World Model Self-Distillation: Training World Models to Solve General Tasks

Anonymous Author(s)
Affiliation
Address
email

Abstract

Pretrained video generators are promising visual world models that exhibit emergent task-solving abilities; however, their reliance on detailed textual descriptions limits their direct use for planning and decision-making. Existing approaches either outsource this reasoning to language or vision-language models, or rely on supervised fine-tuning with paired task-execution videos, which are costly to collect and difficult to scale. We propose a scalable framework that elicits task-solving ability in such models by combining self-distillation with reinforcement learning. Given an unlabeled scene image, a vision-language model generates a candidate task and a detailed step-by-step solution. The solution conditions a pretrained video diffusion model, the *Demonstrator*; we distill its behavior into an *Executor* conditioned only on the image and a short task prompt. This transfers execution knowledge from caption-guided generation to instruction-conditioned task solving without curated task-video supervision. We further improve the Executor with reinforcement learning from VLM feedback, exploiting the asymmetry between judging whether a sampled video satisfies a task and generating the solution. Experiments on *WorldTasksBench* and the DreamGen robotics benchmark show that the Executor surpasses the Demonstrator, generalizes to held-out tasks and scenes, and transfers competitively to robotic tasks.

1 Introduction

World models are a promising paradigm for enabling agents to reason about their environment by internally simulating possible action sequences and selecting those that best achieve a desired goal Ha and Schmidhuber [2018]. Recent advances in visually pretrained world models, particularly video generative models, have demonstrated striking emergent capabilities that resemble intelligent behavior Guo et al. [2025], Wiedemer et al. [2025], Acuaviva et al. [2025].

Common instantiations of such world models are pretrained text- or image-to-video generators He et al. [2025a], Hassan et al. [2024], Hong et al. [2025]. However, their reliance on textual conditioning, typically requiring a detailed description of the scene or action, limits their direct applicability to task solving. In practice, they do not autonomously infer how to execute a task; instead, they depend on the reasoning of external models such as language models or vision-language models (VLMs) to specify the solution. Ideally, we would like the world model to be able to accept a high-level task description and internally generate a plausible sequence of actions, thereby directly leveraging the knowledge acquired during pretraining.

One direct way to close this gap is supervised fine-tuning: collect pairs of task instructions and videos that demonstrate successful executions, and train the video model to generate the corresponding trajectory. However, this approach requires a large and diverse set of successful demonstrations, covering many environments, objects, and levels of task abstraction. Acquiring such data is costly,

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

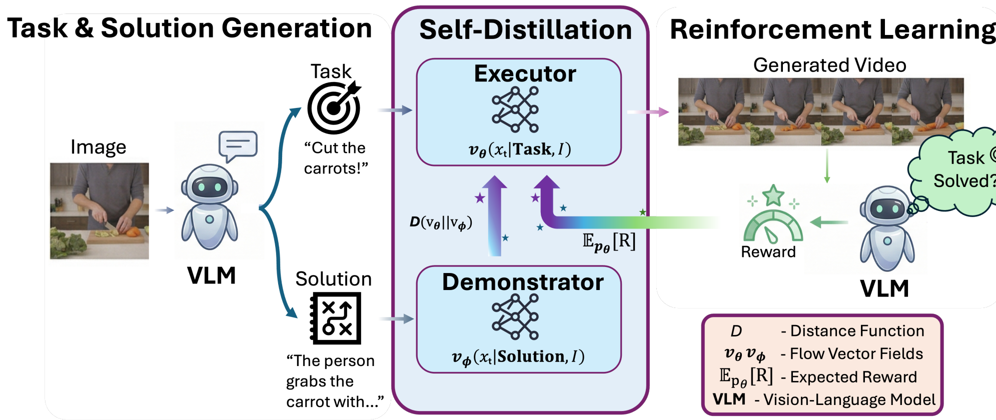


Figure 1: Overview of WMSD. The method addresses general tasks via a two-stage pipeline. **(Left)** A vision–language model (VLM) generates task descriptions along with corresponding solution prompts. **(Bottom → Top)** These solutions supervise the distillation of a video diffusion model (**Demonstrator**) into a task-conditioned video model (**Executor**), enabling the Executor to reproduce effective reasoning strategies. **(Right)** To further improve performance, reinforcement learning is applied: the VLM evaluates generated solution videos and provides feedback to refine the Executor.

especially when tasks are long-horizon or when success depends on fine-grained object interactions. Large-scale world-model platforms and video curation pipelines reduce this burden, but they do not remove the need for scalable task supervision Agarwal et al. [2025].

Reinforcement learning offers a complementary route. Instead of imitating only fixed demonstrations, a model can sample candidate solutions, receive feedback, and improve the probability of generations that satisfy the task. This paradigm has been central to preference-based training of language models Christiano et al. [2017], Ouyang et al. [2022], and recent work has begun to adapt RL objectives to diffusion and flow-based generative models Black et al. [2024], Liu et al. [2025a]. In the video domain, however, this strategy faces a severe computational bottleneck. The most successful video generators are commonly based on diffusion or flow matching, and producing even a short clip may require many denoising or integration steps Lipman et al. [2022]. Since RL requires many rollouts per update, naively applying RL to multi-step video generators is prohibitively expensive.

Few-step distillation helps address this bottleneck. Distribution Matching Distillation (DMD) trains a fast student to match a slower diffusion teacher by minimizing an approximate distributional divergence between student and teacher samples Yin et al. [2023]. Because the objective can be evaluated on student-generated samples, it is attractive for iterative improvement without paired real videos at every update Agarwal et al. [2023], Yin et al. [2024].

We argue that a similar framework can be leveraged beyond efficiency gains and used instead to elicit task-solving capabilities in video world models. First, by conditioning the student model, which we call the *Executor*, on high-level task instructions (e.g., "cut the carrots") together with an initial observation, and training it to match teacher, or *Demonstrator*, outputs conditioned on detailed execution descriptions, the student learns to map instructions directly to plausible action sequences. This effectively transforms the generator into an instruction-following, task-solving world model. Because this approach operates in a self-distillation setting, it remains constrained by the task-solving ability of the demonstrator, effectively placing an upper bound on performance. To move beyond this limitation, reinforcement learning is introduced into the process. Generated rollouts can then be evaluated by a VLM, which assesses whether the produced video successfully fulfills the given instruction.

This relies on a generation-verification asymmetry: for many structured tasks, finding a valid solution can be much harder than checking a proposed one Song et al. [2025]. In our setting, we instantiate this verifier with a vision-language model, following work showing that VLMs can serve as zero-shot reward models for language-specified visual tasks Rocamonde et al. [2023], Wang et al. [2024], Jiang et al. [2025]. Nevertheless, raw VLM rewards can be noisy and inconsistent, especially for ambiguous visual tasks. We therefore view VLM feedback not as a standalone ground-truth reward, but as a weak verification signal to be combined with distributional regularization from the teacher. The combination with self-distillation provides a natural way to stabilize this signal. We call our method *World Model Self-Distillation* (WMSD) and give a general overview in Fig. 1.

To summarize, our **main contributions** are:

1. We propose a self-distillation method that turns pretrained caption-conditioned video diffusion models into instruction-conditioned task solvers, without requiring paired task-execution videos.
2. We augment this distillation procedure with reinforcement learning from VLM feedback, allowing the task-executing model to surpass its teacher and remain competitive with methods trained using curated task-specific supervision.
3. We provide a task-solution prompt dataset that leverages VLMs to derive tasks and detailed execution descriptions from unlabeled scene images.
4. We provide a benchmark for evaluating general task solving in generated videos.

2 Related Work

Task-Conditioned World Models Prior work conditions world models on language, actions, or task specifications, including large-scale video foundation models Agarwal et al. [2025] and planning systems that combine video generation with vision-language models Du et al. [2023], Pan et al. [2024]. We focus on a weaker inference-time interface: the Executor receives only an image and a short task instruction, while privileged step-by-step descriptions are used only during training through the Demonstrator.

Self-Distillation and Distribution Matching On-policy self-distillation and iterative refinement can improve models under distribution shift, especially when combined with reinforcement learning Agarwal et al. [2023], Hubotter et al. [2026], Shenfeld et al. [2026]. Distribution matching similarly aligns student and teacher generative distributions, often for efficiency and stability Yin et al. [2023]. We use this asymmetry for task transfer rather than only acceleration: the teacher sees detailed execution descriptions, whereas the student must solve the task from a compact instruction.

Reinforcement Learning for Flow Models Recent methods adapt policy optimization to diffusion and flow-based generators and improve training stability through flow-specific refinements Liu et al. [2025a], He et al. [2025b], Xue et al. [2025b], Li et al. [2025]. In contrast to reward-only alignment, we combine VLM task rewards with a Demonstrator-derived distillation reward and anchor loss, so RL improves task success while teacher guidance regularizes visual dynamics.

3 Method

Setup We use conditional flow-matching video models Lipman et al. [2022] in a teacher–student setting. Each example contains an initial observation $\mathcal{I}$, a short task instruction $\mathcal{T}$, and a detailed execution description $\mathcal{D}$. The student, or *Executor*, is conditioned only on $c_{\mathrm{E}} = (\mathcal{I}, \mathcal{T})$, whereas the teacher, or *Demonstrator*, is conditioned on the richer description $c_{\mathrm{D}} = (\mathcal{I}, \mathcal{D})$. The teacher is fixed with parameters θ', while the student has trainable parameters θ.

Let $x_t \in \mathbb{R}^d$ be the latent video state at flow time $t \in [0, 1]$, with $x_0 \sim p_0$, where p_0 is the Normal distribution, and $x_1 \sim p_1$, where p_1 denotes the latent video data distribution. At inference time, x_1 is decoded into the generated video. The student and teacher define velocity fields $v_\theta(x_t, t \mid c_{\mathrm{E}})$ and $v_{\theta'}(x_t, t \mid c_{\mathrm{D}})$. A student flow trajectory satisfies

$$\frac{dx_t}{dt} = v_\theta(x_t, t \mid c_{\mathrm{E}}), \qquad x_0 \sim p_0. \tag{1}$$

Teacher trajectories are analogous, replacing v_θ, c_{E} with $v_{\theta'}, c_{\mathrm{D}}$. Let $\tau = \{x_t\}_{t \in [0,1]}$ denote a trajectory, with $p_\theta(\tau \mid c_{\mathrm{E}})$ and $p_{\theta'}(\tau \mid c_{\mathrm{D}})$ denoting the trajectory distributions induced by the student and teacher samplers. With a small abuse of notation, we write $p_\theta(x_t, t \mid c_{\mathrm{E}})$ for the corresponding student state-time occupancy distribution obtained by sampling $\tau \sim p_\theta(\cdot \mid c_{\mathrm{E}})$ and $t \sim \mathcal{U}[0, 1]$.

The goal is to train the student to solve tasks from c_{E}, using the teacher under c_{D} as dense guidance.

Off-policy distillation Matching the student velocity to the teacher velocity at teacher states gives

$$\mathcal{L}_{\mathrm{off}} = \mathbb{E}_{(x_t,t)\sim p_{\theta'}(\cdot|c_{\mathrm{D}})}\left[\|v_\theta(x_t,t\mid c_{\mathrm{E}}) - v_{\theta'}(x_t,t\mid c_{\mathrm{D}})\|_2^2\right]. \tag{2}$$

This objective is stable because sampled states do not depend on the student Lipman et al. [2024], but it constrains the student only on teacher trajectories, so errors may compound during student rollouts.

On-policy distillation To reduce this mismatch, we evaluate teacher–student discrepancy on student trajectories. Define

$$\ell_\theta(x_t,t;c_{\mathrm{E}},c_{\mathrm{D}}) = \|v_\theta(x_t,t\mid c_{\mathrm{E}}) - v_{\theta'}(x_t,t\mid c_{\mathrm{D}})\|_2^2. \tag{3}$$

The on-policy objective is

$$\mathcal{L}_{\mathrm{on}} = \mathbb{E}_{(x_t,t)\sim p_\theta(\cdot|c_{\mathrm{E}})}\left[\ell_\theta(x_t,t;c_{\mathrm{E}},c_{\mathrm{D}})\right] = \mathbb{E}_{\tau\sim p_\theta(\cdot|c_{\mathrm{E}})}\left[\int_0^1 \ell_\theta(x_t,t;c_{\mathrm{E}},c_{\mathrm{D}})\,dt\right]. \tag{4}$$

Unlike $\mathcal{L}_{\mathrm{off}}$, $\mathcal{L}_{\mathrm{on}}$ depends on θ through both the velocity field and the student rollout distribution. Let

$$C_\theta(\tau) = \int_0^1 \ell_\theta(x_t,t;c_{\mathrm{E}},c_{\mathrm{D}})\,dt \tag{5}$$

denote the trajectory-level distillation cost. The score-function decomposition gives

$$\nabla_\theta\mathcal{L}_{\mathrm{on}} = \mathbb{E}_{\tau\sim p_\theta(\cdot|c_{\mathrm{E}})}\left[C_\theta(\tau)\nabla_\theta \log p_\theta(\tau\mid c_{\mathrm{E}})\right] + \mathbb{E}_{\tau\sim p_\theta}\left[\nabla_\theta C_\theta(\tau)\right]. \tag{6}$$

The first term changes the likelihood of trajectories according to their teacher–student discrepancy and has the form of a policy-gradient update with negative reward $-C_\theta(\tau)$. The second is direct vector-field regression on student states. We then show that matching teacher velocity on student states, under shared initial noise, bounds student–teacher trajectory drift.

Proposition 1 (Informal on-policy control)**.** *Assume the teacher velocity field is Lipschitz in x and that the student and teacher flows are initialized from the same base noise $x_0 \sim p_0$. If the student matches the teacher's velocity field on its own trajectories, namely*

$$\mathcal{L}_{\mathrm{on}} \le \varepsilon^2$$

, then the terminal distributions induced by the student and teacher are close. In particular, under the natural coupling given by the shared initial noise,

$$W_2\left(p_\theta(x_1\mid c_{\mathrm{E}}), p_{\theta'}(x_1\mid c_{\mathrm{D}})\right) \le e^L\varepsilon, \tag{7}$$

where L is a Lipschitz constant of the teacher flow, and $\varepsilon \ge 0$.

The proof is a standard Grönwall argument (see Appendix).

Distillation as a reward Eq. (6) suggests an RL view: trajectories with low teacher–student discrepancy should become more likely. We therefore define

$$r_{\mathrm{distill}}(\tau) = -\int_0^1 \|\mathrm{sg}[v_\theta(x_t,t\mid c_{\mathrm{E}})] - v_{\theta'}(x_t,t\mid c_{\mathrm{D}})\|_2^2\,dt, \tag{8}$$

where $\mathrm{sg}[\cdot]$ denotes stop-gradient. Detaching the student makes this term act through trajectory likelihood rather than direct velocity-field backpropagation, up-weighting rollouts whose dynamics agree with the Demonstrator.

Reinforcement learning for task solving Pure distillation imitates the teacher but cannot systematically improve beyond it. Since eq. 6 has a score-function form, we add task-level feedback. Let $r_{\mathrm{task}}(\tau;\mathcal{I},\mathcal{T})$ denote whether the generated video solves $\mathcal{T}$ from $\mathcal{I}$, as judged by a VLM.

The total reward is then

$$R(\tau) = \lambda_{\mathrm{task}} r_{\mathrm{task}}(\tau;\mathcal{I},\mathcal{T}) + \lambda_{\mathrm{distill}} r_{\mathrm{distill}}(\tau), \tag{9}$$

with $\lambda_{\mathrm{task}} > 0$ and $\lambda_{\mathrm{distill}} > 0$ controlling task success versus teacher agreement.

The teacher now acts as a stabilizing prior rather than a hard target: task reward can favor student trajectories that better satisfy the instruction even when they deviate from the teacher.

Optimization objective For the direct regression component $\mathbb{E}_{\tau \sim p_\theta}[\nabla_\theta C_\theta(\tau)]$ in eq. (6), full backpropagation through all sampler steps is impractical. We therefore use the anchor loss

$$\mathcal{L}_{\text{anchor}} = \mathbb{E}_{\tau \sim p_\theta(\cdot \mid c_{\text{E}})} \left[\int_0^1 \| v_\theta(\bar{x}_t, t \mid c_{\text{E}}) - v_{\theta'}(\bar{x}_t, t \mid c_{\text{D}}) \|_2^2 \, dt \right], \tag{10}$$

where $\bar{x}_t = \text{sg}(x_t)$ is a sampled state treated as fixed. The reward term selects trajectories; the anchor keeps the student velocity close to the teacher on those states.

We optimize the student with a policy-gradient objective for flow-matching models, $\mathcal{L}_{\text{RL}}$, using eq. (9) and implement it via several RL approaches Liu et al. [2025a], Xue et al. [2025a], Zheng et al. [2025]: groups of rollouts for the same task define relative advantages that increase the likelihood of higher-reward rollouts, see Sec. A.8.

Finally, we combine reward optimization with teacher anchoring with $\beta_d > 0$ in our *full self-distillation objective*

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{RL}} + \beta_d \mathcal{L}_{\text{anchor}}. \tag{11}$$

Self-distillation transfers detailed execution knowledge, RL improves task success, and the Demonstrator anchor prevents uncontrolled drift while still allowing the Executor to surpass the Demonstrator.

4 Experiments

We evaluate our method along three main axes. First, we compare the two proposed self-distillation variants and examine whether on-policy self-distillation provides a competitive alternative to standard off-policy self-distillation. Second, we study the interaction between self-distillation and reinforcement learning, asking whether the student can improve beyond the teacher's capabilities. Finally, we evaluate generalization to unseen robotic tasks.

4.1 Experimental Setup

We operate in the Advantage-Weighted Matching (AWM) setting, a variant of GRPO better suited to flow-matching models Xue et al. [2025a]. Unless otherwise stated, all experiments use a group size of 24 and a batch size of 32, with LTX-2 HaCohen et al. [2026] as the baseline model. Training alternates between on-policy rollout generation, reward computation, and joint policy optimization with self-distillation. Additional implementation details are provided in Sec. A.1.

Rewards. For experiments involving VLM-based reward signals, we use two complementary components: a task-completion reward and a consistency reward. Task success is evaluated with Qwen3.5-27B Team [2026], which produces a binary judgment indicating whether a generated video completes the specified task. We define the reward as the log-probability difference

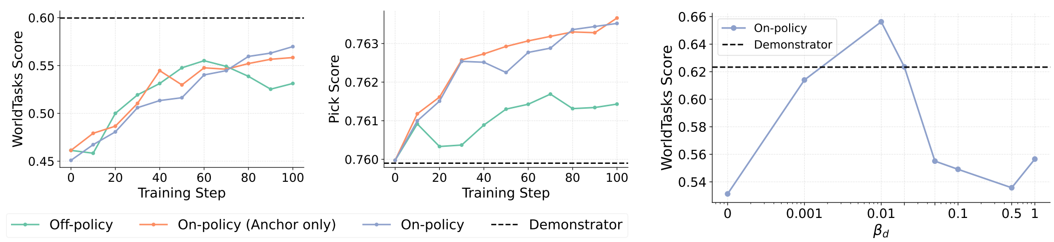
$$R(x) = \log p_{\text{VLM}}(\text{`yes'} \mid x) - \log p_{\text{VLM}}(\text{`no'} \mid x),$$

which incorporates both the predicted outcome and the model's uncertainty. However, optimizing this signal alone can lead to reward hacking, such as unrealistic object appearances or disappearances. Inspired by Agarwal et al. [2025], we introduce a consistency reward to mitigate this issue that penalizes violations of physical plausibility and temporal coherence. Full prompts and implementation details are provided in Appendix Sec. A.2.2 and Boxes 6–7.

4.1.1 WorldTasks Dataset

We construct a dataset of 20,000 images from video-game environments and real-world scenes, largely based on MiraData Ju et al. [2024]. Standard filtering removes low-quality frames and those with limited agentic potential (i.e., no meaningful interaction possible). For each image, we pre-generate eight task–solution pairs using Qwen3.5-27B, covering diverse instruction-following scenarios across environments and task complexities. Further details are provided in Appendix Sec. A.5.

To support learning beyond initial frames and VLM annotations, tasks are designed to be unambiguous yet general. The world model represents all visible entities, not just an egocentric view, enabling settings such as ego-exo modeling and general planning. Instructions are formatted as "[Agent description]: [Task instruction]" to specify the acting agent.



(a) (b)

Figure 2: **Two ablations on *WorldTasksBench*.** Left: Ablation on self-distillation methods, showing average WorldTasks score and PickScore. Right: Ablation of average WorldTasks score vs. β_d.

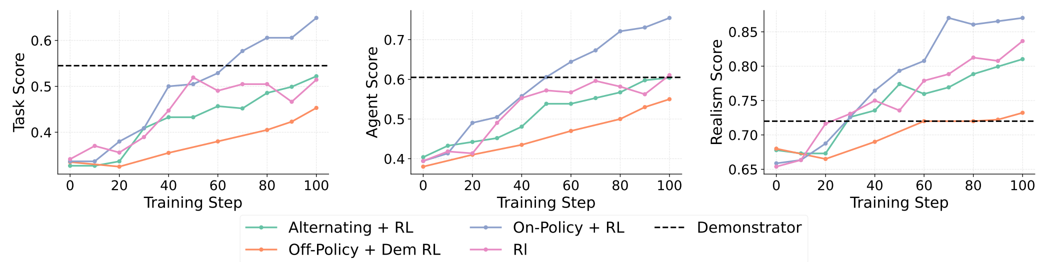


Figure 3: **Ablation across training settings on *WorldTasksBench*.** We report the three evaluation dimensions.

4.1.2 WorldTasks Benchmark

We first study the core properties of *WMSD* in a controlled setting. The corresponding benchmark, **WorldTasks-Bench**, consists of 200 randomly selected image–task pairs. Each generated video is evaluated by a VLM according to three criteria: (1) whether the task is completed, (2) whether the correct agent attempts the task, and (3) whether the video exhibits consistent physics and realistic dynamics. The evaluation prompts are provided in Appendix Sec. A.6.

We report three metrics throughout all experiments. The *Task Score* measures the success rate of task completion as judged by the VLM. The *Agent Score* captures whether the intended agent engages in goal-directed interaction within the scene. The *Realism Score* evaluates physical plausibility and temporal coherence.

4.2 On-policy vs. Off-policy Self-Distillation

We begin by comparing the three self-distillation variants introduced in Sec. 3: off-policy self-distillation, on-policy self-distillation using only the anchor loss between student and teacher, and the full on-policy self-distillation objective eq. (11). In Fig. 2, we report evaluation results every 10 training steps over 100 training steps. We show both the average WorldTasks score and PickScore Kirstain et al. [2023], which measures overall generation quality.

All three methods yield substantial improvements. However, after approximately 60 training steps, off-policy self-distillation saturates, whereas both on-policy variants continue to improve on both metrics and ultimately surpass the off-policy baseline. The full on-policy self-distillation objective, which includes the distillation reward introduced in eq. 4, achieves the best overall performance.

4.3 Surpassing the Demonstrator with RL Training

We investigate whether augmenting self-distillation with reinforcement learning (RL) enables the student to surpass the demonstrator's task-solving performance. To this end, we consider four training settings: (i) standard RL without an anchor loss, (ii) on-policy self-distillation with RL applied to the student, (iii) off-policy distillation with RL applied to the teacher, and (iv) an alternating optimization strategy in which teacher and student updates are interleaved according to a fixed schedule. The

Table 1: **Comparison of *WMSD* against baselines on *WorldTasksBench*.** We report task completion, agent correctness, physical consistency, their average, and end-to-end inference time. *WMSD* consistently improves performance across both base models while preserving the inference cost of the underlying model. * Trained with GRPO for 25 steps.

Method	Task ↑	Agent ↑	Consistency ↑	Avg. ↑	E2E Time (s) ↓
HY1.5	0.463	0.540	0.780	0.597	112
HY1.5+*WMSD**	0.574	0.630	0.828	0.673	112
LTX2	0.315	0.395	0.690	0.467	52.2
LTX2+SFT	0.292	0.389	0.682	0.454	52.2
LTX2+*WMSD**	0.452	0.500	0.693	0.548	52.2
LTX2 (8-Step)	0.285	0.391	0.694	0.455	10.1
LTX2 (8-Step)+VLM	0.495	0.572	0.732	0.598	10.1
LTX2 (8-Step)+*WMSD*	**0.605**	**0.691**	**0.882**	**0.726**	10.1

full procedure is detailed in Alg. 1 (see Appendix). As an additional baseline, we include the *Demonstrator* setting, in which reasoning is entirely outsourced to the VLM for solution generation.

We evaluate all approaches on the three components of *WorldTasksBench*: task-solving performance, agent correctness, and physical consistency. The results are shown in Fig. 3.

Our results show that combining on-policy self-distillation with RL substantially improves task-solving performance and enables the student to surpass the demonstrator. In contrast, standard RL alone achieves comparable performance during early training, up to approximately 50 steps, but quickly plateaus and yields no further gains. The remaining approaches exhibit slower learning dynamics and do not reach the same level of performance.

4.4 Comparison to Baselines

We compare *WMSD* against several baselines. We first examine whether *WMSD* generalizes across different base models, reward functions, and RL optimization settings. To this end, we use Hunyuan-Video 1.5 Team [2025] as the base model, Qwen3-VL-8B as the reward model, and FlowGRPO Liu et al. [2025b] as the RL optimizer, training for 25 steps.

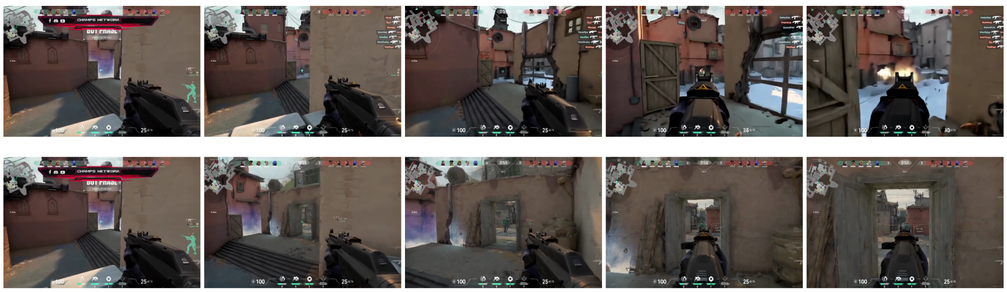
For the LTX-2 HaCohen et al. [2026] 8-step model, using the setup described in Sec. 4.1, we compare against multiple baselines. First, we consider direct solution generation by outsourcing reasoning to a VLM. In this setting, the first frame and task description are provided to the VLM, which generates an image-to-video solution prompt that is then used for video generation (**+VLM**). We also investigate whether unannotated videos can be converted into task–video pairs by labeling them with corresponding tasks and subsequently fine-tuning the model via supervised fine-tuning (**+SFT**). Finally, we compare against *WMSD*. All results on *WorldTasksBench* are reported in Tab. 1.

The results show that, even after only 25 training steps, applying *WMSD* to the HY1.5 baseline yields substantial improvements across all metrics, highlighting the robustness and applicability of our method across different training settings. Applying *WMSD* to the 8-step distilled LTX-2 model further leads to significant gains across all evaluation metrics. In contrast, the SFT baseline provides little to no improvement and, in some cases, degrades performance. We hypothesize that this is due to limited task diversity in the automatically annotated data: many tasks are overly simple or repetitive, such as "walk forward", and therefore fail to capture meaningful real-world interaction scenarios.

4.5 Ablation Studies

Self-distillation strength. We study the effect of the self-distillation strength on the performance of *World-Model Self-Distillation* by varying the KL coefficient between demonstrator and executor, β_d in eq. 11, over the range $[0, 1]$. As shown in Fig. 2, the best performance is obtained around $\beta_d = 0.01$. Both smaller and larger values perform worse: too little regularization weakens the distillation signal, whereas too much regularization dominates the RL objective and limits learning.

(a) Prompt: "*[Lara Croft]: Turn left and position yourself directly in front of the central arch.*"



(b) Prompt: "*[First-person view]: Aim the weapon at the doorway entrance.*"

Figure 4: Two examples: the first row uses the consistency reward, while the second row does not. The second row shows that the model generates (a) the arch + Lara Croft and (b) the doorway as a consequence of reward hacking.

Consistency reward. We further investigate the effect of the additional consistency reward, which is designed to mitigate reward hacking. Without this reward, the model can exploit the VLM reward by producing implausible generations, such as objects appearing or disappearing without physical justification. The exact prompt used for this reward is provided in Box 7. Fig. 4 shows qualitative examples with and without the consistency reward.

Resolution and inference steps. Following prior work of Liu et al. [2025b], Ping et al. [2025], we decouple the number of denoising steps and the resolution used during training and evaluation to improve training efficiency. However, we find that this introduces a trade-off: lower generation quality during training increases the risk of reward hacking, especially in our setting where the VLM requires clear and unambiguous visual evidence to assign reliable rewards.

Additional ablations are provided in the Appendix.

4.6 Generalization to Robotic Tasks

An important application of world models for planning lies in robotics, where data collection is particularly expensive. We therefore evaluate whether *WMSD* trained on *WorldTasks* can achieve competitive performance without task-specific supervision, compared to supervised fine-tuning (SFT) on the Gr00t dataset using the DreamGen benchmark Jang et al. [2025] (Tab. 2).

We compare our LTX-2-based model against several baselines, including HunyuanVideo (Huny) Kong et al. [2024], CogVideoX (CogX) Hong et al. [2022], Wan Wang et al. [2025], and Cosmos Agarwal et al. [2025], across zero-shot and SFT settings.

We observe that, despite operating in a data-free regime, *WMSD* achieves performance comparable to SFT-trained Cosmos, while substantially improving over the LTX-2 baseline.

Table 2: Performance on the DreamGen benchmark. We compare zero-shot and SFT baselines against *WMSD*. Despite not using task-specific supervision, *WMSD* achieves competitive performance with SFT-trained models. Best results are in bold, second-best underlined.

	Zero-shot					**SFT**				***WMSD***
Metric	Huny.	CogX	Wan	Cosmos	LTX2	Huny.	CogX	Wan	Cosmos	LTX2
Object	0.0	0.0	2.0	32.0	20.0	26.0	38.0	58.0	62.0	**70.0**
Behavior	2.1	0.0	2.1	31.9	29.8	10.6	28.0	55.3	**61.7**	57.4
Env	0.0	0.0	6.7	24.1	41.4	27.6	41.4	**65.5**	**65.5**	58.6

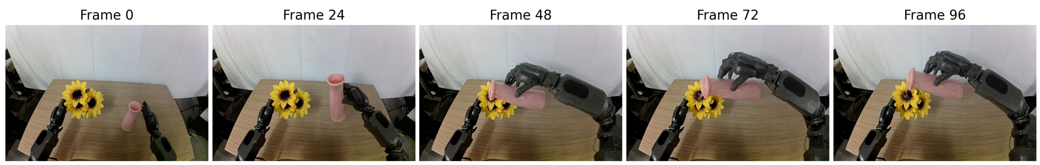


Figure 5: Example video generated with *WMSD* and LTX-2 on the DreamGen benchmark. Task: *"Use the right hand to pick up the pink bottle and pour water on the flower."*

4.7 Discussion & Limitations

Generalizability Training with *WMSD* leads to substantial improvements on *WorldTasksBench* as well as on robotics-related tasks (Sec. 4.6), achieving performance competitive with supervised fine-tuning. Furthermore, recent advances in distilling video generators into few-step models enable efficient RL-based optimization. We show that *WMSD* can effectively leverage these distilled models, resulting in significant gains in training efficiency.

WMSD cannot make up for the lack of information The results on the DreamGen benchmark highlight an inherent limitation of the data-free setting. In particular, the model cannot recover accurate robot-specific dynamics without access to corresponding data. While *WMSD* generates plausible task solutions, it lacks detailed knowledge of the appearance and motion characteristics of a specific robotic platform beyond the initial frame; see Fig. 5. This limitation is intrinsic to the data-free nature of *WMSD*. Extending *WMSD* to video continuation and in-context learning (ICL) could resolve this issue.

Out-of-Distribution Tasks We further investigate performance on out-of-distribution tasks, such as puzzle-based games Wang et al. [2026]. As detailed in Appendix Sec. A.4, when the Demonstrator fails to produce coherent solutions, *WMSD* still yields improvements, albeit with diminished gains. This observation motivated the alternating RL training strategy (Sec. 1); however, as shown in Fig. 3, this approach introduces additional instability.

5 Conclusion

In summary, the experiments show that *WMSD* consistently improves task-solving ability, agent correctness, and physical consistency across a wide range of settings. A key strength of the framework is that it converts the detailed execution knowledge available to caption-guided video generation into a compact instruction-following interface, without requiring curated task-execution videos. In particular, combining on-policy self-distillation with reinforcement learning proves especially effective, enabling the model to surpass the Demonstrator while maintaining efficient inference. The VLM-based reward further lets the model exploit the asymmetry between generating a correct future and recognizing one, turning noisy task-level feedback into measurable gains. At the same time, the Demonstrator anchor preserves useful pretrained behavior and prevents reinforcement learning from drifting toward visually implausible solutions. Beyond controlled benchmarks, the competitive transfer to robotic tasks further highlights the robustness and generality of the approach, suggesting that *WMSD* is a promising direction for scalable and data-efficient world model training.

References

Pablo Acuaviva, Aram Davtyan, Mariam Hassan, Sebastian Stapf, Ahmad Rahimi, Alexandre Alahi, and Paolo Favaro. Rethinking visual intelligence: Insights from video pretraining. *ArXiv*, abs/2510.24448, 2025.

Nvidia Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, Daniel Dworakowski, JiaoJiao Fan, Michele Fenzi, Francesco Ferroni, Sanja Fidler, Dieter Fox, Songwei Ge, Yunhao Ge, Jinwei Gu, Siddharth Gururani, Ethan He, Jiahui Huang, Jacob Samuel Huffman, Pooya Jannaty, Jingyi Jin, Seung Wook Kim, Gergely Kl'ar, Grace Lam, Shiyi Lan, Laura Leal-Taixé, Anqi Li, Zhaoshuo Li, Chen-Hsuan Lin, Tsung-Yi Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Arsalan Mousavian, Seungjun Nah, Sriharsha Niverty, David Page, Despoina Paschalidou, Zeeshan Patel, Lindsey Pavao, Morteza Ramezanali, Fitsum A. Reda, Xiao-Shuai Ren, Vasanth Rao Naik Sabavat, Ed Schmerling, Stella Shi, Bartosz Stefaniak, Shitao Tang, Lyne P. Tchapmi, Przemek Tredak, Wei-Cheng Tseng, Jibin Rajan Varghese, Hao Wang, Haoxiang Wang, Hengyi Wang, Tingwei Wang, Fangyin Wei, Xinyue Wei, Jay Zhangjie Wu, Jiashu Xu, Wei Yang, Lin Yen-Chen, Xiaohui Zeng, Yuan Zeng, Jing Zhang, Qinsheng Zhang, Yuxuan Zhang, Qingqing Zhao, and Artur Zolkowski. Cosmos world foundation model platform for physical ai. *ArXiv*, abs/2501.03575, 2025.

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stańczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2023.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=YCWjhGrJFD`.

Paul Francis Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *ArXiv*, abs/1706.03741, 2017.

Yilun Du, Mengjiao Yang, Peter R. Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Josh Tenenbaum, Leslie Pack Kaelbling, Andy Zeng, and Jonathan Tompson. Video language planning. *ArXiv*, abs/2310.10625, 2023.

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*, abs/2510.26802, 2025.

David R Ha and Jürgen Schmidhuber. World models. *ArXiv*, abs/1803.10122, 2018.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, Eitan Richardson, Guy Shiran, Itay Chachy, Jonathan Chetboun, Michael Finkelson, Michael Kupchick, Nir Zabari, Nitzan Bitton Guetta, Noa Kotler, Ofir Bibi, Ori Gordon, Poriya Panet, Roi Benita, Shahar Armon, V. M. Kulikov, Yaron Inger, Yonatan Shiftan, Zeev Melumian, and Zeev Farbman. Ltx-2: Efficient joint audio-visual foundation model. *ArXiv*, abs/2601.03233, 2026.

Mariam Hassan, Sebastian Stapf, Ahmad Rahimi, Pedro Martelleto Bressane Rezende, Yasaman Haghighi, David Brüggemann, Isinsu Katircioglu, Lin Zhang, Xiaoran Chen, Suman Saha, Marco Cannici, Elie Aljalbout, Botao Ye, Xi Wang, Aram Davtyan, Mathieu Salzmann, Davide Scaramuzza, Marc Pollefeys, Paolo Favaro, and Alexandre Alahi. Gem: A generalizable ego-vision multimodal world model for fine-grained ego-motion, object dynamics, and scene composition control. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22404–22415, 2024.

Haoran He, Yang Zhang, Liang Lin, Zhongwen Xu, and Ling Pan. Pre-trained video generative models as world simulators. *ArXiv*, abs/2502.07825, 2025a.

Xiaoxuan He, Siming Fu, Yuke Zhao, Wanli Li, Jian Yang, Dacheng Yin, Fengyun Rao, and Bo Zhang. Tempflow-grpo: When timing matters for grpo in flow models. *ArXiv*, abs/2508.04324, 2025b.

Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *ArXiv*, abs/2205.15868, 2022.

Yicong Hong, Yiqun Mei, Chongjian Ge, Yiran Xu, Yang Zhou, Sai Bi, Yannick Hold-Geoffroy, Mike Roberts, Matthew Fisher, Eli Shechtman, Kalyan Sunkavalli, Feng Liu, Zhengqi Li, and Hao Tan. Relic: Interactive video world model with long-horizon memory. *ArXiv*, abs/2512.04040, 2025.

Jonas Hubotter, Frederike Lubeck, Lejs Deen Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. *ArXiv*, abs/2601.20802, 2026.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, Loic Magne, Ajay Mandlekar, Avnish Narayan, You Liang Tan, Guanzhi Wang, Jing Wang, Qi Wang, Yinzhen Xu, Xi Zeng, Kaiyuan Zheng, Ruijie Zheng, Ming-Yu Liu, Luke S. Zettlemoyer, Dieter Fox, Jan Kautz, Scott Reed, Yuke Zhu, and Linxi Jim Fan. Dreamgen: Unlocking generalization in robot learning through video world models. In *Proceedings of The 9th Conference on Robot Learning*, pages 5170–5194, 2025. URL `https://proceedings.mlr.press/v305/jang25a.html`.

Dengyang Jiang, Dongyang Liu, Zanyi Wang, Qilong Wu, Liuzhuozheng Li, Hengzhuang Li, Xin Jin, David Liu, Zhen Li, Bo Zhang, Mengmeng Wang, Steven Hoi, Peng Gao, and Harry Yang. Distribution matching distillation meets reinforcement learning. *ArXiv*, abs/2511.13649, 2025.

Xu Ju, Yiming Gao, Zhaoyang Zhang, Ziyang Yuan, Xintao Wang, Ailing Zeng, Yu Xiong, Qiang Xu, and Ying Shan. Miradata: A large-scale video dataset with long durations and structured captions. *ArXiv*, abs/2407.06358, 2024.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *ArXiv*, abs/2305.01569, 2023.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jia-Liang Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fan Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Peng-Yu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhen Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yang-Dan Tao, Qinglin Lu, Songtao Liu, Daquan Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. Hunyuanvideo: A systematic framework for large video generative models. *ArXiv*, abs/2412.03603, 2024.

Junzhe Li, Yutao Cui, Tao Huang, Yi-Ting Ma, Chun Fan, Miles Yang, and Zhao Zhong. Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde. *ArXiv*, abs/2507.21802, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *ArXiv*, abs/2210.02747, 2022.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *ArXiv*, abs/2412.06264, 2024.

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*, abs/2505.05470, 2025a.

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*, abs/2505.05470, 2025b.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155, 2022.

Chenbin Pan, Burhaneddin Yaman, Tommaso Nesti, Abhirup Mallik, Alessandro Allievi, Senem Velipasalar, and Liu Ren. Vlp: Vision language planning for autonomous driving. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14760–14769, 2024.

Bowen Ping, Chengyou Jia, Minnan Luo, Changliang Xia, Xin Shen, Zhuohang Dang, and Hangwei Qian. Paco-rl: Advancing reinforcement learning for consistent image generation with pairwise reward modeling. *ArXiv*, abs/2512.04784, 2025.

Juan Rocamonde, Victoriano Montesinos, Elvis Nava, Ethan Perez, and David Lindner. Vision-language models are zero-shot reward models for reinforcement learning. *ArXiv*, abs/2310.12921, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Jun-Mei Song, Mingchuan Zhang, Y. K. Li, Yu Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *ArXiv*, abs/2402.03300, 2024. URL `https://api.semanticscholar.org/CorpusID:267412607`.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *ArXiv*, abs/2601.19897, 2026.

Yuda Song, Hanlin Zhang, Carson Eisenach, Sham M. Kakade, Dean Foster, and Udaya Ghai. Mind the gap: Examining the self-improvement capabilities of large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=mtJSMcF3ek`.

Qwen Team. Qwen3.5: Accelerating productivity with native multimodal agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

Tencent Hunyuan Foundation Model Team. Hunyuanvideo 1.5 technical report, 2025.

Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Xiaofeng Meng, Ningying Zhang, Pandeng Li, Ping Wu, Ruihang Chu, Rui Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wen-Chao Zhou, Wente Wang, Wen Shen, Wenyuan Yu, Xianzhong Shi, Xiaomin Huang, Xin Xu, Yan Kou, Yan-Mei Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhigang Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *ArXiv*, abs/2503.20314, 2025.

Maijunxian Wang, Ruisi Wang, Juyi Lin, Ran Ji, Thaddaus Wiedemer, Qingying Gao, Dezhi Luo, Yaoyao Qian, Lianyu Huang, Ze-Wen Hong, Jiahui Ge, Qianli Ma, Hang He, Yifan Zhou, Lingzi Guo, Lantao Mei, Jiacheng Li, Hanwen Xing, Tianqi Zhao, Feng Yu, Wei Xiao, Yizheng Jiao, Jian Hou, Danyang Zhang, Pengcheng Xu, Boyang Zhong, Ze Zhao, Gaoyun Fang, John Kitaoka, Yile Xu, Hua Xu, Kenton Blacutt, Tin Nguyen, Siyuan Song, Haoran Sun, Shao-Zhi Wen, Linyang He, Runming Wang, Yanzhi Wang, Mengyu Yang, Ziqiao Ma, Raphaël Millière, Freda Shi, Nuno Vasconcelos, Daniel Khashabi, Alan L. Yuille, Yilun Du, Ziming Liu, Bo Li, Dahua Lin, Ziwei Liu, Vikash Kumar, Yijiang Li, Lei Yang, Zhongang Cai, and Hokin Deng. A very big video reasoning suite. *ArXiv*, abs/2602.20159, 2026.

Yufei Wang, Zhanyi Sun, Jesse Zhang, Zhou Xian, Erdem Biyik, David Held, and Zackory Erickson. Rl-vlm-f: Reinforcement learning from vision language foundation model feedback. In *International Conference on Machine Learning*, 2024.

Thaddaus 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*, abs/2509.20328, 2025.

Shuchen Xue, Chongjian Ge, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning rl with pretraining in diffusion models. *ArXiv*, abs/2509.25050, 2025a.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, and Ping Luo. Dancegrpo: Unleashing grpo on visual generation. *ArXiv*, abs/2505.07818, 2025b.

Tianwei Yin, Michael Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6613–6623, 2023.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Frédo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22963–22974, 2024.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Mingying Liu. Diffusionnft: Online diffusion reinforcement with forward process. *ArXiv*, abs/2509.16117, 2025.

A Technical appendices and supplementary material

A.1 Further Implementation Details

In Tab. 3, we report the hyperparameters used for self-distilling LTX-2 and HunyuanVideo 1.5 in the experiments from Sec. 4.

A.2 Compute Resources

The primary results presented in Sec. 4.4 were obtained using a large-scale cluster comprising 128 GH200 GPUs. In contrast, the ablation studies in Sec. 4.5 were conducted on a smaller setup of 16 GH200 GPUs over a 12-hour period.

A.2.1 Distribution-Matching Self-Distillation

We also investigated Distribution Matching Distillation (DMD) as an alternative on-policy distillation objective combined with RL, following Jiang et al. [2025].

As background, diffusion models generate high-quality samples by iteratively denoising Gaussian noise. However, this multi-step sampling process is computationally expensive, motivating distillation into a one-step generator $G_\theta(z)$. Distribution Matching Distillation trains G_θ to match the distribution of a pretrained teacher diffusion model, rather than exactly reproducing its full denoising trajectory. The core objective is to align the generator-induced distribution with the teacher distribution by minimizing the KL divergence:

$$D_{\mathrm{KL}}(p_{\mathrm{fake}} \,\|\, p_{\mathrm{real}}) = \mathbb{E}_{x\sim p_{\mathrm{fake}}}\left[\log \frac{p_{\mathrm{fake}}(x)}{p_{\mathrm{real}}(x)}\right]. \tag{12}$$

The corresponding score-based gradient is given by

$$\nabla_\theta D_{\mathrm{KL}} = \mathbb{E}_z\left[-\left(s_{\mathrm{real}}(x) - s_{\mathrm{fake}}(x)\right)\frac{\partial G_\theta(z)}{\partial \theta}\right], \qquad x = G_\theta(z). \tag{13}$$

Because diffusion models estimate scores on noisy samples, DMD perturbs generated samples according to

$$q_t(x_t \mid x) = \mathcal{N}(\alpha_t x, \sigma_t^2 I), \tag{14}$$

and estimates the real and fake scores using diffusion denoisers:

$$s_{\mathrm{real}}(x_t, t) = -\frac{x_t - \alpha_t \mu_{\mathrm{base}}(x_t, t)}{\sigma_t^2}, \tag{15}$$

$$s_{\mathrm{fake}}(x_t, t) = -\frac{x_t - \alpha_t \mu^{\phi}_{\mathrm{fake}}(x_t, t)}{\sigma_t^2}. \tag{16}$$

Category	LTX-2	HunyuanVideo-1.5
Base model	`LTX-2-19b-distilled`	`HunyuanVideo-1.5-480p_i2v`
Model type	`ltx2_i2v`	`hy15_i2v`
Fine-tuning method	LoRA	LoRA
LoRA rank	64	64
LoRA alpha	128	128
Trainer type	`AWM-demo`	`GRPO-demo`
Loss type	`exp_first`	–
Advantage weighting	`ghuber`	–
Advantage aggregation	`gdpo`	–
Learning rate	2×10^{-4}	1×10^{-4}
Optimizer	Adam	Adam
Adam betas	$(0.9,\ 0.999)$	$(0.9,\ 0.999)$
Adam epsilon	10^{-8}	10^{-8}
Weight decay	10^{-4}	10^{-4}
Max grad norm	1.0	1.0
EMA decay	0.96	0.9
EMA decay schedule	constant	power
EMA update interval	1	4
Number of inner epochs	1	1
Unique samples per epoch	32	8
Group size	24	16
Training resolution	384×576	480×848
Evaluation resolution	512×768	480×848
Number of frames	121	121
Inference steps (train)	8	10
Inference steps (eval)	8	40
Training timesteps	4	10
Timestep range	$[0,\ 1]$	$[0,\ 0.9]$
Clip range	$[-1, 1]$	$[-3\times10^{-4},\ 3\times10^{-4}]$
Advantage clip range	$[-5, 5]$	$[-5, 5]$
β_d	0.008	1.0
	Reward	
Reward model	`Qwen3.5-27B-FP8`	`Qwen3-VL-8B-Instruct`
Reward type	multi (task/consistency/pick)	task+consistency
Reward weight	0.5 / 0.225 / 0.225	0.95
Distillation reward weight	0.05	0.05

Table 3: Main training hyperparameters used for fine-tuning LTX-2 and HunyuanVideo 1.5. For details on implementation-specific parameters, we refer to the official codebase.

The fake-score denoiser is trained online with the denoising objective

$$\mathcal{L}^{\phi}_{\text{denoise}} = \left\| \mu^{\phi}_{\text{fake}}(x_t, t) - x \right\|_2^2, \tag{17}$$

while the generator is updated using the approximate distribution-matching gradient

$$\nabla_\theta D_{\text{KL}} \simeq \mathbb{E}_{z,t,x,x_t} \left[w_t \alpha_t \left(s_{\text{fake}}(x_t, t) - s_{\text{real}}(x_t, t) \right) \frac{\partial G_\theta(z)}{\partial \theta} \right]. \tag{18}$$

We adapt this objective to our demonstrator–executor self-distillation setting by minimizing the KL divergence between the executor distribution $p_\theta(x_t, t \mid c_{\text{E}})$ and the demonstrator distribution $p_{\theta'}(x_t, t \mid c_{\text{D}})$:

$$D_{\text{KL}} \left(p_\theta(x_t, t \mid c_{\text{E}}) \,\|\, p_{\theta'}(x_t, t \mid c_{\text{D}}) \right). \tag{19}$$

Model	Mean ↑	Abstr. ↑	Categ. ↑	Navig. ↑	Perc. ↑	Physics ↑	Transform. ↑
LTX-2	0.5993	0.6195	**0.6238**	0.5966	0.6118	0.5742	0.5732
LTX-2+*WMSD*	<u>0.6137</u>	<u>0.6407</u>	0.6207	**0.6039**	<u>0.6307</u>	**0.5807**	<u>0.6193</u>
LTX-2+*WMSD* *	**0.6223**	**0.6587**	<u>0.6218</u>	**0.6039**	**0.6621**	<u>0.5796</u>	**0.6212**

Table 4: VBVR evaluation results across models and categories (250 samples each). * indicates the prompt-rewrite variant.

Taking the gradient with respect to the executor parameters θ yields the approximation

$$\nabla_\theta D_{\mathrm{KL}}(p_\theta(x_t, t \mid c_{\mathrm{E}}) \,\|\, p_{\theta'}(x_t, t \mid c_{\mathrm{D}})) \simeq \mathbb{E}_{z,t,x,x_t}\left[w_t \alpha_t \left(s_\theta(x_t, t \mid c_{\mathrm{E}}) - s_{\theta'}(x_t, t \mid c_{\mathrm{D}}) \right) \times \frac{\partial G_\theta(z, c_{\mathrm{E}})}{\partial \theta} \right], \tag{20}$$

$$x = G_\theta(z, c_{\mathrm{E}}), \qquad x_t \sim q_t(x_t \mid x). \tag{21}$$

Although this objective is conceptually appealing, we found it unstable in practice. Across our experiments, the DMD-based self-distillation objective consistently diverged, and we therefore did not use it in the final method.

A.2.2 Reward Prompts

We present the reward prompts used during training, shown in Fig. 6 and Fig. 7. These prompts provide binary supervision for task success and visual-temporal consistency, enabling stable reward computation from generated videos.

A.3 Further Experiments

We compared a per-step distillation reward with the trajectory-level distillation reward in Eq. 8 and found only minor differences in final performance. We therefore use the trajectory-level form in the main experiments for simplicity. We also investigated sharing weights between the Executor and Demonstrator. Across hyperparameter settings and EMA schedules, using the Executor weights as the Demonstrator led to unstable training, so all main results use a fixed Demonstrator.

A.4 VBVR Evaluation

The VBVR tasks are substantially out of distribution for our setting because they are longer and more abstract than the short task instructions in *WorldTasks*. We therefore evaluate the vanilla model, the *WMSD*-trained model, and a prompt-rewrite variant that converts the long benchmark query into a shorter task instruction before generation.

A.5 Further Details on *WorldTasks*

A.5.1 Dataset Filtering

We construct the dataset from pre-extracted images. We first remove incomplete or already processed entries, and then apply an image-quality filter. This filter rejects frames that are excessively blurry, underexposed, overexposed, or nearly empty. Concretely, we compute the variance of the Laplacian as a blur indicator, the mean luminance to detect overly dark or bright images, and the fraction of near-black and near-white pixels to remove degenerate frames. In our main setup, we use thresholds of `min_laplacian_var` $= 12.0$, `min_mean_luma` $= 20.0$, `max_mean_luma` $= 235.0$, `max_black_ratio` $= 0.85$, and `max_white_ratio` $= 0.85$.

To further improve visual quality, we rank the surviving frames with an aesthetic score based on a CLIP-based scoring function and keep only the top 90% of samples according to the combined

Task Success Evaluation Prompt

Instruction: *{instruction}*
Target agent: *{agent_name}*

You are given generated video frames in correct temporal order (frames 0..N-1).
You are judging whether a **target agent** in this temporally ordered video successfully completes an instruction.
If the prompt refers to first-person view, camera perspective, or first-person perspective, interpret that as referring to the camera movement and viewpoint.
Analyze the video carefully and reason strictly from visible evidence. Do not assume intent. Do not infer unseen events. Do not guess.

Evaluation Criteria

1. **Correct Agent Attribution**
   - The required action must be performed by the target agent.
   - If another agent performs the action, the task is **NOT** successful.
2. **Action Progress and Completion**
   - The target agent must clearly complete the instructed action.
   - The instruction must be fully satisfied by the final frame (or earlier with a stable and persistent completed state).
   - For first-person camera view: no other agent should be performing the action.
3. **Realism and Physical Consistency**
   - The outcome must be grounded in objects present in earlier frames.
   - If video quality is too poor to determine completion reliably, the task is **NOT** successful.

Decision Rule
Answer **Yes** only if all of the following are true:

- The target agent performs the required action
- The action is fully completed by the final frame
- The action is physically realistic and consistent

If any condition fails, answer **No**.

Output Format
Return exactly one word: **Yes** or **No**. Do not include any explanation or additional text.

Figure 6: Prompt used for task reward during training.

quality score. We also apply a vision-language quality screening step that discards frames deemed unsuitable for agent-based video generation.

The VLM is prompted to assess whether an image is appropriate for agent-based video generation based on the prompt given in Box 8.

Samples for which the VLM responds negatively are discarded. This additional semantic filtering step complements the low-level image quality criteria by removing visually valid but uninformative or non-actionable scenes, resulting in a dataset that is both visually and semantically suitable for downstream task-conditioned video generation.

A.5.2 Example Task and Solution Prompts from *WorldTasks*

To qualitatively illustrate the structure of *WorldTasks*, we present four representative samples below. Each example contains the first frame together with the first two task prompts and their corresponding descriptive solution prompts. Examples are shown in Fig. 9 and Fig. 10.

Visual Quality and Temporal Consistency Prompt

You are given generated video frames in correct temporal order (frames 0..N-1).
You are judging whether this temporally ordered video is successful in terms of visual quality and temporal consistency.
Analyze the video carefully and reason strictly from visible evidence. Do not infer hidden causes. Do not guess missing frames. Do not speculate beyond what is visible.

Evaluation Criteria

1. **Visual Quality**
    - Frames should be clear, coherent, and stable.
    - Severe blur, flicker, distortions, broken rendering, or major artifacts mean the video is **NOT** successful.
2. **Temporal Consistency**
    - Motion and state changes should be smooth and physically coherent over time.
    - No teleportation, popping, identity instability, discontinuous motion, or implausible changes.
3. **Reliability of Evidence**
    - Judge only from visible evidence in the frames.
    - If frames are too unclear to assess reliably, the video is **NOT** successful.
4. **Consistency with Initial Frame**
    - The video must remain consistent in style and quality with the first frame.

Decision Rule
Answer **Yes** only if all of the following are true:

- Visual quality is acceptable overall without severe artifacts
- Temporal consistency is acceptable without severe continuity failures
- Frames are clear enough for reliable judgment

If any condition fails, answer **No**.

Output Format
Return exactly one word: **Yes** or **No**. Do not include any explanation or additional text.

Figure 7: Prompt used for the consistency reward during training.

A.6 Evaluation Prompts for *WorldTasksBench*

In this section, we present the evaluation prompts used in *WorldTasksBench* to assess generated videos along three complementary dimensions. The first prompt evaluates whether the instructed task is successfully completed, focusing strictly on end-state correctness. The second prompt verifies correct agent attribution, ensuring that the intended actor performs the specified action. The third prompt measures physical realism and temporal consistency, capturing whether the video exhibits plausible motion and coherent dynamics. Together, these prompts provide a structured and binary evaluation framework that isolates task success, agent correctness, and physical validity.

A.7 Alternating Training Algorithm

We present the alternating training procedure (Alg. 1).

VLM-Based Dataset Quality Filtering Prompt

Look at this image carefully. Is this image suitable as a training sample for an agent-based video generation model?

Consider whether:

- The scene contains meaningful, diverse visual content (not blank, corrupted, or trivially simple)
- The image quality is acceptable (not severely blurry, overexposed, underexposed, or distorted)
- The image depicts a scene where an agent could plausibly perform tasks

Output Format
Answer with ONLY **Yes** or **No**. Do not include any explanation or additional text.

Figure 8: Prompt used for VLM-based semantic filtering of dataset images.

Algorithm 1 GRPO/AWM with Demonstrator Anchoring

Require: pretrained base model f_0 with velocity field v_0
Require: Executor sampler $p_\theta(\cdot \mid c_{\mathrm{E}})$ and velocity field v_θ
Require: Demonstrator sampler $p_\phi(\cdot \mid c_{\mathrm{D}})$ and velocity field v_ϕ; fixed for the main *WMSD* setting
Require: VLM-generated condition pairs $\mathcal{S} = \{(c_{\mathrm{E}}, c_{\mathrm{D}})\}$
Require: reward weights $\lambda_{\mathrm{task}}, \lambda_{\mathrm{distill}}$
Require: Executor anchor coefficient β_d and optional Demonstrator anchor coefficient β_ϕ
Require: optional alternation period N; set $N = 0$ to keep the Demonstrator fixed

```
1:  for iteration e = 1, 2, ... do
2:      sample (c_E, c_D) ~ S
3:      if N > 0 and e mod N = 0 then
4:          Optional Demonstrator round
5:          sample rollout group {τ_i}_{i=1}^G ~ p_φ(· | c_D)
6:          compute task rewards r_task(τ_i; I, T)
7:          compute group-relative RL loss L_RL(φ) from the task rewards
8:          compute base-model anchor loss L_base against f_0
9:          update φ by minimizing L_RL(φ) + β_φ L_base
10:     else
11:         Executor round
12:         sample rollout group {τ_i}_{i=1}^G ~ p_θ(· | c_E)
13:         compute task rewards r_task(τ_i; I, T)
14:         compute distillation rewards r_distill(τ_i) using Eq. 8
15:         set R_i = λ_task r_task(τ_i) + λ_distill r_distill(τ_i)
16:         compute group-relative RL loss L_RL(θ) from {R_i}_{i=1}^G
17:         compute Demonstrator anchor loss L_anchor using Eq. 10
18:         update θ by minimizing L_RL(θ) + β_d L_anchor
19:     end if
20: end for
21: Evaluation: use p_θ(· | c_E)
```

A.8 Theoretical Background

Group-relative policy optimization. We optimize the student model using a group-relative reinforcement learning objective inspired by GRPO Shao et al. [2024]. For each task instruction $\mathcal{T}$, we sample G trajectories

$$\tau_1, \ldots, \tau_G \sim p_\theta(\cdot \mid \mathcal{T}),$$

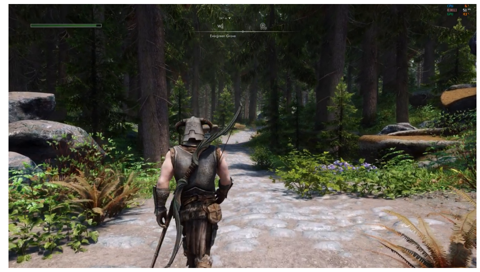
Sample ID: `100_4`

1. **Task 1:** [Man in blue shirt]: Step onto the yellow lane marking and stop exactly at the white arrow's tip.

   Description 1: The man in the blue shirt begins walking forward along the center of the road, his feet deliberately stepping onto the double yellow lane marking, and continues moving straight ahead until he reaches the tip of the white directional arrow painted on the asphalt, where he halts and stands still.

2. **Task 2:** [Person in blue shirt]: Move forward to the nearest building.

   Description 2: The person in the blue shirt begins walking forward along the center of the road, maintaining a steady pace toward the building on the left side of the street, their body oriented directly ahead as they cross the yellow double lines; after a few steps, they continue moving forward until they reach the sidewalk adjacent to the building, then they halt beside the American flag mounted on the building's facade, coming to a complete stop with their feet planted on the pavement.



Sample ID: `145_4`

1. **Task 1:** [Character with horned helmet]: Use the bow to aim at the tree trunk directly ahead.

   Description 1: The character with the horned helmet slowly turns their upper body toward the tree trunk directly ahead, simultaneously drawing the bowstring back with their right hand while keeping their left hand steady on the bow's grip, their gaze fixed on the target as the bowstring tenses and the arrow nocks align with the trunk.

2. **Task 2:** [Character with horned helmet]: Move to the largest boulder and stop beside its left edge.

   Description 2: The character with the horned helmet begins walking forward along the stone path, their body oriented toward the largest boulder visible to the left, and after a few steps, they decelerate, shifting their weight slightly as they turn their head to the left to align their gaze with the boulder's edge, then halt precisely beside its left side, their right hand resting on their hip while their left hand remains near the hilt of their weapon.

Figure 9: Two representative samples from *WorldTasks*. Each sample includes an initial frame, task prompts, and corresponding descriptive solutions.

and compute their rewards $r(\tau_i)$. These rewards are normalized within the group to produce relative advantages

$$A_i = \frac{r(\tau_i) - \mu_r}{\sigma_r + \varepsilon}, \quad \mu_r = \frac{1}{G}\sum_{j=1}^{G} r(\tau_j), \quad \sigma_r^2 = \frac{1}{G}\sum_{j=1}^{G}(r(\tau_j) - \mu_r)^2.$$

The resulting objective is

$$\mathcal{L}_{\text{GRPO}} = -\mathbb{E}_{\mathcal{T}}\left[\frac{1}{G}\sum_{i=1}^{G} A_i \log p_\theta(\tau_i \mid \mathcal{T})\right].$$

This formulation reinforces trajectories that outperform their peers on the same task while suppressing weaker ones. Unlike standard distillation, it enables improvements beyond the teacher whenever the reward function favors better solutions.

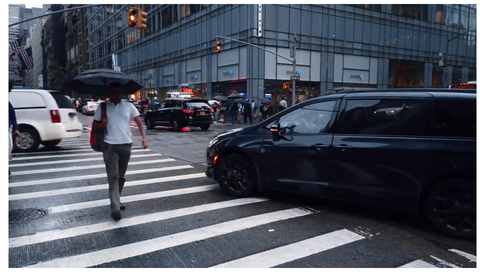
Sample ID: `6888_1`

1. **Task 1:** [Driver in racing suit]: Press the red button on the steering wheel's right side.

   Description 1: The driver's right hand, clad in a black racing glove, moves slightly forward and inward, pressing the red button located on the right side of the steering wheel, while the left hand remains steady on the left side of the wheel, and the vehicle continues forward along the track with the dashboard displaying 186 MPH and an overtaking indicator active.

2. **Task 2:** [First-person view]: Align the car's front bumper with the white track curb ahead.

   Description 2: The driver's hands grip the steering wheel firmly, thumbs pressing the paddle shifters while the left hand subtly adjusts its position to maintain control; simultaneously, the right hand makes a slight inward rotation of the wheel to initiate a gentle steering correction toward the white track curb ahead, and the car's front bumper begins to approach the curb as the vehicle decelerates slightly, aligning its front edge with the curb's edge while the dashboard display updates to reflect the new position and speed.



Sample ID: `7637_6`

1. **Task 1:** [Man holding black umbrella]: Step off the crosswalk and hand the umbrella to the sidewalk curb.

   Description 1: The man holding the black umbrella continues walking forward, stepping off the crosswalk onto the sidewalk, then lowers his arm and extends his hand toward the curb, releasing the umbrella to rest against the sidewalk edge.

2. **Task 2:** [Black minivan]: Align its front bumper with the white pedestrian lane marking.

   Description 2: The black minivan advances forward while maintaining its current trajectory, its front bumper gradually moving closer to the white pedestrian lane marking on the asphalt, adjusting its position as it proceeds along the crosswalk.

Figure 10: Two representative samples from *WorldTasks*. Each sample includes an initial frame, task prompts, and corresponding descriptive solutions.

FlowGRPO. Flow matching models learn a continuous transport from data $x_0 \sim p_{\text{data}}$ to noise $\epsilon \sim \mathcal{N}(0, I)$ via

$$x_t = (1-t)x_0 + t\epsilon,$$

and are trained with the objective

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{x_0,\epsilon,t}\left[\|v_\theta(x_t,t,c) - (\epsilon - x_0)\|^2\right].$$

This enables deterministic sampling through the ODE

$$dx_t = v_\theta(x_t,t)\,dt.$$

Task Solved Evaluation Prompt

Instruction: *{eval_prompt}*

You are a strict video benchmark judge.
Decide whether the video successfully solves the instructed task by the end.

Decision Rule

- Return **1** only if the requested action is clearly completed.
- Return **0** if the task is incomplete, incorrect, ambiguous, only partially done, or a different action occurs.

Output Format
Reply with only **0** or **1**.

Figure 11: Prompt used to evaluate whether a generated video successfully completes the instructed task.

Correct Agent Attribution Prompt

Full instruction: *{eval_prompt}*
Expected acting agent: *{agent_name}*
Expected task for that agent: *{agent_task_text}*

You are a strict video benchmark judge.
Decide whether the correct agent is the one performing the instructed action.

Decision Rule

- Return **1** only if the expected agent clearly performs the task.
- Return **0** if another agent performs it, the acting agent is unclear, or the expected agent is absent or incorrect.

Output Format
Reply with only **0** or **1**.

Figure 12: Prompt used to verify that the correct agent performs the instructed action.

Flow-GRPO Liu et al. [2025a] extends this framework by casting denoising as a multi-step MDP. The state, action, and policy are defined as

$$s_t = (c, t, x_t), \quad a_t = x_{t-1}, \quad \pi_\theta(a_t \mid s_t) = p_\theta(x_{t-1} \mid x_t, c).$$

To introduce exploration, the deterministic flow is converted into an SDE:

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

The model is then optimized using a clipped GRPO-style objective:

$$J_{\text{Flow-GRPO}}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{T}\sum_{t=0}^{T-1}\left(\min\Big(r_t^i(\theta)\hat{A}_t^i, \text{clip}(r_t^i(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t^i\Big) - \beta D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})\right)\right],$$

where

$$r_t^i(\theta) = \frac{p_\theta(x_{t-1}^i \mid x_t^i, c)}{p_{\theta_{\text{old}}}(x_{t-1}^i \mid x_t^i, c)},$$

Physical Realism and Temporal Consistency Prompt

You are a strict video benchmark judge.
Decide whether the video shows physically realistic execution.

Decision Rule

- Return **1** only if motion, contact, timing, object interactions, and scene dynamics are physically plausible and temporally coherent.
- Return **0** if there is teleportation, impossible motion, severe temporal inconsistency, broken dynamics, identity drift, or obvious non-physical behavior.

Output Format
Reply with only **0** or **1**.

Figure 13: Prompt used to evaluate physical realism and temporal consistency of generated videos.

and

$$\hat{A}_t^i = \frac{R(x_0^i, c) - \mathrm{mean}(\{R(x_0^j, c)\}_{j=1}^G)}{\mathrm{std}(\{R(x_0^j, c)\}_{j=1}^G)}.$$

Advantage Weighted Matching. Advantage Weighted Matching (AWM) Xue et al. [2025a] addresses a mismatch between diffusion-style reinforcement learning objectives and the original flow-matching training objective. Methods such as DDPO effectively optimize noisy reverse-step likelihoods, which increases variance and slows convergence.

AWM instead preserves the original flow-matching objective and incorporates rewards through advantage weighting. The prompt c defines the state, and the final sample x_0 is treated as the action with policy

$$\pi_\theta(x_0 \mid c).$$

The sequence likelihood is approximated by the negative flow-matching loss:

$$\log \hat{\pi}_\theta(x_0 \mid c) \approx -\mathbb{E}_t \left[w(t) \| v_\theta(x_t, t, c) - (\epsilon - x_0) \|^2 \right].$$

This yields the likelihood ratio

$$\frac{\hat{\pi}_\theta(x_0 \mid c)}{\hat{\pi}_{\theta_{\mathrm{old}}}(x_0 \mid c)} = \exp\!\left(-\mathbb{E}_t \left[w(t) \| v_\theta(x_t, t, c) - (\epsilon - x_0) \|^2 - w(t) \| v_{\theta_{\mathrm{old}}}(x_t, t, c) - (\epsilon - x_0) \|^2 \right] \right).$$

The corresponding policy-gradient update is

$$\nabla_\theta \log \hat{\pi}_\theta(x_0 \mid c)\, A = -\nabla_\theta \mathbb{E}_t \left[w(t) \| v_\theta(x_t, t, c) - (\epsilon - x_0) \|^2 \right] A.$$

Positive advantages reduce the flow-matching loss for high-reward samples, while negative advantages suppress low-reward ones. AWM further includes a velocity-space KL regularizer

$$D_{\mathrm{KL}} \approx w(t) \| v_\theta(x_t, t, c) - v_{\mathrm{ref}}(x_t, t, c) \|^2,$$

which stabilizes updates by constraining deviations from a reference model.

In contrast to Flow-GRPO, which introduces stochastic trajectory optimization, AWM directly aligns reinforcement learning with the original flow-matching objective, resulting in lower variance and improved training efficiency.

A.9 Method Derivations and Proofs

In this section we provide the derivations and proofs from Sec. 3.

Gradient Decomposition We begin with deriving the gradient decomposition in Eq. 6.

$$
\begin{aligned}
\mathcal{L}_{\text{on}} &= \int p_\theta(\tau \mid c_{\text{E}})\, C_\theta(\tau)\, d\tau, \\
\nabla_\theta \mathcal{L}_{\text{on}} &= \int \nabla_\theta \big[p_\theta(\tau \mid c_{\text{E}})\, C_\theta(\tau)\big] d\tau \\
&= \int \Big[C_\theta(\tau) \nabla_\theta p_\theta(\tau \mid c_{\text{E}}) + p_\theta(\tau \mid c_{\text{E}}) \nabla_\theta C_\theta(\tau)\Big] d\tau \quad \text{(product rule)} \\
&= \int p_\theta(\tau \mid c_{\text{E}}) C_\theta(\tau) \nabla_\theta \log p_\theta(\tau \mid c_{\text{E}})\, d\tau \quad \text{(score function trick)} \\
&\quad + \int p_\theta(\tau \mid c_{\text{E}}) \nabla_\theta C_\theta(\tau)\, d\tau \\
&= \mathbb{E}_{\tau \sim p_\theta}\big[C_\theta(\tau) \nabla_\theta \log p_\theta(\tau \mid c_{\text{E}})\big] + \mathbb{E}_{\tau \sim p_\theta}\big[\nabla_\theta C_\theta(\tau)\big] \quad \text{(rewrite as expectations).}
\end{aligned}
$$

Proposition We continue with the proof of Proposition 1:

Proof. Let x_t^θ and $x_t^{\theta'}$ denote the student and teacher flows initialized from the same $x_0 \sim p_0$, so that $x_0^\theta = x_0^{\theta'}$. Define $\Delta_t := x_t^\theta - x_t^{\theta'}$. Then

$$\frac{d}{dt}\Delta_t = v_\theta(x_t^\theta, t \mid c_{\text{E}}) - v_{\theta'}(x_t^{\theta'}, t \mid c_{\text{D}}).$$

Adding and subtracting $v_{\theta'}(x_t^\theta, t \mid c_{\text{D}})$ and using the L-Lipschitzness of $v_{\theta'}(\cdot, t \mid c_{\text{D}})$ gives

$$\frac{d}{dt}\|\Delta_t\| \le \big\|v_\theta(x_t^\theta, t \mid c_{\text{E}}) - v_{\theta'}(x_t^\theta, t \mid c_{\text{D}})\big\| + L\|\Delta_t\|.$$

Since $\Delta_0 = 0$, Grönwall's inequality implies

$$\|\Delta_1\| \le e^L \int_0^1 \big\|v_\theta(x_t^\theta, t \mid c_{\text{E}}) - v_{\theta'}(x_t^\theta, t \mid c_{\text{D}})\big\|\, dt.$$

Therefore, by Cauchy–Schwarz,

$$\mathbb{E}_{x_0 \sim p_0}\|\Delta_1\|^2 \le e^{2L}\mathbb{E}_{x_0 \sim p_0}\left[\int_0^1 \big\|v_\theta(x_t^\theta, t \mid c_{\text{E}}) - v_{\theta'}(x_t^\theta, t \mid c_{\text{D}})\big\|^2\, dt\right] \le e^{2L}\varepsilon^2.$$

The shared initialization defines a valid coupling between the terminal laws $p_\theta(x_1 \mid c_{\text{E}})$ and $p_{\theta'}(x_1 \mid c_{\text{D}})$. Hence

$$W_2(p_\theta(x_1 \mid c_{\text{E}}), p_{\theta'}(x_1 \mid c_{\text{D}})) \le \big(\mathbb{E}_{x_0 \sim p_0}\|\Delta_1\|^2\big)^{1/2} \le e^L\varepsilon.$$

□

A.10 Visual Examples

In Fig 14 and 15 we showcase random examples of *World-Model Self-Distillation*.

Impact Statement

This paper advances methods for training visual world models that generate task-directed videos from scene images and instructions. The main positive impact of this work is that it may reduce the need for costly task-execution video collection, especially in embodied AI and robotics, where scalable simulation and model-based planning could make research more data-efficient and accessible. The same capability also carries risks. More capable video world models can generate plausible but incorrect futures, and they should not be used as standalone decision-makers in safety-critical settings without external validation, uncertainty estimation, and human or system-level oversight. Because our rewards and task annotations are produced by vision-language models, the trained models may inherit biases, blind spots, or inconsistent judgments from those systems and from the underlying image sources. Improved task-directed video generation could also be misused to create misleading synthetic media or to prototype unsafe actions in simulated environments. We do not deploy the model for real-world control in this work. Released data, code, and model artifacts will include documentation of intended use, limitations, source provenance where available, and usage restrictions discouraging deceptive, unsafe, or non-consensual synthetic video generation.

(a) *[First-person view]: Move to the right to examine the distant house.*

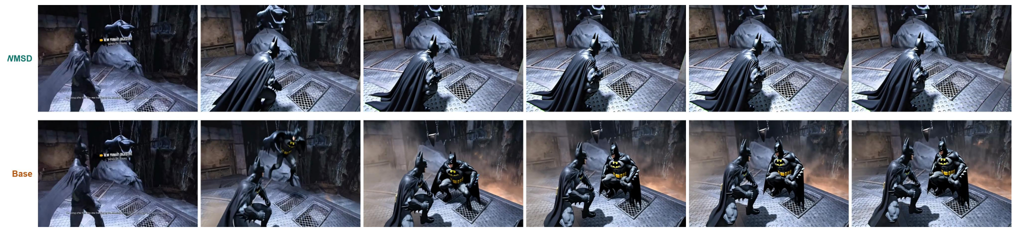
(b) *[First-person view]: Pick up the ammo box on the wooden platform.*

(c) *[First-person view]: Walk toward the building entrance on the right.*

(d) *[First-person view]: Turn right to face the welcome sign on the easel.*

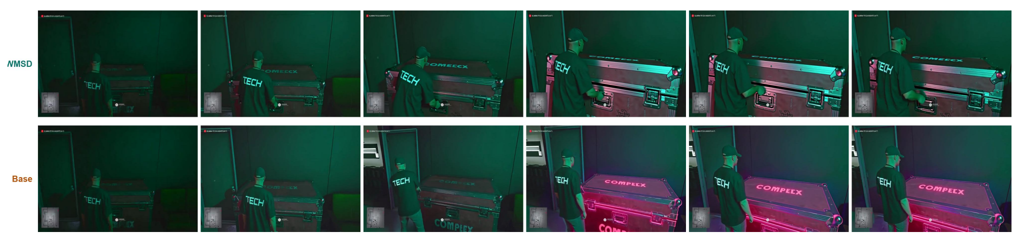
(e) *[First-person view]: Move forward through the tall grass.*

Figure 14: Qualitative comparisons between *WMSD* and the base model. Each subfigure shows six uniformly sampled frames from the generated videos using LTX2.



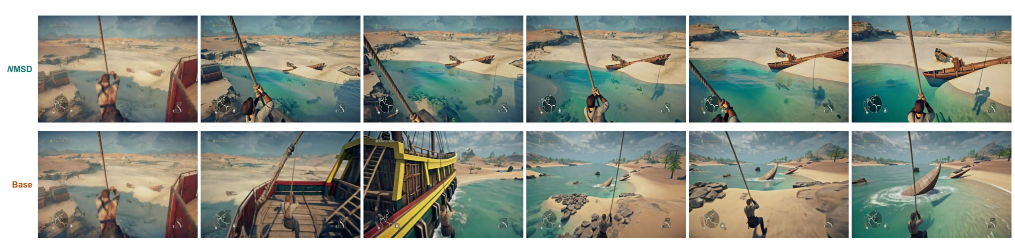
(a) *[Batman]: Crouch near the metal grates on the floor.*

(b) *[Man in center]: Lower one hand slowly.*



(c) *[First-person view]: Interact with the large case labeled COMPLEX.*

(d) *[First-person view]: Turn right and head toward the iron railing.*



(e) *[First-person view]: Use the rope to swing toward the nearest visible shipwreck's hull.*

Figure 15: Additional qualitative HunyuanVideo-1.5 comparisons between *WMSD* and the base model. Each subfigure shows six uniformly sampled frames from the generated videos.

NeurIPS Paper Checklist

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 abstract and introduction state that WMSD trains instruction-conditioned visual task-solving world models via asymmetric self-distillation and VLM-based reinforcement learning, without paired task-execution videos. These claims are supported by the method, WorldTasksBench experiments, baseline comparisons, and DreamGen robotic-task evaluation in Sections 3–5.

   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: [Yes]

   Justification: Section 5.7 discusses limitations including the lack of robot-specific dynamics in the data-free setting, reduced gains on out-of-distribution tasks when the demonstrator fails, and instability introduced by alternating RL training.

   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: [Yes]

Justification: The paper states the assumptions for the informal on-policy control result, including Lipschitz/stability assumptions on the teacher flow and shared initial noise, and provides the corresponding proof or proof background in the appendix. The theoretical claims are used to motivate the method rather than to serve as the sole empirical basis for the paper.

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 paper describes the datasets, models, reward construction, evaluation metrics, baselines, and training setup in Sections 5 and A.1. The authors will release the training code, dataset, and model weights, which provides a direct path for reproducing the main experimental results.

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: [Yes]

Justification: The authors will release the training code, WorldTasks dataset, and model weights, together with instructions for reproducing the main experiments. The supplementary material describes the main hyperparameters, reward prompts, dataset filtering, and evaluation procedures needed to support reproduction.

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: Section 5.1 specifies the experimental setup, including batch size, group size, base models, reward model, reward construction, datasets, and metrics. Appendix A.1 provides the main hyperparameters for LTX-2 and HunyuanVideo-1.5 fine-tuning, including optimizer settings, LoRA configuration, resolution, number of frames, inference steps, reward weights, and KL/self-distillation weights.

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: [N/A]

Justification: The paper reports quantitative comparisons across multiple metrics and benchmarks, but the current version does not report error bars, confidence intervals, or statistical significance tests. Since repeating experiments on a scale where statistics become relevant is not feasible under normal budgets with the model sizes we use, the results should therefore be interpreted as empirical comparisons under the reported evaluation protocol rather than as formal statistical significance claims.

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 current draft acknowledges training resources in the appendix. Since we use publicly available pretrained models, information such as memory requirements and inference time is available on their respective project pages.

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 work uses existing models, automatically generated task-solution annotations, and image/video data for machine learning research, and does not involve deceptive user studies, human-subject interventions, or unsafe deployment. The authors have reviewed the NeurIPS Code of Ethics and are not aware of any deviations.

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: [Yes]

Justification: The paper includes an impact statement and discusses that the work advances scalable and data-efficient world model training, with potential benefits for embodied AI and robotics where data collection is expensive. It also discusses possible negative impacts of improved video world models, including misuse for misleading synthetic video generation, unsafe planning if deployed without validation, and the need for careful evaluation before real-world use.

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: [Yes]

Justification: The released assets are task-solving fine-tuning artifacts, dataset annotations, and model weights built on existing video-generation backbones rather than a new unrestricted foundation model. The paper describes filtering procedures for the WorldTasks data and reward-based consistency checks; the released package should include usage restrictions and documentation discouraging deceptive, unsafe, or non-consensual synthetic video generation.

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: [Yes]

    Justification: The paper cites the existing datasets, benchmarks, models, and methods used, including MiraData, LTX-2, HunyuanVideo, DreamGen/Gr00t, Cosmos, Wan, CogVideoX, Qwen, and related RL/distillation methods. The released version should include the licenses and terms of use for all reused datasets, model checkpoints, and codebases, and the experiments should comply with those terms.

    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: [Yes]

    Justification: The paper introduces WorldTasks, WorldTasksBench, task-solution prompt data, and trained model weights, and describes the dataset construction, filtering pipeline, task-solution generation, reward prompts, and evaluation prompts in Section 5 and Appendix A. The released assets will be accompanied by documentation covering data format, intended use, limitations, and reproduction instructions.

    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 experiments or research with human subjects. Dataset construction and evaluation rely on existing assets and automated VLM-based annotation and judging rather than newly recruited human participants.

    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 or human-subject research, so IRB approval or equivalent review for study participants is not applicable.

    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: [Yes]

    Justification: The paper explicitly describes the use of vision-language models as part of the core methodology: they generate task-solution pairs from images and provide task-completion and consistency rewards during training and evaluation. These uses are described in Sections 3, 5.1, and the appendix reward/evaluation prompts.

    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.