Wan-R1: Verifiable-Reinforcement Learning for Video Reasoning

Ming Liu[1], Yunbei Zhang[2], Shilong Liu[3], Liwen Wang[1], and Wensheng Zhang[1]

[1] Iowa State University, Ames, IA 50011, USA
pkulium@iastate.edu
[2] Tulane University, New Orleans, LA 70118, USA
[3] Princeton University, Princeton, NJ 08544, USA

Abstract. Video generation models produce visually coherent content but struggle with tasks requiring spatial reasoning and multi-step planning. Reinforcement learning (RL) offers a path to improve generalization, but its effectiveness in video reasoning hinges on reward design—a challenge that has received little systematic study. We investigate this problem by adapting Group Relative Policy Optimization (GRPO) to flow-based video models and training them on maze-solving and robotic navigation tasks. We first show that multimodal reward models fail catastrophically in this setting. To address this, we design *verifiable reward functions* grounded in objective task metrics. For structured game environments, we introduce a multi-component trajectory reward that is verifiable against ground-truth optimal paths. For robotic navigation, where ground truth is given only as a reference rollout, we propose an embedding-level *reference-anchored* reward. Our experiments show that RL fine-tuning with verifiable rewards improves generalization. For example, on complex 3D mazes, our model improves exact match accuracy by 29.1% over the SFT baseline, and on trap-avoidance tasks by 51.4%. Our systematic reward analysis reveals that verifiable rewards are critical for stable training, while multimodal reward models could lead to degenerate solutions. These findings establish verifiable reward design as a key enabler for robust video reasoning.

1 Introduction

The emergence of video generation models, such as Veo [9] and Sora [27], offers a new approach to visual reasoning. Instead of generating chains of thought through text tokens [25], these models reason by generating temporally coherent visual sequences [39]. This "reasoning via video" paradigm naturally captures spatial relationships, physical dynamics, and temporal causality within a continuous visual medium.

Recent work [12, 39] shows that video models can learn reasoning tasks like maze-solving. Another work studies whether the same *reasoning-via-video* paradigm can support *real-world* semantic navigation, by evaluating on Target-Bench [36]. These tasks require spatial perception, path planning, and multi-step decision making, which remain challenging for current models [13]. Supervised fine-tuning (SFT) on demonstration videos teaches models to solve specific mazes, but it exposes a critical limitation: SFT models generalize poorly to out-of-distribution scenarios [43]. A model trained on easy mazes struggles with harder configurations. Training on regular grids does not transfer

to irregular or 3D layouts, and changes in visual texture often break performance. This brittleness suggests that SFT encourages the memorization of training patterns rather than robust, transferable reasoning.

We investigate whether reinforcement learning (RL) can bridge this generalization gap. Because RL explores diverse solution strategies during training, it can encourage models to learn generalizable reasoning skills rather than surface-level heuristics. We adapt Flow Group Relative Policy Optimization (GRPO) [20], a recent advance in RL for generative models, to flow-based video generation.

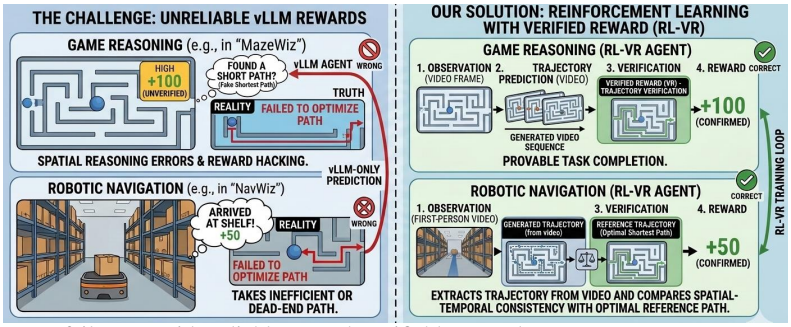

Fig. 1: VLM fails to provide reliable reward, verifiable rewards can guarantee correct supervision.

Applying RL to video reasoning requires careful reward design. Multimodal reward models are susceptible to reward hacking [15, 21, 46]; the generator learns to produce visually convincing videos that satisfy the judge without actually solving the underlying task. We avoid this by grounding the reward in objective ground truth rather than a learned judge. For game environments, we extract the agent's path from the generated video and compare it against the ground-truth solution, yielding a *verifiable* reward that reflects true performance without the pitfalls of learned evaluators. For robotic navigation, where ground truth takes the form of reference rollout videos rather than discrete action sequences, we design an embedding-level *reference-anchored* reward that measures frame-wise visual similarity, temporal consistency, and endpoint fidelity between generated and reference videos. We use the term *reference-anchored* rather than *verifiable* for this reward because it is anchored to a single reference rollout rather than checked against an exhaustive correctness criterion. Our contributions are as follows:

- We adapt Flow-GRPO to flow-matching image-to-video generation with denoising-step reduction, making online RL for video reasoning computationally practical.
- We provide a systematic study of reward design for video generation RL. We show that vision-language reward models fail via reward hacking and that sparse task-completion rewards are insufficient. We introduce multi-component verifiable rewards based on trajectory extraction.
- We propose an embedding-level reference-anchored reward for robotic video reasoning that compares generated rollouts against reference demonstrations via frame similarity, temporal-order consistency, and endpoint fidelity. This reward enables RL training for real-world navigation without discrete action supervision.
- We demonstrate that RL with verifiable rewards consistently outperforms SFT baselines, improving exact match accuracy by up to 51.4 percentage points (e.g.,

on trap-avoidance tasks). On Target-Bench for robotic path planning, our model achieves the best navigation score, reducing displacement errors by roughly half.
– We provide practical insights for scaling RL in video reasoning: we analyze KL regularization, denoising reduction, cold-start requirements, and test-time scaling, offering guidelines for practitioners applying RL to video reasoning.

2 Related Work

2.1 Video Generation Models

The field of video generation has seen rapid progress. Sora-2 [27] demonstrated the ability to produce controllable, physically plausible videos with coherent audio and dialogue integration. Closed-source models—including Runway's Gen-3 [32], Pika Labs [30], Luma AI [23], and Google DeepMind's Veo models [4]—have pushed visual fidelity and realism further, though their implementations are not publicly available. Meanwhile, open-source alternatives like Stable Video Diffusion [2], OpenSora [45], Hunyuan-Video [18], and the Wan series [35] have broadened community access by providing efficient model designs and scalable training pipelines for high-quality video generation.

2.2 Reasoning via Visual Generation

Chain-of-Thought (CoT) prompting has substantially improved the reasoning capabilities of language models [11, 37, 38]. By incorporating reinforcement learning, CoT-style reasoning can be directly embedded into the training process, allowing models to develop internalized multi-step reasoning abilities [26]. Recently, the emergence of unified architectures capable of both generation and comprehension has introduced a novel reasoning framework built around *interleaved vision-language outputs* [5, 40, 41], offering a more visually grounded and expressive approach to tackling complex multimodal reasoning tasks. Recent studies also explore the potential of video generative models in video reasoning [12, 36, 39, 43].

2.3 RL for Generative Models

Reinforcement learning has demonstrated remarkable effectiveness in aligning large language models (LLMs) with human preferences [3, 29]. Group Relative Policy Optimization (GRPO) [11] has recently emerged as a scalable alternative to PPO [33] for reasoning tasks. In the domain of visual generation, PPO-style policy gradients also demonstrated impressive generalization [1, 6, 14]. More recently, DanceGRPO [42] and Flow-GRPO [20] pioneered the adaptation of GRPO to visual synthesis, providing a unified framework for diffusion and flow models through SDE reformulation and achieving stable training across text-to-image, text-to-video, and image-to-video generation.

3 Method

We present our approach for improving video generation reasoning through reinforcement learning. We first describe the task formulation, then detail our adaptation of GRPO to flow-based video models, and finally present our verifiable reward design.

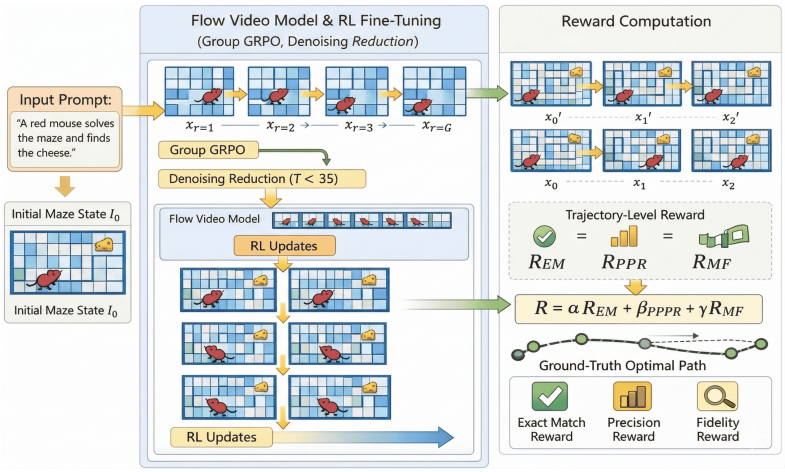

Fig. 2: Flow-GRPO with verified reward.

3.1 Task Formulation

We consider the visual trace reasoning (VTR) task as shown in Figure 2, where a model receives a prompt c and an initial maze image I_0, and must generate a video $V = \{f_1, f_2, \ldots, f_N\}$ showing an agent (e.g., a colored ball) navigating from a start position to a goal position. The task requires the model to perceive the maze structure from the initial image, plan a valid path avoiding obstacles, and generate temporally consistent motion to the goal.

3.2 Background: Flow Matching

Flow matching models [19, 22] define a continuous transformation between noise and data through a velocity field $v_\theta(x_t, t)$. The forward process interpolates between data x_0 and noise x_1:

$$x_t = (1-t)x_0 + tx_1, \quad t \in [0,1] \tag{1}$$

The model is trained to predict the velocity field by minimizing:

$$\mathcal{L}(\theta) = \mathbb{E}_{t,x_0,x_1}\left[\|v - v_\theta(x_t,t)\|^2\right] \tag{2}$$

For inference, samples are generated by solving ordinary differential equation (ODE):

$$dx_t = v_\theta(x_t,t)dt \tag{3}$$

3.3 RL for Flow Matching

Group Relative Policy Optimization (GRPO) [11] provides a lightweight alternative to PPO [33] by using group-relative advantage estimation, eliminating the need for a separate value network.

ODE to SDE Conversion. A key challenge in applying GRPO to flow models is that the deterministic ODE sampling prevents the stochastic exploration required for RL. Flow-GRPO [20] converts the ODE to an equivalent stochastic differential equation (SDE) that preserves the marginal distribution while introducing randomness:

$$dx_t = \left(v_t(x_t) - \frac{\sigma_t^2}{2} \nabla \log p_t(x_t) \right) dt + \sigma_t dw \tag{4}$$

where σ_t controls the noise level and dw denotes Wiener process increments. For rectified flow, the discretized update becomes:

$$x_{t+\Delta t} = x_t + \tilde{v}_\theta(x_t, t)\Delta t + \sigma_t \sqrt{\Delta t}\epsilon \tag{5}$$

where $\epsilon \sim \mathcal{N}(0, I)$ and $\tilde{v}_\theta$ includes the score correction term.

Group Relative Advantage. Given a prompt-image pair (c, I_0), we sample a group of G videos $\{V^i\}_{i=1}^G$ using SDE sampling. The advantage for each video is computed by normalizing rewards within the group:

$$\hat{A}^i = \frac{R(V^i, c) - \text{mean}(\{R(V^j, c)\}_{j=1}^G)}{\text{std}(\{R(V^j, c)\}_{j=1}^G)} \tag{6}$$

Policy Optimization. The GRPO objective maximizes:

$$\mathcal{J}(\theta) = \mathbb{E}\left[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{T} \sum_{t=0}^{T-1} \left(\mathcal{L}_{\text{clip}}^{i,t} - \beta_{KL} D_{\text{KL}} \right) \right] \tag{7}$$

where $\mathcal{L}_{\text{clip}}^{i,t}$ is the clipped surrogate objective and β_{KL} controls KL regularization against the reference policy.

Denoising Reduction. Video generation is computationally expensive, requiring many denoising steps per sample. We follow denoising reduction [20]: using fewer steps ($S_{\text{train}} = 30$) during training data collection while retaining full steps ($S_{\text{infer}} = 50$) during evaluation. This significantly accelerates training without sacrificing final performance.

3.4 Verifiable Reward Design for Game

A critical—and underexplored—challenge in applying RL to video generation is reward design. Unlike text-based RL where human preference provides a natural reward signal, video generation tasks present unique difficulties: the output space is high-dimensional

and continuous, success criteria are often spatial and temporal, and learned reward models face severe reward hacking risks specific to the visual domain.

We identify three desiderata for effective video RL rewards: (i) **verifiability**—the reward must reflect genuine task success, not superficial visual quality; (ii) **density**—the reward should provide gradient information beyond binary success/failure; and (iii) **decomposability**—the reward should separate distinct aspects of task performance to prevent conflation. We now describe how our reward design satisfies each criterion.

Trajectory Extraction. Given a generated video, we extract the agent's trajectory using object tracking [43]. We initialize a tracker on the agent's bounding box in the first frame and track its center position across all frames, yielding a trajectory $\tau_{\text{pred}} = \{p_1, p_2, \ldots, p_N\}$. Trajectory samples are presented in Appendix C.

Reward Components. We define rewards based on trajectory-level metrics [43]. Let $\{\hat{v}_{ij}\}_{j=1}^{n_i}$ denote the predicted trajectory steps and $\{v_{ij}\}_{j=1}^{n_i}$ denote the ground-truth optimal path for the i-th sample.

Exact Match Reward (R_{EM}): Measures whether the generated trajectory exactly matches the complete optimal path:

$$R_{\text{EM}} = \prod_{j=1}^{n_i} \mathbb{I}(\hat{v}_{ij} = v_{ij}) \tag{8}$$

This is a strict binary reward that requires every step to be correct. One deviation from the optimal solution yields zero reward.

Precision Reward (R_{PR}): Quantifies the proportion of consecutively correct steps along the optimal path, providing a softer metric than exact match:

$$R_{\text{PR}} = \frac{1}{n_i} \sum_{j=1}^{n_i} \left[\prod_{k=1}^{j} \mathbb{I}(\hat{v}_{ik} = v_{ik}) \right] \tag{9}$$

This reward reflects the model's ability to make steady, meaningful progress toward the complete correct trajectory.

Fidelity Reward (R_{MF}): Measures the structural consistency of the maze layout across video frames, penalizing unrealistic behaviors such as walls changing or disappearing:

$$R_{\text{MF}} = \frac{1}{M} \sum_{m=1}^{M} \left(1 - \frac{|\{p : |I_0(p) - I_m(p)| > \tau\}|}{N_m} \right) \tag{10}$$

where M is the number of sampled frames, I_0 and I_m denote the background regions of the first and m-th frames respectively, τ is a pixel-difference threshold, and N_m is the number of valid overlapping pixels. Higher values indicate better preservation of the static maze structure.

Combined Reward. Our final reward combines these components as shown in Figure 2:

$$R = \alpha R_{\text{EM}} + \beta R_{\text{PR}} + \gamma R_{\text{MF}} \tag{11}$$

where α, β, γ are hyperparameters balancing path correctness, progressive accuracy, and visual consistency. In our experiments, we use $\alpha = 0.3$, $\beta = 0.5$, $\gamma = 0.2$, prioritizing the precision reward which provides denser learning signal while still encouraging exact solutions and physical plausibility.

3.5 Verifiable Reward Design for Robotic Reasoning

For robotic reasoning task, such as Target-Bench [36], supervision is given as a *reference rollout video* collected by a teleoperated robot, rather than a discrete action sequence. We therefore design a reward that directly compares a generated video plan to its reference, without relying on a learned judge.

Problem setting. Each Target-Bench sample includes a semantic goal prompt c, a first-frame observation I_0, and a reference video $V^* = \{f_1^*, \ldots, f_{T^*}^*\}$. Conditioned on (c, I_0), the model generates a video rollout $V = \{f_1, \ldots, f_T\}$ that should depict motion toward the target.

Frame embedding reward. We compute per-frame embeddings using a frozen vision encoder ϕ (DINOv2 [28] or CLIP [31]) and ℓ_2 normalization: $e_t = \mathrm{norm}(\phi(f_t))$ and $e_t^* = \mathrm{norm}(\phi(f_t^*))$. Since T and T^* may differ, we linearly interpolate the shorter embedding sequence to a common length T_c. We then compute (i) mean frame cosine similarity R_{cos}, (ii) a temporal-order consistency score R_{temp} based on normalized cumulative displacement in embedding space, and (iii) endpoint fidelity R_{end}:

$$R_{\text{cos}} = \frac{1}{T_c} \sum_{t=1}^{T_c} \langle e_t, e_t^* \rangle, \tag{12}$$

$$R_{\text{end}} = \frac{1}{2} \Big(\langle e_1, e_1^* \rangle + \langle e_{T_c}, e_{T_c}^* \rangle \Big), \tag{13}$$

$$R_{\text{temp}} = 1 - \frac{1}{T_c - 1} \sum_{t=1}^{T_c - 1} \left| \bar{c}_t - \bar{c}_t^* \right|, \tag{14}$$

where $\bar{c}_t = \frac{\sum_{k=1}^{t} \|e_{k+1} - e_k\|_2}{\sum_{k=1}^{T_c - 1} \|e_{k+1} - e_k\|_2 + \epsilon}$ (and similarly $\bar{c}_t^*$). The embedding reward is a weighted sum:

$$R_{\text{emb}} = \alpha_{\text{emb}} R_{\text{cos}} + \beta_{\text{emb}} R_{\text{temp}} + \gamma_{\text{emb}} R_{\text{end}}, \tag{15}$$

with $(\alpha_{\text{emb}}, \beta_{\text{emb}}, \gamma_{\text{emb}}) = (0.5, 0.2, 0.3)$ in our implementation.

3.6 Training Algorithm

Algorithm 1 summarizes the complete Video-GRPO training procedure.

Algorithm 1 Video-GRPO Training

Require: SFT model π_θ, reference π_{ref}, dataset $\mathcal{D}$, group size G, training epoch T_{train}

for each iteration **do**
  Sample batch of (c, I, τ^*) from $\mathcal{D}$
  for each (c, I, τ^*) in batch **do**
    Sample G videos via SDE with S_{train} steps
    Extract trajectories $\{\hat{\tau}^i\}_{i=1}^G$ via tracking
    Compute rewards $\{r^i\}_{i=1}^G$ against τ^*
    Compute advantages $\{\hat{A}^i\}_{i=1}^G$ via (6)
  end for
  Update θ by maximizing $\mathcal{J}(\theta)$ via (7)
end for
return Trained model π_θ

4 Experimental Setup

4.1 Dataset and Tasks

We conduct experiments on maze-solving tasks from VR-Bench [43], which provides procedurally generated mazes across five types:

- **Regular Maze:** Grid-based layouts testing basic pathfinding
- **Irregular Maze:** Curved paths preventing coordinate shortcuts
- **3D Maze:** Stereoscopic structures requiring depth perception
- **Trapfield:** Obstacle avoidance with trap regions
- **Sokoban:** Box-pushing puzzles requiring rule comprehension

Each maze type includes three difficulty levels (Easy, Medium, Hard) and multiple visual textures. Examples of games are presented in Appendix C.

Target-Bench (robotic path planning). We additionally include **Target-Bench** [36], a real-world benchmark for *mapless* path planning toward text-specified *semantic targets*. Each sample provides an egocentric RGB observation I_0, a goal prompt c (explicit or implicit), and a reference rollout video V^* collected with a quadruped robot, together with SLAM-verified ground-truth trajectories. We follow the official protocol and evaluate using the VGGT world decoder and trajectory metrics.

4.2 Base Model

We use *Wan2.2-TI2V-5B* [35] as our base video generation model. This is a flow-matching model for text-and-image-to-video generation. We start from an SFT checkpoint that has been fine-tuned on maze-solving demonstrations, which we refer to as Wan-SFT [43]. We compare our model Wan-R1 against a diverse set of baseline models across two categories: ***(1)*** Six closed-source video models, including *Veo-3.1-fast*, *Veo-3.1-pro* [10], *Sora-2* [27], *Kling-v1* [17], *Seedance-1.0-pro* [8], and *MiniMax-Hailuo-2.3* [24]; and ***(2)***

three open-source video models, namely *Wan2.5-i2v-preview* [34] and *Wan2.2-TI2V-5B* [35], as well as supervised finetuned *Wan-SFT* [43]. The reported results of Wan-SFT are taken directly from the original VR-Bench paper [43].

4.3 Training Configuration

The training configurations are presented in Appendix A. Besides, we use $\alpha = 0.3$, $\beta = 0.5$, $\gamma = 0.2$ for the combined reward. For Target-Bench, we use the reward described in Sec. 3.5. We set $(\alpha_{\text{emb}}, \beta_{\text{emb}}, \gamma_{\text{emb}}) = (0.5, 0.2, 0.3)$. We use LoRA [16] to finetune the model. All experiments are conducted on machine with H200 GPUs.

4.4 Evaluation Metrics

We evaluate using four metrics from VR-Bench [43]:

- **Exact Match (EM):** Whether generated trajectory exactly matches the optimal path
- **Success Rate (SR):** Whether the agent reaches the goal
- **Precision Rate (PR):** Proportion of consecutively correct steps
- **Step Deviation (SD):** Relative path length redundancy (lower is better)

Target-Bench. We follow the Target-Bench evaluation protocol [36] to assess whether a generated video implies a useful navigation plan. We report five metrics: Average Displacement Error (ADE), Final Displacement Error (FDE), Miss Rate (MR, threshold τ=2.0 m), Soft Endpoint score (SE, tolerance σ=0.6 m), and Approach Consistency (AC), which measures whether the predicted trajectory stays within a progress-dependent corridor around the ground-truth path. These are aggregated into a weighted overall score WO $\in [0, 1]$, with AC and SE jointly carrying 65% of the weight.

5 Results

5.1 Main Results

	Method	EM (↑)					SR (↑)					PR (↑)					SD (↓)				
		Base	Irreg	Trap	3D	Soko	Base	Irreg	Trap	3D	Soko	Base	Irreg	Trap	3D	Soko	Base	Irreg	Trap	3D	Soko
General Video Model	**Closed-Source**																				
	Veo-3.1-fast	0.0	0.0	0.0	0.0	2.8	40.3	36.1	38.9	48.6	43.1	20.2	24.8	28.2	13.4	21.7	195.3	111.5	80.7	33.5	112.3
	Veo-3.1-pro	0.0	4.2	1.4	0.0	0.0	47.2	36.1	59.7	50.0	37.5	24.6	33.9	39.1	18.0	21.4	140.7	94.5	85.4	40.1	141.8
	Sora-2	1.4	5.6	0.0	0.0	4.2	75.0	**72.2**	83.0	37.5	43.1	45.1	45.7	46.6	19.3	27.4	302.9	187.0	145.1	92.4	138.7
	kling-v1	0.0	0.0	0.0	0.0	0.0	2.8	0.0	1.4	27.8	12.5	6.3	8.8	10.4	11.7	9.0	25.2	–	–	69.7	356.1
	Seedance-1.0-pro	0.0	2.8	2.8	0.0	0.0	75.0	45.8	59.7	77.8	13.9	12.8	35.8	42.7	23.6	17.1	162.3	143.4	99.1	84.4	241.9
	MiniMax-Hailuo-2.3	0.0	1.4	2.8	0.0	0.0	68.1	40.3	70.8	55.6	45.8	23.2	24.2	30.3	20.3	15.5	464.0	170.0	90.9	50.1	165.5
	Open-Source																				
	Wan2.5-i2v-preview	0.0	2.8	4.2	0.0	0.0	58.3	26.4	77.8	24.5	22.4	14.3	21.8	34.4	24.5	17.1	378.4	281.8	73.2	119.9	278.0
	Wan2.2-TI2V-5B$^{\diamond}$	0.0	0.0	0.0	0.0	0.0	6.9	12.5	0.0	31.9	11.1	6.6	9.1	7.1	12.8	9.2	388.7	66.1	–	5.4	176.6
	Wan-SFT [43]	33.3	56.9	38.9	65.3	4.2	76.4	69.4	100.0	100.0	69.4	60.6	71.6	79.1	93.5	44.3	10.3	2.4	3.9	3.9	10.2
Train	Wan-R1(ours)	61.1	47.9	90.3	94.4	30.6	75.0	69.7	100.0	100.0	68.1	78.6	68.4	97.7	98.0	46.9	5.2	8.3	2.4	1.9	16.9
	$\Delta\uparrow$	+61.1	+47.9	+90.3	+94.4	+30.6	+68.1	+57.2	+100.0	+68.1	+57.0	+72.0	+59.3	+90.6	+85.2	+37.7	-383.5	-57.8	–	-3.5	-159.7

Table 1: VR-Bench comprises five tasks: Base (Regular Maze), Irreg (Irregular Maze), Trap (TrapField), 3D (3D Maze), and Soko (Sokoban). Last row shows the performance gain compared with Wan2.2-TI2V base model.

Table 1 presents the main comparison between SFT and RL-tuned models across all maze types. The RL-tuned model (Wan-R1) outperforms the SFT baseline on most maze

types and metrics. On Regular Maze, Wan-R1 achieves 61.1% exact match compared to 33.3% for Wan-SFT, representing an absolute gain of 27.8 percentage points. For 3D Maze, the improvement is even more pronounced, with exact match increasing from 65.3% to 94.4%. Trapfield shows the most dramatic gains, where exact match jumps from 38.9% to 90.3%, accompanied by near-perfect precision (97.7%). While success rates remain comparable across both models for most tasks (as both achieve high completion rates), the key improvements manifest in path optimality: step deviation decreases substantially on Regular Maze (from 10.3 to 5.2) and 3D Maze (from 3.9 to 1.9), indicating that RL training produces more efficient trajectories that closely follow optimal solutions.

5.2 Generalization Results

Task	EM			SR			PR			SD		
	E	M	H	E	M	H	E	M	H	E	M	H
Base	0.0	0.0	0.0	8.3	8.3	4.2	13.2	3.8	2.7	154.8	–	–
	0.0 (+0.0)	4.2 (+4.2)	0.0 (+0.0)	83.3 (+75.0)	41.7 (+33.4)	58.3 (+54.1)	40.1 (+26.9)	26.1 (+22.3)	6.0 (+3.3)	28.2 (-126.6)	10.1 (–)	– (–)
	87.5 (+87.5)	66.7 (+66.7)	29.2 (+29.2)	100.0 (+91.7)	87.5 (+79.2)	41.7 (+37.5)	97.0 (+83.8)	85.4 (+81.6)	57.2 (+54.5)	1.4 (-153.4)	2.5 (–)	14.6 (–)
Irrg	0.0	0.0	0.0	29.2	4.2	4.2	13.4	8.1	5.8	48.3	–	39.3
	83.3 (+83.3)	66.7 (+66.7)	54.2 (+54.2)	95.8 (+66.6)	87.5 (+83.3)	62.5 (+58.3)	88.0 (+74.6)	74.8 (+66.7)	68.4 (+62.6)	3.5 (-44.8)	7.8 (–)	3.1 (-36.2)
	79.2 (+79.2)	45.8 (+45.8)	20.8 (+20.8)	100.0 (+70.8)	66.7 (+62.5)	37.5 (+33.3)	82.3 (+68.9)	70.7 (+62.6)	51.6 (+45.8)	10.1 (-38.2)	9.1 (–)	4.3 (-35.0)
Trap	0.0	0.0	0.0	0.0	0.0	0.0	6.0	6.4	8.9	–	–	–
	62.5 (+62.5)	0.0 (+0.0)	12.5 (+12.5)	100.0 (+100.0)	95.8 (+95.8)	62.5 (+62.5)	86.7 (+80.7)	43.7 (+37.3)	56.7 (+47.8)	2.1 (–)	5.9 (–)	3.5 (–)
	79.2 (+79.2)	100.0 (+100.0)	75.9 (+75.9)	100.0 (+100.0)	100.0 (+100.0)	100.0 (+100.0)	94.7 (+88.7)	100.0 (+93.6)	86.4 (+77.5)	4.9 (–)	1.2 (–)	1.0 (–)
3D	0.0	0.0	0.0	54.2	16.7	25.0	7.6	9.8	15.2	53.0	87.9	33.6
	41.7 (+41.7)	0.0 (+0.0)	4.2 (+4.2)	100.0 (+45.8)	79.2 (+62.5)	83.3 (+58.3)	78.7 (+71.1)	47.4 (+37.6)	58.7 (+43.5)	4.6 (-48.4)	10.2 (-77.7)	13.8 (-19.8)
	87.5 (+87.5)	95.8 (+95.8)	91.7 (+91.7)	100.0 (+45.8)	100.0 (+83.3)	100.0 (+75.0)	98.4 (+90.8)	99.1 (+89.3)	96.2 (+81.0)	1.7 (-51.3)	1.6 (-86.3)	2.7 (-30.9)
Soko	0.0	0.0	0.0	20.8	8.3	4.2	14.6	8.3	5.6	354.2	61.4	–
	4.2 (+4.2)	0.0 (+0.0)	0.0 (+0.0)	83.3 (+62.5)	45.8 (+37.5)	33.3 (+29.1)	62.2 (+47.6)	12.5 (+4.2)	10.4 (+4.8)	18.8 (-335.4)	84.5 (+23.1)	16.1 (–)
	75.0 (+75.0)	12.5 (+12.5)	4.2 (+4.2)	91.7 (+70.9)	91.7 (+83.4)	45.8 (+41.6)	87.8 (+73.2)	30.3 (+22.0)	19.4 (+13.8)	5.8 (-348.4)	8.8 (-52.6)	38.6 (–)

Table 2: Difficulty generalization performance of Wan-R1 on VR-Bench. Each task block presents a comparison between the baseline model (Wan2.2-TI2V-5B) and Wan-SFT (trained exclusively on Easy-level data) and Wan-R1 (trained on easy tasks with flow-GRPO) evaluated across three difficulty levels (E/M/H = Easy/Medium/Hard).

Difficulty Generalization. Table 2 shows generalization to harder difficulty levels when trained only on Easy mazes. The GRPO model shows better generalization to Medium and Hard difficulties on most, though not all, maze types. On Regular Maze, while the SFT model degrades catastrophically from Easy to Hard (40.1% to 6.0% PR), Wan-R1 maintains substantially higher performance (97.0% to 57.2% PR). The effect is most pronounced on 3D Maze Hard, where Wan-R1 reaches 96.2% precision versus 58.7% for Wan-SFT and just 15.2% for the base model, and on Trapfield, where it improves Hard-level PR from 56.7% (Wan-SFT) to 86.4%. The one exception is Irregular Maze at the Hard level, where Wan-R1 (51.6% PR) trails Wan-SFT (68.4% PR): we attribute this to the curved, coordinate-free geometry of irregular mazes, where exploration under our trajectory reward appears to help less than on grid-aligned layouts. We therefore qualify our generalization claim as holding per maze type rather than universally. Overall, these results indicate that RL encourages learning of more generalizable reasoning strategies rather than memorizing specific mazes, while leaving headroom on the hardest irregular configurations.

Texture Generalization. Appendix B presents results on unseen visual textures. The RL-tuned model shows substantially better texture generalization, particularly on visual styles that differ most from training. On Regular Maze, Wan-R1 achieves 6.9% and 31.9% exact match on Skin2 and Skin3 compared to 1.4% and 23.6% for Wan-SFT respectively. For 3D Maze, the model maintains 84.7% and 79.2% exact match on these unseen textures compared to 43.1% and 50.0% for the SFT baseline. This texture invariance suggests that RL encourages the model to focus on structural maze features rather than superficial visual patterns, learning more abstract reasoning strategies that transfer across visual domains.

Maze Type Generalization. Our appendix B shows cross-task generalization when trained on a single maze type. While fine-tuning on individual tasks improves in-domain performance, our method further enhances both in-domain accuracy and cross-task transfer. For Regular Maze, our approach nearly doubles the in-domain EM (61.1 vs 33.3) while maintaining competitive cross-domain performance. The most substantial gains appear in 3D Maze, where our method achieves 94.4 EM compared to 65.3 from fine-tuning alone, alongside near-perfect PR (98.0). For Sokoban, the inherently more complex task requiring box-pushing logic, our method improves in-domain EM from 4.2 to 30.6 and dramatically boosts cross-domain SR on Base mazes (48.6 vs 0.5). Notably, Trapfield with our method achieves perfect SR (100.0) across Base, Irregular, and 3D domains, demonstrating improved generalization to visually distinct maze types. These results suggest that our approach not only strengthens task-specific learning but also facilitates more robust feature representations that transfer across maze variants.

5.3 Target-Bench: Mapless Path Planning

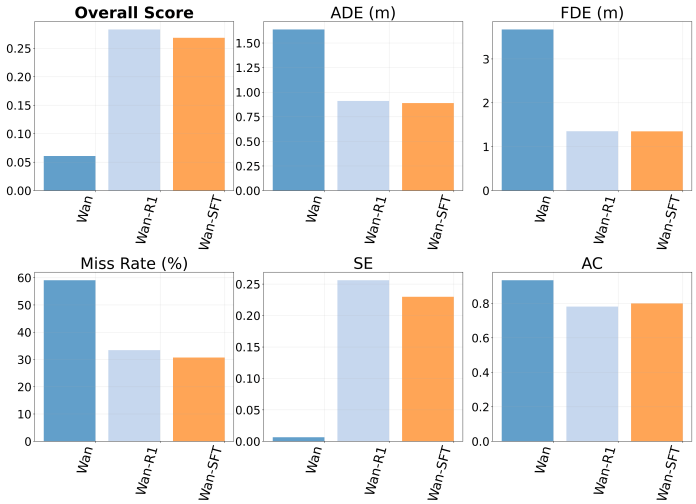

Fig. 3: Target-Bench results. Wan-R1 achieves the highest overall score, with lower displacement errors (ADE, FDE) and miss rate, and higher soft endpoint (SE) and approach consistency (AC) compared to both the base model and Wan-SFT.

We additionally evaluate our models on Target-Bench [36], which measures whether a generated video rollout supports mapless navigation toward a semantic target. Figure 3 provides the results. Wan-R1 achieves the best overall score among all three models, reducing ADE and FDE by roughly half relative to the base model while substantially lowering miss rate. Both SE and AC also improve, indicating that RL training encourages the generated rollouts to better encode goal-directed motion. These gains suggest that the embedding-level reference-anchored reward transfers effectively to real-world robotic navigation beyond structured maze tasks. Notably, the policy improves on the independently measured ADE/FDE/MR metrics, not merely on the reward it was trained against, providing indirect evidence that the reference-anchored reward is not being trivially hacked.

6 Analysis

6.1 Effect of KL Regularization and Denoising Reduction

β	EM	SR	PR	SD
0	62.5	75.0	79.0	6.7
0.04	61.1	75.0	78.6	5.2
0.1	66.7	75.0	81.5	3.7

(a) Effect of KL coefficient β_{KL}.

S_{train}	EM	SR	PR	SD
5	59.7	70.8	77.5	4.0
30	61.1	75.0	78.6	5.2
50	62.5	73.6	79.7	2.9

(b) Effect of training denoising steps (S_{train}).

Table 3: Ablation studies on KL regularization and denoising reduction.

KL Regularization. Table 3a presents the effect of KL regularization strength on model performance. Without KL regularization ($\beta_{KL} = 0$), the model achieves moderate exact match (62.5%) but exhibits the highest step deviation (6.7), indicating a tendency to produce unnecessarily long or redundant paths. Increasing regularization to $\beta_{KL} = 0.1$ yields the best overall performance, achieving the highest exact match (66.7%), precision rate (81.5%), and the lowest step deviation (3.7), while maintaining an identical success rate (75.0%) across all settings. This suggests that stronger KL regularization encourages the policy to stay closer to the reference distribution, producing more efficient and structurally consistent trajectories rather than exploring degenerate solution paths.

Denoising Reduction. Reducing training denoising steps dramatically accelerates training with minimal performance loss. As shown in Table 3b, using $S_{\text{train}} = 5$ degrades performance noticeably (59.7% EM, 70.8% SR), confirming that an excessively aggressive reduction impairs sample quality during rollout collection. However, $S_{\text{train}} = 30$ recovers nearly all of the performance of the full 50-step setting (61.1% vs. 62.5% EM, 78.6% vs. 79.7% PR). Notably, $S_{\text{train}} = 30$ achieves the highest success rate (75.0% vs. 73.6% for full steps), suggesting that moderate denoising reduction may act as a mild regularizer by introducing controlled stochasticity into the generated rollouts.

6.2 Cold Start

Applying Flow-GRPO directly to the base model (Wan2.2-TI2V-5B) without prior supervised fine-tuning yields poor results, as the base model achieves near-zero exact match across all maze types. This highlights a standard characteristic of reinforcement learning: the policy requires a foundational task understanding before RL can effectively guide exploration [7, 44]. Because our verifiable rewards enforce strict logical correctness, they are naturally sparser than subjective visual quality metrics. Therefore, SFT acts as a necessary bootstrap, placing the model within a region of the reward landscape where policy gradient updates become meaningful. Future work could bridge this gap through curriculum learning on simplified environments or offline preference optimization.

6.3 Test-Time Scaling

Appendix B examines whether RL genuinely improves model capability or merely benefits from best-of-K sampling. We evaluate Wan-R1 with varying numbers of samples $K \in \{1, 4, 8, 12, 16\}$ across difficulty levels. The results reveal that performance continues to improve with larger K, demonstrating that the model maintains meaningful diversity in its generations. Importantly, even at K=1 (no sampling advantage), Wan-R1 substantially outperforms the SFT baseline, confirming that RL training improves the underlying policy rather than simply increasing variance for lucky samples. The scaling curves show diminishing returns beyond K=8, suggesting this as a practical inference budget.

6.4 Reward Design Analysis

We conduct a systematic study of reward design choices. This analysis serves two purposes: validating our multi-component verifiable reward and providing general insights on reward design in video reasoning. The results are presented in Appendix B.

When reward models are not reliable We use Qwen2.5-VL as a reward model, prompting it to assess whether the generated video correctly solves the maze. Appendix B reveals fundamental failure mode: the model learns to generate videos where the agent moves confidently along plausible-looking but incorrect paths. Figure in Appendix B illustrates representative examples. The VLM assigns high scores to these incorrect solutions because it responds to visual cues of purposeful motion—smooth trajectories, clear agent visibility, and goal-directed movement—rather than verifying actual path correctness. This confirms that the high-dimensional, continuous nature of video outputs makes learned reward models particularly vulnerable to exploitation, more so than in text-based RL where outputs are discrete and more easily verified.

Sparse rewards provide insufficient learning signal. Using only binary success (SR-Only) or exact match (R_{EM}) as reward yields limited improvement over SFT. With SR-Only, the model learns to reach the goal but via highly suboptimal routes (high SD), because all successful trajectories receive identical reward regardless of efficiency. With R_{EM} alone, the reward is even sparser—few generated trajectories exactly match the optimal path, so most samples receive zero reward, providing negligible gradient signal for early training.

Dense verifiable rewards enable stable learning. The precision reward R_{PR} provides the critical dense signal: it rewards partial progress along the correct path, creating a smooth reward landscape that guides optimization. Training with R_{PR} alone already achieves better EM, substantially outperforming sparse alternatives. Adding R_{EM} provides a bonus for complete solutions, encouraging the model to "close out" trajectories rather than settling for partial correctness. The fidelity term R_{MF} prevents a subtle failure mode where the model achieves high trajectory scores by altering the maze structure in later frames (e.g., removing walls), which constitutes a form of visual "cheating."

Reward-weight robustness. Because our reward combines three components, $R = \alpha R_{\text{EM}} + \beta R_{\text{PR}} + \gamma R_{\text{MF}}$, a natural concern is sensitivity to the weights (α, β, γ). We sweep these weights on Regular Maze (Appendix B). The model is robust to the precise setting: exact match varies by at most 0.056 and step deviation stays below 0.065 across all configurations, and α and β are partially substitutable (swapping them changes EM by less than 0.03). This indicates that our results do not hinge on a finely tuned reward weighting.

6.5 Why Does RL Improve Generalization?

We hypothesize several factors contribute to RL's generalization benefits:

Exploration discovers diverse strategies. Unlike SFT which imitates a single demonstration per maze, RL explores multiple trajectories through stochastic SDE sampling. During training, the model generates G=8 different video solutions for each maze, receiving differentiated feedback based on trajectory quality. This exposure to a broader distribution of valid (and invalid) solutions helps the model learn which features of a solution are essential versus incidental.

Reward provides task-level signal. SFT optimizes frame-level reconstruction loss, which may cause overfitting to superficial visual patterns such as specific texture details or agent appearances. RL optimizes task completion through trajectory-level rewards, encouraging the model to learn more abstract reasoning strategies that capture the essence of pathfinding rather than pixel-level imitation.

Group normalization reduces spurious correlations. By comparing within groups of videos generated for the same maze, GRPO's advantage estimation naturally controls for maze-specific factors. A video is considered "good" only if it outperforms other attempts on the same maze, not if it happens to be generated for an easier maze. This relative evaluation helps the model learn transferable skills rather than maze-specific shortcuts.

7 Conclusion

We explored the application of reinforcement learning to enhance reasoning capabilities in video generation models. By adapting GRPO to flow-based video models and designing verifiable rewards grounded in trajectory matching, we achieved significant

improvements in generalization across difficulty levels, visual textures, and maze types. Beyond structured game environments, we further demonstrated that embedding-level reference-anchored rewards enable effective RL training for robotic video reasoning, where our model improves mapless path planning toward semantic targets on Target-Bench. Our results demonstrate that online RL with carefully designed rewards can strengthen reasoning capabilities across both synthetic and real-world navigation tasks without sacrificing video quality or diversity.

Our work highlights the critical importance of reward design in RL for generative models. While multimodal reward models offer convenience, they are prone to reward hacking in video generation tasks. In contrast, grounding the reward in objective signals—whether a verifiable reward based on discrete trajectory matching for games, or a reference-anchored reward based on continuous embedding similarity for robotic rollouts—provides more reliable training signals and leads to genuine capability improvements.

8 Limitations

Scope of verifiable rewards. Our study focuses on reasoning tasks with strictly definable success criteria, and our reward design depends on either a clean trajectory tracker (games) or a reference rollout (robotics). This constraint is by design: it is precisely what prevents the reward hacking we demonstrate in Sec. 6.4, since the reward cannot be satisfied without genuinely solving the task. The flip side is narrow applicability—the approach does not directly extend to open-ended tasks lacking an objective verifier. We see several concrete paths to relaxing this requirement: learned verifiers gated by agreement with weak signals, pseudo-ground-truth construction via model-based planning, self-consistency across multiple rollouts, and hybrid schemes that admit a VLM reward only when it agrees with a verifiable check. We frame the extension to less constrained settings as an open problem rather than a solved one.

Cold start. RL fine-tuning in our setting requires an SFT initialization: applying Flow-GRPO directly to the base model yields near-zero reward because our verifiable rewards are sparse near a cold start, leaving the policy without informative gradients (Appendix B). We treat cold start as an open research question rather than a passing limitation. Promising mitigations include curriculum learning over progressively harder mazes, offline preference optimization on demonstration pairs, and reward relaxation that anneals from dense shaped signals toward strict verifiability as the policy improves.

Single reference path. Ground-truth paths in VR-Bench are single BFS-shortest paths, so a generated trajectory that takes an *alternative* optimal (equally short) route is penalized from the point of divergence under R_{EM} and R_{PR}. This makes our reward conservative: it can undercount correct solutions. Rewarding against the set of all shortest paths is a natural extension that would remove this bias.

Bibliography

[1] Black, K., Janner, M., Du, Y., Kostrikov, I., Levine, S.: Training diffusion models with reinforcement learning (2024), https://arxiv.org/abs/2305.13301

[2] Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023)

[3] Christiano, P.F., Leike, J., Brown, T., Martic, M., Legg, S., Amodei, D.: Deep reinforcement learning from human preferences. Advances in neural information processing systems **30** (2017)

[4] DeepMind, G.: Veo 2 (December 2024), accessed: 2024

[5] Duan, C., Fang, R., Wang, Y., Wang, K., Huang, L., Zeng, X., Li, H., Liu, X.: Got-r1: Unleashing reasoning capability of mllm for visual generation with reinforcement learning. arXiv preprint arXiv:2505.17022 (2025)

[6] Fan, Y., Watkins, O., Du, Y., Liu, H., Ryu, M., Boutilier, C., Abbeel, P., Ghavamzadeh, M., Lee, K., Lee, K.: Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models (2023), https://arxiv.org/abs/2305.16381

[7] Feng, K., Gong, K., Li, B., Guo, Z., Wang, Y., Peng, T., Wu, J., Zhang, X., Wang, B., Yue, X.: Video-r1: Reinforcing video reasoning in mllms (2025), https://arxiv.org/abs/2503.21776

[8] Gao, Y., Guo, H., Hoang, T., Huang, W., Jiang, L., Kong, F., Li, H., Li, J., Li, L., Li, X., et al.: Seedance 1.0: Exploring the boundaries of video generation models. arXiv preprint arXiv:2506.09113 (2025)

[9] Google: Veo 2 (2024), https://docs.cloud.google.com/vertex-ai/generative-ai/docs/models/veo/2-0-generate, accessed on June 28, 2026

[10] Google: Veo 3 (2025), accessed on June 28, 2026

[11] Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al.: Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948 (2025)

[12] Guo, Z., Chen, X., Zhang, R., An, R., Qi, Y., Jiang, D., Li, X., Zhang, M., Li, H., Heng, P.A.: Are video models ready as zero-shot reasoners? an empirical study with the mme-cof benchmark (2025), https://arxiv.org/abs/2510.26802

[13] Guo, Z., Chen, X., Zhang, R., An, R., Qi, Y., Jiang, D., Li, X., Zhang, M., Li, H., Heng, P.A.: Are video models ready as zero-shot reasoners? an empirical study with the mme-cof benchmark. arXiv preprint arXiv:2510.26802 (2025)

[14] Gupta, S., Ahuja, C., Lin, T.Y., Roy, S.D., Oosterhuis, H., de Rijke, M., Shukla, S.N.: A simple and effective reinforcement learning method for text-to-image diffusion fine-tuning (2025), https://arxiv.org/abs/2503.00897

[15] Hong, Y., Kao, K.C., Zhou, H., Hsieh, C.J.: Understanding reward hacking in text-to-image reinforcement learning (2026), https://arxiv.org/abs/2601.03468

[16] Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models (2021), https://arxiv.org/abs/2106.09685

[17] Kling AI: Kling Video Generation Platform (2025), accessed on June 28, 2026
[18] Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J., Li, X., Wu, B., Zhang, J., et al.: Hunyuanvideo: A systematic framework for large video generative models. arXiv preprint arXiv:2412.03603 (2024)
[19] Lipman, Y., Chen, R.T.Q., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling (2023), https://arxiv.org/abs/2210.02747
[20] Liu, J., Liu, G., Liang, J., Li, Y., Liu, J., Wang, X., Wan, P., Zhang, D., Ouyang, W.: Flow-grpo: Training flow matching models via online rl. arXiv preprint arXiv:2505.05470 (2025)
[21] Liu, M., Zhang, W.: Is your video language model a reliable judge? In: The Thirteenth International Conference on Learning Representations (2025), https://openreview.net/forum?id=m8yby1JfbU
[22] Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow (2022), https://arxiv.org/abs/2209.03003
[23] LumaLabs: Dream machine (June 2024), accessed: 2024
[24] MiniMax: Minimax hailuo 2.3. Online (Dec 2024), https://www.minimax.io/news/minimax-hailuo-23
[25] OpenAI: GPT-5 (2025), accessed on June 28, 2026
[26] OpenAI: OpenAI o3 and o4-mini System Card. Tech. rep., OpenAI (April 2025), accessed: 2025-11-01
[27] OpenAI: Sora 2 system card. Tech. rep., OpenAI (September 2025), https://cdn.openai.com/pdf/50d5973c-c4ff-4c2d-986f-c72b5d0ff069/sora_2_system_card.pdf
[28] Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.Y., Li, S.W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: Dinov2: Learning robust visual features without supervision (2024), https://arxiv.org/abs/2304.07193
[29] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. Advances in neural information processing systems **35**, 27730–27744 (2022)
[30] PikaLabs: Pika 1.5 (October 2024), accessed: 2024
[31] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision (2021), https://arxiv.org/abs/2103.00020
[32] Runway: Introducing gen-3 alpha: A new frontier for video generation (June 2024), accessed: 2024
[33] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms (2017), https://arxiv.org/abs/1707.06347
[34] Wan: Wan2.5-i2v-preview. Online (2025), https://www.alibabacloud.com/en/events/wan2-5-preview-launch?_p_lc=1
[35] Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)

[36] Wang, D., Ye, H., Liang, Z., Sun, Z., Lu, Z., Zhang, Y., Zhao, Y., Gao, Y., Seegert, M., Schäfer, F., Qin, H., Li, W., Palmieri, L., Jahncke, F., Piccinini, M., Betz, J.: Target-bench: Can world models achieve mapless path planning with semantic targets? (2025), https://arxiv.org/abs/2511.17792

[37] Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., Zhou, D.: Self-consistency improves chain of thought reasoning in language models. arXiv preprint arXiv:2203.11171 (2022)

[38] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems **35**, 24824–24837 (2022)

[39] Wiedemer, T., Li, Y., Vicol, P., Gu, S.S., Matarese, N., Swersky, K., Kim, B., Jaini, P., Geirhos, R.: Video models are zero-shot learners and reasoners (2025), https://arxiv.org/abs/2509.20328

[40] Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., Yin, S.m., Bai, S., Xu, X., Chen, Y., et al.: Qwen-image technical report. arXiv preprint arXiv:2508.02324 (2025)

[41] Xie, J., Yang, Z., Shou, M.Z.: Show-o2: Improved native unified multimodal models. arXiv preprint arXiv:2506.15564 (2025)

[42] Xue, Z., Wu, J., Gao, Y., Kong, F., Zhu, L., Chen, M., Liu, Z., Liu, W., Guo, Q., Huang, W., et al.: Dancegrpo: Unleashing grpo on visual generation. arXiv preprint arXiv:2505.07818 (2025)

[43] Yang, C., Wan, H., Peng, Y., Cheng, X., Yu, Z., Zhang, J., Yu, J., Yu, X., Zheng, X., Zhou, D., Wu, C.: Reasoning via video: The first evaluation of video models' reasoning abilities through maze-solving tasks (2025), https://arxiv.org/abs/2511.15065

[44] Yue, Y., Chen, Z., Lu, R., Zhao, A., Wang, Z., Yue, Y., Song, S., Huang, G.: Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? (2025), https://arxiv.org/abs/2504.13837

[45] Zheng, Z., Peng, X., Yang, T., Shen, C., Li, S., Liu, H., Zhou, Y., Li, T., You, Y.: Open-sora: Democratizing efficient video production for all. arXiv preprint arXiv:2412.20404 (2024)

[46] Zhou, J., Ji, J., Chen, B., Sun, J., Chen, W., Hong, D., Han, S., Guo, Y., Yang, Y.: Generative rlhf-v: Learning principles from multi-modal human preference (2025), https://arxiv.org/abs/2505.18531