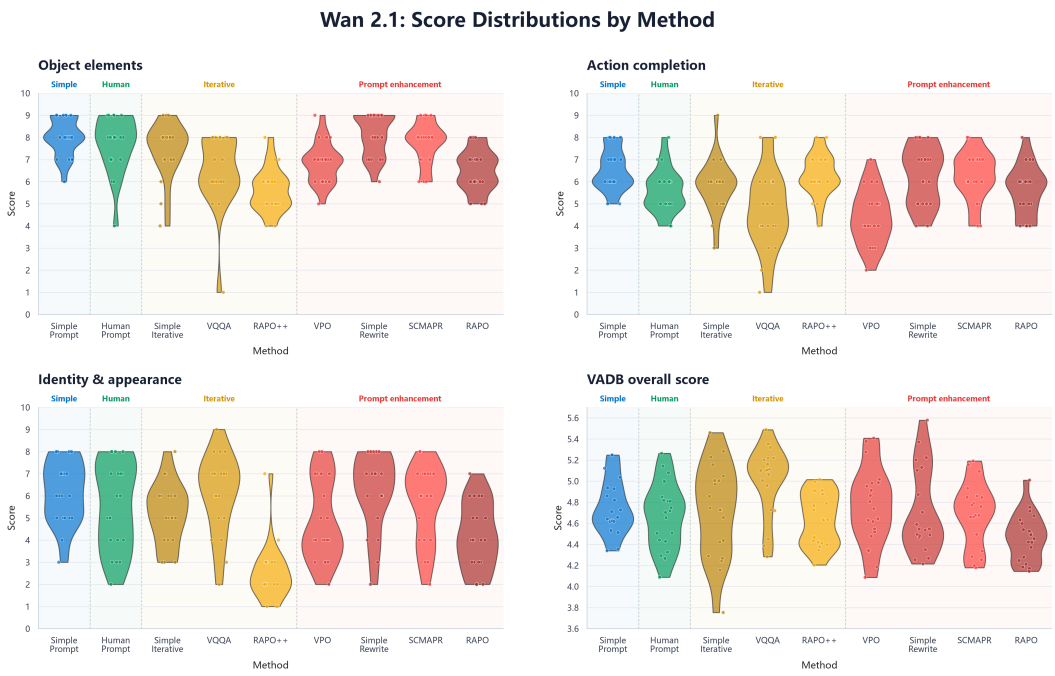

Figure 10: **Score distributions across prompting methods on Wan-2.1.**

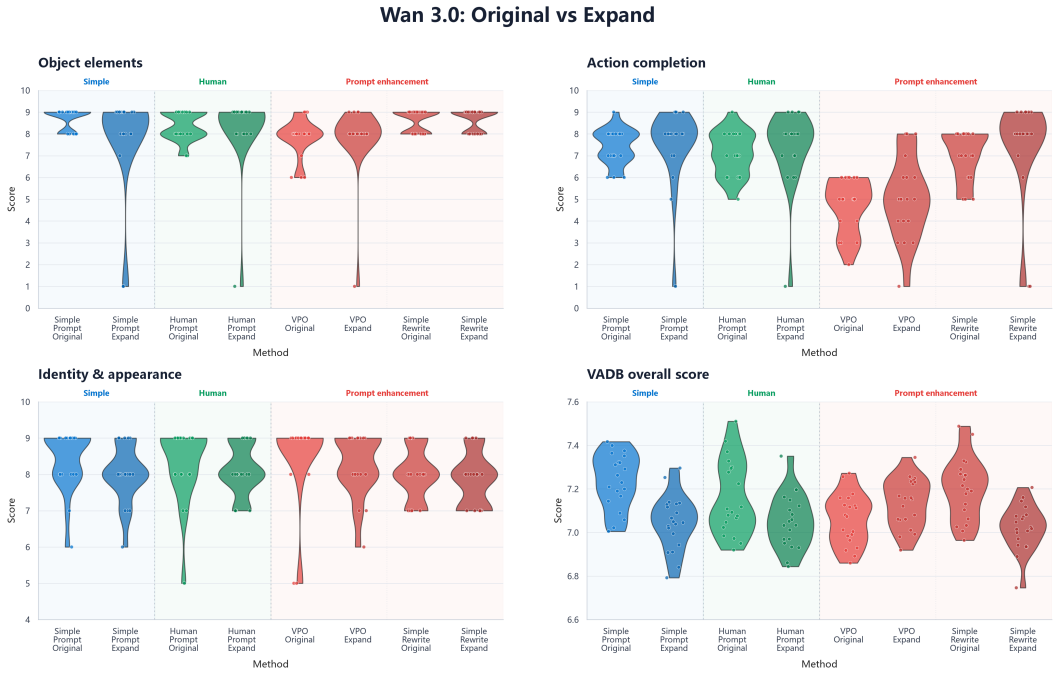

Figure 11: **Effect of the official prompt-enhancement mechanism on Wan-3.0.** Score distributions are compared with the generator-native prompt enhancement disabled (Original) and enabled (Expand) for the same external prompting methods.

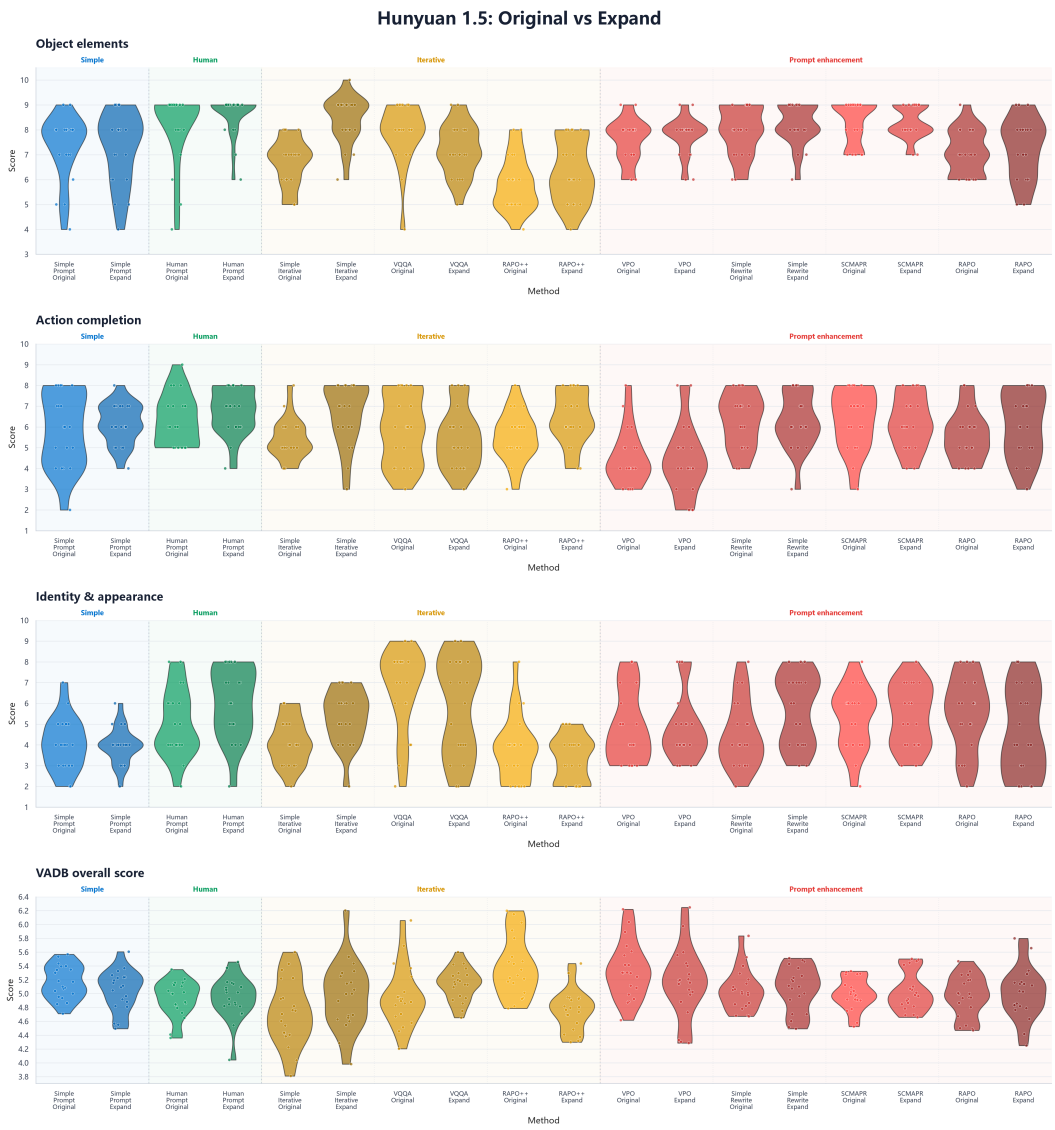

Figure 12: **Effect of the official prompt-enhancement mechanism on Hunyuan-1.5.** Score distributions are compared with the generator-native prompt enhancement disabled (Original) and enabled (Expand) for the same external prompting methods.

B.2 Iteration Trajectories

Each point in Figures 13–18 is the score of one video generated at that round for the specified method and generator. These curves describe observed trajectories; they do not estimate the mean quality of each intermediate prompt across repeated generations.

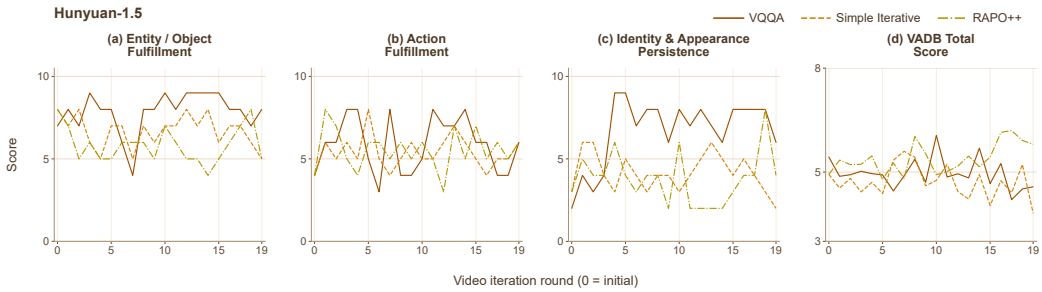

Figure 13: Evaluation scores across prompt iterations for Hunyuan-1.5. Round 0 denotes the initial video from Simple Prompt; rounds 1–19 are successive iterations. The curves compare VQQA, Simple Iterative, and RAPO++ on (a) Entity/Object Fulfillment, (b) Action Fulfillment, (c) Identity/Appearance Persistence, and (d) VADB Total Score. Panel (d) uses a different y-axis scale.

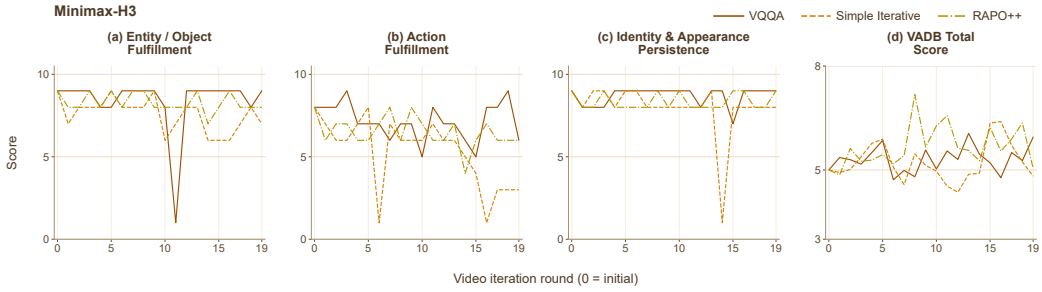

Figure 14: Evaluation scores across prompt iterations for MiniMax-H3. Round 0 denotes the initial video from Simple Prompt; rounds 1–19 are successive iterations. The curves compare VQQA, Simple Iterative, and RAPO++ on (a) Entity/Object Fulfillment, (b) Action Fulfillment, (c) Identity/Appearance Persistence, and (d) VADB Total Score. Panel (d) uses a different y-axis scale.

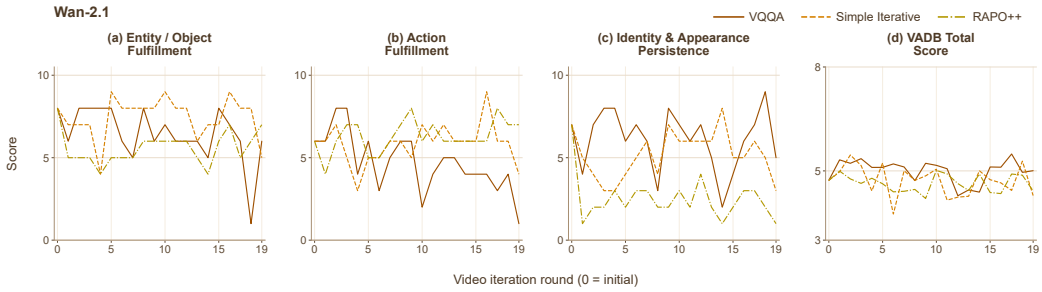

Figure 15: Evaluation scores across prompt iterations for Wan-2.1. Round 0 denotes the initial video from Simple Prompt; rounds 1–19 are successive iterations. The curves compare VQQA, Simple Iterative, and RAPO++ on (a) Entity/Object Fulfillment, (b) Action Fulfillment, (c) Identity/Appearance Persistence, and (d) VADB Total Score. Panel (d) uses a different y-axis scale.

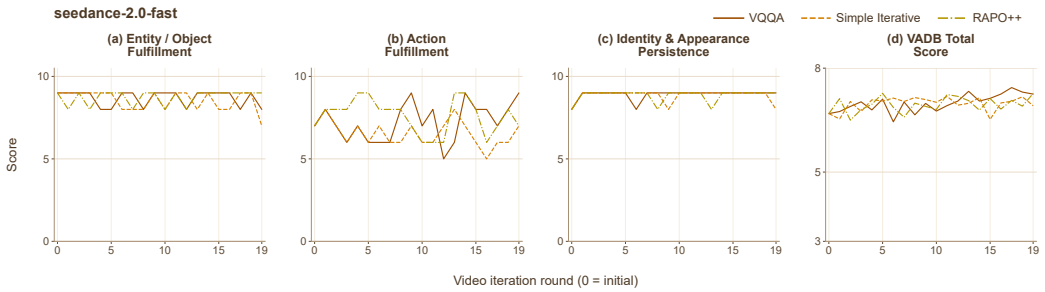

Figure 16: Evaluation scores across prompt iterations for Seedance-2.0-fast. Round 0 denotes the initial video from Simple Prompt; rounds 1–19 are successive iterations. The curves compare VQQA, Simple Iterative, and RAPO++ on (a) Entity/Object Fulfillment, (b) Action Fulfillment, (c) Identity/Appearance Persistence, and (d) VADB Total Score. Panel (d) uses a different y-axis scale.

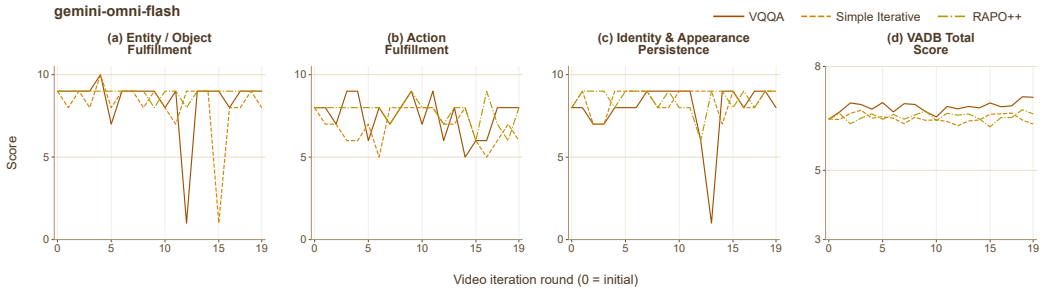

Figure 17: Evaluation scores across prompt iterations for Gemini Omni Flash. Round 0 denotes the initial video from Simple Prompt; rounds 1–19 are successive iterations. The curves compare VQQA, Simple Iterative, and RAPO++ on (a) Entity/Object Fulfillment, (b) Action Fulfillment, (c) Identity/Appearance Persistence, and (d) VADB Total Score. Panel (d) uses a different y-axis scale.

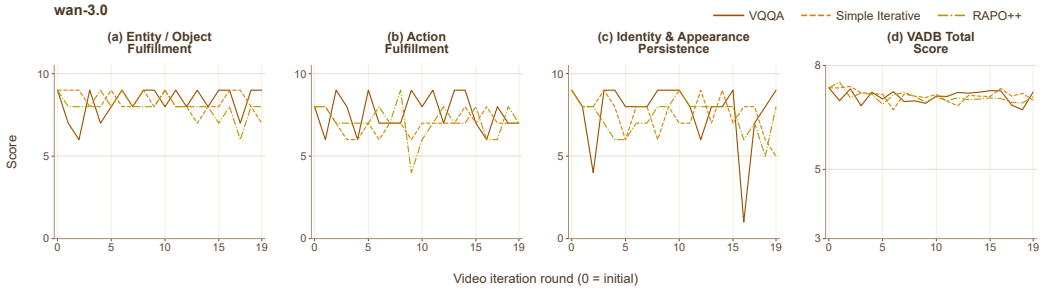

Figure 18: Evaluation scores across prompt iterations for Wan-3.0. Round 0 denotes the initial video from Simple Prompt; rounds 1–19 are successive iterations. The curves compare VQQA, Simple Iterative, and RAPO++ on (a) Entity/Object Fulfillment, (b) Action Fulfillment, (c) Identity/Appearance Persistence, and (d) VADB Total Score. Panel (d) uses a different y-axis scale.

B.3 Top-5 Mean and Final-Round Radar Graphs

For scores $s_1, \ldots, s_6$ in the plotted generator order—Hunyuan-1.5, MiniMax-H3, Wan-2.1, Seedance-2.0-fast, Gemini Omni Flash, and Wan-3.0—we compute the normalized radar area as

$$A(s_1, \ldots, s_6) = \frac{1}{6} \sum_{i=1}^{6} s_i s_{i+1}, \qquad s_7 = s_1.$$

Thus six scores of 10 give an area of 100. Because adjacent scores are multiplied, the area depends on the generator axis order.

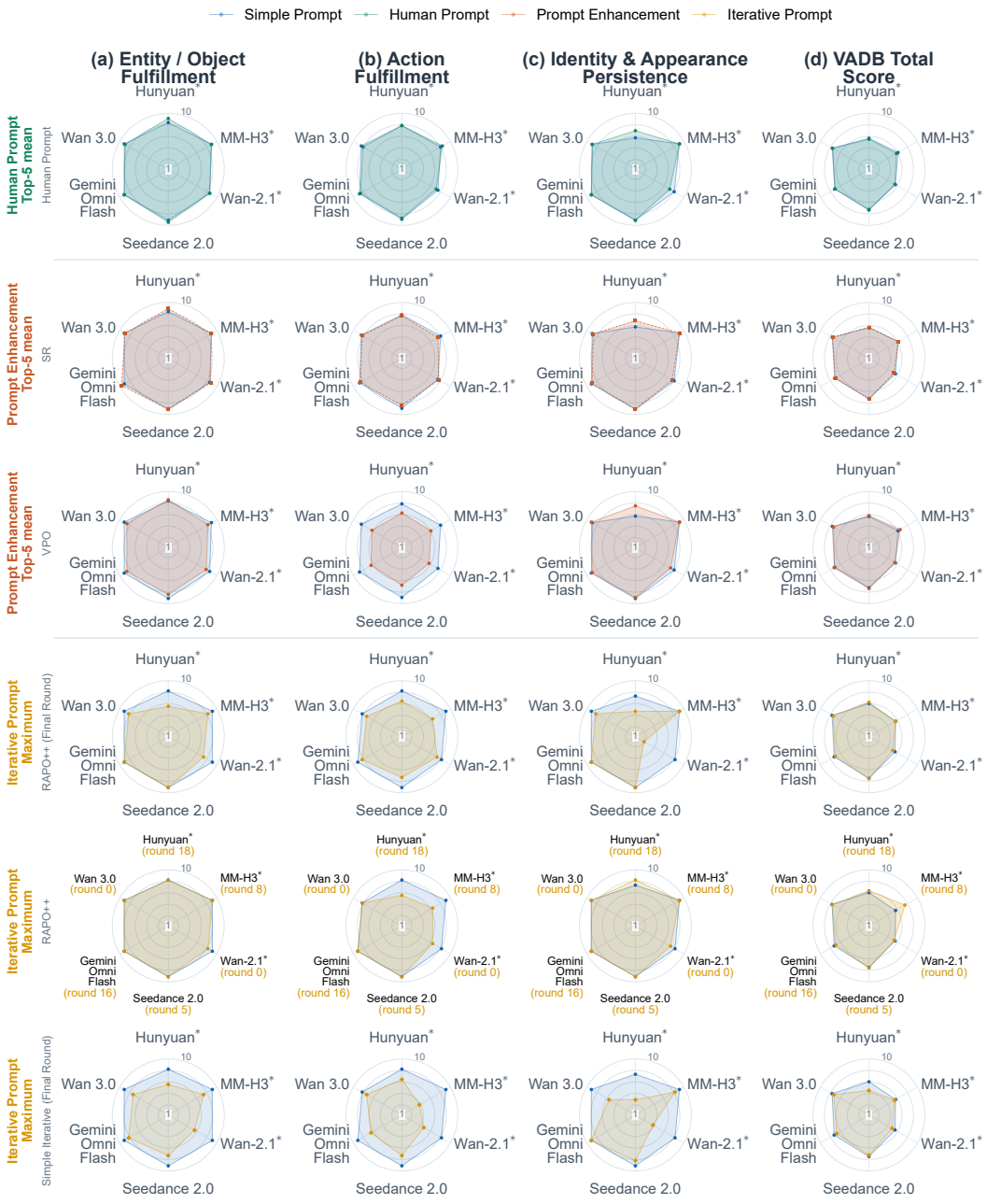

Figure 19: **Additional radar comparisons across four evaluation dimensions.** From top to bottom, the rows show Human Prompt, Simple Rewrite, and VPO Top-5 Mean; RAPO++ Final Round and Maximum; and Simple Iterative Final Round. Each row compares with Simple Prompt under the same selection rule: Top-5 Mean against Top-5 Mean, and Maximum or Final Round against Maximum. Top-5 Mean averages the five videos with the highest sums of four raw scores within each method–generator pool; Maximum uses the highest-scoring video, and Final Round is the round-19 video. The same selected videos provide all four dimension scores. Round labels identify the selected RAPO++ Maximum videos; asterisks mark open-source generators.



Figure 20: **Open-source prompt-enhancement comparisons.** Triangular radars compare RAPO and SCMAPR with Simple Prompt (SP) on Hunyuan-1.5, MiniMax-H3 (MM-H3), and Wan-2.1. The upper two rows show Top-5 Mean; the lower two show Maximum. Within each method–generator pool, videos are ranked by the sum of their three MLLM scores and VADB total score. Top-5 Mean averages the five highest-ranked videos, while Maximum uses the highest-ranked video; the same selection supplies all four displayed scores. Numbers below each radar give its area as a percentage of the full score-10 triangle.

C Task Specification

C.1 Task Inputs

All methods and generators in the principal comparison use the same reference-conditioned task. The fixed Simple Prompt used to define that task is reproduced verbatim below.

Fixed Simple Prompt supplied to the generators (verbatim).

```
Reference Images:
Image 1 shows char1, a thin and stern man in a dark trench coat with short hair and a sharp
↪  gaze.
Image 2 shows char2, a bald man wearing glasses and a one-piece workwear jacket, with a gentle
↪  but serious look.
Image 3 shows char2's home, a rundown dimly lit interior with two connected rooms.
In this scene, char1 takes a small scanner from his pocket and activates it, drawing char2's
↪  attention. char2 suddenly attacks with a scalpel. char1 dodges, disarms him, and the two
↪  engage in a violent close-quarters struggle. char2 overpowers char1, pins him against the
↪  wall, and repeatedly slams him into it until the wall breaks and both of them crash into
↪  the adjacent room.
```



Figure 21: The two character references and the scene reference used in video generation.

Table 5: Common task inputs and their use. "Generation" means that the item was passed to the video-generation interface; it does not imply that every prompt-rewriting method directly consumed the image files.

Input	Fixed content	Use
Simple Prompt	Fixed text printed above, including scanner, action, camera, and appearance specifications	Baseline generations; common starting text and semantic evaluation target
Character references 1–2	Appearance and clothing of each protagonist	Generation; identity assessment
Scene reference	Rundown, dimly lit, connected rooms	Generation; identity/scene assessment

```
The camera is static at first, then slowly pushes forward as char2 gains control and smashes
↪ char1 into the wall. The scene is dark and cool-toned, with strong side-top lighting,
↪ heavy shadows, rough wall textures, realistic dust and debris, and grounded material
↪ detail on skin, clothing, and metal objects.
```

The fixed Simple Prompt specifies the following event order: (1) character 1 activates the scanner; (2) character 2 attacks with a scalpel; (3) character 1 dodges and disarms him; (4) they struggle at close range; (5) character 2 pins and repeatedly slams character 1 into the wall; and (6) the wall breaks and both reach the adjacent room. Table 5 summarizes the inputs, and Figure 21 shows the visual references.

C.2 Video Allocation and Candidate Pools

The principal comparison contains 48 method–generator conditions, each with an $N = 20$ video candidate pool, for 960 scored videos in total. Videos used for preview or iterative feedback count toward the corresponding pool. Table 6 gives the allocation for each method.

The shared initial video is counted once within each condition's $N = 20$ pool. In Human Prompt, this is the Simple Prompt video that the experts could inspect before writing their prompts. For each iterative method, that initial video is round 0; rounds 1–19 each contribute one video. The video generated at a step also supplies feedback for the next prompt revision.

Within a generator, the official PE setting was held fixed across the principal method conditions, as reported in Table 2. This setting does not add videos to any $N = 20$ candidate pool.

C.3 Prompt-Refinement Method Implementations

Implemented procedures and departures. Table 7 summarizes the implemented procedures and their scope. The model identifiers and feedback access follow Table 1.

The three iterative implementations share one schedule. Let p_0 be the fixed Simple Prompt and v_0 its initial video. For $t = 0, \ldots, 18$, the method uses feedback on v_t to form p_{t+1} and generates one video v_{t+1}. Thus each pool contains $v_0, \ldots, v_{19}$; the methods differ in how they obtain feedback.

Table 6: Allocation of the $N = 20$ videos in each method–generator condition. The initial video in the human and iterative conditions is drawn from the corresponding Simple Prompt pool. RAPO and SCMAPR were evaluated on the three open-source generators; the other methods were evaluated on all six.

Method	Video allocation	N
Simple Prompt	20 videos generated from one fixed prompt	20
Human Prompt	1 video from Simple Prompt + 9 from expert prompt 1 + 10 from expert prompt 2	20
Simple Rewrite	20 prompt–video pairs	20
RAPO	20 prompt–video pairs	20
SCMAPR	20 prompt–video pairs	20
VPO	20 prompt–video pairs	20
Simple Iterative	1 video from Simple Prompt + 1 video at each of 19 iterative steps	20
RAPO++	1 video from Simple Prompt + 1 video at each of 19 iterative steps	20
VQQA	1 video from Simple Prompt + 1 video at each of 19 iterative steps	20

Table 7: Implemented prompt procedures and departures from the source methods. "No video" refers to the prompt-revision step, not the subsequent video-generation evaluation.

Method	Revision component	Actual procedure and scope
Simple Rewrite	Gemini-3.7-Flash	Constrained one-shot rewrite without generated-video feedback.
RAPO	Mistral-7B-Instruct-v0.3	The formal video prompts came from the instruction-rewrite branch; the evaluated condition covers that branch of RAPO.
SCMAPR	DeepSeek-V4-Flash; BGE-M3 embeddings	Scenario routing, policy-conditioned refinement, and semantic verification, without generated-video feedback in the revision loop.
VPO	Released `CCCCCC/VPO-5B` optimizer	Pretrained prompt optimizer used at inference to produce saved variants. This study did not train the optimizer or rerun its supervised/preference-learning stages.
Simple Iterative	Gemini-3.7-Flash	At each of 19 steps, inspect one generated video and revise the prompt using video feedback.
RAPO++	Qwen2.5-VL-7B; Qwen2.5-7B	Reproduction of the SSPO test-time stage. At each of 19 steps, diagnose one generated video and revise the prompt.
VQQA	Gemini-3.1-Pro-Preview	Reproduction of the question-generation, question-answering, and prompt-refinement loop, using one generated video at each of 19 steps.

Simple Iterative: critique the current video's content, motion, and visual quality, then update the prompt while preserving the requested scene.

RAPO++ / SSPO: diagnose defects in the current video and use that diagnosis in the SSPO test-time prompt update.

VQQA: generate questions about the current video, answer them against that video, and update the prompt using the resulting feedback.

Rounds 0–4 of a 19-step RAPO++ trajectory on Hunyuan-1.5. The complete condition contains round 0 and rounds 1–19. The excerpt below shows only the first five prompt states, p_0–p_4; rounds 5–19 are part of the same iterative condition but are not reproduced here. Round 0 is the fixed Simple Prompt printed above. Each subsequent round uses feedback from one video to produce the next prompt. The text of p_1–p_4 is reproduced below.

Round 1 prompt (p_1; verbatim).

```
In this scene, char1 takes a small scanner from his pocket and activates it, drawing char2's
↪ attention. char2 suddenly attacks with a scalpel. char1 dodges, disarms him, and the two
↪ engage in a violent close-quarters struggle. char2 overpowers char1, pins him against the
↪ wall, and repeatedly slams him into it until the wall breaks and both of them crash into
↪ the adjacent room. The camera remains in one continuous uninterrupted shot in one
↪ continuous space, capturing the entire struggle without any cuts or jumps. The scene is
↪ dark and cool-toned, with strong side-top lighting, heavy shadows, rough wall textures,
↪ realistic dust and debris, and grounded material detail on skin, clothing, and metal
↪ objects.
```

Round 2 prompt (p_2; verbatim).

```
In this scene, char1 takes a small scanner from his pocket and activates it, drawing char2's
↪  attention. char2 suddenly attacks with a scalpel. char1 dodges, disarms him, and the two
↪  engage in a violent close-quarters struggle. As the struggle intensifies, the wall behind
↪  them begins to crack under the force of their impact until it finally breaks, sending both
↪  of them crashing into the adjacent room. The camera remains in one continuous
↪  uninterrupted shot in one continuous space, capturing the entire struggle without any cuts
↪  or jumps. The scene is dark and cool-toned, with strong side-top lighting, heavy shadows,
↪  rough wall textures, realistic dust and debris, and grounded material detail on skin,
↪  clothing, and metal objects. The transition from the initial interaction to the violent
↪  struggle should be smoother, blending the calm before the storm with the ensuing chaos to
↪  maintain a seamless and intense viewing experience.
```

Round 3 prompt (p_3; verbatim).

```
In this scene, char1 takes a small scanner from his pocket and activates it, drawing char2's
↪  attention. To maintain a seamless and intense viewing experience, ensure the transition
↪  from the initial interaction to the violent struggle is smoother, blending the calm before
↪  the storm with the ensuing chaos. The two engage in a violent close-quarters struggle as
↪  the wall behind them cracks under the force of their impact until it finally breaks,
↪  sending both of them crashing into the adjacent room. The camera remains in one continuous
↪  uninterrupted shot in one continuous space, capturing the entire struggle without any cuts
↪  or jumps. The scene is dark and cool-toned, with strong side-top lighting, heavy shadows,
↪  rough wall textures, realistic dust and debris, and grounded material detail on skin,
↪  clothing, and metal objects. The transition from the initial interaction to the violent
↪  struggle should be smoother, blending the calm before the storm with the ensuing chaos to
↪  maintain a seamless and intense viewing experience.
```

Round 4 prompt (p_4; verbatim).

```
In one continuous uninterrupted shot, in one continuous space, char1 takes a small scanner
↪  from his pocket and activates it, drawing char2's attention. Ensure the transition from
↪  the initial interaction to the violent struggle is smoother, blending the calm before the
↪  storm with the ensuing chaos to maintain a seamless and intense viewing experience. The
↪  two engage in a violent close-quarters struggle as the wall behind them cracks under the
↪  force of their impact until it finally breaks, sending both of them crashing into the
↪  adjacent room. The camera remains fixed, capturing the entire struggle without any cuts or
↪  jumps. The scene is dark and cool-toned, with strong side-top lighting, heavy shadows,
↪  rough wall textures, realistic dust and debris, and grounded material detail on skin,
↪  clothing, and metal objects.
```

Within this excerpt, p_3 and p_4 omit the explicit scalpel attack, dodge, and disarm from p_0. This example shows how an individual revision can change action specificity; the method-level comparisons use the complete candidate pools.

D QUALITATIVE VIDEO EXAMPLES



Figure 22: **Qualitative comparisons across prompt-refinement methods and generators.** (a,b) Simple Prompt sample 5 and SCMAPR sample 20 on Hunyuan-1.5. (c,d) Simple Prompt sample 7 and VPO sample 14 on MiniMax-H3. (e,f) Simple Prompt sample 3 and RAPO sample 2 on Wan-2.1. Each row shows a distinct video, with six frames progressing from left to right in video time. These selected examples illustrate individual outcomes rather than method-level averages.

On Hunyuan-1.5, the Simple Prompt video begins with one clearly visible protagonist and moves into fragmented close-up action, without a clear two-person wall breach (Figure 22(a)). The SCMAPR video establishes both characters and proceeds from confrontation to an impact at the wall (Figure 22(b)). Their Entity/Action/Identity scores are $5/2/3$ and $9/8/8$, although the latter has a lower VADB visual score (4.91 versus 5.39).

On MiniMax-H3, Simple Prompt shows the scanner, attack, and struggle progressing toward the broken wall (Figure 22(c)). VPO holds the characters in largely static positions for much of the sampled sequence and does not show a comparably clear breach (Figure 22(d)). Action Fulfillment falls from 9 to 4, while VADB rises from 5.06 to 6.58, illustrating a semantic-alignment and visual-quality tradeoff in these selected videos.

On Wan-2.1, Simple Prompt retains the two-character encounter and reaches a visible wall impact (Figure 22(e)). RAPO turns into reference-sheet-like character images in its final sampled frames, breaking scene continuity (Figure 22(f)). Their Entity/Action/Identity scores are $9/8/8$ and $5/4/2$.



Figure 23: **Non-monotonic outcomes across VQQA refinement rounds.** (a,b) Rounds 9 and 12 on Seedance-2.0-fast. (c,d) Rounds 11 and 13 on Gemini Omni Flash. (e,f) Rounds 9 and 16 on Wan-3.0. Each row shows a distinct video, with six frames progressing from left to right in video time. These selected pairs illustrate trajectories rather than average changes across rounds.

On Seedance-2.0-fast, VQQA round 9 shows the scanner, confrontation, wall breach, and landing in the adjacent room (Figure 23(a)). Round 12 still reaches the adjacent room, but a long metal object occupies several frames and the intended fight sequence is less clearly expressed (Figure 23(b)). Action Fulfillment decreases from 9 to 5, while VADB changes only slightly, from 6.98 to 7.05.

On Gemini Omni Flash, round 11 follows a two-character confrontation through the wall breach (Figure 23(c)). In round 13, an additional figure appears near the doorway early in the clip, making the cast and roles less stable (Figure 23(d)). Identity/Appearance Persistence drops from 9 to 1, although the VADB scores are nearly identical (6.8448 and 6.8450).

On Wan-3.0, round 9 maintains a legible two-character sequence through the breach (Figure 23(e)). Round 16 loses clear two-character continuity after the impact; its final sampled frame centers on a single figure (Figure 23(f)). Identity/Appearance Persistence falls from 9 to 1 and Action Fulfillment from 9 to 6, despite a VADB increase from 6.91 to 7.27. Together, these examples show that later refinement rounds do not consistently improve semantic alignment, and a higher visual score can accompany weaker fulfillment of the requested action or characters.