Title

RoboTALES: Learning Reasoning-Guided Robot Policies via Task-Aligned Simulated Futures

Authors

Gani, Hanan

Kulkarni, Tejal

Chodavarapu, Madhoolika

et al.

Publication Date

2026

DOI

10.1007/978-3-032-37447-9_26

RoboTALES: Learning Reasoning-Guided Robot Policies via Task-Aligned Simulated Futures

Hanan Gani, Tejal Kulkarni, Madhoolika Chodavarapu, Nicklas Hansen, and Manmohan Chandraker

University of California, San Diego, CA 92093, USA
Correspondence to: hgani@ucsd.edu

Abstract. Pretrained video generative models are promising backbones for visuomotor control, but their imagined futures often drift from task intent and are not reliably action-conditional. As a result, these models can be difficult to use for planning or policy extraction. To address these limitations, we propose RoboTALES, a single-stage framework that learns task-aligned simulated futures and uses them to train robot policies. Our approach introduces two key innovations: (1) a hierarchical LLM-based planner that breaks complex tasks into a sequence of subgoals to guide the model's imagination; and (2) a VLM-based critic that evaluates these "imagined" futures and uses reward-based feedback to keep the model's internal representations focused on the goal. By anchoring the video generator in abstract reasoning, we produce temporally consistent rollouts and more coherent actions. We evaluate RoboTALES on diverse manipulation tasks from RoboCasa and LIBERO10, and show that our method consistently outperforms existing methods, especially in long-horizon tasks. Our code and models are publicly available at https://github.com/hananshafi/RoboTALES.

Keywords: Video Generation· Diffusion Models · Hierarchical Reasoning · Policy Optimization

1 Introduction

Humans rarely act purely reactively; instead, they decompose goals into ordered subtasks, mentally simulate possible futures, and commit to an action only after rejecting undesirable outcomes [9, 38]. This capacity for hierarchical, goal-directed imagination remains an ongoing challenge for autonomous agents. Agents operating on high-dimensional visual observations must reason over extended horizons, anticipate the consequences of their actions, and select among many plausible futures. While model-free reinforcement learning can achieve strong task performance [20, 39, 48], it offers no explicit reasoning substrate and remains sample-inefficient. Model-based approaches address this by learning a world model that enables an agent to "imagine before acting" [19, 21, 22, 24, 47, 51].

The idea of leveraging diffusion-based [8, 27, 50] video generation models [11, 13] directly for robot control has recently gained considerable momentum.

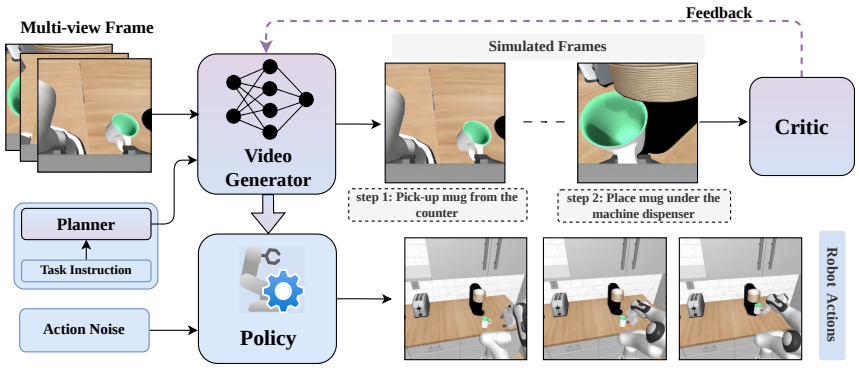


Fig. 1: Overview. An LLM Planner decomposes the task instruction (*e.g.*, "make a coffee") into ordered subtasks that condition the video generator to generate future frames for each milestone. A frozen VLM Critic scores these imagined rollouts against the goal instruction, steering the video generator model toward semantically aligned futures. The resulting goal-conditioned features drive the policy to produce precise robot actions for long-horizon manipulation.

Works such as Video-Policy [35], Gen2Act [6], and ViPRA [46] demonstrate that video generators can serve as powerful priors for policy learning, using generated rollouts to either supervise or directly condition action generation. However, video generation alone does not yet constitute a world model for control. This is because the underlying video generators are still optimized primarily for visual realism or reconstruction objectives, hence their simulated futures can remain weakly grounded in task intent; moreover, making generated videos faithfully follow robot actions is itself nontrivial [34].

In parallel, LLMs have demonstrated a remarkable ability to decompose long-horizon tasks into grounded subtasks and guide robot behavior through natural language [3, 10, 29, 49]. Yet, most prior works treat language and prediction as loosely coupled modules: language influences *which* action to execute, but does not directly shape the predictive representations inside the world model. This decoupling prevents closed-loop alignment between what the agent imagines and what it ultimately does. Recent generative video world models further illustrate the promise of learned rollouts for control [12, 28], but their integration with language-grounded reasoning and policy learning remains largely unexplored. Consequently, there is no closed-loop mechanism ensuring that imagined futures, high-level reasoning, and low-level actions remain semantically aligned.

These motivate a mechanism that can explicitly *(i)* impose hierarchical structure on imagination and *(ii)* enforce semantic faithfulness of simulated futures—both of which are crucial if imagined rollouts are to reliably benefit robot action generation. To address this gap, we introduce **RoboTALES**: Learning Reasoning-Guided **Robo**t Policies via **T**ask-**AL**igned Simulat**E**d Future**S**, a unified framework that formalizes the alignment between high-level task logic and low-level visual dynamics. As illustrated in Figure 1, RoboTALES operates

through two tightly coupled components. An **LLM Planner** that decomposes the natural task instruction into an ordered sequence of temporal subtasks. This hierarchical task decomposition, as studied in both cognitive science [9, 38] and machine learning [4, 43], provides an explicit temporal scaffold, analogous to human subgoal reasoning [9]. Conditioning the video diffusion model on these subtasks transforms rollout generation from an undifferentiated prediction problem into a structured, milestone-driven simulation process. However, generating structured rollouts is necessary but not sufficient. Even with planner conditioning, a generative model may produce futures that satisfy low-level visual statistics while violating high-level intent. Therefore, to enforce semantic faithfulness, we introduce a VLM-based **Critic** that closes this gap by evaluating the generated rollouts against the target task instruction, producing a language-conditioned alignment score. Crucially, this feedback is incorporated in the video generator's hidden states during training via a differentiable policy optimization [7]. This representational steering refines the internal dynamics of the diffusion VideoUNet, encouraging it to encode task-relevant semantics rather than purely visual statistics. These semantically enriched features then serve as the foundation for generating purposeful, precise motor actions.

The joint integration of planner and critic transforms the diffusion VideoUNet into a goal-conditioned transition function where the planner constrains which futures are imagined, and the critic ensures those futures are worth acting upon. We show that our goal-aware features provide a much stronger foundation for robot control. By linking the robot's actions directly to these reasoned 'imaginations', the policy produces smoother and more purposeful movements. This effectively eliminates the erratic behaviors common in traditional diffusion policies [17], ensuring the robot stays on track even during long, complex tasks. Empirically, RoboTALES improves long-horizon manipulation on RoboCasa and LIBERO. In particular, on challenging Pick-and-Place tasks in RoboCasa, RoboTALES reaches a mean success rate of 48%, outperforming [35] and the strongest prior baseline [23]. On multi-step tasks involving turning and pressing, RoboTALES achieves mean success rates of 64% and 96%, respectively, surpassing competing methods. Overall, these results support our core thesis: *aligning imagination with hierarchical task logic yields more reliable rollouts and more coherent downstream actions.*

In summary, our contributions are:

- *Planner-Conditioned Video Generation.* We integrate hierarchical subtask plans from an LLM planner into the latent space of a diffusion-based video generator model. This induces structured, temporally organized rollouts that mirror abstract, goal-directed reasoning.
- *Critic-Guided Representational Steering.* We leverage VLM-based reward signal within an RL-based differentiable optimization framework enabling language-aligned feedback to directly refine the video generator's latent dynamics. This ensures semantic faithfulness to the task instruction.

– *Reasoning-Aligned Policy Learning.* We show that conditioning action generation on reasoning-aligned predictive features produces more precise and semantically consistent trajectories.

2 Related Work

Diffusion Policy Models. Due to over-conservatism, limited expressiveness, and compounding errors in offline policy learning and planning, a growing body of work adopts diffusion models as policy parameterizations. Diffuser [30] first proposed trajectory-level diffusion to jointly predict all timesteps of a plan, improving temporal consistency and reducing compounding errors. Subsequent methods [2, 16, 25, 26, 53] represent policies as diffusion models to capture multimodal action distributions and increase expressiveness. Diffusion Policy [17] brought this paradigm to robotics, with follow-ups extending to 3D settings [32, 55, 57], scaling [59], improving efficiency [31, 54], and introducing architectural refinements [44, 45, 52, 58]. Despite strong empirical performance, these approaches are primarily reactive and do not explicitly endow the policy with a reasoning substrate: they learn to map observations to actions, but do not train an internal predictive model whose rollouts are structured by plans or validated for semantic faithfulness. In contrast, our RoboTALES learns robot policies from a reasoning-guided diffusion predictive model, where planning and critique actively shape the representations used for control.

Video Generation for Robot Actions. A rapidly growing body of work leverages large-scale video generative models as world models for robot control [11, 41, 42, 56], motivated by the observation that video prediction implicitly captures physical dynamics, object interactions, and causal structure that can be repurposed for policy learning [12, 28]. VideoPolicy [35] builds on Stable Video Diffusion (SVD) [8] and conditions a 1D action UNet on hidden embeddings extracted from the video UNet's decoder layers, demonstrating strong generalization to unseen objects, backgrounds, and tasks. Gen2Act [6] similarly synthesizes video demonstrations of unseen tasks to guide policy learning, while ViPRA [46] uses video predictions as affordance signals for action generation. These works establish video generation as a powerful proxy for policy learning and form the direct foundation that our proposed RoboTALES builds upon.

Planning via Language. LLMs and VLMs have shown strong capacity for task decomposition and high-level reasoning, motivating their use as planning modules for embodied agents. [3] grounds LLM plans in robot affordances, [29] incorporates environment feedback into an LLM reasoning loop for closed-loop replanning, and [49] decomposes navigation instructions into visual waypoints using LLMs and VLMs. More recently, [10] embeds reasoning directly into action prediction by fine-tuning VLMs end-to-end on robot demonstrations, while [14] performs high-level planning entirely in language space by predicting interleaved actions and state descriptions. While these approaches demonstrate that language is a powerful medium for abstract task reasoning, they share a

common limitation: language reasoning remains decoupled from the agent's predictive representations. Plans influence which actions are selected but do not shape what the agent imagines about the future. In RoboTALES, the Planner conditions the generative process itself, embedding structured task reasoning into the latent visual dynamics rather than treating it as an external directive.

3 RoboTALES

3.1 Problem Setup and Formulation

We consider an embodied agent tasked with completing a long-horizon goal described by a natural-language instruction τ. At each time step t, the agent receives a visual observation $s_t \in \mathcal{S}$ and must produce a sequence of continuous actions $\mathbf{a}_{t:t+H} \in \mathcal{A}^H$ that maximizes task progress, where H is the action prediction horizon.

Our framework decomposes this problem into a hierarchical decision-making process. First, a high-level planner $\mathcal{F}_P$ maps the goal instruction τ to a structured plan consisting of semantically coherent sub-goals. Second, a generative world model $\mathcal{G}_\theta$ predicts a sequence of future latent states $\hat{s}_{t:t+\Delta}$ conditioned on the current state s_t and the plan. A reward function $\mathcal{F}_R(\hat{s}, \tau)$ evaluates these imagined futures to provide a semantic alignment signal. Finally, an action policy π_ϕ maps the internal representations of the model to executable control signals $\mathbf{a}_{t:t+H}$. The objective is to jointly optimize the world model and the policy such that the generated actions align with the high-level reasoning provided by the planner.

A natural question is whether to train these components separately or jointly. Decoupled (two-stage) training—where the video generator is first trained in isolation and the action policy is subsequently trained on frozen features—avoids gradient interference between heterogeneous objectives but introduces a fundamental limitation: the video model has no awareness of which latent features are most useful for control, and the action policy must adapt to a fixed representation that was never optimized for action generation. In contrast, RoboTALES adopts a coupled, single-stage training strategy in which the video generator and action policy are optimized simultaneously. This end-to-end design allows action-level gradients to flow back into the video generator's decoder layers, encouraging it to produce latent representations that are not only visually faithful and semantically aligned but also directly informative for downstream control. The result is a tighter co-adaptation between imagination and action: the video model learns to "imagine for acting," while the policy learns to "act from imagination."

3.2 Architecture and Components

RoboTALES consists of four components (Fig. 2): a reasoning-based LLM Planner ($\mathcal{F}_P$), a video generator ($\mathcal{G}_\theta$), an action generation model (π_ϕ), and an RL-based VLM critic ($\mathcal{F}_\mathcal{R}$). While these elements resemble hierarchical model-based

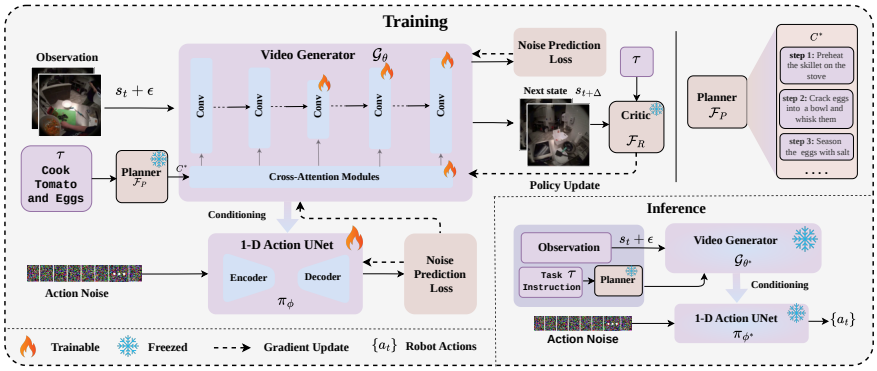


Fig. 2: RoboTALES Architecture and Training. The Planner $\mathcal{F}_P$ decomposes the task instruction τ into ordered subtasks, forming the augmented plan C^*. The video generator $\mathcal{G}_\theta$ is jointly trained with the Action UNet π_ϕ in a single stage. $\mathcal{G}_\theta$ receives a video diffusion loss and reward feedback from a frozen VLM Critic $\mathcal{F}_\mathcal{R}$, while π_ϕ is trained with an action diffusion loss conditioned on the decoder features of $\mathcal{G}_\theta$. Crucially, action-level gradients propagate back into the selected decoder layers of $\mathcal{G}_\theta$, enabling the world model to co-adapt its representations for both visual prediction and downstream control. *Inference (bottom-right):* Given a new observation and task instruction, the Planner produces C^*, the adapted world model $\mathcal{G}_{\theta^*}$ generates future states, and the trained policy π_{ϕ^*} outputs executable robot actions $\{a_t\}$ from the co-adapted latent representations.

RL architectures, our contribution lies in the interfaces between them, which enable task-aligned imagination. Specifically, the planner produces semantic subgoal tokens that condition the diffusion model, guiding the generation of future visual states. The imagined trajectories are evaluated by a vision-language critic that assesses their semantic alignment with the task description. Crucially, the action policy is conditioned on hierarchical representations extracted from the predicted trajectories, and its training signal flows back to shape these representations. This coupled design allows language reasoning, generative prediction, and control to co-evolve during training, enabling the agent to imagine, evaluate, and act upon task-consistent futures. We next describe each component.

Planner $\mathcal{F}_P$. We instantiate $\mathcal{F}_P$ with a strong LLM [18] to serve as the high-level reasoning engine. The planner decomposes a long-horizon task instruction τ into a sequence of K semantically coherent sub-tasks $C = \{c^{(1)}, \ldots, c^{(K)}\}$ based on the current observation such that $c^{(1:K)} \sim \mathcal{F}_P(\cdot \mid s_t, \tau)$. This converts a compressed instruction into an explicit milestone sequence, providing dense semantic structure that improves long-horizon coherence compared to using τ alone [15]. In practice, we use $K \in [2, 5]$ depending on task complexity.

Video Generator $\mathcal{G}_\theta$. We adopt the pretrained Stable Video Diffusion (SVD) backbone [8] as the next state generator. Conditioned on the visual state s_t and the planner output, the model predicts a short-horizon future rollout of Δ frames represented as $\hat{s}_{t+1:t+\Delta}$.

VLM Critic $\mathcal{F}_{\mathcal{R}}$. A frozen vision-language model, parameterized by ψ, evaluates the generated future rollouts $\hat{s}_{t+1:t+\Delta}$ against the goal instruction τ. The critic operates in pixel space to produce a scalar reward signal r that reflects semantic alignment, realism, and goal satisfaction: This signal is used to optimize the targeted parameters of $\mathcal{G}_\theta$ via differentiable policy optimization [7], reinforcing denoising transitions that yield high-reward visual states.
Action Generator π_ϕ. The policy π_ϕ is implemented as a 1D diffusion UNet [17]. The action generator is conditioned on initial action noise a_i (where i denotes the denoising step) and the hidden state embeddings extracted from the video generator, and generates continuous executable action sequences $\mathbf{a}_{t:t+H}$.

3.3 Planning with Hierarchical Reasoning

A central insight of our framework is that raw task instructions are often too compressed and ambiguous to directly guide a generative model over long horizons. To bridge this semantic gap, we introduce an explicit planner that decomposes high-level goals into structured, actionable hierarchical sub-goals before any visual prediction or action generation takes place. In RoboTALES, however, the decomposition serves an additional role beyond simplifying control: the planner produces semantic subgoal tokens that condition the diffusion world model during generation. These tokens act as intermediate anchors that guide the imagined trajectory of future states, ensuring that predicted visual rollouts remain aligned with task intent. This integration enables task-conditioned visual imagination, where the video model predicts not only physically plausible futures but also futures that reflect the evolving semantic goals of the task.

Given an original task instruction τ, the planner $\mathcal{F}_P$ generates a sequence of K semantically coherent sub-goals: $c^{(1:K)} \sim \mathcal{F}_P(\cdot \mid s_t, \tau)$, where each sub-goal $c^{(k)}$ describes a concrete, short-horizon objective grounded in the agent's current visual context. For instance, a compressed instruction such as $\tau =$ `"Prepare the ingredients for Zucchini Curry"` is decomposed into specific directives: $c^{(1)} =$ `"Wash, peel, and chop the zucchini"`, $c^{(2)} =$ `"Chop the onions and tomatoes"`, and so forth. In practice, we extract $K \in [2, 5]$ sub-goals per instruction, scaling with the inherent complexity of the task. The sub-goals are then concatenated with the original instruction to form an augmented, reasoning-rich plan,

$$C^* = \left[\tau \; ; \; c^{(1)} \; ; \; c^{(2)} \; ; \; \ldots \; ; \; c^{(K)} \right]. \tag{1}$$

Unlike the initial compressed instruction, C^* provides a detailed roadmap that makes implicit task knowledge explicit. It provides a dense semantic signal to the video generator, enabling it to anticipate future states that are not merely visually plausible but also causally aligned with the task's logical progression.

3.4 Task-Aligned Predictive Model Steering and Action Decoding

RoboTALES trains the video generator $\mathcal{G}_\theta$ and the action policy π_ϕ jointly in a single stage. This coupled optimization is motivated by a key observation: a predictive model becomes most useful for control when its internal representations

are shaped not only by visual fidelity and semantic alignment, but also by the downstream action objective. In a decoupled regime, the video model's decoder features are fixed before the action policy ever sees them, meaning the model has no incentive to produce representations that are maximally informative for control. Joint training closes this loop by enabling the action-level gradients to flow back into the video generator's decoder layers, encouraging them to encode features that are simultaneously good for prediction and for acting.

During training, the planner $\mathcal{F}_P$ provides the augmented plan C^*, and we adapt the video diffusion UNet $\mathcal{G}_\theta$ while preserving its pretrained visual prior. Specifically, we freeze all parameters except *(i)* the cross-attention modules that ingest CLIP embeddings of C^* and *(ii)* a targeted subset of decoder layers whose hidden states condition the action UNet. Let $\theta^* \subset \theta$ denote these trainable parameters. This targeted update constrains learning to the interfaces where task semantics enter the model (text cross-attention) and where task-relevant abstractions emerge (selected decoder blocks), enabling the video generator to become instruction-aware without sacrificing generative stability.

The coupled objective comprises three complementary losses: a video diffusion loss for visual fidelity, a policy optimization loss for semantic alignment, and an action diffusion loss for precise control.

Video Diffusion Loss. We optimize the selected parameters using the standard noise-prediction objective,

$$L_{\text{video}}(\theta^*) = \mathbb{E}_{x_0, C^*, t, \epsilon}\big\|\epsilon - \epsilon_{\mathcal{G}_{\theta^*}}(x_t, \sigma_t, C^*)\big\|^2, \qquad x_t = \alpha_t x_0 + \sigma_t \epsilon,\ \epsilon \sim \mathcal{N}(0, \mathbf{I}). \tag{2}$$

Task Alignment via VLM Critic. While the diffusion loss encourages fidelity to the training distribution, it does not directly optimize for task-level semantic alignment. To enforce this alignment, we incorporate a pretrained VLM-based critic $\mathcal{F}_\mathcal{R}$ that scores decoded future states against goal instruction τ. We cast the iterative denoising process as a multi-step MDP and apply differentiable policy optimization [7], where each denoising step constitutes an action taken by the video model. Let $x_t \in \mathcal{Z}$ be the latent state at noise level σ_t. The VideoUNet first predicts a clean latent estimate: $x_{0,\theta} = \mathcal{G}_{\theta^*}(x_t, \sigma_t, C^*)$. Using Euler–Ancestral updates, the transition mean to the next less noisy state becomes,

$$\mu_\theta(x_t, \sigma_t, C^*) = x_t + (\sigma_{t-1} - \sigma_t)\frac{x_t - x_{0,\theta}}{\sigma_t}. \tag{3}$$

We treat each sampler step as a Gaussian transition,

$$p_{\mathcal{G}^*_\theta}(x_{t-1} \mid x_t, C^*) = \mathcal{N}\big(\mu_\theta(x_t, \sigma_t, C^*), \Sigma_t\big), \quad \Sigma_t = (\sigma_t^{\text{ker}})^2 \mathbf{I}, \tag{4}$$

where $\sigma_t^{\text{ker}} = \eta\sqrt{\sigma_t^2 - \sigma_{t-1}^2}$ controls the stochasticity of the sampler and η is a tunable noise coefficient.

Per-step Log-Likelihood. For a realized transition in a d-dimensional latent space, the log-likelihood of transitioning from x_t to x_{t-1} is,

$$\ell_\theta(x_{t-1} \mid x_t, C^*) = -\frac{\|x_{t-1} - \mu_\theta(x_t, \sigma_t, C^*)\|_2^2}{2(\sigma_t^{\text{ker}})^2} - \frac{d}{2}\log\big(2\pi(\sigma_t^{\text{ker}})^2\big). \tag{5}$$

Reward Computation. Given a complete denoising rollout $(x_T, \ldots, x_0)$, we decode selected keyframes into pixel-space observations represented as: $s_{t:t+\Delta}$ and evaluate them with the frozen VLM critic which gives a scaler reward r,

$$r = F_{\mathcal{R}}(s_{t:t+\Delta}, \tau) \in \mathbb{R}. \tag{6}$$

We compute a per-instruction baseline $b(\tau)$ as a running mean of rewards for instruction τ, yielding the advantage $A = r - b(\tau)$.

Policy Optimization Objective. Following [7], we optimize a REINFORCE-style objective over the denoising trajectory. Let $\mathcal{T}_K$ denote a uniformly sampled subset of K denoising steps,

$$J_{\text{DDPO}}(\theta^*) = \mathbb{E}_{(x_T \ldots x_0)}\left[A \cdot \frac{1}{K} \sum_{t \in \mathcal{T}_K} \overline{\ell_\theta(x_{t-1} \mid x_t, \tau)}\right], \; L_{\text{DDPO}}(\theta^*) = -J_{\text{DDPO}}(\theta^*). \tag{7}$$

Action Diffusion Loss. Simultaneously, the action UNet π_ϕ receives initial action noise $\mathbf{a}_i$ (where i denotes the denoising step) and hidden state embeddings $f^{\{l\}}_{\mathcal{G}_{\theta^*}}$ from the selected decoder layers of the video generator, and produces executable actions represented as $a_t \sim \pi_\phi\left(. \mid \mathbf{a}_i, f^{\{l\}}_{\mathcal{G}_{\theta^*}, C^*}\right)$. The action UNet is trained using the standard noise prediction objective,

$$L_{\text{action}}(\phi, \theta^*) = \mathbb{E}_{(a,\varepsilon,i)}\left[\left\|\varepsilon - \pi_\phi\left(\cdot \,\middle|\, \mathbf{a}_i,\, f^{\{l\}}_{\mathcal{G}_{\theta^*}, C^*}\right)\right\|^2\right]. \tag{8}$$

Critically, unlike decoupled approaches, we do not place a stop-gradient between π_ϕ and $\mathcal{G}_{\theta^*}$. Gradients from L_{action} propagate back through the conditioning features $f^{\{l\}}_{\mathcal{G}_{\theta^*}}$ into the video generator's trainable decoder layers θ^*. This means the video model receives a direct training signal from the action objective; its decoder features are shaped not only to produce visually faithful and semantically aligned futures, but also to be maximally informative for action prediction. The video generator thus learns to "imagine for acting"—encoding in its latent space the spatial, temporal, and semantic structure that the action policy needs for precise control.

Joint Objective. The full training loss combines all three objectives,

$$L_{\text{total}}(\theta^*, \phi) = L_{\text{video}}(\theta^*) + \beta\, L_{\text{DDPO}}(\theta^*) + \gamma\, L_{\text{action}}(\phi, \theta^*), \tag{9}$$

where β and γ are scaler weights balancing the generative, semantic, and control objectives.

This coupled formulation offers three advantages over decoupled training: *(i)* the video generator's decoder features co-adapt to the action objective, producing representations that encode control-relevant information that a purely generative loss would not incentivize; *(ii)* the action policy benefits from features that are continuously refined by both semantic and visual losses, rather than being constrained to a frozen representation that was never optimized for control; and *(iii)* the shared gradient flow creates an implicit feedback loop where improvements in action prediction inform better imagination, and better imagination in turn yields more informative features for action prediction. We validate this design choice through an ablation in the Appendix.

4 Experimentation

Implementation Details. Our implementation is built upon the codebase and experimental settings of [35]. We utilize Gemini-2.5-Pro [18] as the high-level Planner $\mathcal{F}_P$. The framework was trained on the RoboCasa dataset [40] using two NVIDIA A100 80GB GPUs. Joint training required approximately 6–7 days to complete with a batch size of 1 per GPU. Hyperparameters, including the learning rate and optimizer settings, were kept consistent with those in [35]. The VideoUNet and the Action-UNet were initialized with the Stage 2 weights from the official repository of [35]. For the reward module, we employed the reward choices from [7], [5] and [37], serving as the reward metric for DDPO. Rewards were computed on decoded keyframes every 4 steps and remained frozen throughout training. Typically, the input state is a single frame with 3 views, and 8 future rollouts are obtained per view.

Simulation Setup and Baselines. We evaluate our method on the RoboCasa [40] and LIBERO10 [36] benchmarks, spanning a total of 34 manipulation tasks. RoboCasa provides a large-scale, realistic simulation environment for complex manipulation research. For each task, both benchmarks provide 50 human demonstrations. Demonstrations in RoboCasa were replayed at a resolution of 256×256 pixels, and LIBERO10 demonstrations were resized to match this resolution. The action space is defined as $\mathbf{a}_t \in \mathbb{R}^7$, comprising the 6-DoF gripper pose and a scalar for the gripper state (open/closed). For RoboCasa, we follow the evaluation protocol of [35], executing 50 rollouts across five distinct scenes per task. We use the baselines reported in [35]—including the RoboCasa-trained versions of [33] and ResNet- and CLIP-based variants of [17]. For LIBERO10, we adhere to the protocols defined in [33, 35].

4.1 Quantitative Results

Tables 1 and 2 summarize success rates across a diverse set of long-horizon manipulation tasks. Overall, on RoboCasa simulation dataset (Table 1), **RoboTALES** attains the highest average success and improves consistently across categories, indicating that its gains are not confined to a narrow subset of tasks. Notably, the largest margins appear on *structure-sensitive* tasks (*e.g.*, doors and drawers) where the agent must execute temporally ordered subgoals (approach $\rightarrow$ grasp $\rightarrow$ actuate $\rightarrow$ release) and small deviations compound into failure. We also observe decent improvements on *long horizon* tasks (*e.g.*, Pick and Place) and *contact-rich* behaviors (*e.g.*, turning knobs/levers and insertion), suggesting that aligning simulated futures to the instruction yields predictive features that better disambiguate fine-grained action choices. Notably, [23] and [1] employ substantially more demonstrations for behavior cloning (300 demos), yet our method surpasses both using only 50 demonstrations, underscoring the sample efficiency. In contrast to prior video-based action extraction pipelines [6, 35, 46][1], Robo-

[1] Reproduced numbers have been verified via correspondence with the VideoPolicy [35] authors; they acknowledge a discrepancy in performance between their paper and

Table 1: Comparison to the state of the art on the validation set of the RoboCasa simulation benchmark. Success rates are computed over 50 rollouts per task. Our method uses 50 demonstrations.

Category	Task	3DA	DP3	DP-ResNet	DP-CLIP	GR00T	FPV	DP-VLA	UVA	Video-Policy	Ours
Pick and Place	PnPCabToCounter	0.00	0.04	0.06	0.00	0.20	0.10	0.10	0.26	0.16	**0.44**
	PnPCounterToCab	0.00	0.02	0.06	0.02	0.36	0.14	0.32	0.18	0.38	**0.44**
	PnPCounterToMicrowave	0.00	0.06	0.06	0.02	0.13	0.10	**0.56**	0.10	0.40	0.44
	PnPCounterToSink	0.00	0.00	0.14	0.08	0.10	0.08	0.30	0.16	0.32	**0.50**
	PnPCounterToStove	0.00	0.00	0.00	0.02	0.24	0.04	0.22	0.16	0.52	**0.54**
	PnPMicrowaveToCounter	0.00	0.06	0.06	0.06	0.16	0.12	0.18	0.18	0.16	**0.26**
	PnPSinkToCounter	0.00	0.00	0.10	0.22	0.33	0.30	0.56	0.38	0.52	**0.56**
	PnPStoveToCounter	0.00	0.00	0.02	0.06	0.29	0.26	0.62	0.24	0.64	**0.66**
Doors	OpenSingleDoor	0.00	0.24	0.42	0.32	0.59	0.74	0.42	0.54	0.78	**0.80**
	OpenDoubleDoor	0.00	0.20	0.70	0.82	0.15	0.92	0.80	0.90	0.92	**0.94**
	CloseDoubleDoor	0.00	0.56	0.78	0.84	0.75	0.78	0.84	0.76	0.90	**0.94**
	CloseSingleDoor	0.14	0.62	0.78	0.48	0.83	0.84	**1.00**	0.88	0.98	0.98
Drawers	OpenDrawer	0.00	0.36	0.64	0.60	0.79	0.72	0.66	0.28	0.40	**0.8**
	CloseDrawer	0.00	0.48	0.82	0.96	0.99	0.94	1.00	0.72	0.88	**1.0**
Twisting Knobs	TurnOnStove	0.10	0.24	0.38	0.28	0.56	**0.66**	0.64	0.50	0.34	0.40
	TurnOffStove	0.02	0.06	0.16	0.08	**0.27**	0.20	0.16	0.14	0.06	0.10
Turning Levers	TurnOnSinkFaucet	0.06	0.32	0.66	0.66	0.63	0.70	0.56	0.62	0.80	**0.84**
	TurnOffSinkFaucet	0.28	0.42	0.68	0.70	0.73	0.78	0.72	0.64	0.68	**0.78**
	TurnSinkSpout	0.26	0.54	0.62	0.26	0.53	0.52	**0.90**	0.64	0.26	0.32
Pressing Buttons	CoffeePressButton	0.08	0.16	0.76	0.68	0.85	0.90	0.86	0.84	0.92	**0.96**
	TurnOnMicrowave	0.06	0.38	0.68	0.88	0.78	0.68	0.84	0.94	0.86	**0.96**
	TurnOffMicrowave	0.32	0.54	0.62	**1.00**	0.71	0.96	0.86	0.96	0.94	0.96
Insertion	CoffeeServeMug	0.00	0.18	0.44	0.60	0.73	0.48	0.64	**0.78**	0.74	0.60
	CoffeeSetupMug	0.00	0.04	0.10	0.12	0.23	0.16	**0.30**	0.20	0.18	0.20
Avg. Success		0.06	0.23	0.41	0.43	0.50	0.51	0.57	0.50	0.575	**0.64**

Table 2: Average success rates for tasks in the LIBERO10 benchmark.

Model	DP-C	DP-T	OpenVLA	π_0	π_0-FAST	UVA	VideoPolicy	Ours
Avg.	0.53	0.58	0.54	0.85	0.60	0.90	0.94	**0.97**

TALES explicitly enforces *task-aligned imagination* providing reasoning-rich features for a more reliable conditioning signal for action diffusion, translating into higher task completion rates across the benchmark. This consistent improvement across diverse tasks in the RoboCasa distinct benchmarks spanning 24 manipulation tasks confirms the generalizability of our approach. Similarly, in Table 2, our method achieves a mean success rate of 97% across all 10 tasks in the LIBERO10 benchmark [36], outperforming all baselines by a substantial margin as shown in Table 2. Among prior approaches, VideoPolicy represents the strongest competitor with a success rate of 94%, followed by UVA at 90% and π_0 at 85%, while standard diffusion policy variants (DP-C: 53%, DP-T: 58%) and OpenVLA (54%) lag considerably further behind.

released codebase. To ensure a fair comparison, we use the exact same experimental setup for all experiments in this paper.

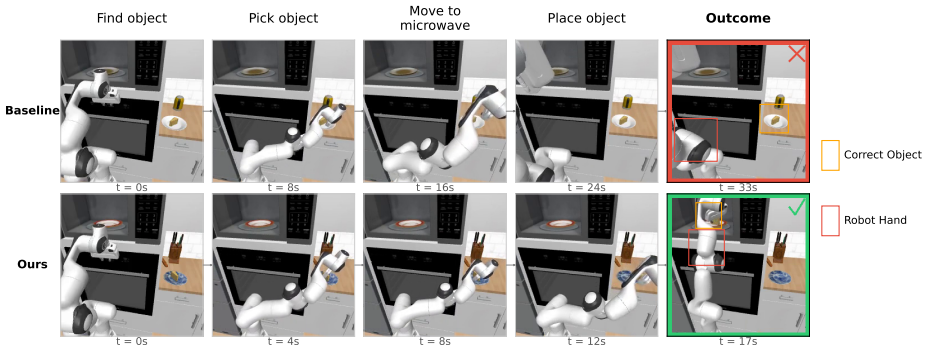


Fig. 3: Qualitative Comparison of Task Execution. While the baseline model (top) fails to successfully pick up the target object and subsequently loses environmental consistency due to semantic drift, our framework (bottom) maintains structural integrity throughout the sequence and accurately executes the transition from picking the object to successfully placing it inside the microwave. *(Best viewed zoomed in)*

4.2 Qualitative Analysis

We conduct a qualitative comparison of our framework against the current state-of-the-art baseline, VideoPolicy [35], focusing on task execution stability and semantic grounding. As illustrated in Fig. 3, we evaluate both models on manipulation tasks in the RoboCasa environment. A primary failure in the baseline is semantic drift during long-horizon execution. While the baseline can initiate a grasp, it fails to maintain environmental integrity during complex transitions, such as moving an object toward a microwave. As seen in Fig. 3 (top), the baseline exhibits physical distortions, leading to task failure. In contrast, our framework maintains high fidelity. By conditioning the video model on reasoning-rich sub-goals the agent retains a strong semantic anchor. This prevents execution drift, ensuring trajectories remain physically plausible and goal-oriented.

4.3 Ablations and Discussion

In this section, we evaluate the individual contributions of our core components and discuss key properties.

Effect of the Planner. We compare three variants: *(i)* [35] augmented with our Planner at inference, *(ii)* our model trained with Planner conditioning but without the Critic, and *(iii)* our full model with both. As shown in Fig. 4 (averaged across 3 RoboCasa tasks), simply injecting structured sub-goals into an unadapted video model yields the lowest success rate, confirming that a pre-trained model cannot exploit hierarchical plans without being explicitly trained to reason over them. Training video model with planner conditioning alone improves performance, confirming that the decomposed plan provides a stronger semantic signal than the raw instruction. Finally, our full method, which additionally incorporates VLM Critic feedback to align the video model's imagined futures with the task goal, achieves the highest success rate. This demonstrates

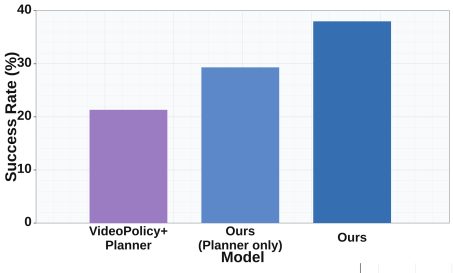
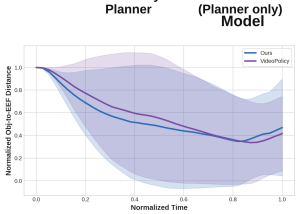
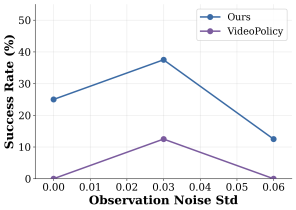


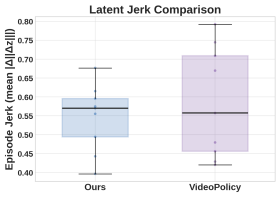
Fig. 4: Impact of the Planner. Planner provides structured subgoals, improving performance when trained with the video generator. Naively augmenting planner with baseline (purple graph) without adaptation results in performance degradation.



(a) Trajectory Quality

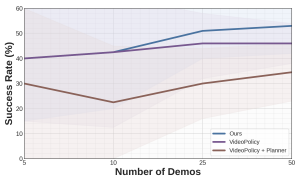


(b) Robustness

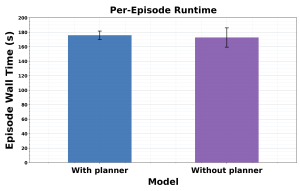


(c) Latent jerk

Fig. 5: Analysis.(a) Normalized object-to-end-effector distance over time. Our method shows consistent purposeful behavior and lower variance, **(b)** Success rate (%) under increasing observation noise. Our method maintains significantly higher performance across all noise levels, demonstrating robust semantic representations. **(c)** RoboTALES produces smoother latent dynamics with lower jerk, indicating more temporally coherent representations that translate to smoother robot actions.



(a) Demo Efficiency



(b) Inference Time

Figure 6: (a) Our method scales consistently with fewer demos, outperforming both baselines. **(b)** The added overhead due to Planner is negligible (∼1s), confirming execution efficiency.

that the Planner is necessary but not sufficient on its own; its full benefit is realized when the video generator is trained to ground structured sub-goals into semantically coherent future predictions.

Effect of Policy Optimization. We examine whether RL-based refinement via the VLM Critic yields representations that improve downstream control beyond pixel-level prediction. We compare against [35], which uses only a video diffusion loss with no semantic reward, and evaluate along three axes (Fig. 5). In terms of *trajectory quality* (Fig. 5a), both methods progressively reduce the object-to-end-effector distance, but ours exhibits significantly lower variance, indicating more consistent reaching behavior from critic-aligned features. For *robustness* (Fig. 5b), we inject Gaussian noise $\sigma \in \{0.00, 0.03, 0.06\}$ into observations at inference; our method maintains substantially higher success rates across all noise levels while [35] collapses to near zero at $\sigma = 0.06$. Finally, for *action smoothness* (Fig. 5c), we measure end-effector jerk as a proxy for motion quality, where our

method produces lower jerk values reflecting smoother, more physically plausible actions. These results confirm that the Critic actively shapes the video model's internal representations to be more stable and robust for action generation, not merely improving visual fidelity.

Demonstration Efficiency. Fig. 6a reports the demonstration efficiency of our method against two baselines, averaged over four RoboCasa tasks (PnPSinkToCounter, PnPStoveToCounter, PnPCounterToStove, PnPCounterToCab), across varying numbers of evaluation demonstrations. Even with as few as 10 demonstrations, our method achieves a competitive success rate of approximately 43%, comparable to VideoPolicy [35] while significantly outperforming its planner-augmented variant. As the number of demonstrations increases, our method scales consistently, reaching approximately 51% at 25 demonstrations and 53% at 50, whereas VideoPolicy saturates around 46% and its planner-augmented variant peaks at only 34%. The shaded confidence intervals further reveal that our method exhibits reduced variance at higher demonstration counts, indicating more stable and reliable learning.

Inference Time Comparison. A major concern with hierarchical models is the potential for slower performance due to multiple processing steps. Since our model only utilizes the Planner at the inference, therefore, the added cost is minimal. We compute the mean time taken by the model with and without Planner across 3 RoboCasa tasks spanning across 50 demos and 30 episodes. Fig. 6b shows minimal increment in per-episode time at inference, which directly verifies the execution efficiency of our approach.

We refer reader to the Appendix for further results and discussions.

5 Conclusion and Future Work

We present RoboTALES, a framework that tightly couples hierarchical reasoning, video generation, and action generation for long-horizon robotic manipulation. By decomposing compressed task instructions into structured sub-goals via an LLM Planner and using them to condition a video model, our approach enables task-aligned visual imagination that goes beyond mere pixel-level fidelity. A frozen VLM Critic steers the video generator toward semantically faithful futures, producing latent representations that are simultaneously optimized for motor control. Extensive experiments across challenging benchmarks demonstrate that RoboTALES achieves impressive performance, outperforming strong baselines in task success rate, demonstration efficiency, long-horizon ability, and robustness to visual perturbations.

Future Work. Our framework currently relies on a frozen VLM critic that provides a coarse semantic reward. A natural extension is to design task-aware reward functions that capture finer-grained progress signals such as sub-goal completion and physical plausibility. More ambitiously, we aim to explore learnable reward models that are jointly optimized with the video generator and the action policy, enabling the critic to co-adapt as the agent's capabilities improve.

Bibliography

[1] Abeyruwan, S., Ainslie, e.a.: Gemini robotics: Bringing ai into the physical world. arXiv preprint arXiv:2503.20020 (2025)

[2] Ada, S.E., Oztop, E., Ugur, E.: Diffusion policies for out-of-distribution generalization in offline reinforcement learning. IEEE Robotics and Automation Letters **9**(4), 3116–3123 (2024). https://doi.org/10.1109/LRA.2024.3363530

[3] Ahn, M., et al.: Do as i can, not as i say: Grounding language in robotic affordances. arXiv preprint arXiv:2204.01691 (2022)

[4] Andreas, J., Klein, D., Levine, S.: Modular multitask reinforcement learning with policy sketches. In: International Conference on Machine Learning (ICML). pp. 166–175. PMLR (2017)

[5] Bahng, H., Chan, C., Durand, F., Isola, P.: Cycle consistency as reward: Learning image-text alignment without human preferences. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 22934–22946 (2025)

[6] Bharadhwaj, H., Dwibedi, D., Gupta, A., Tulsiani, S., Doersch, C., Xiao, T., Shah, D., Xia, F., Sadigh, D., Kirmani, S.: Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. arXiv preprint arXiv:2409.16283 (2024)

[7] Black, K., Janner, M., Du, Y., Kostrikov, I., Levine, S.: Training diffusion models with reinforcement learning. arXiv preprint arXiv:2305.13301 (2023)

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

[9] Botvinick, M.M.: Hierarchical models of behavior and prefrontal function. Trends in Cognitive Sciences **12**(5), 201–208 (2008)

[10] Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., Dubey, A., Finn, C., et al.: RT-2: Vision-language-action models transfer web knowledge to robotic control. arXiv preprint arXiv:2307.15818 (2023)

[11] Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., Ramesh, A.: Video generation models as world simulators. https://openai.com/index/video-generation-models-as-world-simulators/, last accessed 2026-06-29 (2024)

[12] Bruce, J., et al.: Genie: Generative interactive environments. In: International Conference on Machine Learning (ICML). PMLR (2024)

[13] Chandra, A.L., Nematollahi, I., Huang, C., Welschehold, T., Burgard, W., Valada, A.: Diwa: Diffusion policy adaptation with world models. Conference on Robot Learning (CoRL) (2025)

[14] Chen, D., Moutakanni, T., Chung, W., Bang, Y., Ji, Z., Bolourchi, A., Fung, P.: Planning with reasoning using vision language world model. arXiv preprint arXiv:2509.02722 (2025)
[15] Chen, D., Moutakanni, T., Chung, W., Bang, Y., Ji, Z., Bolourchi, A., Fung, P.: Planning with reasoning using vision language world model. arXiv preprint arXiv:2509.02722 (2025)
[16] Chen, H., Lu, C., Ying, C., Su, H., Zhu, J.: Offline reinforcement learning via high-fidelity generative behavior modeling. In: The Eleventh International Conference on Learning Representations (2023)
[17] Chi, C., Xu, Z., Feng, S., Cousineau, E., Du, Y., Burchfiel, B., Tedrake, R., Song, S.: Diffusion policy: Visuomotor policy learning via action diffusion. The International Journal of Robotics Research (2024)
[18] Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al.: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261 (2025)
[19] Ha, D., Schmidhuber, J.: World models. arXiv preprint arXiv:1803.10122 (2018)
[20] Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al.: Soft actor-critic algorithms and applications. arXiv preprint arXiv:1812.05905 (2018)
[21] Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., Davidson, J.: Learning latent dynamics for planning from pixels. In: International Conference on Machine Learning (ICML). pp. 2555–2565 (2019)
[22] Hafner, D., Pasukonis, J., Ba, J., Lillicrap, T.: Mastering diverse domains through world models. arXiv preprint arXiv:2301.04104 (2023). https://doi.org/10.48550/arXiv.2301.04104
[23] Han, B., Kim, J., Jang, J.: A dual process vla: Efficient robotic manipulation leveraging vlm. arXiv preprint arXiv:2410.15549 (2024)
[24] Hansen, N., Su, H., Wang, X.: Td-mpc2: Scalable, robust world models for continuous control (2024)
[25] Hansen-Estruch, P., Kostrikov, I., Janner, M., Kuba, J.G., Levine, S.: Idql: Implicit q-learning as an actor-critic method with diffusion policies (2023)
[26] He, L., Shen, L., Zhang, L., Tan, J., Wang, X.: Diffcps: Diffusion model based constrained policy search for offline reinforcement learning. arXiv preprint arXiv:2310.05333 (2024)
[27] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 33, pp. 6840–6851 (2020)
[28] Hu, A., Russell, L., Yeo, H., Murez, Z., Fedoseev, G., Kendall, A., Shotton, J., Corrado, G.: Gaia-1: A generative world model for autonomous driving. arXiv preprint arXiv:2309.17080 (2023)
[29] Huang, W., Xia, F., Xiao, T., Chan, H., Liang, J., Florence, P., Zeng, A., Tompson, J., Mordatch, I., Chebotar, Y., et al.: Inner monologue: Embodied reasoning through planning with language models. In: Conference on Robot Learning (CoRL). PMLR (2022)

[30] Janner, M., Du, Y., Tenenbaum, J.B., Levine, S.: Planning with diffusion for flexible behavior synthesis. In: International Conference on Machine Learning (2022)
[31] Jia, X., Wang, Q., Donat, A., Xing, B., Li, G., Zhou, H., Celik, O., Blessing, D., Lioutikov, R., Neumann, G.: Mail: Improving imitation learning with selective state space models. In: Agrawal, P., Kroemer, O., Burgard, W. (eds.) Proceedings of The 8th Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 270, pp. 3888–3907. PMLR (06–09 Nov 2025)
[32] Ke, T.W., Gkanatsios, N., Fragkiadaki, K.: 3d diffuser actor: Policy diffusion with 3d scene representations. Arxiv (2024)
[33] Li, S., Gao, Y., Sadigh, D., Song, S.: Unified video action model. arXiv preprint arXiv:2503.00200 (2025)
[34] Li, Y., Zhu, Y., Wen, J., Shen, C., Xu, Y.: Worldeval: World model as real-world robot policies evaluator. arXiv preprint arXiv:2505.19017 (2025)
[35] Liang, J., Tokmakov, P., Liu, R., Sudhakar, S., Shah, P., Ambrus, R., Vondrick, C.: Video generators are robot policies. arXiv preprint arXiv:2508.00795 (2025)
[36] Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., Stone, P.: Libero: Benchmarking knowledge transfer for lifelong robot learning. arXiv preprint arXiv:2306.03310 (2023)
[37] Ma, Y.J., Kumar, V., Zhang, A., Bastani, O., Jayaraman, D.: Liv: Language-image representations and rewards for robotic control. In: International Conference on Machine Learning. pp. 23301–23320. PMLR (2023)
[38] Miller, G.A., Galanter, E., Pribram, K.H.: Plans and the Structure of Behavior. Holt, Rinehart and Winston, New York (1960)
[39] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
[40] Nasiriany, S., Maddukuri, A., Zhang, L., Parikh, A., Lo, A., Joshi, A., Mandlekar, A., Zhu, Y.: Robocasa: Large-scale simulation of everyday tasks for generalist robots. arXiv preprint arXiv:2406.02523 (2024)
[41] NVIDIA, :, Agarwal, N., Ali, A., Bala, M., Balaji, Y., Barker, E., Cai, T., Chattopadhyay, P., Chen, Y., Cui, Y., Ding, Y., Dworakowski, D., Fan, J., Fenzi, M., Ferroni, F., Fidler, S., Fox, D., Ge, S., Ge, Y., Gu, J., Gururani, S., He, E., Huang, J., Huffman, J., Jannaty, P., Jin, J., Kim, S.W., Klár, G., Lam, G., Lan, S., Leal-Taixe, L., Li, A., Li, Z., Lin, C.H., Lin, T.Y., Ling, H., Liu, M.Y., Liu, X., Luo, A., Ma, Q., Mao, H., Mo, K., Mousavian, A., Nah, S., Niverty, S., Page, D., Paschalidou, D., Patel, Z., Pavao, L., Ramezanali, M., Reda, F., Ren, X., Sabavat, V.R.N., Schmerling, E., Shi, S., Stefaniak, B., Tang, S., Tchapmi, L., Tredak, P., Tseng, W.C., Varghese, J., Wang, H., Wang, H., Wang, H., Wang, T.C., Wei, F., Wei, X., Wu, J.Z., Xu, J., Yang, W., Yen-Chen, L., Zeng, X., Zeng, Y., Zhang, J., Zhang, Q., Zhang, Y., Zhao, Q., Zolkowski, A.: Cosmos world foundation model platform for physical ai. arXiv preprint arXiv:2501.03575 (2025)

[42] Parker-Holder, J., Ball, P., Bruce, J., Dasagi, V., Holsheimer, K., Kaplanis, C., Moufarek, A., Scully, G., Shar, J., Shi, J., Spencer, S., Yung, J., Dennis, M., Kenjeyev, S., Long, S., Mnih, V., Chan, H., Gazeau, M., Li, B., Pardo, F., Wang, L., Zhang, L., Besse, F., Harley, T., Mitenkova, A., Wang, J., Clune, J., Hassabis, D., Hadsell, R., Bolton, A., Singh, S., Rocktäschel, T.: Genie 2: A large-scale foundation world model. `https://deepmind.google/discover/blog/genie-2-a-large-scale-foundation-world-model/`, last accessed 2026-06-29 (2024)
[43] Pateria, S., Subagdja, B., Tan, A.h., Quek, C.: Hierarchical reinforcement learning: A comprehensive survey. ACM Computing Surveys **54**(5), 1–35 (2021)
[44] Prasad, A., Lin, K., Wu, J., Zhou, L., Bohg, J.: Consistency policy: Accelerated visuomotor policies via consistency distillation. In: Robotics: Science and Systems (2024)
[45] Reuss, M., Yağmurlu, Ö.E., Wenzel, F., Lioutikov, R.: Multimodal diffusion transformer: Learning versatile behavior from multimodal goals. In: Robotics: Science and Systems (2024)
[46] Routray, S., Pan, H., Jain, U., Bahl, S., Pathak, D.: Vipra: Video prediction for robot actions. arXiv preprint arXiv:2511.07732 (2025)
[47] Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., et al.: Mastering atari, go, chess and shogi by planning with a learned model. Nature **588**(7839), 604–609 (2020)
[48] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
[49] Shah, D., Osinski, B., Ichter, B., Levine, S.: LM-nav: Robotic navigation with large pre-trained models of language, vision, and action. In: Conference on Robot Learning (CoRL). PMLR (2023)
[50] Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
[51] Sutton, R.S.: Dyna, an integrated architecture for learning, planning, and reacting. ACM SIGART Bulletin **2**(4), 160–163 (1991)
[52] Wang, Y., Zhang, Y., Huo, M., Tian, R., Zhang, X., Xie, Y., Xu, C., Ji, P., Zhan, W., Ding, M., et al.: Sparse diffusion policy: A sparse, reusable, and flexible policy for robot learning. CoRL (2024)
[53] Wang, Z., Hunt, J.J., Zhou, M.: Diffusion policies as an expressive policy class for offline reinforcement learning. International Conference on Learning Representations (2023)
[54] Wang, Z., Li, Z., Mandlekar, A., Xu, Z., Fan, J., Narang, Y., Fan, L., Zhu, Y., Balaji, Y., Zhou, M., Liu, M.Y., Zeng, Y.: One-step diffusion policy: Fast visuomotor policies via diffusion distillation. arXiv preprint arXiv:2410.21257 (2024)
[55] Yan, G., Wu, Y.H., Wang, X.: Dnact: Diffusion guided multi-task 3d policy learning. arXiv preprint arXiv:2403.04115 (2024)
[56] Yang, M., Du, Y., Ghasemipour, K., Tompson, J., Schuurmans, D., Abbeel, P.: Learning interactive real-world simulators. International Conference on Learning Representations (2024)

[57] Ze, Y., Zhang, G., Zhang, K., Hu, C., Wang, M., Xu, H.: 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. arXiv preprint arXiv:2403.03954 (2024)
[58] Zhao, T.Z., Tompson, J., Driess, D., Florence, P., Ghasemipour, S.K.S., Finn, C., Wahid, A.: Aloha unleashed: A simple recipe for robot dexterity. In: Agrawal, P., Kroemer, O., Burgard, W. (eds.) Proceedings of The 8th Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 270, pp. 1910–1924. PMLR (06–09 Nov 2025)
[59] Zhu, M., Zhu, Y., Li, J., Wen, J., Xu, Z., Liu, N., Cheng, R., Shen, C., Peng, Y., Feng, F., et al.: Scaling diffusion policy in transformer to 1 billion parameters for robotic manipulation. arXiv preprint arXiv:2409.14411 (2024)

RoboTALES: Learning Reasoning-Guided Robot Policies via Task-Aligned Simulated Futures

A Appendix

The appendix is organized as below:

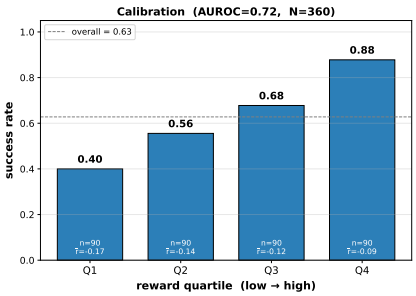
A.1 Quantitative Results on LIBERO10 Benchmark (Sec. A.1)
A.2 Additional Ablations and Analysis (Sec. A.2)
A.3 Additional Qualitative Results (Sec. A.3)
A.4 Planner Details (Sec. A.4)
A.5 Limitations (Sec. A.5)

A.1 Quantitative Results on Libero10 Benchmark

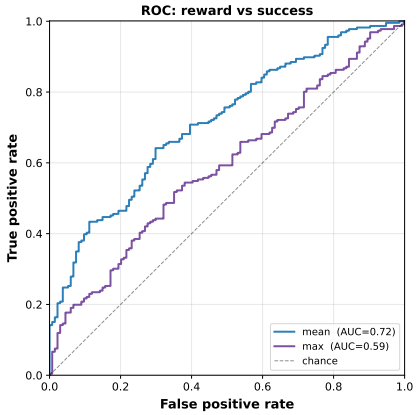
Table 2 of the main paper demonstrated the mean success rate comparison across all 10 tasks in the LIBERO10 benchmark [36], where RoboTALES outperforms all baselines by a substantial margin The per-task breakdown in Table 3 reveals that our method achieves perfect or near-perfect success on the majority of tasks, including a perfect 1.0 on both "KITCHEN SCENE3 turn on the stove and put the moka pot on it" and "LIVING ROOM SCENE1 put both the alphabet soup and the cream cheese box in the basket," with only "KITCHEN SCENE8 put both moka pots on the stove" showing a relatively lower score of 0.88. Notably, the tasks where our model performs strongest tend to involve sequential multi-step manipulation requiring coherent long-horizon reasoning — precisely the setting where critic-guided semantic grounding and hierarchical planning provide the greatest benefit. The consistent high performance across diverse scene configurations (kitchen, living room, and study environments) further suggests that our reasoning-guided world model generalizes robustly rather than overfitting to a narrow set of visual contexts.

Table 3: Per-task success rates out of 50 trials for the 10 tasks in the Libero10 benchmark, following the evaluation protocol in [35].

Libero10 Tasks	**Success**
LIVING ROOM SCENE2 put both the alphabet soup and the tomato sauce in the basket	0.98
LIVING ROOM SCENE2 put both the cream cheese box and the butter in the basket	0.98
KITCHEN SCENE3 turn on the stove and put the moka pot on it	1.0
KITCHEN SCENE4 put the black bowl in the bottom drawer of the cabinet and close it	0.96
LIVING ROOM SCENE5 put the white mug on the left plate and put the yellow and white mug on the right plate	0.98
STUDY SCENE1 pick up the book and place it in the back compartment of the caddy	0.98
LIVING ROOM SCENE6 put the white mug on the plate and put the chocolate pudding to the right of the plate	0.94
LIVING ROOM SCENE1 put both the alphabet soup and the cream cheese box in the basket	1.0
KITCHEN SCENE8 put both moka pots on the stove	0.88
KITCHEN SCENE6 put the yellow and white mug in the microwave and close it	0.98
Average	**0.97**



(a) Success rate increases monotonically with the critic's predicted reward quartile (AUROC = 0.72), showing the reward is well calibrated to task success.



(b) ROC for predicting success from the critic reward. Mean aggregation (AUC = 0.72) clearly outperforms single-frame max (AUC = 0.59).

Fig. 7: Critic reward predicts task success. Calibration by reward quartile (left) and ROC comparing reward aggregations (right) over 360 rollouts.

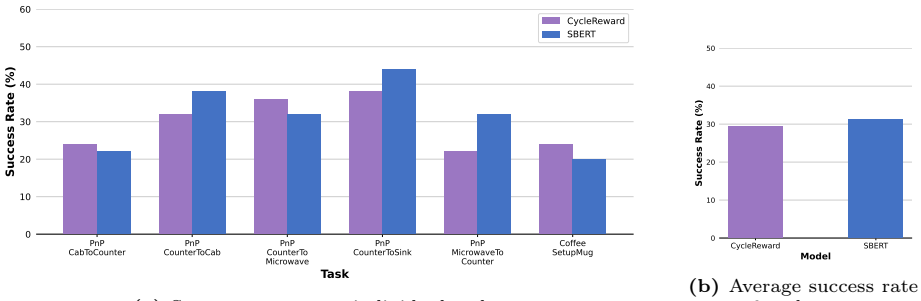
A.2 Additional Ablations and Analysis

Effect of Critic The critic assigns each rollout a scalar reward intended to predict task success. To validate this signal, we evaluate it against ground-truth outcomes over 360 rollouts (overall success rate 0.63). Binning rollouts into reward quartiles, success rate rises monotonically from 0.40 in the lowest quartile to 0.88 in the highest (Fig. 7a), confirming that the critic is well calibrated: higher predicted reward reliably corresponds to a higher chance of success. Treating the reward as a binary success classifier yields an AUROC of 0.72, well above the chance level of 0.5.

How the per-step rewards are aggregated into a trajectory score matters. We compare the *mean* reward over the trajectory against the single-frame *max* (Fig. 7b). The mean aggregation achieves an AUROC of 0.72, substantially outperforming the max (0.59, near chance). This indicates the critic's signal reflects sustained agreement across the trajectory rather than a spurious peak at an individual frame, motivating our use of the mean-aggregated reward.

Effect of different VLM critic choices. To determine the most effective reward signal for guiding our diffusion video generator, we ablate two distinct VLM critic architectures: CycleReward [5] and SBERT-based embeddings. The critic is responsible for providing high-level reasoning feedback by measuring the semantic alignment between the generated video frames and the goal state.

As illustrated in Figure 8, we observe that the optimal critic choice varies by task geometry and semantic complexity. For instance, CycleReward shows a slight edge in highly structured tasks like Coffee SetupMug, whereas SBERT significantly outperforms in varied PnP scenarios such as PnP CounterToSink and PnP MicrowaveToCounter. Ultimately, we select SBERT as our primary reason-



(a) Success rate across individual tasks.

(b) Average success rate across 6 tasks.

Fig. 8: Performance comparison of different VLM-based reward models for reasoning guidance. We evaluate the impact of CycleReward [5] versus BLIP-SBERT score [7] on success rates across six challenging manipulation tasks including five Pick-and-Place (PnP) variants and a mug setup task as shown in (a). While performance is task-dependent, SBERT variant demonstrates slightly superior robustness and a higher overall average success rate as shown in (b).

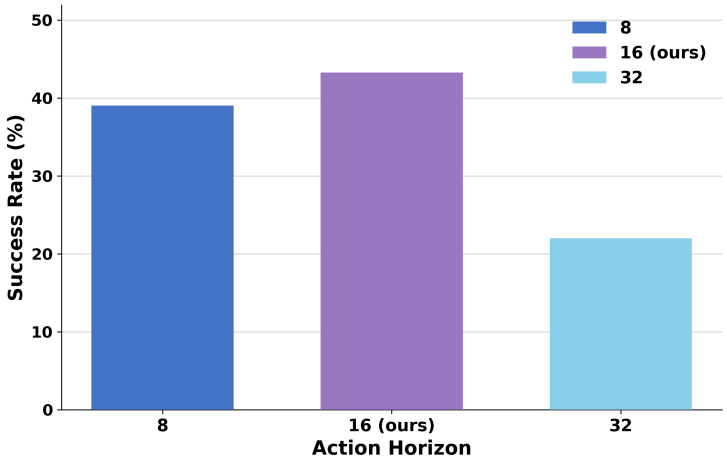


Fig. 9: Effect of changing action horizon. Our choice of action horizon 16 performs best overall computed across 3 tasks with 50 demos each.

ing critic because it yields a higher average success rate across the full task suite, indicating a more stable semantic representation for evaluating environmental transitions.

Effect of changing action horizon. We perform evaluation of our model changing the action horizon on 150 demos across 3 tasks. As shown in Fig. 9, we observe a clear non-monotonic effect: horizon 16 (ours) performs best while a shorter horizon 8 drops the performance and a longer horizon 32 drops the performance further. This suggests horizon 16 gives the best tradeoff between

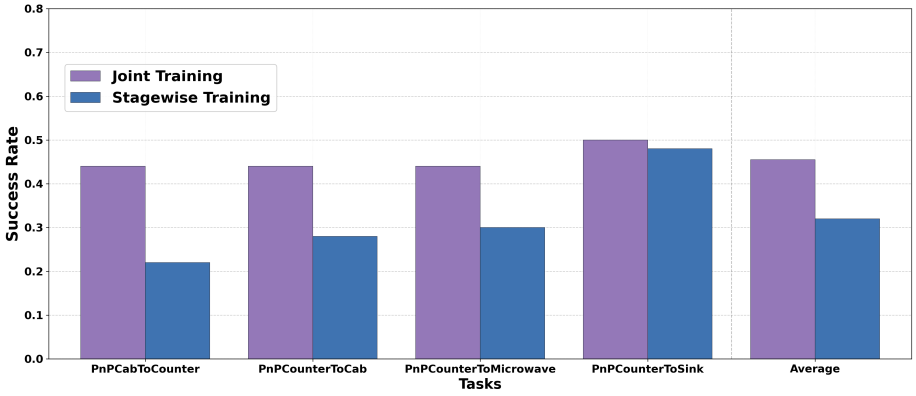


Fig. 10: Joint training vs Stage-wise training: Our proposed joint training setup shows enhanced task-wise success-rates compared to stage-wise training computed across 4 RoboCasa tasks. The Average scores (right) further strengthen our claim.

re-planning frequency and action commitment; very short horizons can make control too reactive, while very long horizons increase open-loop drift and error accumulation.

Joint Training vs. Stage-wise Training. As described in the main paper, RoboTALES adopts a coupled, single-stage joint training strategy in which the video generator and action policy are optimized simultaneously. This end-to-end design allows action-level gradients to flow back into the video generator's decoder layers, encouraging it to produce latent representations that are not only visually faithful and semantically aligned but also directly informative for downstream control. We verify this design choice by comparing the decoupled stage-wise training with our proposed joint training in Fig. 10. As is obvious from the plot, our proposed joint training pipeline promotes tighter co-adaptation between imagination and action, leading to overall improvements in action prediction.

Planner Decomposition Quality. To assess planning quality, we perform a blind pairwise evaluations on 50 randomly sampled task instructions using GPT-4o as a judge, comparing our planner-generated decompositions against a prompt-only baseline in Fig. 11. For each instruction, the decomposition is judged on following 5 criteria: correctness, completeness, ordering, executability, and non-redundancy. The left plot (pairwise winner rate) shows a clear preference for the planner output: it is selected as better in 60% of comparisons, versus 38% for prompt-only, with only 2% ties. The right plot (decomposition length) explains an important part of this gain: planner outputs are substantially shorter on average (2.80 steps vs 5.28), indicating less redundant and more compact task structure. Taken together, the two plots suggest that the planner is not just producing longer, more verbose plans to "game" the judge; instead, it generates concise decompositions that are judged better overall for task execution

quality. This supports the claim that planner structure improves decomposition usefulness and efficiency relative to direct prompt-only generation.

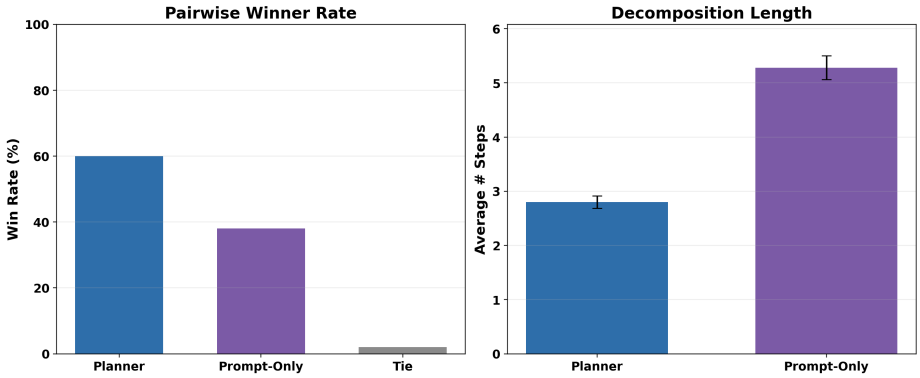


Fig. 11: LLM-judge evaluation of decomposition quality (N=50): planner decompositions outperform prompt-only (60% vs 38% wins, 2% ties) while using fewer steps (2.80 vs 5.28), indicating better plan efficiency and overall preference.

Retaining and Improving General Video Generation Capabilities. To validate whether our joint task-aligned training preserves the underlying generative video prior, we evaluate the original SVD model and our jointly trained video model on non-robotics real videos (from UCF101 dataset), using 200 matched conditioning samples and 30 denoising steps. We compare generated futures to ground-truth future clips in feature space. As shown in Fig. 12, our model is consistently closer to real-video dynamics: Fréchet distance decreases from 0.6715 to 0.2437 (63.7% lower), MMD-RBF decreases from 0.2419 to 0.0631 (73.9% lower), and per-sample cosine alignment to ground truth increases from 0.5955 to 0.8535 (+0.258 absolute, 43.3% relative). Distributional fidelity also improves: diversity-ratio-to-GT moves from 0.7201 (under-dispersed) to 1.0249 (near ideal), and covariance-trace ratio improves from 0.5418 to 1.0494, indicating much less variance collapse after control training. Overall, these results show that joint training does not collapse SVD into a narrow task-specific predictor; instead, it retains and improves general real-world video generation quality relative to the original SVD baseline.

A.3 Additional Qualitative Results.

We present extended qualitative comparisons and visualizations to further demonstrate the robustness of our framework. Figure 13 highlights our method's ability to maintain goal-directed behavior over long horizons compared to the baseline, while Figure 14 showcases successful execution across a diverse array of complex kitchen manipulation tasks.

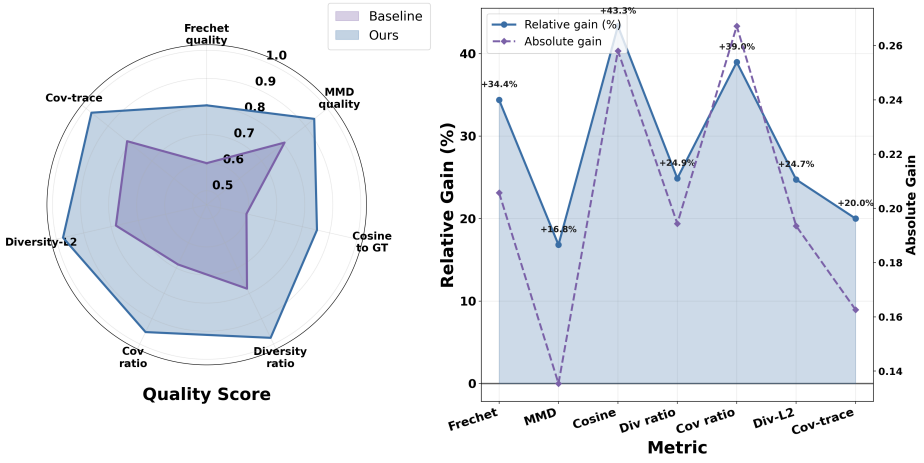


Fig. 12: Generative retention on UCF101. We compare the original SVD with our jointly trained video model using 200 matched conditioning samples from UCF101 dataset with 30 denoising steps each, evaluated against ground-truth future clips. The radar plot shows absolute quality profiles across alignment and distributional metrics, while the gain-area plot reports per-metric improvements (relative gain % with absolute-gain overlay). Our model improves all metrics, with the largest gains in Fréchet and MMD alignment, and substantially better diversity/covariance matching to the real-video distribution.

A.4 Details about Planner

To convert high-level RoboCasa task instructions into executable sequences for video rollout, we utilize `Gemini 2.5 Pro` as a offline task planner. The planner follows the *VLWM-style* abstraction, ensuring each step represents a meaningful and observable transition in the environment state. We impose a constraint on the planner to generate between 2 and 5 action steps per instruction. This range ensures that atomic tasks are sufficiently decomposed for the policy to learn meaningful sub-goals, while complex composite tasks are compressed into high-level transitions that remain manageable for the video rollout horizon.

To ensure computational efficiency during training and consistency across inference rollouts, we pre-generate these decompositions for the entire dataset. These steps are stored in a persistent local cache, indexed by a unique hash of the instruction and model version. This avoids redundant LLM API calls and ensures that the downstream policy is always trained on a stable set of sub-goals.

Below, we provide the system prompt employed to obtain the decompositions of the task instruction.

System Prompt

You are a planner that converts ONE RoboCasa task instruction into a short, ordered list of ACTIONS for video rollout.

Planning principle
- Follow VLWM-style abstraction: each step is an ACTION that causes a meaningful, observable world-state change. Keep state implicit.

Hard constraints
- Always return between 2 and 5 actions inclusive. - Output ONLY a strict JSON object with this shape: "steps": ["<action 1>", "<action 2>", "..."]
- Each action is a concise, imperative verb phrase (<= 12 words). - Preserve explicit ordering from the instruction.

Granularity rules
- Composite instructions: 3-5 actions capturing key transitions.
- Atomic instructions: Split into two micro-actions (e.g., reach/align).

Instruction Prompt & In-Context Examples

Task instruction:
{{SYSTEM_PROMPT}}

Return ONLY a strict JSON object of 2-5 concise action strings: "steps": ["...", "..."]

In-context examples:
Instruction: Open the microwave, put the bowl inside, close it, then start it.
Expected:
"steps": [
"open the microwave door",
"place the bowl inside the microwave",
"close the microwave door",
"press the start button"
]

Table 4 showcases how the `Gemini 2.5 Pro` planner translates high-level semantic instructions into sequential, observable sub-goals while adhering to a 2–5 step granularity constraint. This cached decomposition enables consistent, multi-stage task execution by grounding complex human requests into a manageable sequence of intermediate world-state transitions.

Table 4: Examples of task instructions and their corresponding planner-generated step decompositions for the Libero10 dataset

Original Task Instruction	**Decomposed Steps**
Turn on the stove and put the moka pot on it	1. Turn on the stove 2. Place the moka pot on the stove burner
Put the black bowl in the bottom drawer of the cabinet and close it	1. Pick up the black bowl 2. Open the cabinet bottom drawer 3. Place the black bowl in the drawer 4. Close the cabinet bottom drawer
Put the yellow and white mug in the microwave and close it	1. Put the yellow and white mug in the microwave 2. Close the microwave door
Put both moka pots on the stove	1. Place the first moka pot on the stove 2. Place the second moka pot on the stove
Put both the alphabet soup and the cream cheese box in the basket	1. Pick up the alphabet soup 2. Place the alphabet soup in the basket 3. Pick up the cream cheese box 4. Place the cream cheese box in the basket
Put both the alphabet soup and the tomato sauce in the basket	1. Place the alphabet soup in the basket 2. Place the tomato sauce in the basket
Put both the cream cheese box and the butter in the basket	1. Place the cream cheese box in the basket 2. Place the butter in the basket
Put the white mug on the left plate and put the yellow and white mug on the right plate	1. Put the white mug on the left plate 2. Put the yellow and white mug on the right plate
Put the white mug on the plate and put the chocolate pudding to the right of the plate	1. Put the white mug on the plate 2. Place the chocolate pudding to the right of the plate
Pick up the book and place it in the back compartment of the caddy	1. Pick up the book 2. Place the book in the caddy back compartment

A.5 Limitations

While RoboTALES demonstrates strong performance across diverse manipulation tasks, it is worth noting certain limitations. First, our VLM Critic is frozen throughout training and provides a relatively coarse, task-level reward signal. A learnable reward model that co-adapts with the video generator and action policy could provide richer, more nuanced feedback – particularly for contact-rich tasks where subtle differences in grasp quality or object pose are difficult to assess from pixel-level text-image similarity alone. Second, our framework inherits the computational cost of video diffusion models. The DDPO-based policy optimization requires multiple denoising rollouts per training step to estimate

gradients, making training expensive. Third, our setup purely simulation-based in current state. We are actively working on addressing these limitations as part of our ongoing and future work.

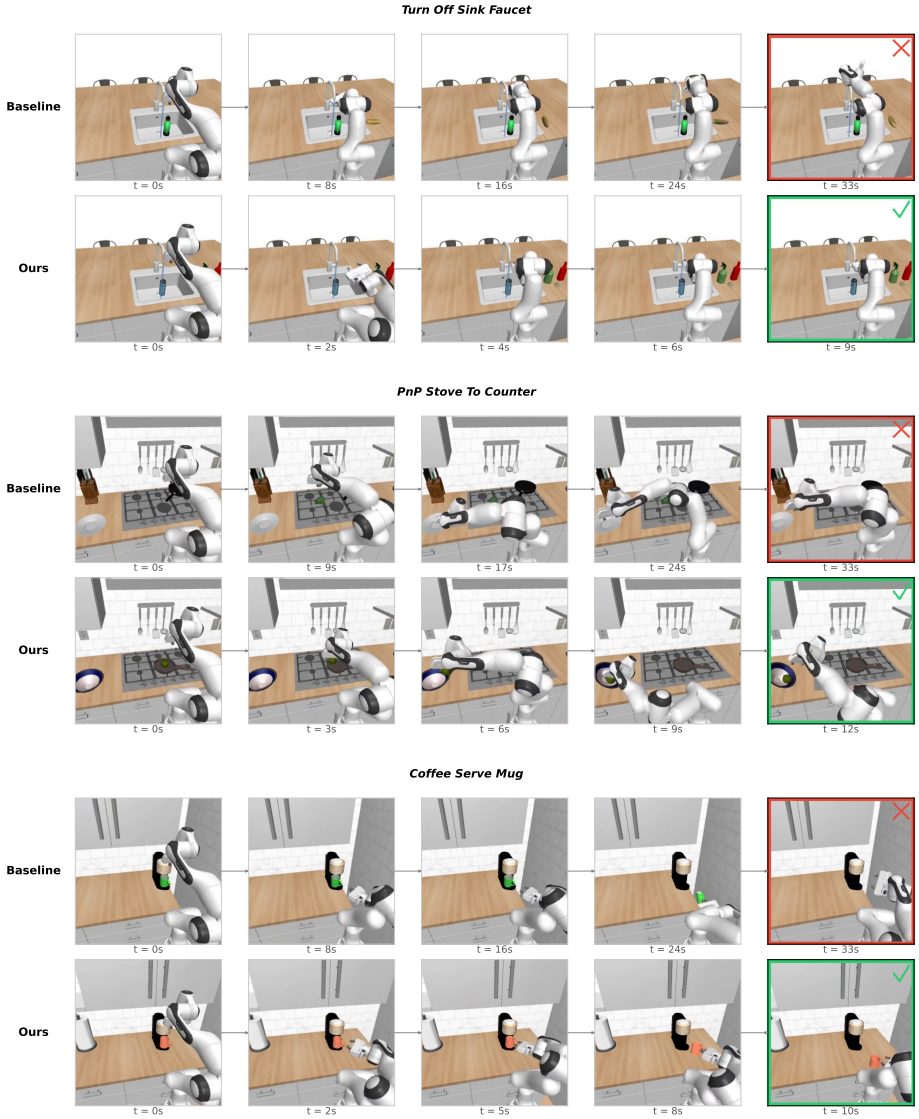


Fig. 13: Comparing our method to the baseline across different tasks . We show how our model performs against the baseline on tasks like turning off a faucet, moving items from a stove to a counter, and serving coffee. In each case, the baseline (top rows) loses track of the goal over time, leading to distorted movements or failed actions. Our method (bottom rows) uses high-level planning to stay on track, completing the tasks accurately and more quickly.

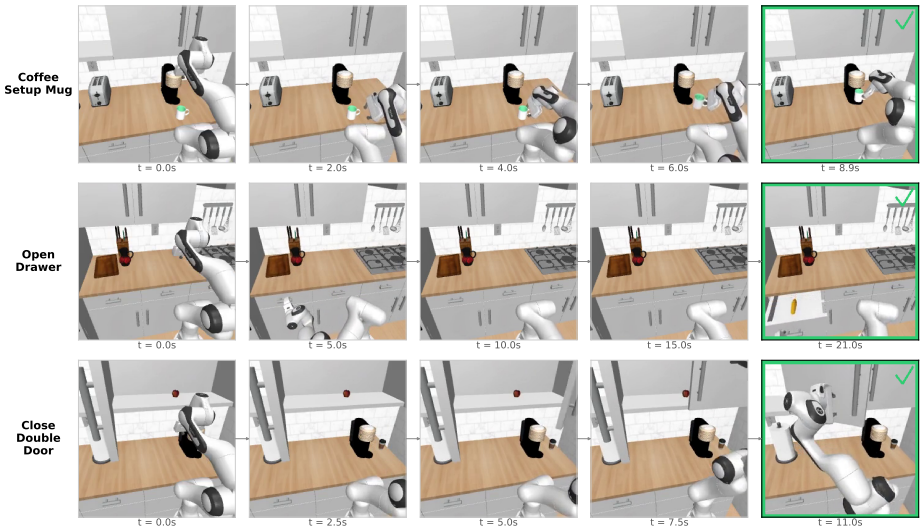


Fig. 14: Success of our method in various kitchen scenarios . This figure demonstrates our model's ability to handle different complex tasks, such as setting up a coffee mug, opening drawers, and closing double doors. Our model consistently creates smooth, goal-oriented movements while managing multiple camera views and complex objects. These successful results show that combining smart reasoning with a video generator leads to more reliable robot performance in real-world settings