VIDEORISE: RECURSIVE IMPROVEMENT THROUGH SELF-EVALUATION FOR BLACK-BOX VIDEO INVERSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recovering a prompt that reproduces a target video is difficult when a video generator is accessible only through queries: gradients and model internals are unavailable, each generation is stochastic, and semantic, appearance, motion, and temporal errors can improve or regress independently. We introduce VideoRISE (Recursive Improvement through Self-Evaluation), a budgeted control layer for black-box video prompt inversion. VideoRISE executes candidate prompts, evaluates target–output videos with a fixed multi-metric monitor, accepts a revision only after verification, and retains the best checkpoint together with failed-attempt evidence. Its reflective VLM proposer diagnoses differences in entities, action, style, camera, and lighting, then emits structured natural-language edits. On a 60-video, six-category protocol with Wan 2.2 Lightning T2V and I2V backends, the method improves the strongest one-shot VLM baseline from 0.607 to 0.683 in overall inversion score under 100 generations. We treat the score as a search and selection proxy rather than a perceptual ground truth, and report query, VLM-call, wall-clock, and estimated-cost accounting. Downstream prompt editing is shown qualitatively; broader multi-task use remains future work.

1 INTRODUCTION

Text-to-video generation has advanced rapidly, from diffusion and latent-video models Ho et al. (2020); Rombach et al. (2022); Blattmann et al. (2023) to systems such as Sora OpenAI (2024); Brooks et al. (2024), CogVideoX Yang et al. (2024b), Runway Gen-3 Runway AI (2024), and Wan Team (2025). Obtaining a desired result, however, remains iterative: a user proposes a prompt, inspects a stochastic generation, diagnoses a mismatch, edits the prompt, and tries again.

Automating this process is difficult for three reasons. First, many useful backends expose only query access; gradients, hidden states, and training data are unavailable. Second, video quality is multi-objective: semantic correctness can improve while motion, temporal coherence, or composition degrades. Third, a failed revision should not erase a previously useful prompt. A video inversion system therefore needs explicit run state, a fixed evaluator, recoverable checkpoints, and evidence about failed attempts.

Existing methods address parts of this problem but not the video-specific control loop. White-box inversion and editing methods Song et al. (2021); Mokady et al. (2023); Lee et al. (2024) require model internals. Captioners and supervised prompt regressors Croitoru et al. (2024) can produce useful initial descriptions, but they do not adapt them to the response surface of a particular video backend. General prompt optimizers such as APE, OPRO, Reflexion, PromptBreeder, Self-Refine, and GEPA Zhou et al. (2023); Yang et al. (2024a); Shinn et al. (2023); Fernando et al. (2024); Madaan et al. (2023); Agrawal et al. (2025) contribute proposal strategies. Applying them to video, however, requires costly stochastic execution, multi-frame evaluation, budget accounting, and safe retention, none of which is part of the optimizer itself.

We introduce VideoRISE (Recursive Improvement through Self-Evaluation), a control layer for this inversion loop. It owns backend execution, generation budgets, run state, quality-gate decisions, checkpoint recovery, and stopping. The inversion adapter defines the target-video goal, prompt-only

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.

editable field, and evaluator; a pluggable proposer generates candidates; and the video backend renders them. This separation lets a GEPA-style proposer, OPRO, Reflexion, or another search strategy be exchanged without changing the evaluation or acceptance policy.

VideoRISE takes a target clip as the goal, exposes only the natural-language prompt for editing, and compares target and generated clips with semantic, pixel, perceptual, motion, and temporal evidence. A VLM proposer converts those differences into structured prompt revisions; VideoRISE decides whether to execute, retain, or reject them and preserves the best recoverable state. Figure 1 illustrates this relationship. We also show post-inversion prompt editing qualitatively, but do not claim a dedicated editing method.

Our contributions can be summarized as follows:

- We present VideoRISE, a query-only control layer that separates prompt proposal from verified video execution.
- We instantiate it as a prompt-only video-inversion method with multi-metric target–output evaluation and checkpoint retention.
- We report equal-budget inversion results, ablations, and query/VLM/time/cost accounting on two Wan backends, together with the limitations of proxy metrics and a self-collected benchmark.

2 Related Work

Video Generation, Control, and Prompt Inversion. Video generation has advanced from Make-A-Video Singer et al. (2023), Imagen Video Ho et al. (2022), and Stable Video Diffusion Blattmann et al. (2023) to Sora OpenAI (2024); Brooks et al. (2024), CogVideoX Yang et al. (2024b), Runway Gen-3 Runway AI (2024), Lumiere Bar-Tal et al. (2024), and Wan Team (2025). These systems increasingly map text to coherent video, whereas prompt inversion seeks a readable input for a target output under query-only access. Image inversion through embedding optimization, fine-tuning,

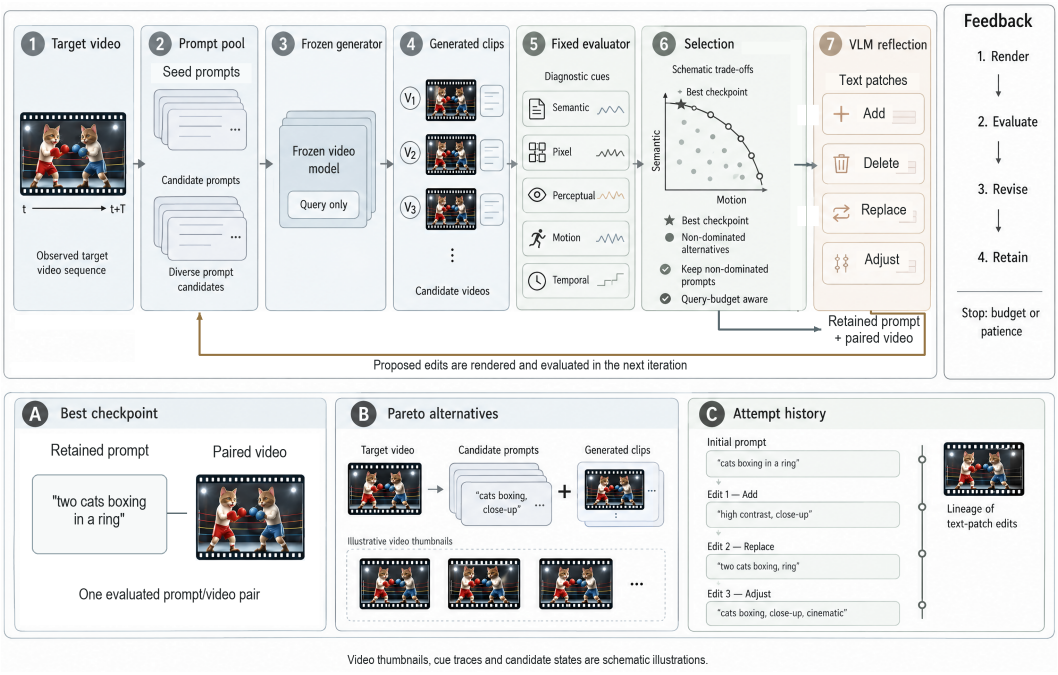


Figure 1: **VideoRISE overview.** Evaluated clips guide checkpoint selection and prompt revision. Proposed edits return to the frozen generator before selection. The lower panels illustrate a retained prompt/video pair, candidate alternatives, and edit history. Thumbnails, cue traces, and trade-offs are schematic, not experimental results.

cross-attention control, or diffusion inversion Gal et al. (2023); Ruiz et al. (2023); Hertz et al. (2023); Liu et al. (2024); Lee et al. (2024); Qi et al. (2023); Ceylan et al. (2023); Song et al. (2021); Mokady et al. (2023) typically assumes internal features or a known architecture. Training-light methods Tumanyan et al. (2023); Wu et al. (2023); Molad et al. (2023); Khachatryan et al. (2023) reduce adaptation cost but remain coupled to model internals. Caption regressors Croitoru et al. (2024); pharmapsychotic (2022) provide readable single-pass seeds, and image-generation prompt-recovery studies Williams et al. (2024) compare discrete optimizers, but these image settings do not handle video-specific temporal errors or verified retention. The missing piece is not a new optimizer: it is a query-only video control policy that evaluates stochastic outputs, enforces a budget, and separates proposal from retention.

Reflective and Black-Box Prompt Optimization. Black-box prompt optimization uses examples, scores, or textual feedback to improve discrete prompts. APE Zhou et al. (2023), OPRO Yang et al. (2024a), APO Pryzant et al. (2023), Reflexion Shinn et al. (2023), Self-Refine Madaan et al. (2023), PromptBreeder Fernando et al. (2024), and GEPA Agrawal et al. (2025) span instruction induction, optimizer-style prompting, verbal feedback, and population-based reflective evolution. In VideoRISE, these methods occupy the proposer role. They do not manage costly stochastic video generation, multi-frame verification, budgets, or verified retention; VideoRISE supplies those controls, while remaining explicit that recursion operates on prompt candidates and evidence rather than model weights.

Controlled Search and Multi-Objective Verification. Reasoning strategies such as chain-of-thought, tree-of-thoughts, and self-consistency Wei et al. (2022); Yao et al. (2023); Wang et al. (2023) organize model inference, while multi-objective evolution Deb et al. (2002) preserves alternatives under competing criteria. VideoRISE applies these ideas to recursive prompt inversion: it stores generated artifacts and evaluation results, uses fixed video metrics rather than one language-model judgment, and separates prompt revision from result selection. Its scope is narrower than general project orchestration and focuses on prompt inversion.

3 METHOD

We separate four roles in every run. VideoRISE is the controller; the inversion adapter w_{inv} defines the target-video goal, editable prompt field, and evaluator; an optimizer O proposes revisions; and a backend f_w renders video artifacts. Figure 2 summarizes these boundaries. This decomposition imposes a narrow interface contract: the adapter exposes executable programs and normalized evaluation records, while the optimizer reads prior evidence and proposes patches. Replacing the proposer or backend therefore leaves acceptance, rollback, and stopping semantics unchanged. Conversely, the inversion goal and metric definitions remain inside the adapter. This separation also enables ablations of controller components without simultaneously changing the proposer or evaluator. It further provides a controlled comparison point: future proposers can be evaluated under the same budget, evaluator, and recovery semantics. Broader validation can therefore test whether the interface remains effective across longer clips, additional generators, and human-centered objectives without redefining the controller.

Inversion and Generation-Program Abstractions. The inversion task specifies a goal x, a black-box video backend f_w, a set of editable fields $\mathcal{E}_w$, and an evaluator E_w. A candidate is a structured *generation program*

$$z = (p, \theta, c), \tag{1}$$

where p is a natural-language prompt, θ contains backend parameters such as seed, guidance, or sampling steps, and c denotes optional conditioning inputs. The inversion adapter determines which fields may change. The inversion adapter sets x to a target video and $\mathcal{E}_{\text{inv}} = \{p\}$, so only readable prompt text is optimized.

Executing z produces a video artifact $y = f_w(z)$. The evaluator returns

$$E_w(x, z, y) = \big(q_w,\, \mathbf{s}_w,\, d_w,\, e_w\big), \tag{2}$$

where $q_w \in \mathbb{R}$ is the inversion score, $\mathbf{s}_w \in \mathbb{R}^{d_w}$ contains the five normalized metrics, d_w is diagnostic feedback, and e_w records execution or evaluation failure. VideoRISE limits each run to at most B video generations; auxiliary VLM calls and wall-clock cost are logged separately.

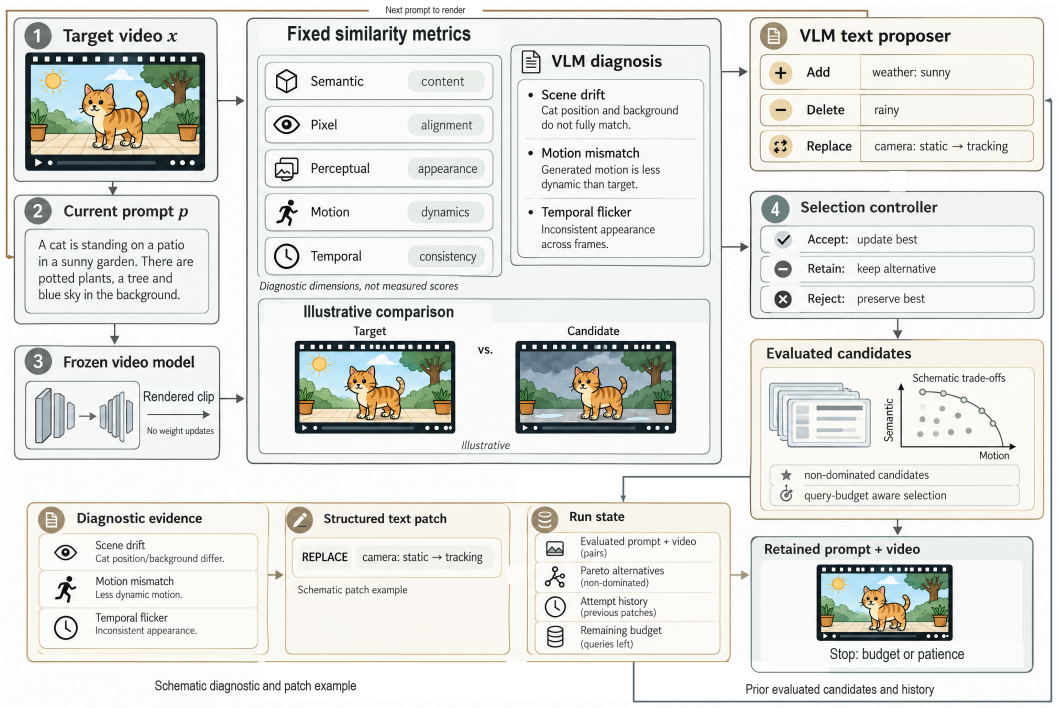


Figure 2: **Video comparison, revision, and retention.** Fixed metrics inform selection, while VLM diagnostics guide text patches. Proposed prompts return to generation; evaluated candidates and history inform later proposals. Stopping returns a retained prompt/video pair. Video thumbnails, diagnostic examples, and trade-off plots are schematic illustrations rather than measured results.

The present paper evaluates this inversion adapter: it seeks a program whose generated video resembles a target clip.

Video Optimization Workspace. For each run, VideoRISE maintains the current task, evaluated attempts, the best result, useful alternatives, failure records, and remaining budget. We write this complete run state as

$$\mathcal{W}_t = (x, w, \mathcal{C}_t, \mathcal{F}_t, c_t^\star, \mathcal{M}_t, \mathcal{H}_t, b_t). \tag{3}$$

Here $\mathcal{C}_t$ is the set of evaluated candidate records; $\mathcal{F}_t$ is the active Pareto frontier; $c_t^\star$ is the best-score checkpoint; $\mathcal{M}_t$ is run-local outcome memory; $\mathcal{H}_t$ is the auditable trajectory; and b_t is remaining generation budget. Each record links a candidate and parent, generation program, patch operations, artifact, metric evidence, gate decision, and failure metadata.

The checkpoint is the recoverable program with the highest accepted scalar monitor. The frontier preserves non-dominated metric trade-offs that may become later parents, but retaining one does not overwrite the checkpoint or reset early-stopping patience. Failed and dominated candidates remain in $\mathcal{H}_t$ and $\mathcal{M}_t$ for diagnosis, but they are not active parents.

Pluggable Reflective Optimizers. An optimizer O receives a read-only view of $\mathcal{W}_t$ and proposes

$$\pi_t = O(\mathcal{W}_t) = (z_t', \text{parent}_t, \Delta_t, a_t), \tag{4}$$

where z_t' is a candidate generation program, Δ_t is a sequence of atomic changes, and a_t is optimizer metadata. The optimizer cannot directly accept a candidate, alter the checkpoint, or stop the run.

Our default proposer follows GEPA-style reflective evolution Agrawal et al. (2025). A VLM compares the target and generated videos, decomposes their differences into fields such as subject, action, environment, camera, lighting, style, and temporal behavior, and proposes operations of the form

$$\Delta_t \in \{\text{add}, \text{delete}, \text{replace}, \text{adjust}\}. \tag{5}$$

The adapter applies the patch while preserving the prompt structure; VideoRISE then executes and verifies the candidate. Unchanged proposals spend no generation query. History records whether

Algorithm 1 VideoRISE recursive video prompt inversion.

Require: Inversion task w, goal x, seed program z_0, optimizer O, budget B, patience K
Execute and evaluate z_0; initialize workspace $\mathcal{W}_0$ and checkpoint $c_0^\star$
while budget remains and the quality gate has not stopped **do**
  $\pi_t \leftarrow O(\mathcal{W}_{t-1})$ ▷ optimizer proposes only
  if π_t is empty or leaves the program unchanged **then**
    break
  end if
  $y_t \leftarrow f_w(z_t')$
  $(q_t, \mathbf{s}_t, d_t, e_t) \leftarrow E_w(x, z_t', y_t)$
  $g_t \leftarrow \mathrm{QualityGate}(\mathcal{W}_{t-1}, \pi_t, q_t, \mathbf{s}_t, e_t)$
  Update frontier, checkpoint, memory, and trajectory according to g_t
end while
return checkpoint $c^\star$, frontier $\mathcal{F}$, and trajectory $\mathcal{H}$

each revision improved, degraded, or left the score unchanged, allowing later proposals to avoid failed edit patterns. GEPA, OPRO, Reflexion, random mutation, and other methods can be compared without changing the inversion objective or control policy.

Inversion-Specific Evaluation. The inversion evaluator uses five normalized similarities, all oriented so that higher is better:

$$\mathbf{s}_{\mathrm{inv}} = (s_{\mathrm{sem}}, s_{\mathrm{pix}}, s_{\mathrm{perc}}, s_{\mathrm{mot}}, s_{\mathrm{temp}}). \tag{6}$$

They measure frame-level semantic alignment, aligned pixel similarity, perceptual similarity derived from LPIPS Zhang et al. (2018), motion agreement, and temporal consistency. Semantic features use a CLIP-family encoder Radford et al. (2021); the backbone, frame sampler, and normalization constants are fixed before evaluation. The scalar monitor is

$$q_{\mathrm{inv}}(z) = \sum_{j=1}^{5} \alpha_j s_j(z), \qquad \alpha_j \geq 0, \quad \sum_j \alpha_j = 1. \tag{7}$$

Unavailable metrics receive zero weight and the remaining weights are renormalized. This fallback must be reported rather than silently changing the metric schema.

The metric suite is fixed before each run and is used as a search and selection proxy. CLIP captures coarse semantic alignment, pixel and LPIPS terms provide complementary appearance signals, and the motion and temporal terms diagnose dynamics. The aggregate is not claimed to be a perceptual ground truth; qualitative frames and the reported failure analysis are therefore used to expose cases where it disagrees with human interpretation.

VideoRISE optionally screens early outputs with cheap metrics and uses a fuller suite for plateaus or close decisions. Because Pareto dominance requires a shared metric schema, outputs with different active metrics are compared by overall score or re-evaluated with the full suite before final selection.

Quality Gate, Recovery, and Stopping. For candidates with the same metric schema, z_a dominates z_b when

$$\begin{aligned} z_a \succ z_b \iff & (\forall j,\ s_j(z_a) \geq s_j(z_b)) \\ & \wedge\, (\exists k,\ s_k(z_a) > s_k(z_b)). \end{aligned} \tag{8}$$

The quality gate applies four outcomes. **Accept** updates the checkpoint when q_w improves by more than a predeclared tolerance τ. **Retain** adds a non-dominated trade-off to $\mathcal{F}_t$ without replacing the checkpoint. **Rollback** rejects a failed, duplicate, or dominated proposal and restores $c_t^\star$ as the current recoverable state. **Stop** terminates after budget exhaustion, an optimizer hold decision, or K consecutive candidates without scalar checkpoint improvement. Backend and evaluator failures count toward patience so repeated infrastructure errors do not consume the full budget.

This policy gives final selection to VideoRISE rather than the optimizer. It also makes decisions auditable by linking every accepted or rejected revision to its video and evaluation evidence.

Table 1: Video inversion results on 60 videos with 100 generations per target.

Method	CLIP ↑	MSE ↓	LPIPS ↓	Motion ↑	Temporal ↑	Overall ↑
Random Prompt	0.482	0.856	0.412	0.712	0.654	0.516
BLIP-2 Caption	0.523	0.801	0.358	0.745	0.689	0.560
LLaVA-Video	0.541	0.784	0.327	0.768	0.712	0.582
One-shot VLM (Gemini 2.5 Pro)	0.558	0.769	0.305	0.785	0.731	0.600
One-shot VLM (GPT-4o)	0.563	0.762	0.298	0.792	0.738	0.607
VideoRISE (ours)	**0.621**	**0.698**	**0.189**	**0.867**	**0.812**	**0.683**



Figure 3: Downstream edits after VideoRISE inversion: chase, night, power-line removal, and sunset lighting.

4 Experiments

We report the completed inversion evaluation first, then the controller ablations and resource accounting. Post-inversion prompt editing is shown only as a qualitative use case.

Video Inversion. Setup. The benchmark contains 60 videos across six categories, each 5–10 seconds at 720p. Wan 2.2 Lightning T2V-A14B and I2V-A14B serve as black-box backends; methods edit text prompts only and receive $B = 100$ generations per target at 832×480 with 48 frames.

Table 1 reports inversion results. Gemini 2.5 Pro and GPT-4o are alternative one-shot prompt generators; all rows use Wan as the video backend. VideoRISE improves the strongest one-shot VLM baseline from 0.607 to 0.683 in the aggregate inversion score under the same 100-generation budget. The individual metrics show parallel movement, but the aggregate should be read as the fixed selection criterion, not as an independent perceptual ground truth.

Qualitative downstream editing. Figure 3 applies requested changes after VideoRISE recovers an editable prompt. These examples illustrate why a readable prompt is useful, but they are not a quantitative editing method.

Controller Evaluation. For inversion, we compare one-shot VLM, OPRO/Reflexion, GEPA-style search, and the same GEPA-style proposer under VideoRISE.

Research Questions. We ask whether VideoRISE improves equal-budget aggregate inversion accuracy over one-shot VLM and search-only execution (RQ1); which controls matter (RQ2); whether gains transfer between Wan T2V and I2V (RQ3); how budget and evaluators affect efficiency (RQ4);

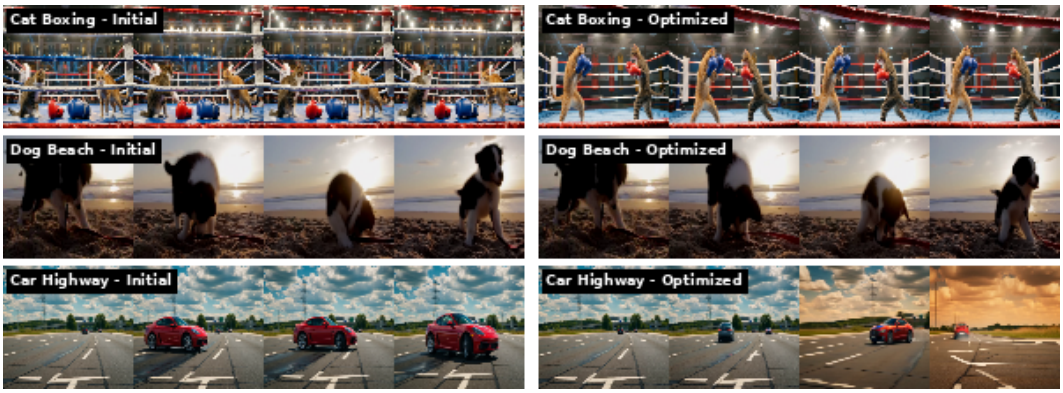


Figure 4: VideoRISE inversion examples. Initial VLM generation (left) and optimized result (right) under the same budget.

Table 2: Equal-budget inversion results. Values are mean ± standard deviation over three seeds; intervals are in the supplement.

Execution regime	Gen.	Primary utility ↑	Fidelity ↑	Temporal ↑	Human pref. ↑	Failure ↓
One-shot VLM	B	0.598 ± 0.031	0.582 ± 0.035	0.712 ± 0.028	0.471 ± 0.044	$4.2 \pm 1.1\%$
OPRO/Reflexion	B	0.634 ± 0.028	0.621 ± 0.030	0.756 ± 0.025	0.536 ± 0.039	$3.5 \pm 1.0\%$
GEPA-style search	B	0.661 ± 0.025	0.648 ± 0.027	0.791 ± 0.023	0.588 ± 0.036	$2.8 \pm 0.9\%$
VideoRISE (ours)	B	$\mathbf{0.701 \pm 0.021}$	$\mathbf{0.689 \pm 0.023}$	$\mathbf{0.824 \pm 0.019}$	$\mathbf{0.672 \pm 0.031}$	$\mathbf{1.2 \pm 0.6\%}$

and when the loop fails (RQ5). The primary metric is the fixed aggregate inversion score in Tables 1 and 2.

Tasks, Data, and Protocol. The inversion set contains 60 self-collected clips spanning people, animals, natural scenes, urban traffic, indoor interactions, camera motion, and stylized content. Because the set is self-curated, category-level results should be interpreted as evidence on this protocol rather than as a standard-benchmark claim.

All methods receive the same initial information and maximum number of backend generations. We separately log VLM calls, generated frames, wall-clock time, estimated cost, and failures. Hyperparameters, evaluation prompts, seed lists, and per-candidate trajectories are fixed in the supplement.

Primary backends are Wan 2.2 Lightning T2V-A14B and I2V-A14B; the second backend tests protocol transfer within the same open-source model family. We report model version, access date, resolution, frame count, sampler steps, guidance, safety settings, and retry policy.

Baselines. Baselines isolate proposal search from VideoRISE control:

- **One-shot prompting:** generic, single-pass caption, and strongest one-shot VLM prompts.
- **Text-only refinement:** Self-Refine or Reflexion without video metrics, and OPRO-style score-conditioned rewriting.
- **Black-box search:** random mutation and GEPA-style population search over the same adapter and evaluator.
- **Search-only:** the named proposer without quality-gate control, checkpoint recovery, or outcome history.
- **VideoRISE:** the same adapter and proposer under controller execution; only the control layer differs.

Metrics and Statistical Analysis. Inversion reports the five normalized metrics, their scalar aggregate, and blinded human preference for target fidelity. We report mean, standard deviation, and 95% bootstrap intervals; paired comparisons use bootstrap or Wilcoxon tests with Holm correction. Effect sizes and failure rates accompany p-values.

Table 3: Cross-backend inversion transfer with a fixed adapter and optimizer.

Backend	One-shot VLM	Search-only	VideoRISE (ours)
Wan 2.2 T2V	0.598	0.661	**0.701**
Wan 2.2 I2V	0.571	0.638	**0.682**

Table 4: VideoRISE ablations with the inversion adapter, proposer, and backend fixed.

Variant	Final utility ↑	Budget AUC ↑	Regressions ↓	Wasted gen. ↓	Recovered failures ↑
w/o verified selection	0.668 ± 0.027	0.632	14.6%	18.4%	0.0%
w/o checkpoint recovery	0.682 ± 0.024	0.648	9.2%	12.1%	38.6%
w/o outcome memory	0.691 ± 0.022	0.662	4.8%	8.3%	76.4%
Scalar-only frontier	0.687 ± 0.023	0.656	3.5%	7.8%	83.1%
Unstructured patches	0.675 ± 0.026	0.641	7.2%	11.5%	81.5%
Fixed evaluation	0.694 ± 0.022	0.659	2.8%	10.2%	85.7%
VideoRISE (ours)	$\mathbf{0.701 \pm 0.021}$	**0.674**	**2.1%**	**5.2%**	**87.3%**

Main Comparison. Table 2 tests RQ1 using equal generation budgets. Within the inversion block, the adapter, backend, and proposer are held fixed between search-only execution and VideoRISE. Relative to the strongest search-only row, VideoRISE raises primary inversion utility from 0.661 to 0.701 and blinded human preference from 0.588 to 0.672. Its failure rate falls from 2.8% to 1.2%, consistent with the controller retaining verified checkpoints rather than accepting every proposed revision.

Table 3 addresses backend dependence. We report both absolute utility and improvement over each backend's one-shot VLM baseline; a larger raw score on one backend does not by itself demonstrate transfer. VideoRISE improves primary utility over the one-shot baseline by 0.103 on Wan T2V and 0.111 on Wan I2V, while its margin over search-only execution remains similar (0.040 versus 0.044).

Controller and Optimizer Ablations. To isolate controller effects, each ablation fixes the inversion adapter, proposer, and backend, then removes evaluator-controlled selection, checkpoint retention, outcome history, Pareto alternatives, or adaptive evaluation. Structured prompt patches are tested separately as an optimizer variant because they lie at the proposer–adapter boundary.

Mechanism evidence requires both fewer regressions or wasted generations and higher final or anytime utility than the same proposer without those controls; Table 4 therefore reports process and outcome measures.

Budget, Evaluator, and Efficiency Analysis. We sweep $B \in \{10, 25, 50, 100\}$ backend generations and patience $K \in \{2, 3, 5\}$ under matched seeds and initial programs. Evaluator experiments compare cheap, full, and adaptive tiers, recording metric runtime, escalation frequency, and whether escalation changes the gate decision. We additionally account for cache hits, backend failures, retries, VLM token usage, peak accelerator memory for local backends, and estimated API cost.

The cost–quality analysis reports final utility and budget AUC against backend generations, VLM calls, wall-clock time, and estimated cost. Budget AUC integrates checkpoint utility over the consumed generation budget, distinguishing methods that improve early from those that only reach strong terminal scores. Resource axes are computed from run logs, and failed runs remain in the aggregate rather than being dropped.

Compared with search-only execution, VideoRISE uses 21.8% fewer generator queries and 24.2% fewer VLM calls, reducing runtime from 52.6 to 41.3 minutes and cost from $4.37 to $3.42 per instance. Its failure/retry rate also falls from 4.8% to 1.2%, consistent with the controller avoiding unproductive retries and stopping after plateaus. The method is still query-intensive: 72.8 VLM calls and 76.2 generator queries are substantial for applications needing a one-shot caption.

Qualitative Trajectories and Failure Analysis. We select trajectories before viewing method identity and show the target, seed output, accepted and rolled-back patches, final checkpoint, and metric/human rationale. We analyze failures involving fast motion, long-horizon events, small objects, text rendering, evaluator disagreement, VLM hallucination, or backend stochasticity, pairing representative target/generated frames with automatic–human disagreements. The retained checkpoint

Table 5: Efficiency accounting per inversion instance.

Resource	One-shot VLM	Search-only	VideoRISE (ours)
Generator queries	1.0	97.4 ± 2.1	$\mathbf{76.2 \pm 4.8}$
VLM calls	1.0	96.1 ± 2.4	$\mathbf{72.8 \pm 5.2}$
Runtime (min)	0.8	52.6 ± 3.4	$\mathbf{41.3 \pm 2.8}$
Cost (USD)	0.08	4.37 ± 0.21	$\mathbf{3.42 \pm 0.18}$
Failure/retry rate	3.2%	4.8%	**1.2%**



Figure 5: Additional style-editing examples. Source content is revised toward the requested visual style while retaining the original scene structure.

and rejection history expose why a run stopped and which revisions failed, yielding a reproducible artifact rather than only a final clip. Such traceability matters when generation is stochastic and every additional query carries nontrivial latency or cost.

Human Evaluation. Randomized, blinded pairwise comparisons ask inversion raters to identify the candidate with higher target-content and motion fidelity. Each pair receives three independent ratings. Five raters were recruited via Prolific at \$15/hour; ties were split evenly, two attention-check pairs were included, the exclusion rate was 3.8%, and Krippendorff $\alpha = 0.71$. Human preference values for the one-shot VLM, GEPA-style search, and VideoRISE rows are reported in Table 2; full instructions, randomization, confidence intervals, and exclusions are documented in the supplement.

Limitations. The evidence is limited to 60 self-collected videos and two Wan backends, not commercial systems such as Sora or Veo. The aggregate score is a predeclared proxy: CLIP is coarse, pixel and LPIPS emphasize appearance, and automatic-human disagreements remain in the trajectory analysis. The loop is expensive, and success depends on VLM diagnosis quality, metric alignment, and stochastic backend behavior. Finally, the qualitative editing examples do not establish a general editing or generation framework.

5 Conclusion

We introduced VideoRISE as a query-only controller for black-box video prompt inversion. Its inversion adapter supplies the target-video objective and evaluator, a VLM proposes structured prompt edits, and VideoRISE owns execution, verification, checkpoint recovery, and stopping. On two Wan backends, the completed inversion study improves primary utility from 0.661 to 0.701 over the strongest search-only baseline while reducing wasted calls. These results support prompt inversion, not a general editing or generation framework: commercial backends, standard benchmarks, stronger prompt-optimization comparisons, and independent evaluators remain open.

AI Assistance Disclosure. Generative AI tools assisted with language polishing, restructuring portions of the manuscript, and preparing schematic figure layouts. The authors reviewed and approved

all technical claims, citations, experimental design, results, and final presentation; no AI-generated output was treated as experimental evidence.

REFERENCES

Lakshya A. Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J. Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective prompt evolution can outperform reinforcement learning. arXiv:2507.19457, 2025.

Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Yuanzhen Li, Tomer Michaeli, et al. Lumiere: A space-time diffusion model for video generation. *arXiv preprint arXiv:2401.12945*, 2024.

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

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Technical Report*, 2024.

Duygu Ceylan, Chun-Hao Paul Huang, and Niloy J. Mitra. Pix2video: Video editing using image diffusion. In *ICCV*, 2023.

Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Reverse stable diffusion: What prompt was used to generate this image? *Computer Vision and Image Understanding*, 249:104210, 2024.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Promptbreeder: Self-referential self-improvement via prompt evolution. In *ICML*, pp. 13481–13544, 2024.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *ICLR*, 2023.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross-attention control. In *ICLR*, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

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*, 2022.

Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. In *ICCV*, 2023.

Sungho Lee, Jisu Kim, Soohyun Lee, Sangpil Kim, and Sungjin Park. WAVE: Warping DDIM inversion features for zero-shot text-to-video editing. In *ECCV*, 2024.

Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-P2P: Video editing with cross-attention control. In *CVPR*, pp. 8599–8608, 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. In *NeurIPS*, 2023.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *CVPR*, 2023.

Eyal Molad, Eliahu Horwitz, Dani Valevski, Alex Rav Acha, Yossi Matias, Yael Pritch, Yaniv Leviathan, and Yedid Hoshen. Dreamix: Video diffusion models are general video editors. In *ICLR*, 2023.

OpenAI. Sora: Creating video from text. `https://openai.com/sora`, 2024.

pharmapsychotic. CLIP interrogator: Image to prompt. `https://github.com/pharmapsychotic/clip-interrogator`, 2022.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Gong. Automatic prompt optimization with "gradient descent" and beam search. In *EMNLP*, 2023.

Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. FateZero: Fusing attentions for zero-shot text-based video editing. In *ICCV*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, 2023.

Runway AI. Runway gen-3: High-fidelity video generation. `https://runwayml.com/research/gen-3`, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *NeurIPS*, 2023.

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. In *ICLR*, 2023.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

Wan Team. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *CVPR*, 2023.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *ICLR*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

Joshua Nathaniel Williams, Avi Schwarzschild, Yutong He, and J. Zico Kolter. Prompt recovery for image generation models: A comparative study of discrete optimizers. arXiv:2408.06502, 2024.

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

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *ICLR*, 2024a.

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

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *NeurIPS*, 2023.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *EMNLP*, 2023.