Video Models Can Reason with Verifiable Rewards

Anonymous Author(s)
Affiliation
Address
email

Abstract

Video diffusion models have made rapid progress in perceptual realism and temporal coherence, but they remain primarily optimized for plausible generation rather than verifiable reasoning. This limitation is especially pronounced in tasks where generated videos must satisfy explicit spatial, temporal, or logical constraints. Inspired by the role of reinforcement learning with verifiable rewards (RLVR) in reasoning-oriented language models, we introduce **VideoRLVR**, a practical recipe for optimizing video diffusion models with rule-based feedback. VideoRLVR formulates video reasoning as the generation of verifiable visual trajectories and consists of an SDE-GRPO optimization backbone, dense decomposed rewards, and an *Early-Step Focus* strategy for efficient training. The Early-Step Focus strategy restricts policy optimization to the early denoising phase, reducing training latency by about 40% while preserving performance. We evaluate VideoRLVR on Maze, FlowFree, and Sokoban, three procedurally generated domains with objective success criteria. Across these tasks, VideoRLVR consistently improves over supervised fine-tuning baselines, with dense decomposed rewards proving especially important in low-success-rate settings. Our RL-optimized model also outperforms the evaluated proprietary and open-source video generation models on these verifiable reasoning benchmarks. These results suggest that verifiable RL can move video models beyond perceptual imitation toward more reliable rule-consistent visual reasoning.

1 Introduction

Recent progress in large language models (LLMs) has reshaped the role of generative models from content producers into increasingly capable reasoning systems [11, 32, 6]. A key intuition behind this shift is that the model can externalize the problem-solving process by generating intermediate states rather than only a final answer. This raises a natural question for video generation: *if language models can reason through sequences of tokens, can video models reason through sequences of frames?* Videos provide an appealing foundation for this idea, where each frame can represent an intermediate visual state in a goal-directed process. In domains such as navigation [7], puzzle solving [14], and embodied planning [27], a generated video can therefore be viewed not merely as motion synthesis, but as a temporally ordered chain of visual states [39] that encodes a visual reasoning trajectory.

Despite this potential, current video diffusion models are still primarily optimized for perceptual quality, temporal coherence, and plausible motion [13, 45, 36]. While large-scale video models have begun to show signs of visual reasoning [39, 12, 37], these abilities remain difficult to elicit reliably and verify under standard training objectives. The core challenge is the mismatch between perceptual plausibility and objective correctness. Supervised fine-tuning (SFT) on ground-truth solution videos can teach the model the visual form of valid trajectories, yet it does not directly optimize the correctness of sampled outputs. As a result, models may imitate solution-like patterns while failing to satisfy the underlying rules that make those solutions valid [9, 28]. This suggests an analogy to reasoning-oriented LLMs where pre-training provides broad generative competence, SFT

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

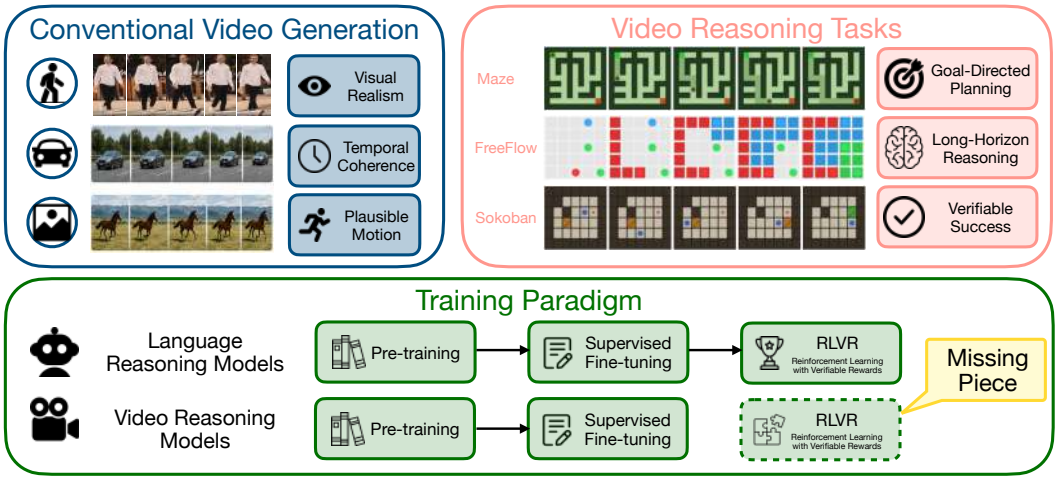

Figure 1: **Evolution towards verifiable video reasoning.** *Top:* Comparison between perception-focused generation and reasoning-intensive tasks. *Bottom:* We introduce **VideoRLVR**, the missing puzzle in the training paradigm for video reasoning models.

teaches the format of reasoning traces, Reinforcement Learning with Verifiable Rewards (RLVR) is the essential third stage required to optimize objective correctness.

In this work, we introduce **VideoRLVR**, a systematic recipe for applying reinforcement learning with verifiable rewards to video models. Our framework has four main components. First, we construct verifiable video reasoning data by generating solution trajectories with rule-based planners and aligning each logical transition with the video frame sequence. Second, we adopt an SDE-GRPO optimization backbone [24] for optimizing flow-matching video models. Third, we propose an *Early-Step Focus* strategy for efficient video RL. Instead of applying stochastic exploration and backpropagation across the entire denoising trajectory, this strategy concentrates optimization on the early denoising phase, where coarse structure and long-range planning are largely determined [38]. Finally, we design dense decomposed rewards that break sparse task success into verifiable structural components, providing informative feedback even when full success is rare.

We evaluate our RLVR recipe on a multi-task suite designed for rule-based verification, including Maze, FlowFree, and Sokoban. Our experiments show that VideoRLVR improves video reasoning beyond supervised imitation. Across all three domains, the RL-optimized model consistently achieves higher success rates than the SFT checkpoint used to initialize training, with gains of 6.1%, 5.5%, and 3.2% on Maze, FlowFree, and Sokoban, respectively. Compared with continued supervised training, VideoRLVR yields larger gains on harder tasks, suggesting that verifiable rewards provide an optimization signal beyond what can be captured by imitation alone. We further evaluate VideoRLVR on the out-of-domain split of VBVR [37], where VideoRLVR shows improved transfer beyond the training domains. Our ablations further show that dense decomposed rewards are crucial in low-success-rate domains, and that Early-Step Focus reduces training time by about 40% while maintaining nearly the same performance Finally, VideoRLVR outperforms several proprietary and open-source video generation models on our verifiable reasoning benchmarks, indicating that targeted verifiable RL can substantially improve the logical correctness of generated visual trajectories.

In summary, our contributions are as follows:

1. We introduce **VideoRLVR**, a reinforcement learning framework that optimizes video diffusion models with verifiable rewards, including dense decomposed reward functions to provide informative feedback for rule-verifiable visual trajectories.
2. We introduce a scalable training pipeline that combines rule-based trajectory generation, SDE-GRPO optimization, dense decomposed rewards, and an Early-Step Focus strategy that reduces training time by about 40% while preserving the performance.
3. We show that VideoRLVR improves over supervised fine-tuning and competitive proprietary and open-source video generation models on Maze, FlowFree, and Sokoban, while also demonstrating improved out-of-domain transfer on VBVR.

2 Related Work

Reinforcement learning for diffusion and flow models. Reinforcement learning has increasingly been used to align diffusion and flow-based generative models with human preferences, perceptual objectives, and task-specific rewards [42]. Prior work formulates denoising as a sequential decision process and applies policy-gradient or preference-optimization methods to improve text-to-image and video generation [2, 8, 35]. For flow-matching models, recent methods address the deterministic nature of ODE sampling by introducing stochastic transitions or alternative preference objectives, enabling likelihood-ratio or GRPO-style optimization [24, 43, 5]. Other extensions apply these ideas to video or embodied objectives [1, 25]. However, existing work optimizes perceptual or preference-based criteria such as aesthetics, text rendering, image fidelity, geometric consistency, or motion quality [21, 22]. In contrast, our work studies reinforcement learning for *verifiable video reasoning*, where rewards are computed from objective task rules and success depends on the logical correctness of the generated visual trajectory.

Reasoning in video generation models. Recent work has begun to investigate whether video generation models can serve as reasoning systems rather than only visual synthesizers. Large-scale video models have shown emerging abilities on visual puzzles and sequential prediction tasks, motivating the view that video generation can be interpreted as a chain of visual states or "chain of frames" [39, 12, 18]. Benchmark efforts [37, 4, 44, 34] further evaluate video models on reasoning-oriented tasks that require temporal consistency, spatial planning, or rule satisfaction. Other studies analyze video models as world simulators or physical reasoners, highlighting both their potential and their limitations in capturing causal and physical structure [3, 19, 27, 28, 47, 33]. These works suggest that video models may contain useful visual reasoning priors, but also show that standard generation objectives do not reliably produce rule-correct trajectories [12, 26]. Our work addresses this gap by directly optimizing video models with verifiable rewards, using rule-based success criteria rather than relying solely on supervised imitation or zero-shot generation.

Verifiable reinforcement learning and reasoning models. Reinforcement learning with verifiable rewards has played an important role in recent progress on reasoning-oriented language models [11, 32, 6]. In these settings, the model is rewarded according to objective correctness signals, such as mathematical equivalence, executable code tests, or rule-based verification, instead of only human preference judgments [23, 46, 16, 17]. This paradigm is attractive because it provides scalable supervision when outcomes can be automatically checked, which facilitates the development of emerging behaviors like searching and backtracking [48, 41]. Our work extends this training from language outputs to video trajectories. Whereas text reasoning is often verified by final-answer correctness, video reasoning requires trajectory-level verification over visual, temporal, and process constraints. We study how verifiable RL can optimize video diffusion models under these criteria.

3 Problem Formulation

RLVR for Video Reasoning. Following [39], we formulate video reasoning as a conditional generation task where a model generates a temporal sequence of visual states whose transitions and terminal state can be checked against task-specific rules. Given an initial image I_0 and a textual instruction T, let $c = (I_0, T)$ denote the conditioning input. The model generates a video $\mathbf{V} = \{I_0, I_1, \ldots, I_{F-1}\}$, where F is the number of frames. Unlike standard video synthesis, which primarily evaluates perceptual quality and temporal coherence, video reasoning requires the generated sequence to satisfy task-specific correctness criteria. This formulation allows us to treat video generation as a search for a valid visual trajectory conditioned on the initial state and instruction.

Video Generation as a Markov Decision Process. To apply reinforcement learning to flow-matching video generation, we formulate the reverse denoising process as a Markov Decision Process (MDP) over latent variables. This MDP is defined over denoising steps rather than reasoning steps, where the reward is computed after the final video is decoded. At denoising step $k \in \{1, \ldots, K\}$, the state is the noisy video latent x_{t_k} at noise level t_k, and the action is the model velocity prediction $\hat{v}_\theta(x_{t_k}, t_k, c)$, which determines the mean update of the next latent. Under the deterministic ODE solver, the transition is given by $x_{t_{k+1}} = x_{t_k} + (t_{k+1} - t_k)a_k$. After the final denoising step, the decoded video $\mathbf{V}$ receives a verifier-derived reward $R(\mathbf{V}, c)$. A fundamental challenge in this formulation is that standard flow matching employs a deterministic ODE solver, making it a deterministic function

of the initial noise x_1. Under this deterministic solver, the next latent is a deterministic function of (x_{t_k}, c), yielding no tractable stochastic transition density $\pi_\theta(x_{t_{k+1}} \mid x_{t_k}, c)$ for likelihood-ratio policy gradients. In Section 4, we address this by adopting an SDE-based formulation that introduces stochastic transitions compatible with flow-matching generation.

Tasks. To evaluate VideoRLVR across different reasoning domains, we instantiate our framework on three rule-verifiable visual reasoning domains: Maze, FlowFree, and Sokoban. We choose these tasks because they satisfy three properties: 1) solution correctness can be checked by rule-based verifiers, 2) large-scale training and test instances can be generated, and 3) the tasks span different levels of reasoning complexity. Maze primarily tests spatial connectivity under explicit obstacle constraints, FlowFree requires globally consistent non-overlapping path connectivity and implicit constraints, and Sokoban introduces object interaction, irreversible transitions, and longer-horizon reasoning.

4 RLVR Recipe for Video Reasoning Models

We present **VideoRLVR**, a systematic recipe for optimizing video models with verifiable rewards. The recipe consists of four components: 1) rule-based data curation, 2) an SDE-GRPO optimization backbone, 3) dense decomposed rewards design, and 4) an Early-Step Focus optimization strategy.

4.1 Data Curation

Existing video reasoning datasets [44, 37] often lack the scale, task diversity, or fine-grained difficulty variation required to study RLVR for high-dimensional video generation. We therefore synthesize task instances with rule-based planners, producing conditioning inputs $c = (I_0, T)$ and valid video trajectories, together with metadata for automatic verification and reward computation. To align visual generation with symbolic reasoning, we map each discrete action to a unique frame transition $I_f \to I_{f+1}$. This action-to-frame mapping makes the generated video directly interpretable as a reasoning trajectory and enables the verifier to evaluate both terminal correctness and process-level validity. Dataset scale and task-specific generation details are provided in Section 5.

4.2 SDE-GRPO for Video Reasoning

GRPO [31] estimates relative advantages from groups of sampled outputs without training a separate critic, making it well suited for verifiable reward settings. However, standard flow-matching models generate samples with a deterministic ODE sampler, which does not provide a tractable stochastic transition density over denoising steps. Following Flow-GRPO [24], we convert the deterministic denoising dynamics into stochastic transitions with Gaussian log-probabilities.

Stochastic denoising transitions.. For a discretized denoising schedule $\{t_k\}_{k=1}^K$, the SDE formulation defines a Gaussian transition:

$$\pi_\theta(x_{t_{k+1}} \mid x_{t_k}, c) = \mathcal{N}\big(x_{t_{k+1}}; \mu_\theta(x_{t_k}, t_k, c), \sigma_k^2 \mathbf{I}\big)\,, \tag{1}$$

where $\mu_\theta(x_{t_k}, t_k, c)$ is the mean update induced by the model and σ_k^2 is the SDE transition variance. This stochastic transition enables closed-form log-probabilities and likelihood-ratio policy gradients.

GRPO objective. Given a group of G sampled videos for each condition, we compute verifier-derived rewards and normalize them within the group to obtain advantages A_i. For each sample i and denoising step k, we compute the dimension-normalized log-ratio:

$$\log \rho_{i,k} = \log \frac{\pi_\theta(x^{(i)}_{t_{k+1}} \mid x^{(i)}_{t_k}, c_i)}{\pi_{\text{old}}(x^{(i)}_{t_{k+1}} \mid x^{(i)}_{t_k}, c_i)} = -\frac{1}{2\sigma_k^2} \cdot \frac{1}{D} \sum_{d=1}^{D} \left[\left(x^{(i)}_{t_{k+1}} - \mu_\theta^{(i,k)} \right)_d^2 - \left(x^{(i)}_{t_{k+1}} - \mu_{\text{old}}^{(i,k)} \right)_d^2 \right], \tag{2}$$

where $\mu_\theta^{(i,k)} = \mu_\theta(x^{(i)}_{t_k}, t_k, c_i)$, $\mu_{\text{old}}^{(i,k)} = \mu_{\text{old}}(x^{(i)}_{t_k}, t_k, c_i)$, and D is the number of latent elements. The policy loss uses PPO-style clipping:

$$\mathcal{L}_{\text{policy}} = -\mathbb{E}_{i,k} \left[\min\left(\rho_{i,k} A_i,\ \text{clip}(\rho_{i,k}, 1-\varepsilon, 1+\varepsilon) A_i \right) \right]. \tag{3}$$

We additionally regularize the policy against the reference model with a closed-form KL penalty:

$$\mathcal{L}_{\text{KL}} = \mathbb{E}_k \left[\frac{1}{D} \frac{\|\mu_\theta - \mu_{\text{ref}}\|_2^2}{2\sigma_k^2} \right]. \tag{4}$$

The final objective is $\mathcal{L}_{\text{VideoRLVR}} = \mathcal{L}_{\text{policy}} + \beta \mathcal{L}_{\text{KL}}$,where β controls the strength of regularization.

4.3 Early-Step Focus for Efficient Video RL

Video RL is substantially more expensive than text RL because each rollout requires generating and backpropagating through high-dimensional spatio-temporal latents. A full SDE-GRPO update over all K denoising steps therefore incurs large memory and time costs. However, not all denoising steps contribute equally to the reasoning objective. Early high-noise steps are primarily responsible for coarse layout, object placement, and long-range structure, whereas later low-noise steps mainly refine local appearance and consolidate the generation into a specific visual trajectory [38].

Motivated by this observation, we introduce *Early-Step Focus*. During RL optimization, we sample the full denoising trajectory for generation and reward evaluation, but restrict stochastic perturbation, log-probability computation, and gradient backpropagation to the first $L < K$ denoising steps. This creates an efficient exploration-exploitation trade-off: early denoising steps receive stochastic perturbations and policy-gradient updates for high-level reasoning, while later steps preserve the generative prior and refine visual details. The policy loss becomes:

$$\mathcal{L}_{\text{ESF}} = -\mathbb{E}_{i,k\le L}\left[\min\left(\rho_{i,k}A_i,\ \text{clip}(\rho_{i,k}, 1-\varepsilon, 1+\varepsilon)A_i\right)\right] + \beta\mathcal{L}_{\text{KL}}^{k\le L}. \tag{5}$$

In our experiments, we use $K = 20$ denoising steps and $L = 10$ early steps. This reduces training latency by about 40% while preserving reasoning performance, suggesting that the early denoising phase carries most of the reward-relevant structural signal.

4.4 Reward Design for Video Reasoning

Verifiable reasoning tasks provide objective rules that can be converted into dense reward signals. Instead of using only a binary success indicator, we decompose each task into structural components that measure partial progress toward a valid solution. This is especially important in low-success-rate domains, where most sampled videos fail completely and binary rewards provide little variation within a GRPO group.

Task-aware Reward Function. We use a task-aware reward function for joint training across heterogeneous domains. For each conditioning input c, the dispatcher identifies the task $\mathcal{T}(c) \in \{\text{Maze}, \text{FlowFree}, \text{Sokoban}\}$ and evaluates the generated video with the corresponding reward:

$$R(\mathbf{V}, c) = R_{\mathcal{T}(c)}(\mathbf{V}, c). \tag{6}$$

This allows mixed-task RL batches while preserving task-specific verification criteria.

Dense Reward Formulations. For each task, we decompose the global objective into measurable rule-based components:

- **Maze.** We define the reward as: $R_{\text{maze}} = c_{\text{conn}} \cdot w_{\text{wall}}$, where c_{conn} measures start-to-goal path connectivity and w_{wall} penalizes wall violations. The multiplicative form couples connectivity with wall consistency, discouraging connected paths that violate maze constraints.
- **FlowFree.** We combine four structural metrics: $R_{\text{ff}} = \lambda_{\text{valid}}p_{\text{valid}} + \lambda_{\text{ep}}\text{ep}_{\text{pres}} + \lambda_{\text{conn}}c_{\text{conn}} + \lambda_{\text{fill}}\text{fill}_{\text{rate}}$, where p_{valid} measures endpoint-to-endpoint path validity, ep_{pres} measures preservation of the given endpoints, c_{conn} measures 4-connected color regions, and $\text{fill}_{\text{rate}}$ measures grid coverage by valid path colors. The weights $\lambda_{\text{valid}}, \lambda_{\text{ep}}, \lambda_{\text{conn}}, \lambda_{\text{fill}}$ balance the relative importance of these components. In our experiments, we set them to 0.15, 0.35, 0.30, and 0.20, respectively.
- **Sokoban.** We use a combination of final-state and process-validity rewards: $R_{\text{sok}} = \lambda_{\text{state}}R_{\text{state}} + \lambda_{\text{proc}}R_{\text{proc}}$, where R_{state} measures box placement on target cells and R_{proc} measures the fraction of valid transitions under Sokoban movement rules. The weights λ_{state} and λ_{proc} balance final-state correctness and process validity. We use $\lambda_{\text{state}} = \lambda_{\text{proc}} = 0.5$ in all experiments.

5 Experiments

In this section, we evaluate **VideoRLVR** from two perspectives. First, we compare against supervised fine-tuning and competitive video generation baselines on three rule-verifiable reasoning domains: Maze, FlowFree, and Sokoban. Then, we test transfer beyond the training domains using the out-of-domain split of VBVR [37]. Together, these experiments assess whether verifiable RL improves both in-domain rule-based correctness and out-of-domain visual reasoning behavior.

Table 1: Comparison of our method with other state-of-the-art method. We report both Success Rate (SR) and GT alignment metrics: Precision (Prec), Recall (Rec), and F1. **Bold** indicates the best and underlined indicates second best.

Model	Maze				FlowFree				Sokoban			
	Prec	Rec	F1	SR	Prec	Rec	F1	SR	Prec	Rec	F1	SR
Proprietary Models												
Sora 2	15.8	17.2	16.5	3.1	10.8	5.1	5.8	0.0	8.5	4.8	5.4	0.0
Kling V3	24.8	15.7	19.2	23.5	18.8	2.7	4.7	0.0	5.7	2.7	3.7	0.0
Veo 3.1	22.8	18.1	20.2	26.0	23.9	4.7	7.5	4.0	22.2	6.0	9.4	0.0
Open-Source Models												
CogVideoX1.5	13.3	10.8	11.9	0.0	18.7	2.2	3.9	0.0	3.2	0.3	0.5	0.0
HunyuanVideo	17.3	11.4	13.8	2.2	12.5	2.9	4.8	0.0	8.2	2.7	3.2	0.0
Wan2.2-TI2V-5B	18.3	12.2	14.6	0.0	17.4	2.0	3.4	0.0	4.1	0.7	1.0	0.0
SFT Models												
Wan-R1	20.9	65.6	31.7	31.9	20.9	3.6	6.1	0.0	7.7	2.1	3.3	0.0
VBVR-Wan2.2	62.7	77.8	69.4	60.8	17.9	5.6	8.5	1.7	16.2	1.7	3.1	0.0
SFT Epoch 5	80.2	83.0	81.6	66.1	42.8	42.2	42.4	2.4	33.6	11.9	17.6	2.9
SFT Epoch 10	80.4	85.1	82.7	69.0	43.1	42.5	42.8	2.5	32.8	11.6	17.1	2.7
RL Model												
Ours	82.1	86.9	84.4	**72.2**	44.3	43.8	44.0	**7.9**	34.0	12.5	29.4	**6.1**

5.1 Experimental Setup

Dataset. We train and evaluate on a multi-task suite of three procedurally generated reasoning domains: Maze, FlowFree, and Sokoban. To prevent the model from overfitting to specific visual features, we apply varied color themes across the dataset, encouraging the model to rely on structural invariants. Each sample consists of an input image, a task instruction, and an 81-frame ground-truth video at 480×832 resolution. The total training dataset consists of 30,000 samples (1,0000 per task). For the test set, we maintain a held-out set of 3,000 samples (1,000 per task) generated with disjoint random seeds. Dataset construction details are provided in Section A.1.

Base Model and SFT Baseline. We use Wan2.2-TI2V-5B [36], a state-of-the-art video generation model, as our base model. It generates $F = 81$ frames at 480×832 resolution. We first establish an SFT baseline by training the model on ground-truth solution videos using the standard flow matching objective. This SFT checkpoint provides the necessary perceptual and structural prior for the model, serving as both the initial policy and the reference policy π_{ref} for RL optimization.

Baselines. To evaluate the effectiveness of our method, we compare our model with competitive proprietary and open-source video generation baselines. For proprietary models, we use Sora 2 [29], Kling V3 [20], and Veo 3.1 [39]. For open-source models, we compare with Wan2.2-TI2V-5B [36], CogVideoX1.5-5B-I2V [13], and HunyuanVideo-I2V [40]. We also compare with specialized SFT-based video reasoning models, including Wan-R1 [44] and VBVR-Wan2.2 [37]. Wan-R1 adopts the same base model as ours and is trained on the Maze and Sokoban domains with LoRA [15]. VBVR-Wan2.2 utilizes Wan2.2-I2V-A14B [36] and is trained on the VBVR dataset with LoRA.

Training Configuration. We train the SFT baseline for 5 epochs with a learning rate of 1×10^{-5}. For VideoRLVR, we initialize from the SFT checkpoint and train on the same training set for 1 epoch using SDE-GRPO as the optimization backbone. We use group size $G = 16$, $T = 20$ denoising steps, learning rate 5×10^{-6}, and KL coefficient $\beta = 0.04$. Following the Early-Step Focus strategy in Section 4.3, backpropagation and SDE injection is restricted to the first $L = 10$ steps of the denoising trajectory. All training experiments are conducted on 8 NVIDIA B200 GPUs.

Evaluation. We evaluate the results using two complementary metric families: 1) trajectory alignment metrics, including Precision (Prec), Recall (Rec), and F1, which measure pixel-, cell-, or action-level alignment with the reference solution, and 2) symbolic success rate, which verifies whether the video satisfies the underlying task rules. Evaluation details are listed in Section A.2

5.2 Main Results

Table 1 compares VideoRLVR with supervised baselines and competitive video generation models on our verifiable reasoning benchmarks.

RLVR consistently outperforms supervised baselines. VideoRLVR yields consistent improvements across all three reasoning domains. Compared with the SFT Epoch 5 checkpoint used to initialize RL training, VideoRLVR improves success rate by 6.1% on Maze, 5.5% on FlowFree, and 3.2% on Sokoban. Notably, VideoRLVR also significantly surpasses the performance of recent state-of-the-art closed-source models on visual reasoning tasks, validating the efficacy of verifiable reinforcement learning in domains where generic video pre-training remains insufficient for complex logical tasks.

Superior scaling on high-complexity tasks. To isolate the specific advantages of RLVR over extended supervised learning, we evaluate a stronger SFT baseline (SFT Epoch 10), representing the result of conducting further supervised training on the same Epoch 5 checkpoint. As shown in Table 1 and Figure 2, VideoRLVR is more robust than continued SFT as task difficulty increases. Within the Maze domain, RLVR establishes a 3.2% margin over the SFT Epoch 10 checkpoint and shows less degradation when the scale of the maze increases. On FlowFree, VideoRLVR improves over SFT Epoch 10 by 5.4%, while continued SFT provides little improvement over the Epoch 5 checkpoint. On Sokoban, continued SFT slightly degrades performance, whereas VideoRLVR improves over SFT Epoch 10 by 3.4% These trends suggest that verifiable rewards provide an optimization signal that is not captured by additional imitation training alone.

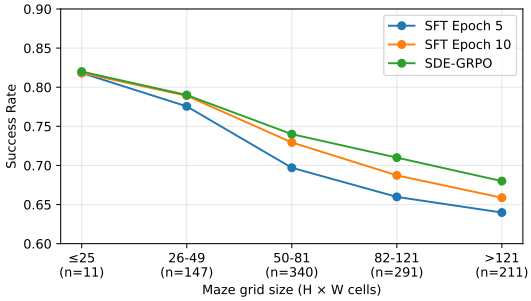

Figure 2: Success rate of different grid size for maze. n represents the number of samples falling into this range

5.3 Comparison with LLMs

To determine if our reasoning domains can be solved by language reasoning alone, we benchmark frontier LLMs on the Maze task. Table 2 presents the results of state-of-the-art models, including GPT-5.5 Pro [30] and Gemini 3.1 Pro [10], compared against our RLVR-optimized video model. Despite their sophisticated reasoning capabilities in textual domains, LLMs exhibit a sharp performance decay in maze tasks. This divergence highlights a representation bottleneck: while LLMs must reason over a tokenized rendering of the maze, our video model operates directly on a visual latent space that inherently preserves the visual topological relationships necessary for complex visual reasoning. These results suggest that, for visual reasoning, directly generating and optimizing visual trajectories can be more effective than solving the task through language-token representations alone.

Table 2: Comparison with LLMs on maze tasks.

Model	Maze			
	Prec	Rec	F1	SR
GPT 4o	11.7	13.0	12.3	0.0
GPT 5.5 Pro	76.0	70.1	72.9	66.0
Gemini 2.5 Flash	11.2	10.5	10.9	0.0
Gemini 3.1 Pro	26.8	27.0	26.9	23.0
Ours	**82.1**	**86.9**	**84.4**	**72.2**

5.4 OOD Results

To evaluate whether VideoRLVR transfers beyond the training domains, we test our model on the out-of-domain split of VBVR [37]. This benchmark covers multiple reasoning categories and therefore provides a broader test of whether VideoRLVR improves general video reasoning behavior beyond Maze, FlowFree, and Sokoban. As shown in Table 3, VideoRLVR substantially improves over the 5B baseline, increasing the average score from 26.2 to 60.2 with gains across all VBVR-OOD categories. VideoRLVR also performs competitively with the larger 14B VBVR-Wan2.2 model, achieving a similar average score despite using a smaller 5B backbone and

Table 3: **OOD evaluation on VBVR.** We report average performance and category-wise scores on the VBVR-OOD split.

Model	**Avg.**	**Abst.**	**Know.**	**Perc.**	**Spat.**	**Trans.**
5B Models						
CogVideoX1.5	26.2	28.1	23.5	25.0	25.4	28.2
VideoRLVR	60.2	65.5	**62.0**	**59.7**	58.8	58.2
14B Models						
Wan2.2-I2V-A14B	32.9	40.5	30.8	34.3	23.6	30.7
VBVR-Wan2.2	**61.0**	**76.8**	57.2	54.7	**61.8**	**61.5**

much less training data. These results suggest that VideoRLVR learns transferable visual reasoning ability that generalizes beyond the generated training tasks.

6 Analysis

In this section, we analyze the main components of VideoRLVR, including GRPO group size, Early-Step Focus, KL regularization, dense reward design, and qualitative generation behavior.

6.1 Ablation Study

Impact of Group Size. In GRPO, the group size G affects the stability of group-relative advantage estimation. We investigate the impact of this hyperparameter within the Maze reasoning domain, as shown in Figure 3. Our results indicate that performance scales positively with group size, primarily due to the stabilization of the reward distribution's statistics. While the expected sample standard deviation of rewards is a property of the policy's diversity, a small group size (e.g., $G \leq 4$) provides a noisy and often biased estimate of this value. This leads to significant fluctuations in the advantage calculation, as the group mean fails to accurately represent the current policy's performance level. Increasing the group size to $G = 16$ provides a more stable comparison set for estimating relative advantages, which improves training stability in our experiments. However, we observe diminishing returns beyond this point. Furthermore, because video generation remains a significant computational bottleneck, scaling G entails a linear increase in rollout time and VRAM overhead. We therefore use $G = 16$ as a practical trade-off between advantage-estimation stability and computational cost.

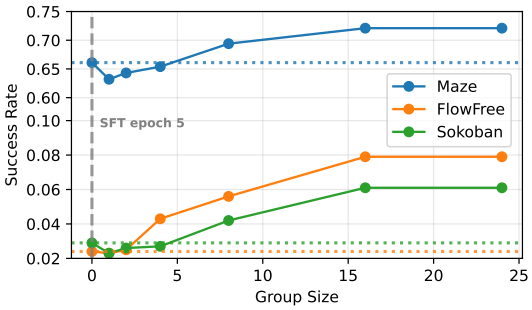

Figure 3: Scaling results with group size.

Early-Step Focus. To validate the efficacy of our Early-Step Focus strategy, we conduct a controlled experiment within the Maze domain. We fix the total inference budget at $T = 20$ denoising steps and compare the reasoning performance when the gradient and noise injection is calculated over the full trajectory ($L = 20$) versus the first $L = 10$ steps. As illustrated in Table 4, the success rates and F1 scores remain nearly unchanged, while training time is substantially reduced. This suggests that the early denoising steps carry much of the reward-relevant structural signal for visual reasoning. Because the later denoising steps primarily govern local textural refinement, they contribute less to the verifier-derived reasoning objective in this setting. Restricting backpropagation and noise injection to these early steps thus serves as a computationally efficient optimization path, significantly reducing the training time without degrading the performance.

Table 4: Comparisons of computing over full denoising steps and early step focus.

Gradient Steps	F1	SR	Time / step
20 (Full)	84.6	72.3	156 s
10 (Early 10 steps)	84.4	72.2	93.5 s

KL Constraint. The KL-divergence constraint is essential for maintaining the model's generative prior. We show a qualitative example in Section B.1. As shown in Figure 5, removing the KL penalty ($\beta = 0$) at an early stage of GRPO optimization can lead to reward-hacking behavior. Without this regularization, the model the model may produce visually implausible patterns that satisfy parts of the verifier while degrading generation quality. Implementing a constant penalty of $\beta = 0.04$ successfully anchors the optimization to the original quality, ensuring that improvements in logical success are achieved without sacrificing the model's inherent visual plausibility.

6.2 Reward Design

To evaluate the necessity of our dense decomposed reward design, we investigate the efficacy of a sparse reward ($R \in \{0, 1\}$) based exclusively on success rate. This ab-

Table 5: Training with a sparse binary success reward.

Steps	0	200	400	600	800	1000
Maze	66.1	67.2	68.9	70.1	71.5	72.9
FlowFree	2.4	2.3	2.5	2.4	2.6	2.5

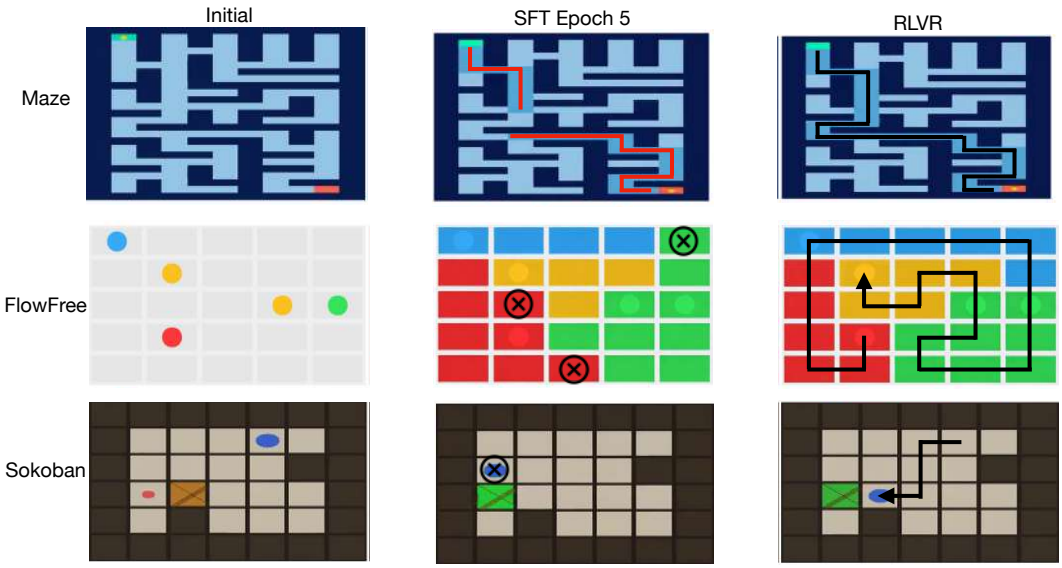

Figure 4: Qualitative case study across three reasoning domains. We compare generations from the SFT baseline (Epoch 5) and the final RLVR-optimized model.

lation aims to determine if binary feedback is sufficient across varying levels of task complexity.

Our results, shown in Table 5, reveal that sparse success rewards behave differently across domains. In domains like maze, where the baseline model already achieves a decent success rate, the sparse reward signal is sufficient to provide an informative gradient. The model is able to encounter success frequently enough during group rollouts to differentiate between advantageous and disadvantageous trajectories. Conversely, on high-complexity tasks like FlowFree and Sokoban, where the initial success rate is near-zero, the sparse reward provides little useful signal. In these environments, the model suffers from extreme gradient sparsity. Since success is rarely encountered within the group rollout G, the advantage estimates remain uninformative. This underscores a critical cold-start problem in video RL: while binary success is the ultimate goal, it is an insufficient signal for exploring high-dimensional latent space from a low-performance starting point. Dense decomposed rewards address this issue by providing partial credit for intermediate structural properties, giving the policy useful feedback before full success becomes frequent.

6.3 Case Study

Figure 4 provides qualitative examples across Maze, FlowFree, and Sokoban. The SFT baseline often captures the visual format of each domain, such as drawing paths, coloring grids, or rendering objects, but it can fail to satisfy the task rules. For example, SFT outputs may contain disconnected paths, inconsistent color connectivity, or invalid object transitions. In the Sokoban example, the SFT model produces a visually plausible but invalid shortcut rather than a valid sequence of box-pushing actions. In contrast, the VideoRLVR-optimized model more consistently satisfies the symbolic constraints checked by our verifiers. Across the shown examples, it produces connected paths, more coherent grid solutions, and more valid object transitions while preserving the overall visual structure of the task. These qualitative results support the quantitative findings: verifiable RL improves rule-based correctness beyond what is achieved by supervised imitation alone.

7 Conclusion

This work studies reinforcement learning with verifiable rewards for video reasoning and introduces **VideoRLVR**, a practical recipe for optimizing video reasoning models. By combining rule-verifiable data generation, an SDE-GRPO optimization backbone, dense decomposed rewards, and Early-Step Focus, VideoRLVR addresses the gap between perceptual video synthesis and task-level logical

correctness. Our experiments show that supervised fine-tuning provides an important visual and structural prior, but can plateau or degrade on harder reasoning tasks when optimized only through imitation. In contrast, verifiable RL improves success rates across Maze, FlowFree, and Sokoban, with dense decomposed rewards proving especially useful in low-success-rate domains. We further show that Early-Step Focus reduces training time by about 40% with little observed loss in reasoning performance. Overall, our results suggest that verifiable RL can substantially improve the logical correctness of generated videos, enabling open-source video models to outperform stronger general-purpose video generation baselines on visual reasoning benchmarks.

Limitations

Broader Impacts

References

[1] Zhaochong An, Orest Kupyn, Théo Uscidda, Andrea Colaco, Karan Ahuja, Serge Belongie, Mar Gonzalez-Franco, and Marta Tintore Gazulla. Vggrpo: Towards world-consistent video generation with 4d latent reward. *arXiv preprint arXiv:2603.26599*, 2026.

[2] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

[3] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Leo Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024.

[4] Zefan Cai, Haoyi Qiu, Tianyi Ma, Haozhe Zhao, Gengze Zhou, Kung-Hsiang Huang, Parisa Kordjamshidi, Minjia Zhang, Wen Xiao, Jiuxiang Gu, et al. Mmgr: Multi-modal generative reasoning. *arXiv preprint arXiv:2512.14691*, 2025.

[5] Wentse Chen, Shiyu Huang, Yuan Chiang, Tim Pearce, Wei-Wei Tu, Ting Chen, and Jun Zhu. Dgpo: discovering multiple strategies with diversity-guided policy optimization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 11390–11398, 2024.

[6] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[7] Yifei Dong, Fengyi Wu, Yilong Dai, Lingdong Kong, Guangyu Chen, Xu Zhu, Qiyu Hu, Tianyu Wang, Johnalbert Garnica, Feng Liu, et al. Language-conditioned world modeling for visual navigation. *arXiv preprint arXiv:2603.26741*, 2026.

[8] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36:79858–79885, 2023.

[9] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[10] Google DeepMind. Gemini 3.1 Pro Model Card. `https://deepmind.google/models/model-cards/gemini-3-1-pro/`, February 2026.

[11] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[12] Ziyu Guo, Xinyan Chen, Renrui Zhang, Ruichuan An, Yu Qi, Dongzhi Jiang, Xiangtai Li, Manyuan Zhang, Hongsheng Li, and Pheng-Ann Heng. Are video models ready as zero-shot reasoners? an empirical study with the mme-cof benchmark. *arXiv preprint arXiv:2510.26802*, 2025.

[13] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022.

[14] Sepidehsadat Sepid Hossieni, Mohammad Amin Shabani, Saghar Irandoust, and Yasutaka Furukawa. Puzzlefusion: Unleashing the power of diffusion models for spatial puzzle solving. *Advances in Neural Information Processing Systems*, 36:9574–9597, 2023.

[15] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

[16] Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*, 2025.

[17] Yixu Huang, Tinghui Zhu, and Muhao Chen. Learning adaptive reasoning paths for efficient visual reasoning. *arXiv preprint arXiv:2604.14568*, 2026.

[18] Ziqi Huang, Ning Yu, Gordon Chen, Haonan Qiu, Paul Debevec, and Ziwei Liu. Vchain: Chain-of-visual-thought for reasoning in video generation. *arXiv preprint arXiv:2510.05094*, 2025.

[19] Bingyi Kang, Yang Yue, Rui Lu, Zhijie Lin, Yang Zhao, Kaixin Wang, Gao Huang, and Jiashi Feng. How far is video generation from world model: A physical law perspective. *arXiv preprint arXiv:2411.02385*, 2024.

[20] Kling AI. All you need to know about kling video 3.0. `https://kling.ai/blog/kling-video-3-0-ai-director-features-guide`, February 2026.

[21] Rui Li, Yuanzhi Liang, Ziqi Ni, Haibing Huang, Chi Zhang, and Xuelong Li. Growing with the generator: Self-paced grpo for video generation. *arXiv preprint arXiv:2511.19356*, 2025.

[22] Yuming Li, Yikai Wang, Yuying Zhu, Zhongyu Zhao, Ming Lu, Qi She, and Shanghang Zhang. Branchgrpo: Stable and efficient grpo with structured branching in diffusion models. *arXiv preprint arXiv:2509.06040*, 2025.

[23] Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, et al. From system 1 to system 2: A survey of reasoning large language models. *arXiv preprint arXiv:2502.17419*, 2025.

[24] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025.

[25] Xiao Liu, Yifan Zhou, Fabian Weigend, Shubham Sonawani, Shuhei Ikemoto, and Heni Ben Amor. Diff-control: A stateful diffusion-based policy for imitation learning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7453–7460. IEEE, 2024.

[26] Yang Luo, Xuanlei Zhao, Baijiong Lin, Lingting Zhu, Liyao Tang, Yuqi Liu, Ying-Cong Chen, Shengju Qian, Xin Wang, and Yang You. V-reasonbench: Toward unified reasoning benchmark suite for video generation models. *arXiv preprint arXiv:2511.16668*, 2025.

[27] Zhiting Mei, Tenny Yin, Ola Shorinwa, Apurva Badithela, Zhonghe Zheng, Joseph Bruno, Madison Bland, Lihan Zha, Asher Hancock, Jaime Fernández Fisac, et al. Video generation models in robotics-applications, research challenges, future directions. *arXiv preprint arXiv:2601.07823*, 2026.

[28] Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models understand physical principles? In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 948–958, 2026.

[29] OpenAI. Sora 2 system card. `https://openai.com/index/sora-2-system-card/`, September 2025.

[30] OpenAI. GPT-5.5 System Card. `https://openai.com/index/gpt-5-5-system-card/`, April 2026.

[31] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[32] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

[33] Selena Song, Ziming Xu, Zijun Zhang, Kun Zhou, Jiaxian Guo, Lianhui Qin, and Biwei Huang. Learning plug-and-play memory for guiding video diffusion models. *arXiv preprint arXiv:2511.19229*, 2025.

[34] Jingqi Tong, Yurong Mou, Hangcheng Li, Mingzhe Li, Yongzhuo Yang, Ming Zhang, Qiguang Chen, Tianyi Liang, Xiaomeng Hu, Yining Zheng, et al. Thinking with video: Video generation as a promising multimodal reasoning paradigm. *arXiv preprint arXiv:2511.04570*, 2025.

[35] Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8228–8238, 2024.

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

[37] Maijunxian Wang, Ruisi Wang, Juyi Lin, Ran Ji, Thaddäus Wiedemer, Qingying Gao, Dezhi Luo, Yaoyao Qian, Lianyu Huang, Zelong Hong, et al. A very big video reasoning suite. *arXiv preprint arXiv:2602.20159*, 2026.

[38] Ruisi Wang, Zhongang Cai, Fanyi Pu, Junxiang Xu, Wanqi Yin, Maijunxian Wang, Ran Ji, Chenyang Gu, Bo Li, Ziqi Huang, et al. Demystifing video reasoning. *arXiv preprint arXiv:2603.16870*, 2026.

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

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

[41] Siye Wu, Jian Xie, Yikai Zhang, Aili Chen, Kai Zhang, Yu Su, and Yanghua Xiao. Arm: Adaptive reasoning model. *arXiv preprint arXiv:2505.20258*, 2025.

[42] Zeyue Xue, Siming Fu, Jie Huang, Shuai Lu, Haoran Li, Yijun Liu, Yuming Li, Xiaoxuan He, Mengzhao Chen, Haoyang Huang, et al. A systematic post-train framework for video generation. *arXiv preprint arXiv:2604.25427*, 2026.

[43] Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

[44] Cheng Yang, Haiyuan Wan, Yiran Peng, Xin Cheng, Zhaoyang Yu, Jiayi Zhang, Junchi Yu, Xinlei Yu, Xiawu Zheng, Dongzhan Zhou, et al. Reasoning via video: The first evaluation of video models' reasoning abilities through maze-solving tasks. *arXiv preprint arXiv:2511.15065*, 2025.

[45] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion probabilistic modeling for video generation. *Entropy*, 25(10):1469, 2023.

[46] Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*, 2025.

[47] Chenyu Zhang, Daniil Cherniavskii, Antonios Tragoudaras, Antonios Vozikis, Thijmen Nijdam, Derck WE Prinzhorn, Mark Bodracska, Nicu Sebe, Andrii Zadaianchuk, and Efstratios Gavves. Morpheus: Benchmarking physical reasoning of video generative models with real physical experiments. *arXiv preprint arXiv:2504.02918*, 2025.

[48] Tinghui Zhu, Kai Zhang, Jian Xie, and Yu Su. Deductive beam search: Decoding deducible rationale for chain-of-thought reasoning. *arXiv preprint arXiv:2401.17686*, 2024.

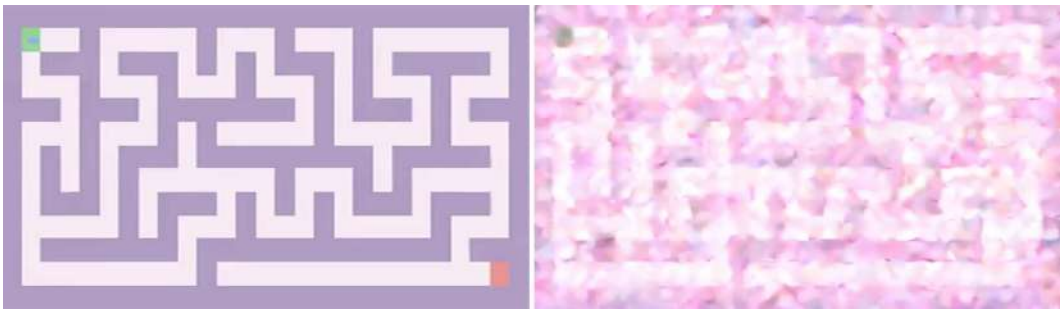

Figure 5: Qualitative example of reward hacking in the absence of KL-divergence regularization. It preserves the wall constraint, and saturates all paths to connect two endpoints, thereby achieving a maximal reward.

A Experimental Setup

A.1 Dataset

- **Maze:** We generate $10,000$ samples with grid dimensions ranging from 7×7 to 21×21. Each instance pairs an unsolved layout containing start and goal markers with a ground-truth video that renders the contiguous path between them.
- **FlowFree:** We generate $10,000$ puzzles with grid sizes between 5×5 and 8×8 by splitting Hamiltonian paths into colored segments, ensuring that a valid solution must occupy all available cells. The initial frame displays only the discrete color-pair endpoints, and the video unrolls the progressive coloring of each path.
- **Sokoban:** This domain includes $10,000$ puzzles with grid sizes from 6×6 to 10×10 and 1–3 boxes. Solution trajectories are capped at 60 moves. The input frame depicts the initial board configuration, while the video unrolls the solution at a resolution of one agent push per frame, ensuring strict alignment between temporal and logical steps.

A.2 Evaluation

Trajectory Alignment Metrics. To measure the alignment with the ground-truth reference, we compute precision, recall, and F1 at the unit most natural to each task's solution manifold:

- **Maze (Pixel-level):** We compute a change mask between the initial and final frames to isolate the generated path from the static background. This ensures the metric captures the model's intervention rather than background reconstruction.
- **FlowFree (Cell-level):** We extract mean colors for each grid cell in the terminal frame. This avoids penalizing anti-aliasing artifacts and focuses on the semantic correctness of the path coloring.
- **Sokoban (Action-level):** Since multiple trajectories can yield the same final state, we decode the video into a symbolic action sequence $a \in \{\mathrm{U(Up)}, \mathrm{D(Down)}, \mathrm{L(Left)}, \mathrm{R(Right)}\}$. We report position-aligned F1, which penalizes out-of-order or invalid moves.

Symbolic Success Rate. Alignment with a single GT reference is insufficient to detect valid alternative solutions or visually plausible but rule-violating outputs. We therefore implement binary success detectors that parse the video $\mathbf{V}$ into symbolic states to verify task-specific rules:

- **Maze Success:** Requires a connected path between markers without violating wall constraints.
- **FlowFree Success:** Validates endpoint preservation, color connectivity, and the grid fill-rate.
- **Sokoban Success:** Evaluates *process validity* over all frames, checking that player and box displacements follow physics-based rules and the final state matches the target.

B Analysis

B.1 KL Constraint

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.

   Guidelines:

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

2. **Limitations**

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

   Answer: **[TODO]**

Justification: **[TODO]**

Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper does not include theoretical results.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The full experimental setup is presented in Section 5 and Section A.

   Guidelines:

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

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

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

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

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

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

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The full experimental setup is presented in Section 5 and Section A.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: Error bars are not reported because it would be too computationally expensive. Inference one time would cost more than 24 GPU hours of B200.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The full experimental setup is presented in Section 5 and Section A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This work does not have such risks.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [N/A]

Justification: The paper does not use existing assets.

Guidelines:

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

13. **New assets**

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

Answer: **[TODO]**

Justification: **[TODO]**

Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

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