REKEY: BENCHMARKING AND LEARNING AGENTIC KEYFRAME REFLECTION FOR VIDEO GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent video generation models have advanced substantially in visual fidelity and motion modeling, yet generated videos still suffer from semantic and temporal inconsistencies. This limitation has motivated Reflective Video Generation (RVG), a paradigm that introduces explicit reflection and correction into the generation process. We introduce ReKey-Bench to evaluate RVG. It comprises 400 video cases and 784 keyframe evaluation samples, covering keyframe verification, rectification, and downstream video improvement. Evaluation of existing vision-language models (VLMs) on ReKey-Bench reveals that they may overlook visual errors or produce rectifications that violate the keyframe plan. To address these limitations, we propose ReKey, an agent framework for evidence-grounded keyframe reflection. We train its reflection agent through on-policy self-distillation, providing a frozen teacher with question-answer-evidence (QAE) as privileged context. ReKey achieves gains of 13.3 points in keyframe verification F1 and 15.9 points in keyframe rectification score over Qwen3.5 on ReKey-Bench, while also improving downstream video generation and achieving consistent gains on T2V-CompBench and VBench-2.0. Our code and models will be publicly available.

1 INTRODUCTION

Recent video generation models have made remarkable progress in visual fidelity, motion modeling, and instruction following (Wan et al., 2025; Jiang et al., 2025; Wei et al., 2026; Yuan et al., 2026b). However, they still struggle with complex prompts, often producing videos with semantic, temporal, and physical inconsistencies (Zhao et al., 2025; Shao et al., 2025; Nam et al., 2025; Feng et al., 2025b). As illustrated in Figure 1(a), these inconsistencies can affect intermediate visual states and object identity. These limitations motivate Reflective Video Generation (RVG) (Xue et al., 2025; Huang et al., 2026a; Lee et al., 2026; Gao et al., 2026b).

RVG builds on the visual understanding and reasoning capabilities of vision-language models (VLMs) and employs them as agents to support planning, visual evaluation, reflection, and iterative refinement (Huang et al., 2026b). Such reflection can be applied to the keyframes that guide video synthesis (Choi et al., 2026; Xu et al., 2026), enabling agents to identify and rectify deviations from the intended visual states before generation. However, existing video benchmarks primarily evaluate generated videos (Huang et al., 2024; Feng et al., 2025a), as illustrated in Figure 2(a), leaving keyframe reflection and its impact on downstream video generation largely unexamined.

ReKey-Bench. To fill this evaluation gap, we introduce ReKey-Bench, comprising 400 video cases and 784 keyframe evaluation samples. For each sample, the agent examines the video prompt, keyframe plan, current target keyframe, and a preceding reference keyframe when available, then predicts whether correction is needed, provides a visual explanation, and specifies a rectification instruction when necessary. Based on the agent's reflection, the target keyframe is either retained or processed by an image editor, and the resulting keyframes are used to condition video generation. As shown in Figure 2(b), ReKey-Bench evaluates keyframe verification, rectification, and downstream video improvement.

Key Observations. Evaluation on ReKey-Bench reveals two limitations in keyframe reflection. As shown in Figure 1(b), an agent may overlook an unsatisfied visual requirement and retain an incorrect keyframe. Even when it correctly determines that rectification is needed, it may produce

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.

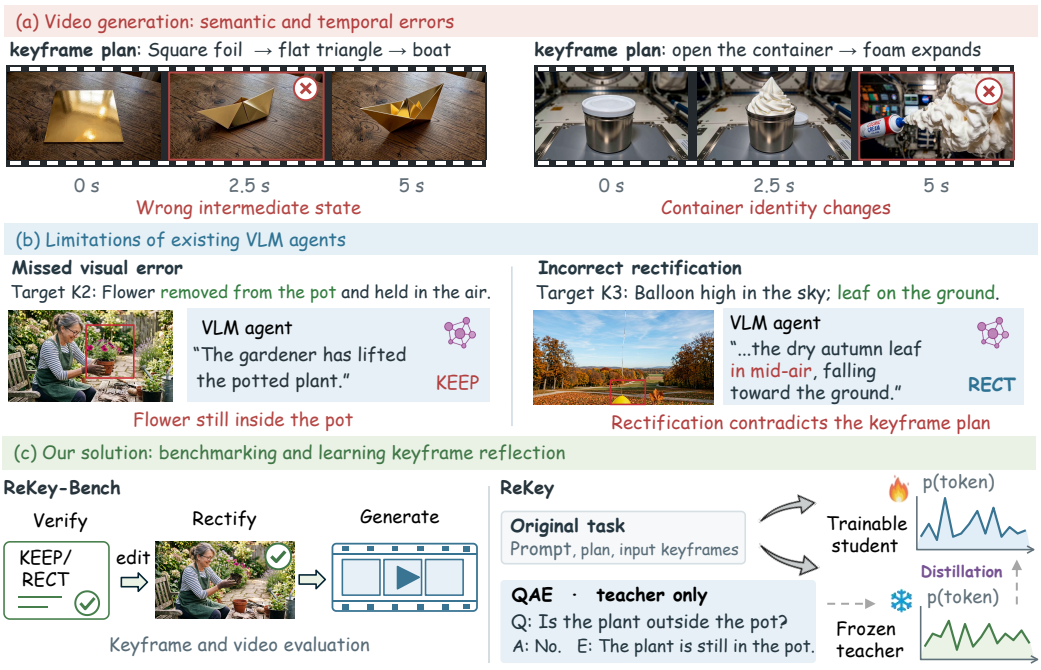

Figure 1: **Method Motivation and Overview. (a)** Video generation models may produce semantic and temporal inconsistencies. **(b)** Existing VLM agents may overlook visual errors and retain incorrect keyframes, or produce rectifications that violate the keyframe plan. **(c)** ReKey-Bench evaluates keyframe verification, rectification, and downstream video improvement, while ReKey learns evidence-grounded keyframe reflection through on-policy self-distillation.

rectifications that violate the keyframe plan. Such failures can leave incorrect keyframes in the generation pipeline and affect downstream video generation. These findings motivate us to ask: *Can improving reflection decisions and corrective actions enhance downstream video generation?*

Our Solution. To address this issue, we propose ReKey, an agent framework that leverages visual evidence to improve reflection decisions and corrective actions. Specifically, we adopt on-policy self-distillation (OPSD) (Zhao et al., 2026) and construct question-answer-evidence (QAE) as privileged context that captures the evidence underlying each reflection decision. Given a video prompt, keyframe plan, and input keyframes, QAE decomposes the planned visual requirements into targeted questions and provides corresponding answers grounded in visual evidence. As shown in Figure 1(c), the student generates responses from the original inputs, while the frozen teacher additionally receives QAE and provides token-level distribution supervision along the same student-generated prefixes. QAE is available only to the teacher during distillation, enabling the student to internalize evidence-grounded reflection and improve its performance without requiring QAE at inference.

Experiments with Qwen3.5-9B and Qwen3.5-27B (Qwen Team, 2026a) show that ReKey consistently improves keyframe verification and rectification on ReKey-Bench, achieving comparable performance to closed-source models such as Gemini 3.1 Pro (Google DeepMind, 2026). Evaluations on T2V-CompBench (Sun et al., 2025) and VBench-2.0 (Zheng et al., 2025) further demonstrate the video-generation performance of ReKey.

In summary, the contributions of this work are as follows:

- We introduce ReKey-Bench, which evaluates RVG through editable keyframes and jointly measures keyframe verification, rectification, and downstream video improvement.
- We find that existing VLMs do not reliably ground their reflection decisions and corrective actions in relevant visual evidence. To address this limitation, we propose ReKey, an agent framework for evidence-grounded keyframe reflection. We train its reflection agent through OPSD, providing a frozen teacher with QAE as privileged context.



Figure 2: **Comparison with existing benchmarks.** **(a)** Existing video benchmarks primarily assess generated videos, leaving intermediate keyframe verification and rectification largely unexamined. **(b)** ReKey-Bench provides keyframe plans, input keyframes, verification labels, and evaluation questions for both keyframes and videos, supporting the evaluation of keyframe verification, rectification, and downstream video improvement.

- ReKey improves keyframe verification F1 by 13.3 points, keyframe rectification score by 15.9 points, and downstream video improvement score by 7.0 points over the Qwen3.5-9B baseline, while achieving performance comparable to closed-source models and improving video generation performance on external T2V benchmarks.

2 PRELIMINARY

This section introduces the task formulation, benchmark, and evaluation protocol for RVG. We first formalize the task and sequential generation process in Sec. 2.1. We then present ReKey-Bench in Sec. 2.2, followed by the evaluation protocol in Sec. 2.3.

2.1 REFLECTIVE VIDEO GENERATION

Given a video prompt P and its corresponding keyframe plan K, RVG processes keyframes sequentially in temporal order. For each target keyframe, the processed keyframe from the preceding temporal position provides the visual context for reflection. Conditioned on P and K, this process follows a first-order Markov dependency, where reflection at each position depends only on the current target keyframe and the preceding processed keyframe. For the k-th target keyframe, we denote the task input as

$$x_k = \left(P, K, \mathbf{I}_k, \tilde{\mathbf{I}}_{k-1}\right), \tag{1}$$

where $\mathbf{I}_k$ is the current target keyframe and $\tilde{\mathbf{I}}_{k-1}$ is the processed keyframe from the preceding temporal position. The latter is omitted for the opening keyframe. A VLM policy π_θ produces a structured response

$$R_k = \pi_\theta(x_k) = \langle v_k, e_k, a_k \rangle, \tag{2}$$

where v_k is the verification verdict, e_k is the visual explanation, and a_k is the rectification instruction.

Keyframe Verification and Rectification. The verdict v_k determines whether the target keyframe is retained (`KEEP`) or rectified (`RECT`). For `RECT`, the rectification instruction a_k specifies either a local edit for minor errors or regeneration for major errors. A shared image generation and editing model Φ executes the rectification using the task context. We denote the retained or rectified result

as the selected keyframe $\tilde{\mathbf{I}}_k$. The selected keyframe $\tilde{\mathbf{I}}_k$ provides the visual context for the next target keyframe. After all target keyframes are processed, the resulting keyframe sequence is used to condition the video generator. We perform one verification and at most one rectification for each target keyframe.

2.2 REKEY-BENCH

In this section, we introduce ReKey-Bench, which evaluates keyframe verification and rectification in RVG, as well as their effects on downstream video generation. Below, we describe the benchmark composition and construction pipeline.

Benchmark Overview. ReKey-Bench comprises 400 video cases designed to evaluate the capabilities required for RVG. Each case contains a video prompt P, a keyframe plan K, and the corresponding keyframes. Each keyframe evaluation sample includes a target keyframe and a fixed preceding reference keyframe when available. Each sample is annotated with a verification label for evaluating v_k and atomic visual questions for assessing the target keyframe $\mathbf{I}_k$ and the resulting selected keyframe $\tilde{\mathbf{I}}_k$. We further provide video-level questions to assess the downstream video generated from the selected keyframe sequence.

Benchmark Construction. We construct ReKey-Bench by creating new video scenarios and collecting or rewriting scenarios from existing video benchmarks (Feng et al., 2025a; Guan et al., 2025; Sun et al., 2025; Zheng et al., 2025; Liu et al., 2023). These scenarios guide the construction of video prompts, keyframe plans, and candidate keyframes. We use model-assisted annotation to obtain verification labels and atomic questions for keyframe and video-level evaluation. Ambiguous or invalid annotations are filtered out, and all retained samples undergo human verification. This process yields 784 keyframe-level samples from 400 video cases. Details of the data sources, construction pipeline, and annotation process are provided in Appendix B.

2.3 EVALUATION PROTOCOL

We evaluate RVG agents at three levels: keyframe verification, keyframe rectification, and downstream video improvement. Together, these evaluations assess whether an agent can correctly identify flawed keyframes, improve them through rectification, and ultimately enhance video generation. We use a fixed image editor Φ and multimodal evaluator $\mathcal{J}$ across all methods.

Phase I: Keyframe Verification. We first evaluate whether the predicted verdict v_k correctly determines if $\mathbf{I}_k$ should be retained or rectified. We compare v_k with the ground-truth verification label and report verification accuracy and F1 score, treating `RECT` as the positive class. Missing or unparseable verdicts are counted as incorrect.

Phase II: Keyframe Rectification. We next evaluate whether the rectification action improves the target keyframe. Following Sec. 2.1, Φ executes a_k for `RECT` predictions to obtain $\tilde{\mathbf{I}}_k$. Otherwise, $\mathbf{I}_k$ is retained. The evaluator $\mathcal{J}$ assesses $\mathbf{I}_k$ and $\tilde{\mathbf{I}}_k$ using the same atomic questions. We compute their case-averaged alignment scores, denoted by $\mathcal{A}^I_{\text{before}}$ and $\mathcal{A}^I_{\text{after}}$. We define the keyframe rectification score as

$$\mathcal{S}^I_{\text{rect}} = \frac{\mathcal{A}^I_{\text{after}} - \mathcal{A}^I_{\text{before}}}{1 - \mathcal{A}^I_{\text{before}}}. \tag{3}$$

$\mathcal{S}^I_{\text{rect}}$ measures the fraction of the initial keyframe alignment deficit recovered by rectification.

Phase III: Downstream Video Improvement. Finally, we evaluate whether the selected keyframe sequence $\{\tilde{\mathbf{I}}_k\}_k$ improves downstream video generation. Following Sec. 2.1, the selected keyframes condition the video generator. We keep the video generator, prompt P, and sampling settings fixed across methods. *Direct* denotes text-to-video generation from P without keyframe conditioning. The evaluator $\mathcal{J}$ assesses each generated video using the corresponding video-level questions. Let $\mathcal{A}^V_m$ and $\mathcal{A}^V_{\text{Direct}}$ denote the resulting case-averaged alignment scores for method m and the Direct baseline, respectively. We define the downstream video improvement score as

$$\mathcal{S}^V_{\text{imp}} = \frac{\mathcal{A}^V_m - \mathcal{A}^V_{\text{Direct}}}{1 - \mathcal{A}^V_{\text{Direct}}}. \tag{4}$$

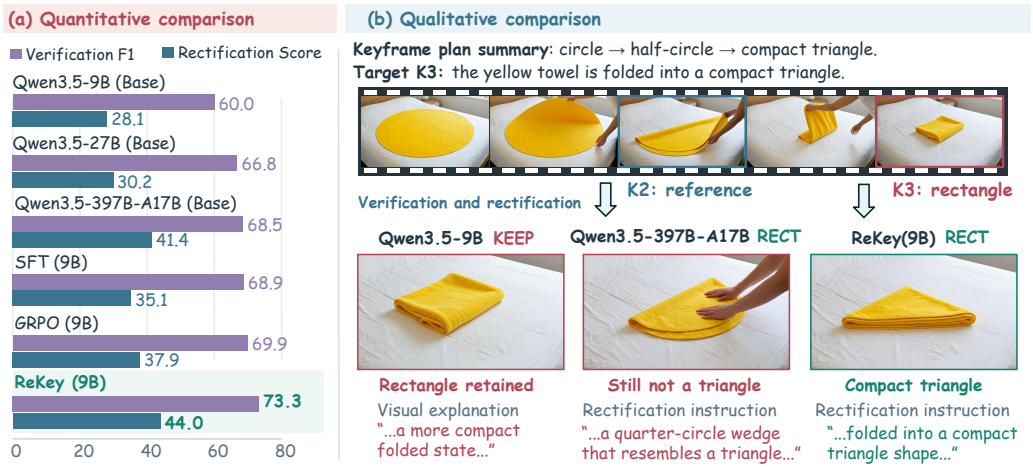

Figure 3: **Empirical Findings on Existing VLMs in RVG. (a)** Quantitative comparison of verification and rectification performance across VLM agents. **(b)** An example contrasting missed errors and ineffective rectification by existing agents with successful rectification by ReKey.

$\mathcal{S}^V_{\text{imp}}$ measures the fraction of the Direct baseline's remaining alignment deficit recovered through keyframe-conditioned generation.

Empirical Observations. Evaluation on ReKey-Bench reveals clear limitations of existing VLMs in keyframe reflection, as shown in Figure 3. They may overlook violations of the intended visual state or produce ineffective corrective actions. Motivated by these observations, we focus on improving downstream video generation through more accurate verification and effective rectification of keyframes. In the following section, we detail how ReKey leverages task-relevant visual evidence to make these decisions and actions more reliable.

3 REKEY

In this section, we present ReKey, an agent framework for evidence-grounded keyframe reflection. We train its reflection agent through OPSD. In OPSD, privileged context is available only to the teacher during training, allowing it to provide stronger supervision while the student operates under the original inference-time context. In our setting, question-answer-evidence (QAE) serves as such privileged context by providing the teacher with task-relevant visual evidence. As shown in Figure 4, the student generates responses from x_k, while the frozen teacher conditions additionally on QAE and provides next-token supervision along the same student-generated prefixes.

Privileged Context Construction. Specifically, for the task input x_k defined in Sec. 2.1, we define the corresponding QAE context h_k as a collection of atomic checks of the visual requirements specified by the video prompt and keyframe plan. Each entry contains a question specifying one visual requirement, a yes-or-no answer indicating whether that requirement is satisfied, and a textual description of the supporting visual evidence, as illustrated in Figure 4. The questions cover both the target state and its change relative to the preceding selected keyframe when available, including satisfied and violated requirements. These atomic answers indicate which individual requirements are satisfied or violated, providing more detailed information than the overall `KEEP`/`RECT` verdict.

On-Policy Teacher Guidance. We use the VLM policy π_θ as the trainable student and its frozen copy π^*_θ as the teacher. Without access to h_k, the student samples the structured response R_k from $\pi_\theta(\cdot \mid x_k)$. Viewing R_k as its serialized token sequence, let L denote its length. At each token position i, let p^S_i and p^T_i denote the student and teacher next-token probability distributions over the vocabulary, respectively. Both policies condition on the same student-generated prefix $R_k^{1:i-1}$, consisting of the first $i-1$ tokens of R_k, while the privileged QAE context h_k is provided only to the teacher:

$$\begin{aligned} p^S_i &= \pi_\theta(\cdot \mid x_k, R_k^{1:i-1}), \\ p^T_i &= \pi^*_\theta(\cdot \mid x_k, h_k, R_k^{1:i-1}). \end{aligned} \tag{5}$$

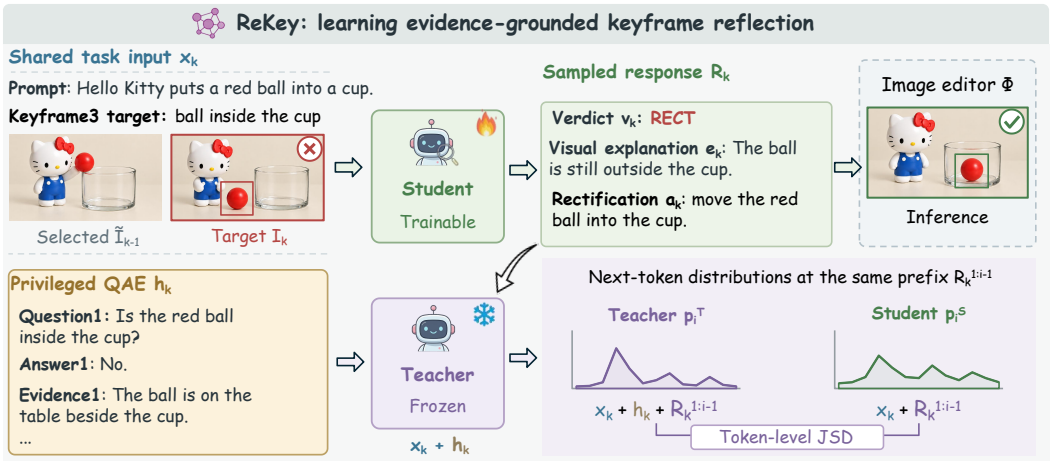

Figure 4: **Overview of ReKey.** Given the video prompt, keyframe plan, preceding reference keyframe, and current target keyframe, the student samples a verification verdict, visual explanation, and rectification instruction. A frozen teacher receives the same task input together with QAE as privileged context. The student is trained by minimizing the Jensen–Shannon divergence between their next-token distributions along student-generated prefixes.

Both distributions are computed over the full vocabulary using the same distillation temperature. Because the prefixes $R_k^{1:i-1}$ are generated by the current student policy, the teacher provides supervision along the student's on-policy trajectory. At inference, the student receives only x_k and produces the structured response R_k defined in Sec. 2.1.

Token-Level Distillation. We minimize the token-averaged Jensen–Shannon divergence (JSD) (Lin, 1991) between the student and teacher distributions:

$$\mathcal{L}_{\text{ReKey}} = \frac{1}{L}\sum_{i=1}^{L} \text{JSD}\left(p_i^S, p_i^T\right). \tag{6}$$

The loss covers the complete structured response R_k, including the verification verdict, visual explanation, and rectification instruction. The teacher is frozen, and gradients are propagated only through the student. With this learning objective, ReKey enables the student to ground keyframe verification and rectification in visual evidence, thereby improving downstream video generation.

4 Experiments

In this section, we examine whether stronger keyframe reflection leads to better video generation and what supervision drives this improvement. We first establish the effectiveness of ReKey on ReKey-Bench through comparisons with existing VLMs and representative training baselines, and further examine whether it scales to a larger base model. We then evaluate ReKey on T2V-CompBench and VBench-2.0 to assess its text-to-video generation performance. Finally, we compare label, QA, QAE, and length-matched QA+ guidance to examine the roles of atomic QA and visual evidence.

4.1 Evaluation on ReKey-Bench

Experimental Setup. We compare ReKey with existing VLMs and representative training baselines on ReKey-Bench. The VLM baselines include Qwen3.5 at multiple scales, Kimi K2.6 (Kimi, 2026), and Gemini 3.1 Pro. On Qwen3.5-9B, we further compare against supervised fine-tuning (SFT), group relative policy optimization (GRPO) (Shao et al., 2024), and label-guided OPSD. SFT is trained on responses generated by the QAE-guided teacher, GRPO uses verification correctness as the reward, and label-guided OPSD provides only the verification label as privileged teacher input.

Table 1: **Results on ReKey-Bench.** Acc. and F1 measure keyframe verification. $\mathcal{S}^I_{\text{rect}}$ and $\mathcal{S}^V_{\text{imp}}$ denote the keyframe rectification score and downstream video improvement score, respectively. In the Training column, – denotes no additional training on our task. Direct generation is the common baseline for $\mathcal{S}^V_{\text{imp}}$. The Original keyframes baseline generates videos directly from the three supplied keyframes without reflection. Best and second-best results are bold and underlined, respectively.

Model	Training	Acc.	F1	$\mathcal{S}^I_{\text{rect}}$	$\mathcal{S}^V_{\text{imp}}$
Original keyframes		–	–	–	0.318
Qwen3.5-9B	–	0.759	0.600	0.281	0.368
Qwen3.5-27B	–	0.823	0.668	0.302	0.416
Qwen3.5-397B-A17B	–	0.809	0.685	0.414	0.422
Kimi K2.6	–	0.825	0.713	0.468	0.423
Gemini 3.1 Pro	–	<u>0.871</u>	**0.829**	**0.559**	**0.493**
Qwen3.5-9B	SFT	0.809	0.689	0.351	0.377
Qwen3.5-9B	GRPO	0.804	0.699	0.379	0.386
Qwen3.5-9B	Label-guided OPSD	0.719	0.545	0.274	0.367
Qwen3.5-9B	ReKey (Ours)	0.832	0.733	0.440	0.438
Qwen3.5-27B	ReKey (Ours)	**0.872**	<u>0.800</u>	<u>0.543</u>	<u>0.477</u>

We also apply ReKey to Qwen3.5-27B to examine its effectiveness with a larger base model. All trained variants use the same training data, image editor, video generator, and evaluation protocol. Additional implementation details are provided in Appendix D.

Main Results. As shown in Table 1, ReKey achieves the strongest overall performance among the Qwen3.5-9B training variants, outperforming SFT, GRPO, and label-guided OPSD. Moreover, the resulting 9B agent surpasses Qwen3.5-397B-A17B across all four metrics. The gains remain consistent when ReKey is applied to Qwen3.5-27B, which improves over the base model across all metrics and further narrows the gap to Gemini 3.1 Pro. Figure 5 provides qualitative comparisons at both keyframe and video levels. Comparison agents may retain flawed keyframes or produce rectified outputs that still violate the planned requirements. ReKey corrects the illustrated state and object-count errors. Compared with generation conditioned on the original keyframes, ReKey also produces videos that better follow the intended motion progression and object-count changes. All agents shown use Qwen3.5-9B. Additional qualitative examples and quantitative comparisons are provided in Appendices E and C, respectively.

4.2 Evaluation on T2V Benchmarks

Experimental Setup. We evaluate ReKey on selected dimensions of T2V-CompBench and VBench-2.0, covering compositional changes, temporal relations, numeracy, and physical consistency. For each prompt, the agent constructs a keyframe plan and the shared image model generates the first keyframe. Conditioned on this keyframe, MiniMax H3 (MiniMax, 2026) generates an initial video, from which the remaining target keyframes are sampled. Following the RVG process in Sec. 2.1, all target keyframes are sequentially verified and either retained or rectified. The resulting keyframes then condition final video generation.

Our ReKey configuration uses Qwen3.5-9B for planning and reflection, with QAE-guided OPSD applied to the reflection agent. We compare this configuration with Direct generation and representative inference-time optimization methods, including SCMAPR (Yang et al., 2026a), VPO (Cheng et al., 2025), SSPO from RAPO++ (Gao et al., 2026a), PhyT2V (Xue et al., 2025), and NEWTON (Feng et al., 2026b). We report both Qwen3.5-9B and Gemini 3.1 Pro configurations for these optimization baselines, except VPO, which uses its trained VPO-5B model. All methods use MiniMax H3. We use Qwen3.8-27B (Qwen Team, 2026b) for VLM-based evaluation on both benchmarks. Further details are provided in Appendix D.2.

Main Results. As shown in Tables 2 and 3, ReKey achieves the highest average scores among the evaluated methods: 0.675 on T2V-CompBench and 0.639 on VBench-2.0, compared with 0.580

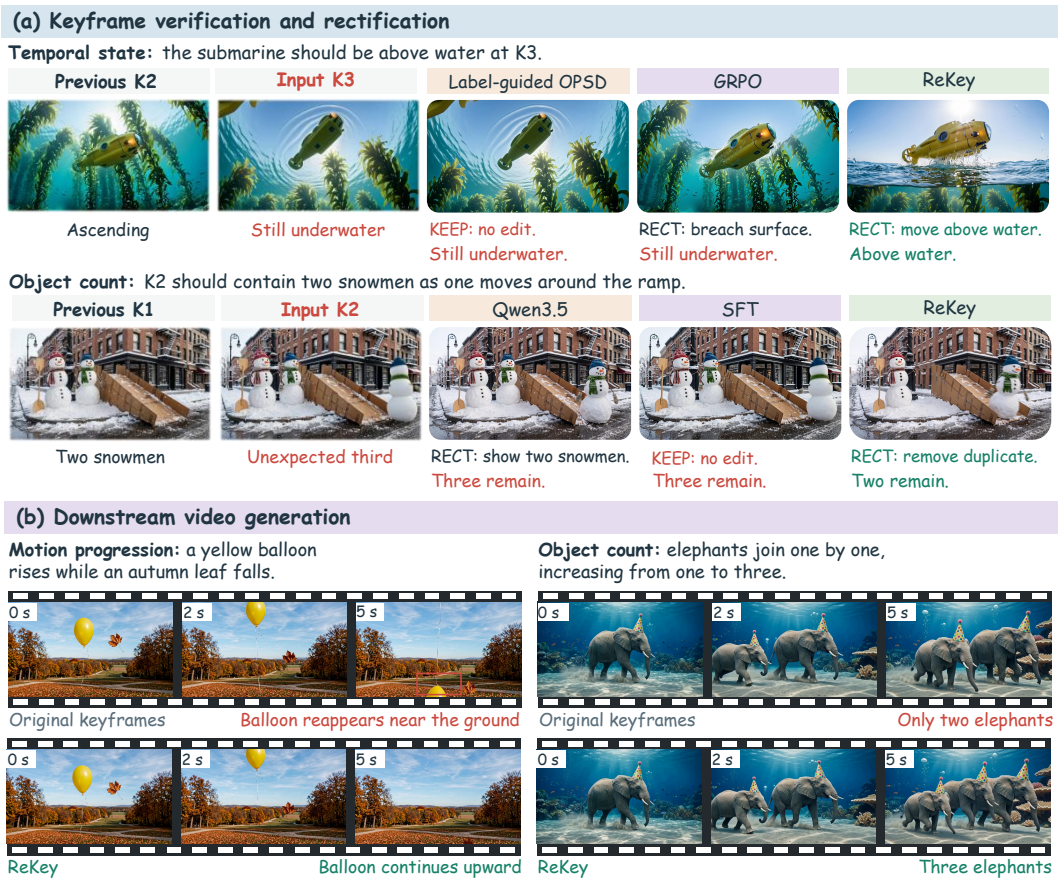

Figure 5: **(a)** Keyframe verification and rectification across different agents. **(b)** Downstream video generation with and without keyframe reflection.

Table 2: **Results on T2V-CompBench.** ReKey uses Qwen3.5-9B trained with QAE-guided OPSD. SSPO uses five inference-time optimization rounds; PhyT2V and NEWTON use one adapted feedback-and-regeneration round. Best and second-best results are bold and underlined, respectively.

Method	Model	Cons. Attr.	Dyn. Attr.	Spatial Rel.	Motion Bind.	Numeracy	Avg.
Direct	–	0.764	0.271	0.669	0.407	0.352	0.492
VPO	VPO-5B	0.863	0.308	0.621	0.397	0.498	0.537
SCMAPR	Qwen3.5-9B	0.855	0.401	0.729	0.388	0.527	0.580
SCMAPR	Gemini 3.1 Pro	0.866	0.412	0.728	0.396	0.519	0.584
SSPO	Qwen3.5-9B	0.830	0.317	0.699	0.393	0.512	0.550
SSPO	Gemini 3.1 Pro	0.866	0.373	0.727	0.423	<u>0.626</u>	0.603
PhyT2V	Qwen3.5-9B	0.814	0.287	0.683	0.410	0.425	0.524
PhyT2V	Gemini 3.1 Pro	0.864	<u>0.418</u>	**0.737**	0.424	0.594	0.608
NEWTON	Qwen3.5-9B	0.823	0.299	0.683	0.376	0.448	0.526
NEWTON	Gemini 3.1 Pro	<u>0.878</u>	0.404	<u>0.736</u>	<u>0.461</u>	0.606	<u>0.617</u>
ReKey (Ours)	Qwen3.5-9B	**0.927**	**0.429**	0.699	**0.522**	**0.800**	**0.675**

and 0.570 for the strongest baselines using the same Qwen3.5-9B backbone, respectively. The resulting 9B agent also outperforms all evaluated Gemini 3.1 Pro-based baselines in average score on both benchmarks. ReKey leads in consistent and dynamic attribute binding, motion binding, and

Table 3: **Results on VBench-2.0.** ReKey uses Qwen3.5-9B trained with QAE-guided OPSD. SSPO uses five inference-time optimization rounds; PhyT2V and NEWTON use one adapted feedback-and-regeneration round. Best and second-best results are bold and underlined, respectively.

Method	Model	Dyn. Attr.	Dyn. Rel.	Motion Order	Material	Mechanics	Thermotics	Avg.
Direct	–	0.604	0.609	0.263	0.240	0.491	0.260	0.411
VPO	VPO-5B	0.637	0.391	0.515	0.460	0.596	0.480	0.513
SCMAPR	Qwen3.5-9B	0.637	0.667	0.404	0.500	0.614	0.600	0.570
SCMAPR	Gemini 3.1 Pro	0.626	**0.739**	0.505	0.580	<u>0.649</u>	0.600	0.617
SSPO	Qwen3.5-9B	0.505	0.478	0.374	0.540	0.561	0.520	0.496
SSPO	Gemini 3.1 Pro	0.648	0.522	0.414	0.520	**0.667**	<u>0.640</u>	0.568
PhyT2V	Qwen3.5-9B	0.516	0.493	0.374	0.440	0.509	0.440	0.462
PhyT2V	Gemini 3.1 Pro	<u>0.681</u>	<u>0.725</u>	0.455	0.580	0.632	**0.660**	0.622
NEWTON	Qwen3.5-9B	0.549	0.536	0.323	0.480	0.456	0.380	0.454
NEWTON	Gemini 3.1 Pro	**0.692**	**0.739**	<u>0.556</u>	<u>0.600</u>	0.596	0.620	<u>0.634</u>
ReKey (Ours)	Qwen3.5-9B	<u>0.681</u>	<u>0.725</u>	**0.576**	**0.640**	0.632	0.580	**0.639**

numeracy on T2V-CompBench, and in motion order and material on VBench-2.0. These results demonstrate the effectiveness of ReKey for text-to-video generation under this evaluation setting.

4.3 Effects of QAE Guidance

We progressively enrich the privileged guidance provided to the frozen teacher, from the overall verification label to atomic QA and further to QAE. As shown in Table 4, atomic QA substantially improves both verification and keyframe rectification over label-only guidance. Incorporating visual evidence further strengthens these capabilities and yields the best downstream video improvement. The length-matched QA+ variant remains inferior to QAE, indicating that increased prompt length alone does not account for QAE's gains. These results demonstrate that atomic decomposition and visual evidence provide complementary supervision for learning more reliable keyframe reflection.

Table 4: **Effects of teacher guidance on ReKey-Bench.** All variants use Qwen3.5-9B with the same training data and training budget. QA+ repeats atomic QA to match the teacher-prompt length of QAE without introducing visual evidence. Best results are bold.

Teacher guidance	Acc.	F1	$\mathcal{S}^{I}_{\text{rect}}$	$\mathcal{S}^{V}_{\text{imp}}$
Verification label	0.719	0.545	0.274	0.367
Atomic QA	0.795	0.649	0.356	0.359
QA+ (length-matched)	0.781	0.628	0.367	0.378
Atomic QAE (ReKey)	**0.832**	**0.733**	**0.440**	**0.438**

5 Conclusion

We study agentic keyframe reflection in RVG and introduce ReKey-Bench, which evaluates keyframe verification, rectification, and downstream video improvement. Evaluation on ReKey-Bench reveals that existing VLMs do not consistently ground their reflection decisions and corrective actions in the visual evidence relevant to each requirement. To address this limitation, we propose ReKey, an agent framework for evidence-grounded keyframe reflection. We train its reflection agent through OPSD, providing a frozen teacher with QAE as privileged context. ReKey consistently improves keyframe verification, rectification, and downstream video generation with both Qwen3.5-9B and Qwen3.5-27B, and further improves performance on T2V-CompBench and VBench-2.0. These results demonstrate the importance of grounding reflection in task-relevant visual evidence for reliable rectification and improved video generation.

AI ASSISTANCE

We used large language models to assist with language editing and manuscript organization, including improving grammar, clarity, and presentation. Generative AI models were also used to support the construction of ReKey-Bench and the training data, including video prompts, keyframe plans, visual samples, and question-answer-evidence annotations, as described in the paper. The authors take full responsibility for the final manuscript, data, analyses, claims, and references.

REPRODUCIBILITY STATEMENT

Data construction and benchmark composition are detailed in Appendix B, and training settings and evaluation details in Appendix D. Appendix F provides the inference, teacher-guidance, and evaluation prompt templates. Additional results and analyses are reported in Appendix C. Code and model weights will be made publicly available.

REFERENCES

Hritik Bansal, Zongyu Lin, Tianyi Xie, Zeshun Zong, Michal Yarom, Yonatan Bitton, Chenfanfu Jiang, Yizhou Sun, Kai-Wei Chang, and Aditya Grover. Videophy: Evaluating physical commonsense for video generation. In *International Conference on Learning Representations*, volume 2025, pp. 102075–102121, 2025.

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.

Minkyu Choi, S. P. Sharan, Harsh Goel, Sahil Shah, and Sandeep Chinchali. We'll fix it in post: Improving text-to-video generation with neuro-symbolic feedback. *arXiv preprint arXiv:2504.17180*, 2025. URL https://arxiv.org/abs/2504.17180v1.

Tae Eun Choi, Sumin Shim, Junhyeok Kim, and Seong Jae Hwang. Anchoring and rescaling attention for semantically coherent inbetweening. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8976–8985, 2026.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. In-video instructions: Visual signals as generative control. *arXiv preprint arXiv:2511.19401*, 2025.

Kaituo Feng, Manyuan Zhang, Shuang Chen, Yunlong Lin, Kaixuan Fan, Yilei Jiang, Hongyu Li, Dian Zheng, Chenyang Wang, and Xiangyu Yue. Gen-searcher: Reinforcing agentic search for image generation. *arXiv preprint arXiv:2603.28767*, 2026a.

Weixi Feng, Jiachen Li, Michael Saxon, Tsu-Jui Fu, Wenhu Chen, and William Yang Wang. TC-bench: Benchmarking temporal compositionality in conditional video generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 4638–4662, July 2025a. doi: 10.18653/v1/2025.findings-acl.241. URL https://aclanthology.org/2025.findings-acl.241/.

Weixi Feng, Chao Liu, Sifei Liu, William Yang Wang, Arash Vahdat, and Weili Nie. Blobgen-vid: Compositional text-to-video generation with blob video representations. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12989–12998, 2025b. doi: 10.1109/CVPR52734.2025.01212.

Yuxiang Feng, Juncheng Wang, Chao Xu, Yijie Qian, Huihan Wang, Wenlong Hou, Yang Liu, Baigui Sun, Yong Liu, and Shujun Wang. Newton: Agentic planning for physically grounded video generation. *arXiv preprint arXiv:2605.18396*, 2026b.

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, 2026a. URL https://arxiv.org/abs/2510.20206.

Jiayi Gao, Changcheng Hua, Jiaqi Tang, Yuxin Peng, and Yang Liu. Identity-preserving text-to-video generation via agentic enhancement and semantic repair, 2026b. URL https://arxiv.org/abs/2608.20749v1.

Google DeepMind. Gemini 3.1 Pro model card, 2026. URL https://deepmind.google/models/model-cards/gemini-3-1-pro/.

Kaisi Guan, Zhengfeng Lai, Yuchong Sun, Peng Zhang, Wei Liu, Kieran Liu, Meng Cao, and Ruihua Song. Etva: Evaluation of text-to-video alignment via fine-grained question generation and answering. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21299–21309, 2025. doi: 10.1109/ICCV51701.2025.01978.

Xuyang Guo, Jiayan Huo, Zhenmei Shi, Zhao Song, Jiahao Zhang, and Jiale Zhao. T2vphysbench: A first-principles benchmark for physical consistency in text-to-video generation. *arXiv preprint arXiv:2505.00337*, 2025a.

Zujin Guo, Size Wu, Zhongang Cai, Wei Li, and Chen Change Loy. Controllable human-centric keyframe interpolation with generative prior, 2025b. URL https://arxiv.org/abs/2506.03119.

Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, Sherman Siu, Haonan Chen, Abhranil Chandra, Ziyan Jiang, Aaran Arulraj, Kai Wang, Quy Duc Do, Yuansheng Ni, Bohan Lyu, Yaswanth Narsupalli, Rongqi Fan, Zhiheng Lyu, Bill Yuchen Lin, and Wenhu Chen. VideoScore: Building automatic metrics to simulate fine-grained human feedback for video generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2105–2123, November 2024. doi: 10.18653/v1/2024.emnlp-main.127. URL https://aclanthology.org/2024.emnlp-main.127/.

Xuan He, Dongfu Jiang, Ping Nie, Minghao Liu, Zhengxuan Jiang, Mingyi Su, Wentao Ma, Junru Lin, Chun Ye, Yi Lu, Keming Wu, Benjamin Schneider, Quy Duc Do, Zhuofeng Li, Yiming Jia, Yuxuan Zhang, Guo Cheng, Haozhe Wang, Wangchunshu Zhou, Qunshu Lin, Yuanxing Zhang, Ge Zhang, Wenhao Huang, and Wenhu Chen. Videoscore2: Think before you score in generative video evaluation, 2025. URL https://arxiv.org/abs/2509.22799.

Xuehan Hou, Meng Fan, Pengchong Qiao, Zesen Cheng, Yian Zhao, Lei Zhu, Kaiwen Cheng, Chang Liu, and Jie Chen. Anchor frame bridging for coherent first-last frame video generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=isNjWnVsUR.

Kaiyi Huang, Yukun Huang, Xuefei Ning, Zinan Lin, Yu Wang, and Xihui Liu. Genmac: compositional text-to-video generation with multi-agent collaboration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 5049–5057, 2026a.

Yinming Huang, Shuyuan Tu, Xi Yan, Jiahao Zhan, Zihan Yang, Zhen Xing, Hui Zhang, Tiehua Zhang, Yu-Gang Jiang, and Zuxuan Wu. Agentic visual generation: From generative models to agentic control, 2026b. URL https://arxiv.org/abs/2609.06758.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818, 2024. doi: 10.1109/CVPR52733.2024.02060.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, Yaohui Wang, Xinyuan Chen, Ying-Cong Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(3):3268–3285, 2026c. doi: 10.1109/TPAMI.2025.3633890.

Sangwon Jang, Taekyung Ki, Jaehyeong Jo, Jaehong Yoon, Soo Ye Kim, Zhe Lin, and Sung Ju Hwang. Frame guidance: Training-free guidance for frame-level control in video diffusion models. In *International Conference on Learning Representations*, volume 2026, pp. 60977–61003, 2026.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17191–17202, 2025.

Kimi. Kimi K2.6: Advancing open-source coding, 2026. URL https://www.kimi.ai/en/blog/kimi-k2-6.

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

Daeun Lee, Jaehong Yoon, Jaemin Cho, and Mohit Bansal. Self-correcting text-to-video generation with misalignment detection and localized refinement. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 36464–36489, July 2026. doi: 10.18653/v1/2026.findings-acl.1817. URL https://aclanthology.org/2026.findings-acl.1817/.

Baiqi Li, Zhiqiu Lin, Deepak Pathak, Jiayao Li, Yixin Fei, Kewen Wu, Tiffany Ling, Xide Xia, Pengchuan Zhang, Graham Neubig, and Deva Ramanan. Genai-bench: Evaluating and improving compositional text-to-visual generation. *arXiv preprint arXiv:2406.13743*, 2024.

Haodong Li, Tianfei Ren, Xiaoxiao Ma, Chunmei Qing, Zhen Fang, Sipeng He, Ziyu Guo, Haoyu Wu, Juanxi Tian, Yihang Zou, Ruichuan An, Dongzhi Jiang, Boxue Yang, Ji Xie, Xu Huang, Wenhao Yan, Jialv Zou, Zhengrong Yue, Yaxin Luo, Xiaotong Li, Yuzhu Wang, Junyan Ye, Jinjing Zhao, Zehui Chen, Lin Chen, Renye Yan, Feng Zhao, and Pheng-Ann Heng. Videococo: Code-as-cot for physically-consistent video generation via an agentic dual-engine system, 2026a. URL https://arxiv.org/abs/2607.27380v2.

Qirui Li, Jinkun Hao, Yibo Li, Ran Yi, Paul L. Rosin, and Yu-Kun Lai. PhysAgent: Reflective agentic physics control for physically plausible video generation, 2026b. URL https://arxiv.org/abs/2607.16355v1.

Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991. doi: 10.1109/18.61115.

Yang Liu, Xilin Zhao, Peisong Wen, Siran Dai, and Qingming Huang. Bootstrapping physics-grounded video generation through VLM-guided iterative self-refinement. *arXiv preprint arXiv:2511.20280*, 2025.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22139–22149. IEEE, 2024.

Yuanxin Liu, Lei Li, Shuhuai Ren, Rundong Gao, Shicheng Li, Sishuo Chen, Xu Sun, and Lu Hou. Fetv: A benchmark for fine-grained evaluation of open-domain text-to-video generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 62352–62387, 2023. doi: 10.52202/075280-2723.

MiniMax. MiniMax H3. Hugging Face model card, 2026. URL https://huggingface.co/MiniMaxAI/MiniMax-H3.

Hyelin Nam, Jaemin Kim, Dohun Lee, and Jong Chul Ye. Optical-flow guided prompt optimization for coherent video generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7837–7846, 2025. doi: 10.1109/CVPR52734.2025.00734.

Quynh Phung, Long Mai, Fabian David Caba Heilbron, Feng Liu, Jia-Bin Huang, and Cusuh Ham. Cineverse: Consistent keyframe synthesis for cinematic scene composition. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2626–2636. IEEE, 2026.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026a. URL https://qwen.ai/blog?id=qwen3.5.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026b. URL https://qwen.ai/blog?id=qwen3.8.

Naina Raisinghani. Nano Banana 2: Combining Pro capabilities with lightning-fast speed. Google Blog, February 2026. URL https://blog.google/innovation-and-ai/technology/ai/nano-banana-2/.

Dian Shao, Mingfei Shi, Shengda Xu, Haodong Chen, Yongle Huang, and Binglu Wang. Finephys: Fine-grained human action generation by explicitly incorporating physical laws for effective skeletal guidance. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1905–1916, 2025. doi: 10.1109/CVPR52734.2025.00184.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Yiwen Song, Tomas Pfister, and Yale Song. VQQA: An agentic approach for video evaluation and quality improvement. *arXiv preprint arXiv:2603.12310*, 2026.

Achint Soni, Sreyas Venkataraman, Abhranil Chandra, Sebastian Fischmeister, Percy Liang, Bo Dai, and Sherry Yang. Videoagent: Self-improving video generation, 2025. URL https://arxiv.org/abs/2410.10076.

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.

Kanghui Tian, Siyuan Liu, Ziang Yan, Sheng Xia, Shuai Dong, and Yi Wang. ViCuR: Visual cues as recoverable privilege for multimodal on-policy distillation. *arXiv preprint arXiv:2606.05718*, 2026.

Aman Tyagi, Hemanth Boinpally, Jonathan Chen, Douglas Gebert, and Steven Hickson. Beyond trial-and-error: Agentic optimization for image-to-video adherence, 2026. URL https://arxiv.org/abs/2608.12290v1.

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

Junjie Wang, Xinghua Lou, Jason Li, Ye Tian, Keyu Chen, Yulin Li, Bin Kang, Jacky Mai, Yanwei Li, Zhuotao Tian, and Liqiang Nie. Benchmarking and evolving reason-reflect-rectify for reflective visual generation, 2026a. URL https://arxiv.org/abs/2605.19639.

Xiaojuan Wang, Boyang Zhou, Brian Curless, Ira Kemelmacher-Shlizerman, Aleksander Holynski, and Steve Seitz. Generative inbetweening: Adapting image-to-video models for keyframe interpolation. In *International Conference on Learning Representations*, volume 2025, pp. 30594–30609, 2025a.

Xingrui Wang, Jiang Liu, Ze Wang, Xiaodong Yu, Jialian Wu, Ximeng Sun, Yusheng Su, Alan Yuille, Zicheng Liu, and Emad Barsoum. Keyvid: Keyframe-aware video diffusion for audio-synchronized visual animation. *arXiv preprint arXiv:2504.09656*, 2025b.

Ziyue Wang, Shiqi Huang, Weiwen Xu, Bihan Wen, and Xudong Jiang. Video-opsd: Exploiting privileged visual evidence for on-policy self-distillation in video large language models, 2026b. URL https://arxiv.org/abs/2608.27065.

Cong Wei, Quande Liu, Zixuan Ye, Qiulin Wang, Xintao Wang, Pengfei Wan, Kun Gai, and Wenhu Chen. Univideo: Unified understanding, generation, and editing for videos, 2026. URL https://arxiv.org/abs/2510.08377.

Chuanzhi Xu, Huiqi Liang, Bang Shi, Huiming Zhang, Yifan Xiao, Guangcheng Lin, Haodong Chen, Qiang Qu, Zhicheng Lu, and Weidong Cai. Drawvideo: Generating long video from storyboard keyframe sketches. *arXiv preprint arXiv:2605.23508*, 2026.

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, 2026a. doi: 10.18653/v1/2026.findings-acl.345. URL https://aclanthology.org/2026.findings-acl.345/.

Yu Yang, Yue Liao, Jianbiao Mei, Baisen Wang, Xuemeng Yang, Licheng Wen, Jiangning Zhang, Xiangtai Li, Liang Lv, Hanlin Chen, Botian Shi, Yong Liu, Shuicheng Yan, and Gim Hee Lee. SPIRAL: Self-evolving action-conditioned video generation via reflective planning agents. *arXiv preprint arXiv:2603.08403*, 2026b.

Yuhang Yang, Ke Fan, Shangkun Sun, Hongxiang Li, Ailing Zeng, FeiLin Han, Wei Zhai, Wei Liu, Yang Cao, and Zheng-Jun Zha. Videogen-eval: Agent-based system for video generation evaluation. *arXiv preprint arXiv:2503.23452*, 2025.

Qianhao Yuan, Jie Lou, Xing Yu, Hongyu Lin, Le Sun, Xianpei Han, and Yaojie Lu. Vision-OPD: Learning to see fine details for multimodal LLMs via on-policy self-distillation. *arXiv preprint arXiv:2605.18740*, 2026a.

Shenghai Yuan, Yuanyang Yin, Zongjian Li, Xinwei Huang, Xiao Yang, and Li Yuan. Helios: Real real-time long video generation model. *arXiv preprint arXiv:2603.04379*, 2026b.

Ziyun Zeng, Zixuan Wang, Yongsheng Yu, Hang Hua, and Jiebo Luo. Videoargus: Agentic rubric-grounded unified evaluation for video generation and editing, 2026. URL https://arxiv.org/abs/2608.05485v1.

Zhida Zhang, Jie Ma, Zhan Peng, Haoxue Wu, Yang Han, Jun Liang, Jie Cao, and Jing Li. Smartdirector: Keyframe-conditioned cinematic video generation with narrative pacing control, 2026. URL https://arxiv.org/abs/2605.27891.

Qi Zhao, Xingyu Ni, Ziyu Wang, Feng Cheng, Ziyan Yang, Lu Jiang, and Bohan Wang. Synthetic video enhances physical fidelity in video synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12135–12146, October 2025.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Lulu Gu, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, Yu Qiao, and Ziwei Liu. Vbench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness, 2025. URL https://arxiv.org/abs/2503.21755.

Le Zhuo, Liangbing Zhao, Sayak Paul, Yue Liao, Renrui Zhang, Yi Xin, Peng Gao, Mohamed Elhoseiny, and Hongsheng Li. From reflection to perfection: Scaling inference-time optimization for text-to-image diffusion models via reflection tuning. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15329–15339, 2025. doi: 10.1109/ICCV51701.2025.01422.

Supplementary Material

A Related Work 15

B Data Construction and ReKey-Bench 16

B.1 Data Sources and Construction 17

B.2 ReKey-Bench Composition 17

B.3 ReKey-Bench Examples 18

C Additional Results and Analysis 18

C.1 Effectiveness of Keyframe Reflection 18

C.2 Reliability of ReKey-Bench Evaluation 21

C.3 Performance under Different Evaluators 22

C.4 Additional Reflection Agent Comparisons 22

D Training and Evaluation Details 23

D.1 Training Setup 23

D.2 Evaluation Implementation 23

E Additional Qualitative Results 24

E.1 Keyframe Verification and Rectification 24

E.2 Downstream Video Comparisons 24

E.3 Failure Cases 29

F Prompt Details 30

F.1 Student Inference Prompt 30

F.2 Teacher Guidance 32

F.3 Keyframe Evaluation Prompt 32

F.4 Video Evaluation Prompt 33

A Related Work

Video Generation. Recent video foundation models, such as Wan and HunyuanVideo, have made substantial progress in visual quality, motion modeling, and text-to-video alignment through large-scale training (Wan et al., 2025; Kong et al., 2024). Beyond text-only generation, visual conditioning provides finer control over appearance and temporal structure (Fang et al., 2025; Wang et al., 2025b). VACE enables reference-conditioned generation and video editing, while UniVideo unifies video understanding, generation, and editing through multimodal instructions (Jiang et al., 2025; Wei et al., 2026). More structured visual conditions further constrain how scene states evolve over time. Generative Inbetweening synthesizes motion between specified keyframes (Wang et al., 2025a), while PoseFuse3D-KI introduces 3D human guidance for controllable keyframe interpolation (Guo et al., 2025b). Anchor Frame Bridging selects intermediate anchor frames to improve semantic continuity between the first and last frames (Hou et al., 2026), and Frame Guidance supports frame-level conditions including keyframes, sketches, and depth maps (Jang et al., 2026). DrawVideo and SmartDirector further organize long-form generation using storyboard-derived or multiple keyframes to control narrative progression (Xu et al., 2026; Zhang et al., 2026). CineVerse uses detailed scene and shot plans to synthesize consistent cinematic keyframes (Phung et al., 2026). Collectively, these methods provide increasingly structured mechanisms for translating textual and visual conditions into coherent videos. Our work focuses on a complementary problem: verifying whether candidate keyframes satisfy the planned visual states and temporal progression, and rectifying violations before video synthesis.

Reflective Video Generation. Reflection and corrective feedback have been explored in image generation through reason–reflect–rectify workflows (Wang et al., 2026a), reflection-guided refinement (Zhuo et al., 2025), and learned agentic search (Feng et al., 2026a). Recent work incorporates planning, evaluation, and iterative refinement into different stages of video generation (Huang et al., 2026b; Soni et al., 2025; Liu et al., 2025; Li et al., 2026b; Yang et al., 2026b; Choi et al., 2025; Song et al., 2026). GENMAC uses multi-agent verification and redesign to iteratively improve compositional video generation (Huang et al., 2026a), while SCMAPR combines multi-agent prompt refinement with semantic verification (Yang et al., 2026a). VPO learns prompt optimization from supervised data and preference feedback (Cheng et al., 2025), while RAPO++ integrates training-data-aligned prompt refinement, test-time feedback, and prompt-rewriter fine-tuning (Gao et al., 2026a). Agentic image-to-video optimization also uses multimodal feedback to refine prompts and search generation settings (Tyagi et al., 2026). PhyT2V iteratively improves physical consistency through feedback, and NEWTON trains a planner to coordinate physics-aware tools and verification (Xue et al., 2025; Feng et al., 2026b). VideoCoCo generates simulated drafts through agent-produced Blender programs to guide subsequent video editing (Li et al., 2026a). Other approaches directly rectify visual errors. VideoRepair detects text-to-video misalignment and selectively regenerates problematic regions (Lee et al., 2026), while AESR uses edited frames as visual references to rectify semantic errors in generated videos (Gao et al., 2026b). Within this broader class, we focus on sequential keyframe verification and rectification conditioned on the video prompt, keyframe plan, and preceding selected keyframe when available.

To train the keyframe verification and rectification policy, we build on on-policy self-distillation (OPSD), where a teacher is conditioned on privileged information and its next-token distributions are distilled along student-generated prefixes (Zhao et al., 2026). Recent multimodal extensions incorporate different forms of privileged visual information, including recoverable visual cues in ViCuR, crop-conditioned perception in Vision-OPD, and evidence frames for video understanding in Video-OPSD (Tian et al., 2026; Yuan et al., 2026a; Wang et al., 2026b). ReKey applies this principle to keyframe verification and rectification. In our setting, the frozen teacher receives question–answer–evidence (QAE) as privileged context, associating each atomic visual requirement with its satisfaction status and supporting visual evidence, while the student receives only the original task input. We distill the teacher's next-token distributions across the full student-generated response, including the verification verdict, visual explanation, and rectification instruction.

Benchmarks for Video Generation and Reflection. Existing video-generation benchmarks evaluate complementary aspects of generated videos (Liu et al., 2024). VBench and VBench++ assess visual quality, temporal consistency, and semantic alignment (Huang et al., 2024; 2026c). T2V-CompBench and GenAI-Bench evaluate compositional requirements, while TC-Bench focuses on temporal changes in objects, attributes, and relations (Sun et al., 2025; Li et al., 2024; Feng et al., 2025a). FETV supports fine-grained video evaluation across prompts categorized by content, controllable attributes, and complexity (Liu et al., 2023), while ETVA measures text-video alignment through atomic question answering (Guan et al., 2025). VBench-2.0 extends video evaluation to intrinsic faithfulness, including physical and commonsense consistency, while VideoPhy and T2VPhysBench further assess physical plausibility and compliance with physical laws (Zheng et al., 2025; Bansal et al., 2025; Guo et al., 2025a). Learned evaluators such as VideoScore and VideoScore2 provide automated assessments (He et al., 2024; 2025), while VideoGen-Eval and VideoArgus use agents to structure video evaluation and feedback (Yang et al., 2025; Zeng et al., 2026). These benchmarks primarily evaluate generated outputs. In contrast, R^3-Bench explicitly evaluates verification and rectification in image generation (Wang et al., 2026a). ReKey-Bench extends reflection evaluation to video generation by separately assessing keyframe verification, keyframe rectification, and downstream video improvement under shared generation and evaluation backends. This staged evaluation reveals whether more accurate keyframe verification and more effective rectification improve downstream video alignment.

B Data Construction and ReKey-Bench

In this section, we first describe the data construction process. We then summarize the composition of ReKey-Bench and present representative benchmark examples.

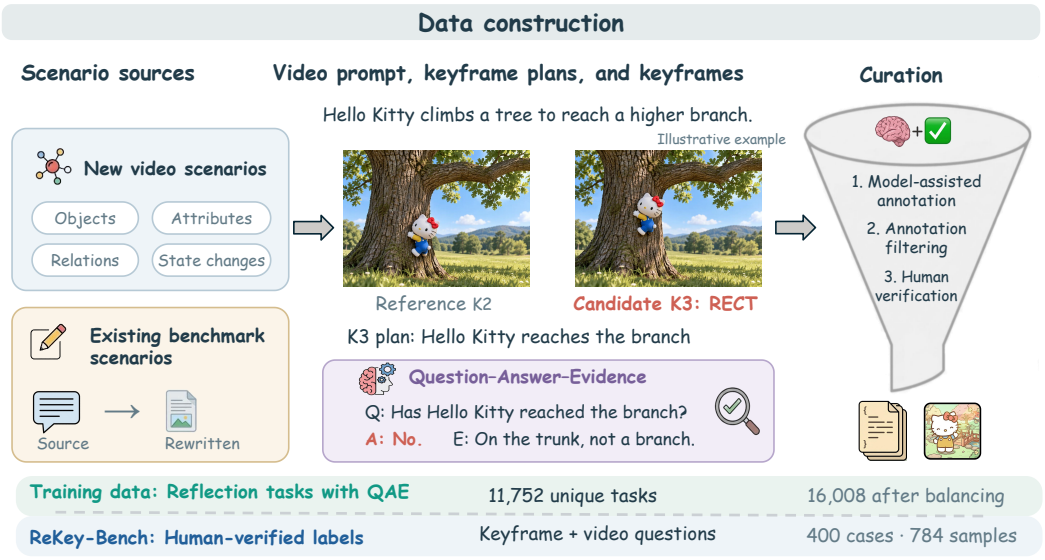

Figure 6: **Data construction.** We construct video prompts and keyframe plans from video scenarios, generate videos, and extract keyframes. Using the videos and keyframes alongside the prompts and plans, we construct and verify questions, answers, and visual evidence. The verified annotations form the training data and ReKey-Bench, with all retained benchmark samples additionally verified by humans.

B.1 Data Sources and Construction

Figure 6 summarizes the data construction process. For training, we create new video scenarios by combining requirements on objects, attributes, relations, and state changes, and collect or rewrite scenarios from TC-Bench, ETVA, and FETV. ReKey-Bench includes newly created scenarios and scenarios collected or rewritten from TC-Bench, ETVA, T2V-CompBench, VBench-2.0, and FETV. We reserve its original scenarios and their rewritten variants for evaluation and exclude all associated samples from training. For both training data and ReKey-Bench, Gemini 3.1 Pro converts each scenario into a video prompt and a keyframe plan specifying opening, intermediate, and final states. After verifying the prompts and plans, we use them to guide video generation and extract the opening, intermediate, and final frames from each source video as candidate keyframes for reflection.

To assess whether the generated content satisfies the intended visual requirements, we construct atomic questions using the video prompt, keyframe plan, source video, and extracted keyframes. Keyframe questions assess the requirements at the corresponding stage, while video questions assess subjects, attributes, states, and transitions throughout the sequence. After the questions are verified, Gemini 3.1 Pro answers them and provides supporting visual evidence. For training, the keyframe questions, their answers, and the supporting evidence form question–answer–evidence (QAE) annotations.

To reduce annotation errors, Gemini 3.1 Pro verifies its generated answers, followed by three additional verification rounds with Qwen3.8-27B. We retain only QAE annotations validated by Gemini and confirmed in all three Qwen rounds. The verified answers directly determine the keyframe verification label: `KEEP` if all atomic requirements for the target keyframe are satisfied, and `RECT` if at least one is violated. This ties each label to the individual visual requirements and their supporting evidence. We further manually verify all retained ReKey-Bench samples.

B.2 ReKey-Bench Composition

Following the construction and verification process described above, we obtain ReKey-Bench with 400 cases and 784 keyframe evaluation samples. The benchmark provides 3,484 keyframe questions and 4,866 video questions. As shown in Figure 7(a), these questions cover diverse visual require-

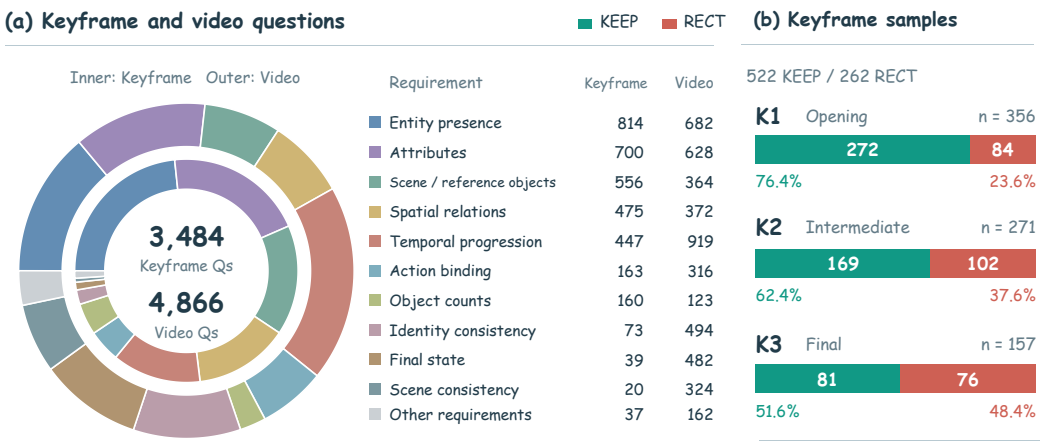

Figure 7: **Composition of ReKey-Bench. (a)** Counts and distributions of 3,484 keyframe questions (inner ring) and 4,866 video questions (outer ring) by visual requirement; each ring is normalized within its question set. **(b)** `KEEP`/`RECT` distributions across opening (K1), intermediate (K2), and final (K3) stages, covering 784 keyframe evaluation samples from 400 cases.

ments, including entity presence, attributes, spatial relations, object counts, and temporal progression. Entity presence and attributes are the two largest categories among keyframe questions, while video questions place greater relative emphasis on temporal progression, identity consistency, and final states. Together, these questions assess visual requirements at individual keyframe stages as well as consistency and temporal progression throughout the video.

To ensure reliable evaluation, we require the preceding keyframes to satisfy their planned visual requirements. Specifically, K2 samples require a valid K1, while K3 samples require valid K1 and K2 keyframes. Samples with inconsistent model verification results are also excluded. After applying these criteria, we retain 784 evaluation samples, of which 522 are labeled `KEEP` and 262 are labeled `RECT`. As shown in Figure 7(b), these samples are distributed across three keyframe stages: 356 opening (K1), 271 intermediate (K2), and 157 final (K3) samples.

B.3 ReKey-Bench Examples

To illustrate the samples and evaluation questions summarized above, we present six ReKey-Bench examples in Figure 8. These examples involve visual requirements such as object counts, attributes, spatial relations, lighting, and shape changes. For each case, we use a video prompt and a three-stage keyframe plan to specify the intended visual states. Against these requirements, we indicate whether the target keyframe needs rectification through its `KEEP`/`RECT` label. In the figure, check marks indicate keyframes that satisfy their planned requirements, crosses indicate violations, and dashes denote keyframes not evaluated in the illustrated task. To complement the overall verification label, we provide atomic keyframe questions that assess individual visual requirements. We also provide video evaluation questions to measure downstream video improvement.

C Additional Results and Analysis

In this section, we first examine the effectiveness of keyframe reflection and the reliability of ReKey-Bench evaluation. We then assess performance under an alternative evaluator and compare additional reflection agents.

C.1 Effectiveness of Keyframe Reflection

Verification and Rectification. Figure 9 contrasts verification and rectification across Qwen3.5-9B agents. ReKey restores all four participants while preserving their appearances and identities; the

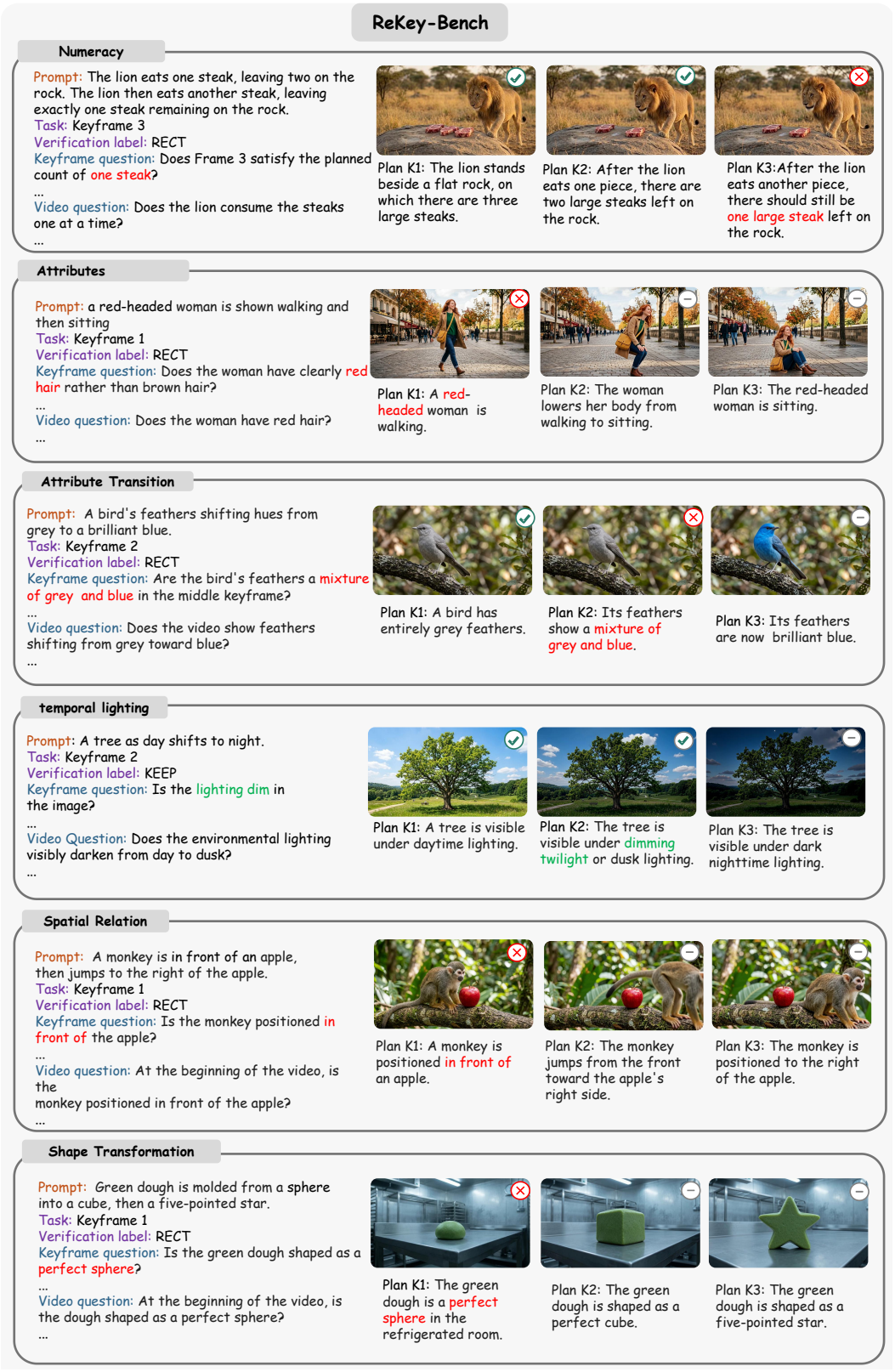

Figure 8: **Visualizations of ReKey-Bench.** Representative examples include video prompts, keyframe plans, corresponding keyframes, verification labels for the target keyframes, and keyframe and video questions.

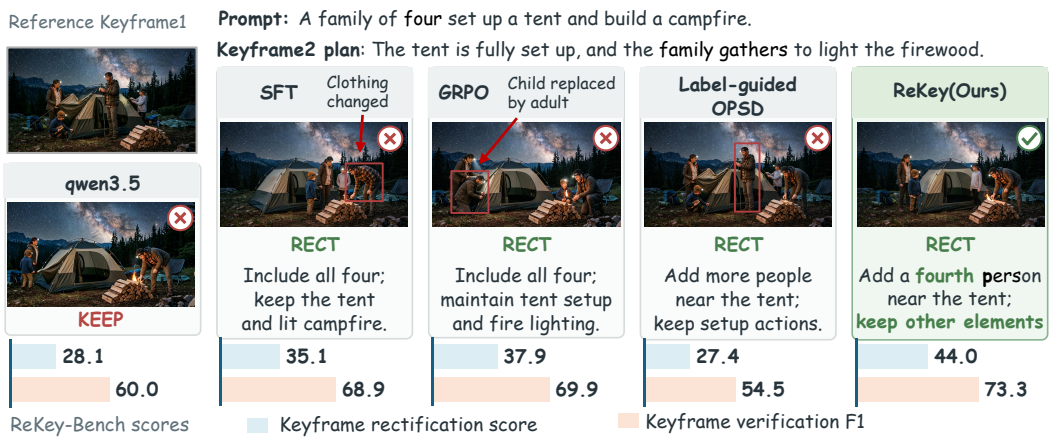

Figure 9: **Keyframe verification and rectification under different training strategies.** The base agent misses the error, while SFT, GRPO, and label-guided OPSD identify the need for rectification but produce edited keyframes that violate participant-count or cross-keyframe consistency requirements. ReKey restores the required participant count while preserving consistency with the preceding keyframe. All agents use Qwen3.5-9B, and instructions are abbreviated.

compared agents miss the error or violate count or consistency requirements. Across ReKey-Bench, we compute the weighted fractions of initially unmet requirements corrected and initially satisfied requirements violated, using the question weights underlying the main alignment scores. ReKey corrects 50.86% of unmet requirements, compared with 38.19% for Base and 46.65% for GRPO, while violating 1.24% of initially satisfied requirements, versus 1.82% and 1.57%, respectively. These results show that ReKey corrects more errors while better preserving valid content.

Sensitivity to Visual Evidence. To examine the role of visual evidence in the verification decisions illustrated above, we use matched pairs of valid and invalid target keyframes from ReKey-Bench. We compare the base Qwen3.5-9B model with OPSD variants using verification labels (Label), atomic question–answer pairs (QA), and question–answer–evidence (QAE) as teacher guidance, with QAE corresponding to ReKey. All models are evaluated without privileged guidance. For each pair, we hold the task text, preceding reference keyframe when available, and response prefix fixed. We then measure how much the probability of predicting `RECT` increases when the valid target keyframe is replaced by the invalid one:

$$\Delta_{\text{evidence}} = p(\texttt{RECT} \mid I_{\text{invalid}}) - p(\texttt{RECT} \mid I_{\text{valid}}), \tag{7}$$

where I_{valid} and I_{invalid} denote the paired target keyframes, and $p(\texttt{RECT} \mid I)$ is the probability of `RECT` normalized over `KEEP` and `RECT`. A larger positive Δ_{evidence} indicates greater separation between valid and invalid keyframes.

Figure 10(a) shows that ReKey achieves the largest mean probability increase, indicating greater sensitivity to visual differences that require rectification. In the example in Figure 10(b), QA-guided OPSD retains an incorrect keyframe, while ReKey identifies the visual discrepancy and produces a rectified keyframe that satisfies the planned requirements. Together, these results support the effectiveness of QAE guidance in grounding verification and rectification in task-relevant visual evidence.

Keyframe-Video Correlation. Finally, we examine whether improvements in keyframe alignment are associated with downstream video gains. We measure ReKey's keyframe alignment changes relative to the input keyframes under independent evaluation, and its video gains over generation conditioned on the original three keyframes under sequential RVG. The gains are positively correlated, with a Spearman correlation of 0.239 (95% case-bootstrap CI: $[0.136, 0.336]$). Relative to generation from the original keyframes, the mean gain in $\mathcal{S}^{V}_{\text{imp}}$ is 0.246 for cases with improved keyframe alignment and 0.065 for the remaining cases. Together, these analyses demonstrate that ReKey improves keyframe verification and rectification and produces downstream videos that better satisfy the intended visual requirements.

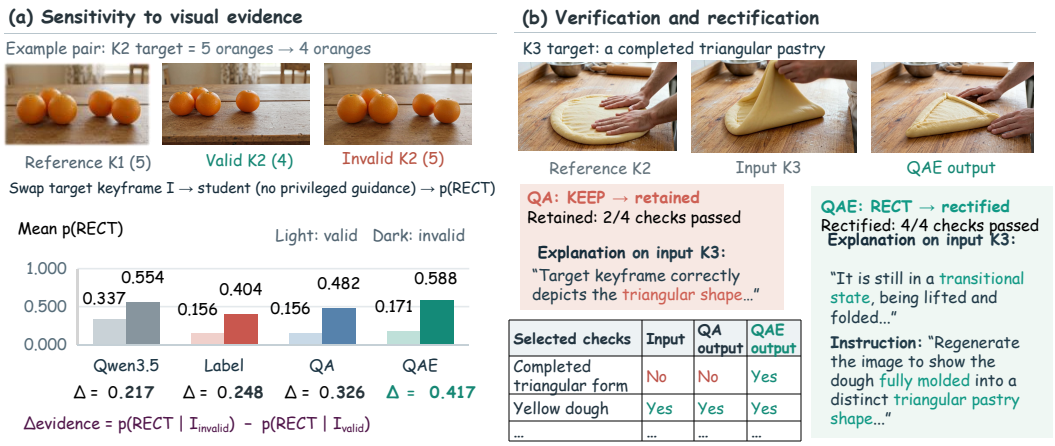

Figure 10: **Visual evidence in keyframe verification and rectification.** **(a)** Mean probabilities of predicting `RECT` for matched valid–invalid target-keyframe pairs under a fixed task context and response prefix. **(b)** Qualitative comparison of verification and rectification with QA-guided OPSD and ReKey. All agents use Qwen3.5-9B. Label, QA, and QAE denote teacher-guidance variants used during OPSD training, with QAE corresponding to ReKey.

C.2 Reliability of ReKey-Bench Evaluation

Paired Confidence Intervals. To assess the statistical reliability of the main comparisons, we estimate paired confidence intervals for the performance differences between ReKey and the comparison methods. Table 5 reports these differences and their 95% bootstrap confidence intervals against Base, SFT, GRPO, Label-guided OPSD, QA, and QA+, all using Qwen3.5-9B. QA+ denotes the length-matched QA guidance variant. We perform 10,000 paired bootstrap resamples over the 400 cases, retaining all tasks and questions within each sampled case and using identical resamples across methods. Each resample recomputes verification accuracy, F1, and both normalized scores, including their baseline denominators.

The intervals for verification accuracy and downstream video improvement lie above zero for all six comparisons. The intervals for verification F1 and keyframe rectification also lie above zero, except in the comparison with GRPO, where they include zero. These results support the statistical reliability of ReKey's gains in verification accuracy and downstream video improvement.

Table 5: **Paired case-level confidence intervals.** For each metric, Δ is ReKey minus the column method; the following row gives its 95% bootstrap interval. Differences are computed from unrounded scores. F1 uses `RECT` as the positive class. Label, QA, and QA+ denote the corresponding OPSD teacher-guidance variants.

Metric	Base	SFT	GRPO	Label	QA	QA+
Δ Acc.	+0.073	+0.023	+0.028	+0.112	+0.037	+0.051
95% CI	[0.0436, 0.1020]	[0.0013, 0.0453]	[0.0051, 0.0516]	[0.0833, 0.1413]	[0.0143, 0.0600]	[0.0254, 0.0767]
Δ F1	+0.132	+0.044	+0.034	+0.187	+0.084	+0.105
95% CI	[0.0850, 0.1815]	[0.0118, 0.0781]	[-0.0038, 0.0707]	[0.1407, 0.2339]	[0.0439, 0.1244]	[0.0620, 0.1500]
$\Delta\mathcal{S}^{I}_{\text{rect}}$	+0.159	+0.089	+0.060	+0.166	+0.084	+0.073
95% CI	[0.0841, 0.2340]	[0.0248, 0.1534]	[-0.0089, 0.1292]	[0.0884, 0.2450]	[0.0149, 0.1499]	[0.0002, 0.1457]
$\Delta\mathcal{S}^{V}_{\text{imp}}$	+0.069	+0.061	+0.052	+0.071	+0.079	+0.060
95% CI	[0.0305, 0.1078]	[0.0194, 0.1047]	[0.0131, 0.0901]	[0.0317, 0.1110]	[0.0411, 0.1196]	[0.0137, 0.1068]

Performance across Keyframe Stages. We further examine how these improvements vary across the opening (K1), intermediate (K2), and final (K3) stages in Table 6. ReKey achieves the highest verification accuracy, F1, and keyframe rectification score at K1, the highest verification scores at K2, and the highest rectification score at K3. GRPO leads in rectification at K2 and verification at K3. Across all three stages, ReKey improves over the base model on every metric, demonstrating consistent benefits throughout the keyframe sequence.

Table 6: **Keyframe verification and rectification by stage.** All methods use Qwen3.5-9B. F1 uses `RECT` as the positive class. Task counts are shown in the column headers. Bold and underlined values indicate the best and second-best results in each column, respectively.

	K1 (356 tasks)			K2 (271 tasks)			K3 (157 tasks)		
Method	Acc.	F1	$\mathcal{S}^I_{\text{rect}}$	Acc.	F1	$\mathcal{S}^I_{\text{rect}}$	Acc.	F1	$\mathcal{S}^I_{\text{rect}}$
Base	0.767	0.435	-0.021	0.779	0.684	0.345	0.707	0.662	0.379
SFT	<u>0.829</u>	<u>0.555</u>	0.162	<u>0.797</u>	<u>0.734</u>	0.393	0.783	0.754	0.472
GRPO	0.815	0.459	0.023	0.764	0.731	**0.465**	**0.847**	**0.842**	<u>0.585</u>
Label	0.666	0.287	0.009	0.775	0.681	0.371	0.745	0.683	0.401
QA	0.806	0.481	<u>0.196</u>	0.775	0.684	0.398	0.803	0.767	0.482
QA+	0.772	0.382	0.153	0.790	0.711	0.414	0.783	0.746	0.520
ReKey	**0.834**	**0.576**	**0.246**	**0.827**	**0.777**	<u>0.450</u>	<u>0.834</u>	<u>0.819</u>	**0.612**

C.3 Performance under Different Evaluators

To examine whether ReKey's gains are consistent across evaluators, we rescore the existing images and videos of all seven Qwen3.5-9B agents with Qwen3.8-27B on all 400 ReKey-Bench cases, using the same 3,484 keyframe and 4,866 video questions as the Gemini 3.1 Pro evaluation. Table 7 shows that ReKey achieves the highest keyframe rectification and downstream video improvement scores under both evaluators. Under Qwen3.8-27B, its keyframe rectification and video improvement scores are 0.489 and 0.616, compared with 0.301 and 0.561 for Base, and 0.411 and 0.566 for QA-guided OPSD. These results show that ReKey's advantages are maintained under the alternative evaluator.

Table 7: **Performance under different evaluators.** Keyframe rectification score and downstream video improvement score for seven Qwen3.5-9B agents.

Evaluator	Method	$\mathcal{S}^I_{\text{rect}}$	$\mathcal{S}^V_{\text{imp}}$
Gemini 3.1 Pro	Base	0.281	0.368
	SFT	0.351	0.377
	GRPO	0.379	0.386
	Label-guided OPSD	0.274	0.367
	QA-guided OPSD	0.356	0.359
	QA+	0.367	0.378
	ReKey(Ours)	**0.440**	**0.438**
Qwen3.8-27B	Base	0.301	0.561
	SFT	0.415	0.564
	GRPO	0.420	0.586
	Label-guided OPSD	0.313	0.574
	QA-guided OPSD	0.411	0.566
	QA+	0.373	0.575
	ReKey(Ours)	**0.489**	**0.616**

C.4 Additional Reflection Agent Comparisons

We compare ReKey with AESR (Gao et al., 2026b) and VideoRepair (Lee et al., 2026) on ReKey-Bench, adapting both methods to keyframe reflection with shared image editing and video generation

Table 8: **Additional reflection agent comparisons on ReKey-Bench.** Base denotes Qwen3.5-9B using the standard keyframe reflection prompt without additional training. The Gemini configuration of ReKey uses the same reflection pipeline without additional training. Bold values indicate the best result within each model group.

Method	Model	Acc.	F1	$\mathcal{S}^I_{\text{rect}}$	$\mathcal{S}^V_{\text{imp}}$
Base	Qwen3.5-9B	0.759	0.600	0.281	0.368
AESR	Qwen3.5-9B	0.334	0.501	0.183	0.350
VideoRepair	Qwen3.5-9B	0.615	0.445	0.018	0.369
ReKey (Ours)	Qwen3.5-9B	**0.832**	**0.733**	**0.440**	**0.438**
AESR	Gemini 3.1 Pro	**0.895**	**0.856**	0.569	0.466
VideoRepair	Gemini 3.1 Pro	0.883	0.843	**0.642**	0.477
ReKey (w/o training)	Gemini 3.1 Pro	0.871	0.829	0.559	**0.493**

backends. Evaluation follows the protocols in Appendix D.2. Table 8 shows that ReKey outperforms Base, AESR, and VideoRepair across all four metrics with Qwen3.5-9B. In our Qwen3.5-9B adaptation, AESR predicts `RECT` for all valid responses. With Gemini 3.1 Pro, AESR and VideoRepair obtain higher verification and keyframe rectification scores, while ReKey achieves the highest downstream video improvement score.

D Training and Evaluation Details

In this section, we first describe the training setup for ReKey. We then explain how we calculate keyframe and video scores and describe the evaluation settings for external T2V benchmarks.

D.1 Training Setup

We train ReKey's reflection agent using QAE-guided OPSD. For Qwen3.5-9B, we apply LoRA on eight RTX 5090 GPUs, keeping the visual encoder and aligner frozen. We report checkpoints after 2,000 training steps for the 9B methods and 1,000 steps for ReKey with Qwen3.5-27B. Table 9 lists the detailed hyperparameters for 9B ReKey. ReKey samples one response per prompt, while GRPO samples eight. Under these training settings, ReKey achieves approximately $6.9\times$ GRPO's prompt throughput while obtaining higher keyframe reflection scores.

D.2 Evaluation Implementation

ReKey-Bench Evaluation. We use Gemini 3.1 Flash Image (Raisinghani, 2026) as the shared image editor Φ. Keyframe evaluation treats the 784 samples independently, using fixed benchmark references when available; some case–stage pairs include alternative candidates. For video evaluation, we process all three keyframes of each case sequentially, using the selected preceding keyframe as the reference for the next decision. For keyframe alignment, we average atomic-question scores within each sample and then across samples within each case. For video alignment, we average video-question scores within each case. We then average each type of alignment score across cases with equal case weights and compute $\mathcal{S}^I_{\text{rect}}$ and $\mathcal{S}^V_{\text{imp}}$ as defined in Sec. 2.3.

External T2V Evaluation. All methods use MiniMax H3 on T2V-CompBench and VBench-2.0. Keyframe-conditioned generation uses Ref2VA, while Direct and prompt-optimization baselines use FL2VA. SSPO performs five optimization rounds. PhyT2V and NEWTON reuse the Direct initial videos and each perform one adapted feedback-and-regeneration round. For NEWTON, we adapt its publicly released inference-time loop using the agent listed in the tables.

We fix the scoring procedures across methods. For T2V-CompBench, we use Qwen3.8-27B for VLM-based evaluation and retain the original specialized evaluators for the remaining dimensions. For VBench-2.0, Qwen3.8-27B evaluates all selected dimensions according to the benchmark criteria. Material, Mechanics, and Thermotics receive zero credit when the required initial condition is not met. We report the arithmetic mean over the dimensions in each table.

Table 9: **Training hyperparameters for ReKey with Qwen3.5-9B.**

Hyperparameter	Value	Description
Model and Adaptation		
Base model	Qwen3.5-9B	Initial policy and teacher backbone.
LoRA rank / alpha / dropout	32 / 64 / 0.05	Adapter rank, scaling, and dropout.
LoRA target modules	LM linear layers	Linear layers in the language model.
Visual encoder and aligner	Frozen	Excluded from optimization.
Optimization and Batch Settings		
Optimizer	Fused AdamW	Fused parameter updates.
Adam (β_1, β_2) / ϵ	$(0.9, 0.95)$ / 10^{-8}	Moment decay and stability constant.
Learning rate	5×10^{-6}	Peak learning rate.
Weight decay	0.1	Decoupled weight regularization.
Scheduler / warmup ratio	Cosine / 0.05	Cosine decay with 5% warmup.
Training steps	2,000	Total optimization updates.
Per-device batch size	1	Tasks per GPU per microbatch.
Gradient accumulation steps	4	Microbatches per update.
Global batch size	32	Tasks per optimization update.
On-Policy Sampling and Distillation		
Samples per prompt	1	Student responses per task.
Rollout temperature / top-p	0.5 / 1.0	Student sampling settings.
Distillation temperature	0.5	Teacher/student logit temperature.
JSD mixture weight	0.5	Mixture coefficient in JSD.
Input Limits and Runtime		
Sequence / response limit	4,096 / 2,048 tokens	Maximum sequence and response lengths.
Maximum image pixels	262,144	Pixel cap per input image.
Base / adapter precision	BF16 / FP32	Precision of weights and adapters.
Gradient checkpointing	Enabled	Reduces activation memory.
Training seed / data seed	42 / 42	Training and data-order randomness.
Number of GPUs	8	Devices used for training.

E Additional Qualitative Results

In this section, we present additional keyframe and video comparisons illustrating how ReKey corrects keyframe errors and improves downstream videos, followed by representative failure cases that highlight its remaining limitations.

E.1 Keyframe Verification and Rectification

Figures 11 and 12 compare keyframe verification and rectification across scene settings, object requirements, and temporal states. The baselines sometimes overlook incorrect scene settings or missing participants, while also requesting unnecessary edits to valid keyframes, such as the five oranges or the STOP sign with the required red lettering. ReKey restores the required participants, scene settings, and intermediate shapes, while avoiding unnecessary edits to valid object counts, attributes, and intermediate states.

E.2 Downstream Video Comparisons

Figures 13 and 14 compare how generated videos follow the planned visual requirements and temporal progression. The baseline videos illustrate incomplete state changes and incorrect action outcomes: lava remains glowing at the end, the sky stays partly cloudy, or the robot moves past its intended stopping point beside the cactus. In these examples, ReKey better preserves object–motion binding and follows the required state changes and action outcomes, including lava cooling into dark rock, foil passing through the triangular folding stage, and a robot stopping beside the cactus.

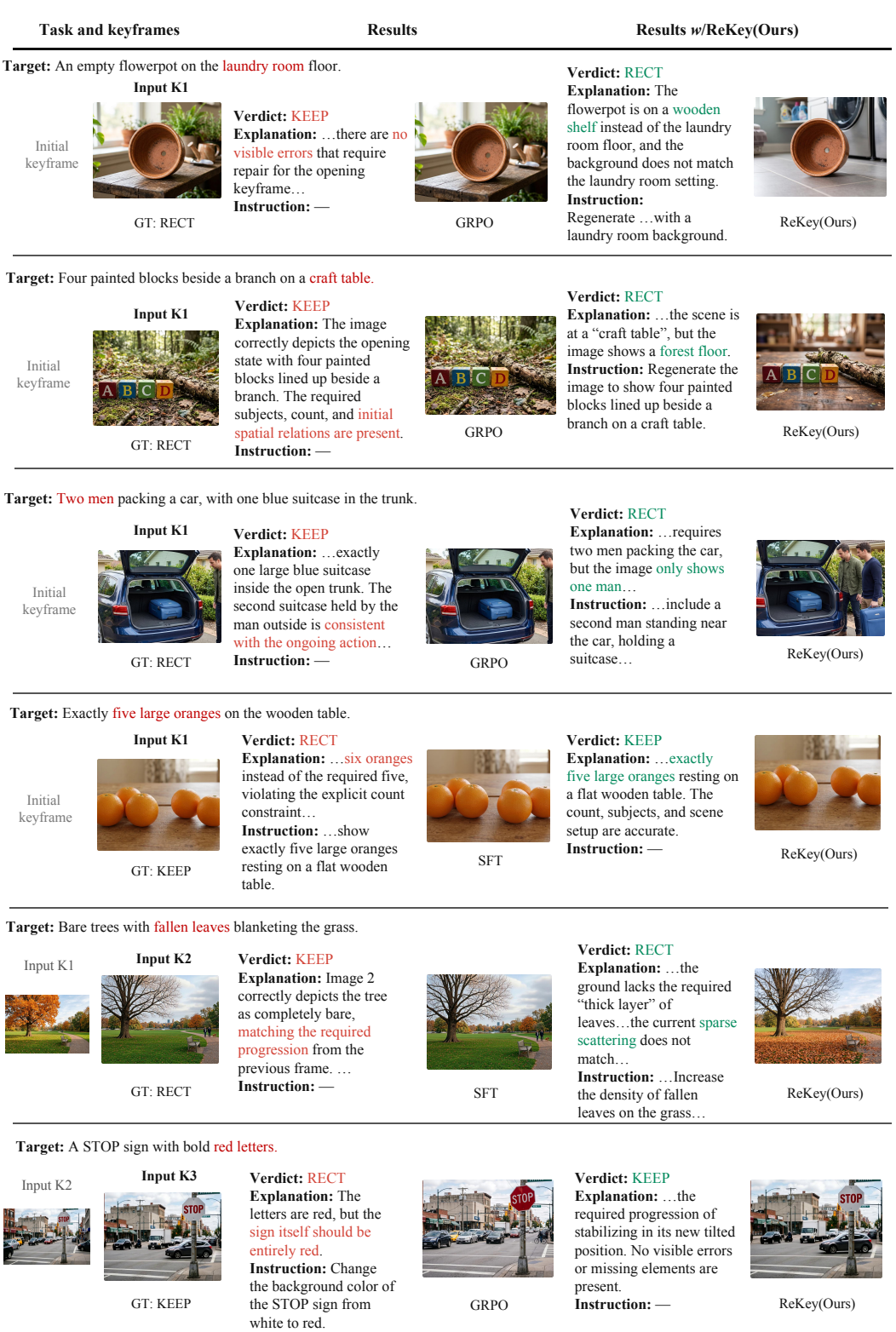

Figure 11: **Scene and object requirements.** Examples show rectification of scene settings, participant counts, and ground coverage, as well as preservation of keyframes with valid object counts and attributes. Each row compares a baseline with ReKey.

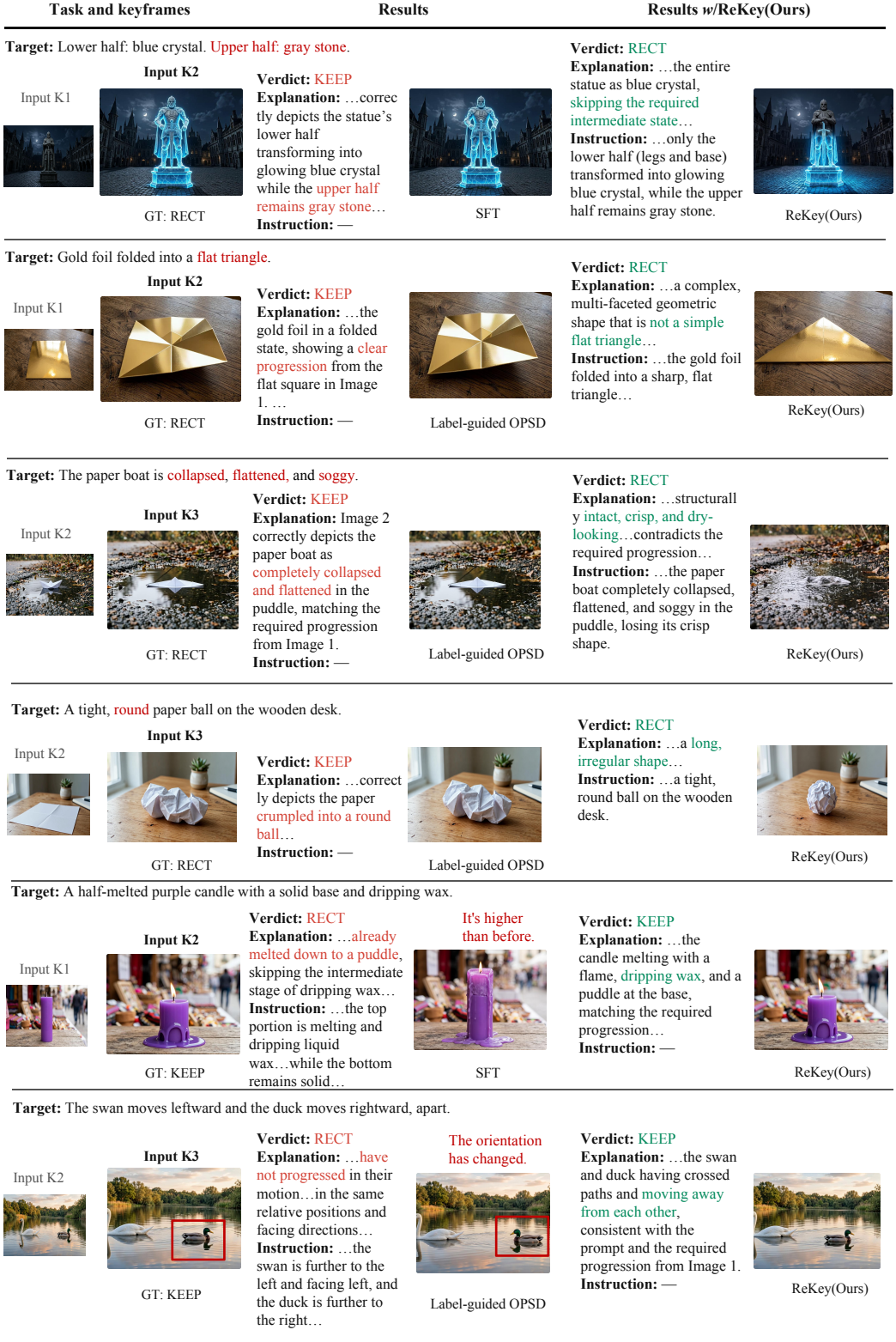

Figure 12: **Temporal states and motion.** Cases cover partial transformations, intermediate shapes, final states, and motion directions. Comparisons include both rectifying invalid keyframes and retaining valid keyframes. Each row compares a baseline with ReKey.



Figure 13: **Additional downstream video comparisons.** Examples show a kangaroo moving behind a board, underwater lava cooling, a clearing sky, and a robot stopping beside a cactus. Paired rows compare the indicated baseline with ReKey at matched timestamps.



Figure 14: **Additional downstream video comparisons.** Examples show an animal carrying a fireball, staged foil folding, sequential steak consumption, and plate breakage. Paired rows compare the indicated baseline with ReKey at matched timestamps.

E.3 FAILURE CASES

Figure 15 highlights two remaining limitations. *(1) Keyframe rectification failures.* The agent may identify an error but overlook other violated requirements, leaving its rectification instruction incomplete. Even with correct rectification instructions, the image editor may fail to produce the intended changes or introduce new artifacts. *(2) Temporal inconsistency in video generation.* Even after the target keyframe errors are corrected, the video generation model may fail to maintain temporal continuity and introduce new inconsistencies between frames.

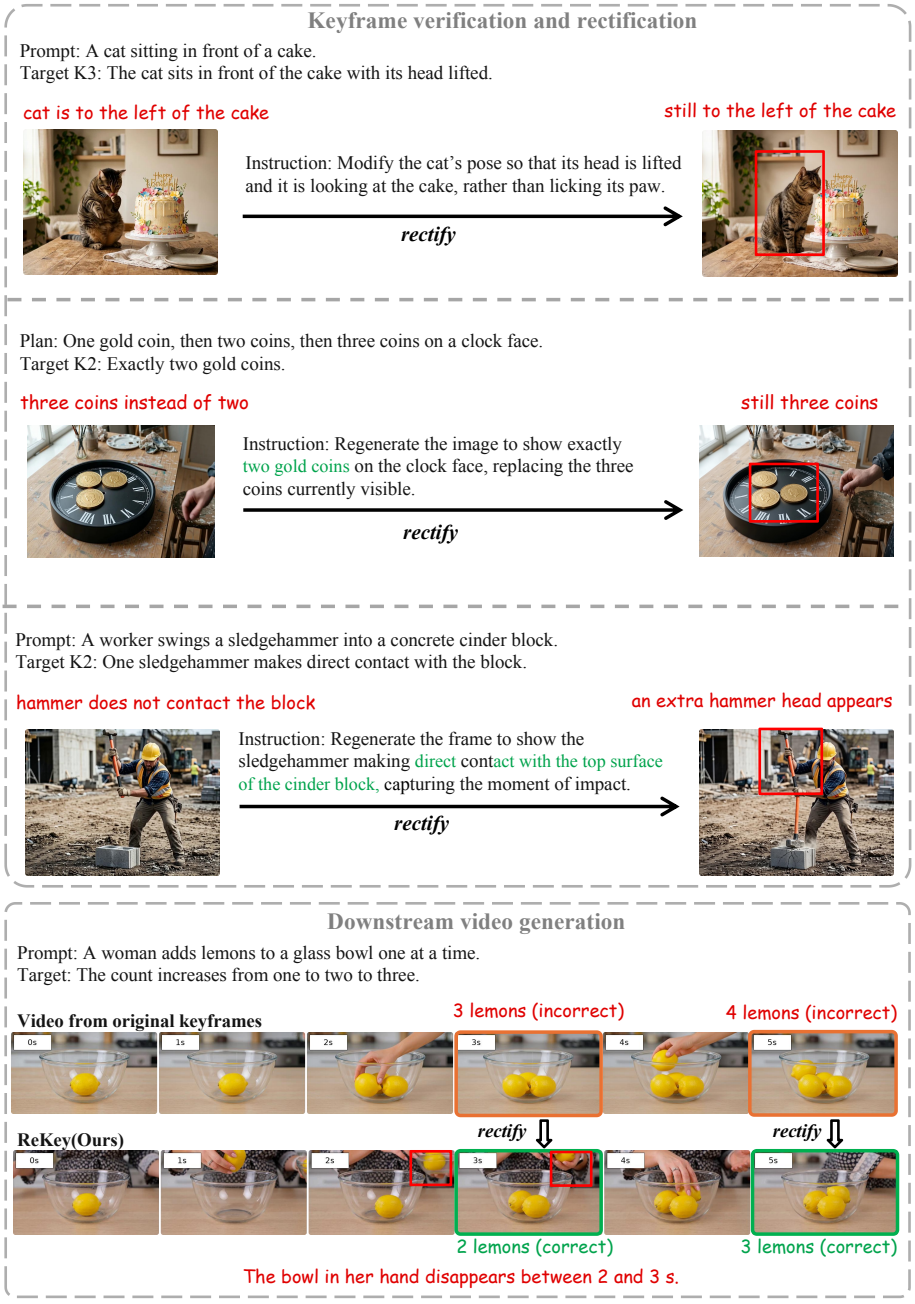

Figure 15: **Failure cases in ReKey-Bench. Top:** The rectification instruction does not address the cat's required position in front of the cake; despite correct instructions, the image editor leaves three coins instead of two or introduces an extra hammer head. **Bottom:** The generated video shows the required lemon counts, but the bowl in the woman's hand disappears between 2 and 3 s.

F PROMPT DETAILS

In this section, we provide the prompt templates used for student inference, teacher guidance, and evaluation. Blue text marks variable content; images are attached separately in the specified order.

F.1 STUDENT INFERENCE PROMPT

For K2 and K3, Image 1 is the preceding reference keyframe and Image 2 is the current candidate. K1 uses only the candidate opening keyframe. The teacher-guidance block is empty during student inference. The shared visual-assessment instruction and JSON schema are shown below the K1 template.

Student inference: intermediate and final keyframes (K2 / K3)

You are verifying a candidate keyframe from a coarse video.
Original prompt: `${prompt}`
Target frame: Frame `${frame_id}`
Previous state or transition context: `${previous_context}`
Required progression from the previous keyframe to this keyframe: `${target_progression}`
Attached images: Image 1 is the previous reference keyframe.
Image 2 is the current candidate keyframe to judge.
Task: Determine whether Image 2 is a valid next keyframe for the original prompt and the intended visual progression from Image 1. If repair is needed, identify the decisive visible error, choose an appropriate repair type, and provide a concrete instruction for correcting Image 2. Use Image 1 only as the previous-state and continuity reference; do not propose a repair for Image 1.
`${teacher_guidance_block}`

Judgment criteria:

- Treat the original prompt as the primary requirement. Use the progression text as auxiliary temporal context, and ignore it if it conflicts with the original prompt.
- Check whether Image 2 contains the required subjects, explicit counts, visible attributes, action-role bindings, spatial relations, and target state.
- Check whether Image 2 shows the required visible progression from Image 1. Do not require an exact pose or exact timing beyond what the prompt and progression clearly require.
- Check continuity of subject identity, motion direction, spatial layout, static anchors, scale, viewpoint, and scene structure.
- Base the decision on clearly visible evidence. Do not infer hidden motion, unseen events, or errors from an ambiguous viewpoint or partial occlusion.
- Do not require repair for minor artifacts that do not change the prompt meaning, required progression, or continuity.

Repair selection:

- If Image 2 is acceptable, set `needs_rect=false` and `repair_type="keep"`.
- Use `repair_type="local_edit"` only when the required subjects, scene, progression, and continuity are already correct and the remaining problem is localized and low risk.
- Use `repair_type="regenerate"` when correcting the frame requires changing a main subject, explicit count, action-role binding, required relation, scene, temporal progression, or overall composition.
- Do not use `local_edit` to create missing motion, substantially relocate a main subject, or reconstruct the overall scene.
- If `needs_rect=true`, make `"instruction"` concrete and actionable by identifying the target and the required visible change.
- If `needs_rect=false`, set `"instruction"=""`.

Respond strictly in the following JSON format: `${reflection_json_schema}`
`${optional_reasoning_instruction}`

Student inference: opening keyframe (K1)

You are verifying the opening keyframe of a keyframe-guided video pipeline.
Original prompt: `${prompt}`
Opening plan: `${initial_state}`
Attached image: The image is the candidate opening keyframe.
Task: Determine whether the image is a valid opening state for the original prompt. If repair is needed, identify the decisive visible error, choose an appropriate repair type, and provide a concrete instruction for correcting the image.
`${teacher_guidance_block}`

Judgment criteria:

- Treat the original prompt as the primary requirement. Use the opening plan only as auxiliary context, and ignore it if it conflicts with the original prompt.
- Check the required main subjects, explicit counts, visible attributes, action-role bindings, initial spatial relations, and scene setup.
- Check for unsupported extra main subjects only when they change the prompt meaning.
- Do not require later motion, completed actions, final outcomes, or late transformations to appear in the opening keyframe.
- Base the decision on clearly visible evidence. Do not infer hidden events or errors from ambiguous or partially visible content.
- Do not require repair for minor artifacts that do not change the prompt meaning or opening state.

Repair selection:

- If the opening keyframe is acceptable, set `needs_rect=false` and `repair_type="keep"`.
- Use `repair_type="local_edit"` only when the required subjects, scene, layout, and opening state are already correct and the remaining problem is localized and low risk.
- Use `repair_type="regenerate"` when correcting the frame requires changing a main subject, explicit count, action-role binding, required relation, scene, opening state, or overall composition.
- If `needs_rect=true`, make `"instruction"` concrete and actionable by identifying the target and the required visible change.
- If `needs_rect=false`, set `"instruction"=""`.

Respond strictly in the following JSON format: `${reflection_json_schema}`
`${optional_reasoning_instruction}`

Shared response instructions and output schema

Visual assessment

Before providing the JSON response, first inspect the visible image evidence, including temporal continuity when multiple images are provided. Write clear and brief visual reasoning inside `<visual_assessment>`clear and brief visual reasoning`</visual_assessment>`, then provide the JSON object in the format above. Do not repeat the prompt, plan, previous assessment, or the same evidence.

JSON output

```
{
  "needs_rect": true or false,
  "repair_type": "keep, local_edit, or regenerate",
  "reason": "visible reason for the decision",
  "instruction": "actionable repair instruction for the current
 image when repair is needed"
}
```

The visual assessment precedes the JSON response. `needs_rect=true` corresponds to `RECT`, and `false` to `KEEP`.

F.2 Teacher Guidance

The teacher receives the same task and images as the student. Label supplies verification labels, QA supplies atomic question–answer pairs, and QAE adds evidence. QA+ appends repeated QA statements to match the QAE teacher-prompt length within two text tokens without adding evidence.

Privileged teacher guidance

Label-guided OPSD
Use the following target-frame decision annotation as an additional check. Verify the candidate decision against the attached images before making the decision. If the annotation conflicts with visible image content, follow the images.
Target-frame `needs_rect`: `${needs_rect}`

QA-guided OPSD
Use the following atomic visual annotations as additional checks. Verify each candidate answer against the attached images before making the decision. If a candidate answer conflicts with visible image content, follow the images.
`${details}`

QA+-guided OPSD: appended to QA guidance
QA restatement (same questions and candidate answers):
`[Repeated QA statements]`

QAE-guided OPSD (ReKey)
Use the following atomic visual annotations as additional checks. Verify each candidate answer and its evidence against the attached images before making the decision. If an annotation conflicts with visible image content, follow the images.
`${details}`

Atomic question format

```
- [scope/criterion]
  Question: atomic question
  Candidate answer: yes or no
  Visual evidence: supporting visual description
```

F.3 Keyframe Evaluation Prompt

The evaluator receives the task, atomic questions, and images with explicit reference and target roles. It returns an answer and brief visual evidence for each question.

Keyframe evaluation: visible requirements and QA output

Evaluate the current benchmark stage using the attached images and the provided yes/no questions. Use only visible image content, the original prompt, and the accepted three-state plan shown below. The original prompt defines the global task semantics; the plan defines the required visible state at this stage.
Attached image mapping: `${image_mapping}`
Answer every question independently. Answer yes only when the question's visible claim is satisfied. Use missing only when an attached image is unavailable or unreadable. Do not infer hidden motion beyond the shown images.

Return exactly one JSON object with this schema:

```
{
  "task_id": "...",
  "question_answers": [
    {"question_id": "...", "answer": "yes|no|missing", "evidence":
  "brief visible evidence"}
  ]
}
```

Task metadata: `${task_metadata}`
Questions: `${questions}`

F.4 VIDEO EVALUATION PROMPT

The main ReKey-Bench video evaluator uses Gemini 3.1 Pro (`gemini-3.1-pro-preview`) at temperature 0, with a 4,096-token response limit. Each question is evaluated separately using 32 uniformly sampled frames, resized to a maximum long edge of 768 pixels and encoded as JPEGs with quality 85. Sampled frames are attached in temporal order.

Video evaluation: system instruction, user prompt, and JSON format

System:

Return one JSON object for the video QA request. Do not include markdown, code fences, notes, or chain-of-thought outside JSON.

User:

Evaluate the attached video against the listed yes/no questions.

Use only visible content in the sampled video frames and the original prompt. Answer each question independently. Minor artifacts or style changes are acceptable unless they change the prompt meaning or contradict the question.

Return exactly one valid JSON object matching this schema:

```
{
  "case_id": "${case_id}",
  "question_answers": [
    {
      "task_id": "same as input question item",
      "question_id": "same as input question item",
      "answer": "yes or no",
      "evidence": "brief visible evidence; no chain-of-thought",
      "attributed_frame_ids": [
        "optional sampled frame ids such as f01, f16, f32;
 otherwise []"
      ]
    }
  ]
}
```

Input:

```
{
  "case_id": "${case_id}",
  "method": "${method}",
  "original_prompt": "${original_prompt}",
  "attached_video": "${video_name}",
  "questions": [
    {
      "task_id": "${task_id}",
      "question_id": "${question_id}",
      "criterion": "${criterion}",
      "question": "${question}",
      "checks": "${checks}",
      "scope": "${scope}"
    }
  ]
}
```