arXiv:2603.23149v2 [cs.AI] 28 Aug 2026

Describe-Then-Act: Proactive Agent Steering via Distilled Language-Action World Models

Massimiliano Pappa[1,2⋆], Luca Romani[1*], Valentino Sacco[1*], Alessio Palma[1], Stéphane Lathuilière[2], Fabio Galasso[1], Xavier Alameda-Pineda[2], Indro Spinelli[1]

[1] Sapienza University of Rome, Italy
[2] Inria, Univ. Grenoble Alpes, CNRS, LJK

Abstract. Deploying safety-critical agents requires anticipating the consequences of actions before they are executed. While world models offer a paradigm for this proactive foresight, current approaches relying on visual simulation incur prohibitive latencies, often exceeding several seconds per step. In this work, we challenge the assumption that visual processing is necessary for failure prevention. We show that a trained policy's latent state, combined with its planned actions, already encodes sufficient information to anticipate action outcomes, making visual simulation redundant for failure prevention. To this end, we introduce **DILLO** (**DI**sti**LL**ed Language-Acti**O**n World Model), a fast steering layer that shifts the paradigm from "simulate-then-act" to "describe-then-act." DILLO is trained via cross-modal distillation, where a privileged Vision Language Model teacher annotates offline trajectories and a latent-conditioned Large Language Model student learns to predict semantic outcomes. This creates a text-only inference path, bypassing heavy visual generation entirely, achieving a 14× speedup over baselines. Experiments on MetaWorld and LIBERO demonstrate that DILLO produces high-fidelity descriptions of the next state and is able to steer the policy, improving episode success rate by up to 15 pp and 9.3 pp on average across tasks. Code is available at `github.com/MaxPappa/DILLO`.

1 Introduction

AI-driven agents are increasingly deployed in settings that demand high reliability, from robotic manipulation to autonomous navigation. However, a critical gap remains: these agents typically operate as "black boxes", selecting and executing actions without explicitly anticipating the consequences. In classical control theory, the standard solution is Model Predictive Control (MPC) [6]: simulate the system forward and commit only to actions whose predicted outcomes are acceptable. The same principle underlies model-based reinforcement learning [9,26], where world models forecast future states to steer policies.

However, deploying world models in the tight control loops of dynamic agents presents a stark dilemma between foresight and latency. Current approaches generally fall into two categories. The first, post-hoc analysis [2, 21, 25], operates

⋆ Equal contribution

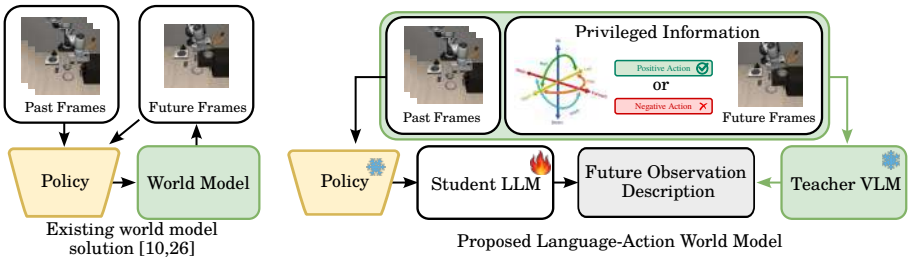

Fig. 1: (*left*) Standard approaches rely on heavy visual world models to simulate future outcomes using expensive rendering. In contrast, (*right*) DILLO avoids this visual dependency entirely. By distilling the foresight of a simulator-privileged Teacher VLM into a fast Language-Action World Model, DILLO predicts future outcome descriptions and success verdicts directly from policy latents (z_t) and action chunks ($a_{t:t+k}$). This shift from "simulate-then-act" to "describe-then-act" enables fast proactive steering on consumer hardware.

as a reactive wrapper, diagnosing failures only after they occur. While computationally inexpensive, this approach is fundamentally incapable of prevention. The second paradigm, proactive visual simulation, utilizes heavyweight world models to generate high-dimensional future states (images/videos or their latent representation) [13,16]. While effective at anticipating risks, these "visual" world models incur prohibitive computational overhead. For instance, recent steering methods report inference latencies of nearly 4 seconds per decision [31] on an enterprise RTX A6000 with 48GB of VRAM, rendering them impractical for real-time control on an embodied agent. We posit that for real-time near-future failure prevention, simulating the visual world is redundant. A policy's internal latent representation is explicitly trained to retain task-critical features: object geometry, relative distances, and contact dynamics. If this representation already contains the information needed to predict whether an action will succeed, then why pay the cost of re-rendering pixels? We call this the *Latent Sufficiency Hypothesis*.

To empirically verify this hypothesis, we propose **DILLO**(**DI**sti**LL**ed Language-Acti**O**n World Model), see Fig. 1, a real-time reliability layer that shifts the paradigm from "simulate-then-act" to "describe-then-act." DILLO is trained via cross-modal knowledge distillation: a privileged Vision-Language-Model (VLM) Teacher annotates offline trajectories with semantic outcomes, having access to the simulation environment. The Student (DILLO) then learns to predict these outcomes directly from the policy's compact latent state and candidate action chunk, without accessing raw visual observations. Architecturally, we leverage the reasoning capacity of Small Language Models, mapping projected latents directly to the language decoder. This vision-free design enables DILLO to perform fast inference, effectively decoding the agent's internal "state of mind" into human-readable foresight.

Specifically, DILLO generates two outputs: **(i) a natural language behavior preview** (d) that describes the expected physical interaction (*e.g.*, *"The gripper goes left and forward, starting to approach the ball"*), and **(ii) a binary verdict** ($c \in \{\text{POSITIVE}, \text{NEGATIVE}\}$). A *Negative* action is one causing stagnation, erratic movement, or progression toward failure; a *Positive* action advances the task. This enables a dual-purpose mechanism: the verdict allows the agent to autonomously reject negative actions via rejection sampling, while the description provides interpretable information for human supervisors.

Our main contributions are:

- We propose the *Latent Sufficiency Hypothesis* and demonstrate empirically that a policy's latent state coupled with the sampled actions is a sufficient statistic for predicting failure-critical outcomes, making visual simulation redundant for proactive failure prevention.
- We introduce DILLO, a Distilled Language-Action World Model that distills a VLM teacher into a latent-conditioned LLM Student, shifting the paradigm from "simulate-then-act" to "describe-then-act." This yields a $\sim$14$\times$ speedup over visual baselines [31], enabling a full correction loop in **0.26 s** on consumer hardware.
- DILLO's latent-based descriptions match or exceed the fidelity of vision-based baselines, including an oracle with access to future observations. Furthermore, we demonstrate effective steering on single- and multi-task policies across MetaWorld and LIBERO, improving episode success rate by up to **15 pp** and **9.3 pp** on average across tasks, achieving a verdict classification accuracy of 91.4% without access to any visual observation at inference time.

2 Related Work

We propose a real-time, language-based failure detection layer within the context of three dominant paradigms: reactive failure analysis, proactive world modeling, and action-language grounding.

Reactive Post-Hoc Analysis. The most common approach to agent interpretability follows an "act-then-describe" paradigm, analyzing failures only *after* they have occurred. Methods such as Aha [2] and REFLECT [21] leverage Vision-Language Models (VLMs) to process offline trajectory data, generating human-readable summaries of past errors. Other works treat VLMs as behavioral critics [8] or as a means to ground historical robotic experiences within a linguistic framework [28]. While runtime monitors [1, 25] operate closer to the point of execution, they are primarily designed to detect anomalies or consistency violations post-occurrence. Similarly, SAFE [7] analyzes the latent states of Vision-Language-Action models (VLAs) to predict task success or failure in a task-agnostic manner. Although these methods are indispensable for data curation and debugging, they remain fundamentally reactive, offering a "failure autopsy" rather than a preemptive intervention. In physical environments where robotic actions are often irreversible, such retrospective detection cannot substitute for robust, real-time prevention.

Proactive World Models for Control. To prevent failures, an agent must anticipate them. The concept of *World Models* champions this proactive paradigm, evolving from early visual foresight [3–5] to powerful latent dynamics models like Dreamer [11–13]. While these models effectively hallucinate future rewards for policy learning, they typically output latent vectors that are opaque to human supervisors. Recently, this paradigm was extended to *semantic* policy steering by Forewarn [31], which utilizes VLMs to narrate and evaluate simulated latent states. While validating the utility of linguistic previews, such "VLM-in-the-loop" methods remain computationally prohibitive. Relying on visual encoders to process predicted latent vectors and their subsequent evaluation incurs high latency (*e.g.*, ∼3.7 seconds per decision [31]), rendering them impractical for the tight control loops of dynamic agents. DILLO takes a different approach: rather than performing expensive, future-state simulation, we distill semantic foresight into a fast, latent-conditioned predictor enabling real-time safety without the associated simulation and evaluation overhead.

Action-Language Grounding. Finally, we distinguish our work from rationale-driven Explainable AI (XAI) [22,24,30]. Methods like Embodied Chain-of-Thought (ECoT) [34] use language to expose an agent's *internal logic* (*e.g.*, "I need the cup, so I will move my arm"). In contrast, DILLO is outcome-driven; it does not explain the agent's reasoning but rather predicts the external physical consequence of an action (*e.g.*, "This action will cause the gripper to move backward, increasing its distance from the cup"). This distinction allows DILLO to function as a controller-agnostic safety module that can be layered onto any policy to provide a fast, interpretable sanity check.

3 Distilled Language Action World Model

We introduce **DILLO** (**DI**sti**LL**ed Language-Acti**O**n World Model), a framework for real-time failure prevention that decouples semantic anticipation from visual simulation. We formulate the problem as cross-modal knowledge distillation, where a lightweight student learns to approximate the semantic foresight of a privileged teacher solely from the policy's internal representations and predicted actions, see Fig. 2.

3.1 Semantic Anticipation

Consider a robotic agent operating in a Partially Observable Markov Decision Process (POMDP). At decision step t, the agent receives an observation $o_t \in \mathcal{O}$ and processes it via the policy's encoder $\mathcal{E}_\pi$ to extract a compact latent state:

$$z_t = \mathcal{E}_\pi(o_t) \in \mathbb{R}^{D_z} \tag{1}$$

Conditioned on this latent state, the policy proposes a sequence of k future actions, the action chunk [20] $a_{t:t+k}$. Our objective is to learn a **Language-Action World Model** modeling a distribution $P_\theta(y_{t+k} \mid z_t, a_{t:t+k})$ that predicts the semantic outcome y_{t+k} directly from these internal representations,

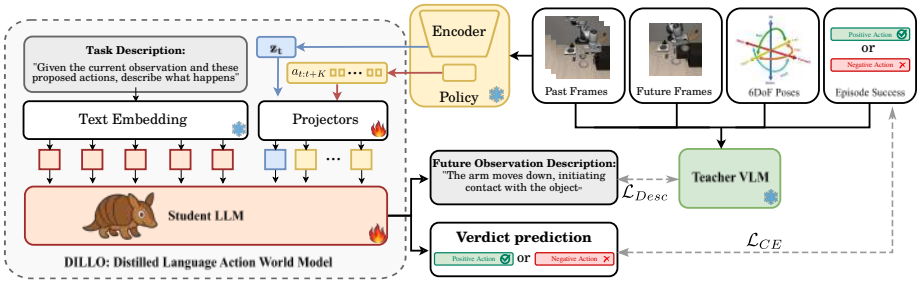

Fig. 2: DILLO Training Pipeline. We distill the foresight of a VLM teacher. The pipeline bypasses visual processing by projecting the fixed policy's internal latents (z_t) and action chunks ($a_{t:t+k}$) directly into the LLM's embedding space.

without accessing the raw visual observation. Standard world models use the state representation, often an RGB image, to predict high-dimensional future observations $\hat{o}_{t+k}$ (pixel space).

The semantic outcome y_{t+k}, consists of a natural language description d_{t+k} (*e.g.*, physical dynamics, contacts) and a verdict $c_{t+k} \in \{\text{POSITIVE}, \text{NEGATIVE}\}$ indicating whether the action chunk advances or hinders the task. The distribution of this outcome is conditioned on the environment dynamics:

$$P_{env}(y_{t+k} \mid o_t, a_{t:t+k}) \tag{2}$$

This is estimated by a teacher distribution $P_{\mathcal{T}}(y_{t+k} \mid o_{t:t+k})$ that has privileged access to a world model or simulator:

$$P_{env}(o_{t+k} \mid o_{t:t+k-1}, a_{t:t+k}) \tag{3}$$

The teacher receives the full ground-truth visual history from the simulated environment, along with 6DoF poses for both the object and the gripper.

3.2 The Latent Sufficiency Hypothesis

Evaluating $P_{\mathcal{T}}$ is computationally prohibitive for real-time control. Our objective is to learn a fast estimator, the student distribution $P_\theta(y_{t+k} \mid z_t, a_{t:t+k})$, conditioned only on the policy's latent state and planned actions. The feasibility of this approach rests on the following hypothesis:

Hypothesis on Latent Sufficiency. *For the task of semantic world modeling, the mutual information between the policy latent-action tuple* $(z_t, a_{t:t+k})$ *and the future outcome* y_{t+k} *approximates the mutual information between the full observation history* $o_{t:t+k}$ *and* y_{t+k}*:*

$$I\big(o_{t:t+k};\, y_{t+k}\big) \;\approx\; I\big((z_t,\, a_{t:t+k});\, y_{t+k}\big). \tag{4}$$

Whether trained via Reinforcement Learning (*e.g.*, SAC [10]) to maximize task reward, or Imitation Learning (*e.g.*, ACT [35]) to reconstruct expert action

intent, the encoder $\mathcal{E}_\pi$ must extract and retain interaction-critical representations, such as object geometry, relative distances, and contact dynamics. We claim that $(z_t, a_{t:t+k})$, derived solely from o_t, carries as much predictive information about y_{t+k} as the full observation history seen by the teacher, making z_t a sufficient statistic for y_{t+k} and rendering generating and re-encoding raw pixels unnecessary. This sufficiency extends to failing policies. When a policy encounters an out-of-distribution state, the resulting latent z_t captures the "context of failure" (*e.g.*, uncertainty or feature mismatch), and DILLO learns to map these latents to Negative verdicts.

By minimizing the Kullback–Leibler divergence between the teacher and student distributions [14]:

$$\min_{\theta} \; D_{KL}(P_{\mathcal{T}}(y_{t+k} \mid o_{t:t+k}) \parallel P_\theta(y_{t+k} \mid z_t, a_{t:t+k})) \tag{5}$$

we distill the teacher's future-aware foresight into a Language-Action World Model that translates the policy's existing manifold into natural language descriptions. This reduces to a cross-entropy minimization over $(z_t, a_{t:t+k})$. DILLO thereby predicts outcomes over the horizon $t \to t+k$ with negligible computational overhead.

3.3 Models

Teacher. We instantiate the teacher $\mathcal{T}$ as a VLM with access to privileged information from the simulator state, eliminating the depth ambiguity and occlusion issues inherent to raw vision. In particular, the model has access to:

1. **Visual Context:** The sequence of RGB observations corresponding to the execution of the action chunk.
2. **Geometric Deltas:** Directional changes and 6DoF poses of the end-effector and task-relevant objects, grounding the VLM in the physics of the scene and mitigating the spatial hallucinations common in vision-only models.
3. **Episode Outcome Signal:** A binary indicator of whether the specific action chunk led to successful task completion.

In physical deployment, this privileged information can be substituted by state-of-the-art 6D pose estimation algorithms [29].

To construct the distillation dataset, the privileged teacher model processes the ground-truth state information alongside the natural language task description and the target goal configuration. We segment policy rollouts at the state-action level. For each transition, the teacher generates a natural language description $d_{\mathcal{T}}$ and assigns a binary verdict $c_{\mathcal{T}} \in \{\textsc{Positive}, \textsc{Negative}\}$. A Negative verdict explicitly penalizes trajectories that exhibit stagnation, erratic actuation, or behaviors otherwise detrimental to task progression. Finally, we pair these high-level semantic labels with the base policy's internal latent state z_t and the predicted action chunk $a_{t:t+k}$. This yields the final distillation tuple:

$$\tau = (z_t, a_{t:t+k}, d_{\mathcal{T}}, c_{\mathcal{T}})$$

Student (DILLO). The student f_θ parameterizes P_θ using the Gemma [27] family of open weights. We implement two variants to validate scalability:

- **DILLO-1B:** Uses the standard language-only Gemma-1B-it LLM backbone.
- **DILLO-4B:** Unlike standard VLM inference, which relies on a 417M-parameter SigLIP encoder, our model uses only the large language model of the Gemma-VLM-4B-it architecture. This ensures that DILLO-4B benefits from the reasoning capacity of a VLM while incurring zero visual overhead.

To map the continuous control space to the language space, we employ two learnable projectors:

- **Latent Projector** $P_z : \mathbb{R}^{D_z} \to \mathbb{R}^{D_{emb}}$, mapping the frozen policy embedding to the LLM input space.
- **Action Projector** $P_a : \mathbb{R}^{D_a} \to \mathbb{R}^{D_{emb}}$, mapping each action chunk $a_{t:t+k}$ to an action vector that is then fed to the LLM.

The input sequence to the Transformer is constructed as:

$$X_{in} = \texttt{[TASK PROMPT]} \oplus P_z(z_t) \oplus P_a(a_{t:t+k}) \tag{6}$$

Here, X_{in} contains no tokens derived from o_t, ensuring independence from the visual encoder during inference.

3.4 Training

End-to-end mapping of continuous control latents into discrete LLM tokens is prone to modality collapse. To stably ground the LLM in the policy's physical manifold, we employ a progressive three-stage curriculum:

Stage 1: Projector Alignment. To bridge the dimensional gap without destabilizing pre-trained weights, we freeze the LLM backbone θ_{LLM} and optimize only the linear projectors $\{P_z, P_a\}$ [19]. We use standard autoregressive next-token prediction on the teacher's descriptions to ensure the projected latents act as valid prompt embeddings.

Stage 2: Description Distillation. To establish a dense semantic prior, we apply Low-Rank Adaptation (LoRA) [15] to the LLM. We jointly optimize the LoRA parameters and projectors to reconstruct the teacher's natural language rationale $d_\mathcal{T}$:

$$\mathcal{L}_{Desc} = -\mathbb{E}_\mathcal{D}\left[\sum_{i=1}^{|d|} \log p_\theta(d_i | d_{<i}, z_t, a_{t:t+k})\right] \tag{7}$$

Stage 3: Verdict Optimization. Finally, we introduce the binary task constraint. The model generates the rationale followed by a verdict token $\hat{c} \in \{\textsc{Positive}, \textsc{Negative}\}$. We minimize the joint objective:

$$\mathcal{L}_{Total} = \mathcal{L}_{Desc} + \lambda \cdot \mathcal{L}_{CE}(c_\mathcal{T}, \hat{c}) \tag{8}$$

where $\mathcal{L}_{CE}$ is the cross-entropy against the ground-truth verdict $c_\mathcal{T}$, ensuring the natural language descriptions are grounded with the task execution.

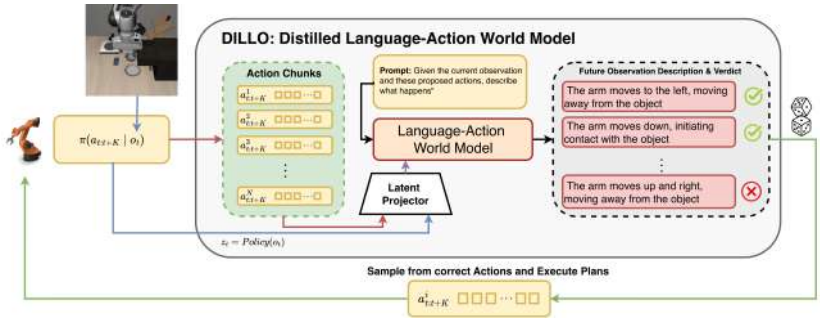

Fig. 3: Proactive Policy Steering via Latent Rejection Sampling. At inference time, DILLO acts as a high-speed safety filter. The policy proposes N candidate action chunks ($a^i_{t:t+k}$). Instead of re-rendering future images for each candidate, DILLO projects the policy latent (z_t) directly into the Language-Action World Model. The model predicts a semantic description and verdict for each chunk. Negative trajectories (red $\times$) are rejected, and the first verified positive plan (green ✓) is executed.

3.5 Proactive Steering via Latent Rejection Sampling

DILLO's distilled verdict $\hat{c}$ enables it to function as a controller-agnostic, zero-overhead safety filter at inference time. Rather than executing the first proposed action chunk, the agent uses DILLO to screen a batch of N candidate plans before committing to any physical action, as illustrated in Fig. 3.

Inference Protocol. At each decision step t, the base policy encodes the current observation into a shared latent state $z_t = \mathcal{E}_\pi(o_t)$, computed *once* regardless of how many candidates are subsequently evaluated. The policy then proposes a budget of N candidate action chunks by sampling independently from its action distribution:

$$\left\{a^i_{t:t+k}\right\}_{i=1}^{N} \sim \pi(\cdot \mid z_t) \tag{9}$$

DILLO evaluates all N candidates in a single batched forward pass. For each candidate, the shared latent z_t and an encoded projection of the proposed action sequence are jointly fed into the model:

$$y^i_{t+k} = \left(d^i_{t+k},\, c^i_{t+k}\right) = f_\theta\left(P_z(z_t),\, P_a(a^i_{t:t+k})\right), \quad i = 1, ...N \tag{10}$$

The agent then selects any candidate that receives a POSITIVE verdict. If no POSITIVE verdict is found within the budget, the agent falls back to executing a candidate from the initial proposal, ensuring the control loop is never stalled. In our experiments we set N=5. Notably, the framework naturally supports multiple resampling rounds: the policy can draw and evaluate successive batches of N candidates up to R rounds (where R is a configurable budget), allowing up to $R \cdot N$ total evaluations before fallback at the cost of added latency (see Sec. 4.4).

4 Experiments

Our evaluation validates two central claims. First, *latent sufficiency*: does DILLO, conditioned solely on latent states (z_t), produce semantic descriptions (y_{t+k}) that faithfully match ground-truth physical outcomes (o_{t+k})? Second, *utility*: does DILLO's verdict effectively steer a base policy toward higher episode success rates? We conduct experiments in **MetaWorld** [33] (`Soccer`, `Sweep-Into`, `Drawer-Open`) and **LIBERO** [18] (`Goal`, `Object`, `Spatial`, `10`, `90`).

4.1 Constructing the Distillation Dataset

To train DILLO, we collect distillation datasets by rolling out pre-trained policies across two distinct algorithmic paradigms, capturing the latent transitions inherent to both specialized and generalist behaviors.

Single-Task Reinforcement Learning (MetaWorld). We roll out SAC-trained policies [10] across three manipulation tasks (Soccer, Sweep-Into, Drawer-Open). Each policy is specialized for a single task; correspondingly, a separate DILLO model is distilled per task. The collected trajectories consist of purely observational latents, $z_t = \mathcal{E}(o_t)$, where the representation is driven entirely by the RL reward signal, with no language conditioning.

Multi-Task Imitation Learning (LIBERO). We roll out action-chunking, imitation-learned policies [35] across five task suites (LIBERO-Goal, LIBERO-Object, LIBERO-Spatial, LIBERO-10, LIBERO-90). Here, a single policy must generalize over a shared set of language conditioned tasks, and the distillation dataset reflects this diversity. A single DILLO model is therefore distilled from trajectories spanning heterogeneous, language-specified goals, testing whether the latent space of a generalist policy retains sufficient structure for cross-task outcome prediction.

Failures. A reliable predictive model must identify failure modes across all levels of agent capability. A classifier trained solely on expert successes and random failures will fail to generalize to subtle execution errors. To guarantee a comprehensive distribution of both catastrophic and marginal failures, we curate our dataset using two complementary strategies:

- **Spectrum of Competency Sampling.** Rather than relying exclusively on converged experts, we sample rollouts from policy checkpoints saved at distinct performance thresholds. We incorporate low-competency models (20% success) to capture erratic exploratory behaviors, mid-competency models (50% success) to provide nuanced "near-miss" examples [1] that are difficult to classify, and high-competency models (80% success) to establish ground-truth task alignment while offering "hard negatives" through their rare failures.
- **Perception Noise Injection.** During the data collection step, we inject Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$ into the observation o_t before querying the policy $\pi(a_{t:t+k}|o_t + \epsilon)$. This represents a standard proxy for sensor uncertainty used in Sim-to-Real domain randomization. The perturbation causes

the agent to misinterpret its state, inducing trajectory drift and compounding execution errors. Crucially, this generates failures that originate from a competent action manifold rather than random initialization, systematically mimicking the covariate shift commonly observed during physical deployment.

4.2 Metrics

To support the hypothesis on "latent sufficiency", we must quantify the alignment between the text predicted from latents (z_t) and the actual physical outcome observed in the simulator (o_{t+k}). We compare DILLO against baselines backed by Gemma-3-1B and Gemma-3-4B:

- **Zero-Shot (ZS):** The model receives the current observation o_t and the task description.
- **Few-Shot (FS):** The model receives o_t along with 3 in-context examples of (o_t, y_{t+k}) pairs.
- **Few-Shot Captioning (CAP):** The model receives the current observation o_t, the *future* observation o_{t+k}, and 3 in-context examples of (o_t,o_{t+k}, y_{t+k}) tuples. It does not need to predict the future; it can caption what actually happened therefore it should be seen as an oracle.

To evaluate baselines and DILLO, we rely on two fidelity metrics:

Text-to-Observation Fidelity (T2O). To quantify low-level dynamic accuracy without the noise of linguistic ambiguity, we map both physical state transitions and generated descriptions into a shared space of canonical directional labels. A deterministic function, $f_{extract}(o_t, o_{t+k})$, computes the true displacement for the gripper and objects, as well as relative relational changes (*e.g.*, *approaching* vs. *receding*) across three orthogonal axes. Axes with negligible movement are assigned a "static" label. Simultaneously, a parser $g_{parse}(d)$ projects the natural language prediction into the same canonical set via robust synonym mapping. The final T2O score is computed as the accuracy of matched labels across all dynamic variables, serving as a direct proxy for physical grounding (see supplementary material for the definition of g_{parse}).

Reasoning-Based Semantic Score. While T2O captures dynamics, it may miss nuanced semantic details. We employ a Judge LLM (Qwen2.5-32B-Instruct [32]) to score the semantic alignment between DILLO's prediction d and the ground-truth reference $d_{\mathcal{T}}$. The judge decomposes the reference into a set of atomic facts $F = \{f_i\}_{i=1}^{|F|}$. Subsequently, it evaluates the extent to which the candidate description d entails each fact, assigning a support score $m_i \in \{0, 0.5, 1\}$ (contradiction, partial implication, and full support). We define the final metric as the mean fact recall:

$$\text{Score}(d_{t+k}|d_{\mathcal{T}}) = \frac{1}{|F|}\sum_{i=1}^{|F|} m_i \tag{11}$$

Table 1: Fidelity Results. We measure the alignment between the predicted description and the actual simulator state. We compare DILLO against LLM and VLM baselines (Few-Shot, Zero-Shot, and CAP) across two model scales (1B and 4B parameters) on both the MetaWorld and LIBERO suites. DILLO consistently achieves high fidelity and outperforms or matches baselines, validating that the latent state is a sufficient statistic for semantic prediction.

		1B Parameter Models				4B Parameter Models			
Environment	**Task**	**DILLO-1B**	FS	ZS	CAP	**DILLO-4B**	FS	ZS	CAP
MetaWorld	Soccer	**0.518**	0.495	0.398	0.477	0.586	0.501	0.540	**0.626**
	Sweep-Into	**0.562**	0.329	0.528	0.271	0.554	0.630	0.687	**0.725**
	Drawer-Open	**0.706**	0.626	0.419	0.612	**0.705**	0.555	0.553	0.609
LIBERO	Goal	**0.727**	0.262	0.287	0.723	0.710	0.422	0.241	**0.905**
	Object	0.622	0.388	0.204	**0.820**	0.587	0.546	0.225	**0.877**
	Spatial	0.668	0.546	0.341	**0.794**	0.656	0.563	0.194	**0.902**
	10	**0.648**	0.214	0.199	0.630	0.686	0.449	0.265	**0.867**
	90	**0.708**	0.258	0.206	0.640	0.650	0.278	0.258	**0.713**

Table 2: Reasoning Based LLM Score. We compare DILLO against VLM baselines (Few-Shot, Zero-Shot, and CAP) across two model scales (1B and 4B parameters). DILLO consistently outperforms standard baselines in semantic reasoning, even approaching or exceeding the CAP in the 1B regime.

		1B Parameter Models				4B Parameter Models			
Environment	**Task**	**DILLO-1B**	FS	ZS	CAP	**DILLO-4B**	FS	ZS	CAP
MetaWorld	Soccer	**0.471**	0.301	0.204	0.277	**0.504**	0.361	0.328	0.425
	Sweep-Into	**0.592**	0.372	0.284	0.388	**0.607**	0.391	0.381	0.364
	Drawer-Open	**0.655**	0.510	0.189	0.427	**0.628**	0.497	0.212	0.252
LIBERO	Goal	**0.834**	0.259	0.271	0.686	**0.830**	0.487	0.278	0.816
	Object	**0.803**	0.380	0.208	0.668	**0.723**	0.519	0.243	0.684
	Spatial	**0.782**	0.448	0.323	0.684	0.758	0.545	0.282	**0.792**
	10	**0.822**	0.227	0.205	0.611	**0.797**	0.493	0.282	0.777
	90	**0.708**	0.258	0.206	0.640	0.650	0.278	0.258	**0.713**

This metric offers a robust, interpretable measure of semantic coverage beyond simple lexical overlap metrics like BLEU [23] or ROUGE [17].

Episode Success Rate (ES). We measure the percentage of episodes where the agent completes the task goal. We report the success rate of the DILLO-Steered Policy versus the Base Policy (no steering). A high ES for DILLO confirms that the distilled verdict $\hat{c}$ is a reliable signal for filtering negative actions.

4.3 Validating Latent Sufficiency

We validate the Latent Sufficiency Hypothesis using the T2O fidelity and LLM-judged semantic scores reported in Tables 1 and 2. The results reveal three consistent trends across both benchmarks.

Latent beats pixels. The "blind" DILLO-4B model consistently and significantly outperforms the pixel-based Few-Shot and Zero-Shot baselines on both

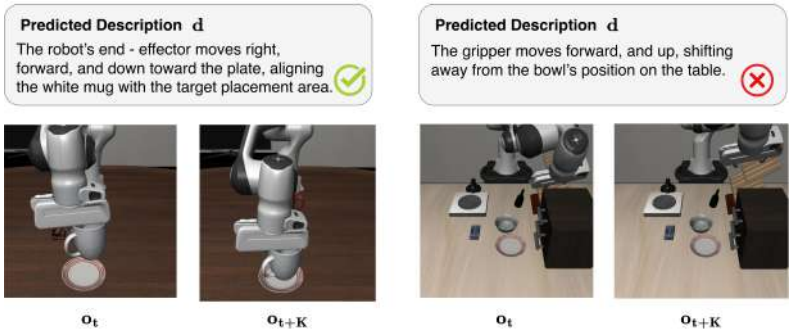

Fig. 4: Qualitative validation of DILLO's foresight for 'Positive' (left) and 'Negative' (right) outcomes. DILLO generates the predicted description (d) and verdict (check/cross) using only the policy's latent state (z_t) and the proposed action chunk. The initial observation (o_t) and ground-truth future (o_{t+K}) are shown purely for visual comparison, confirming the high fidelity of the latent-based prediction.

MetaWorld and LIBERO, confirming that the policy latent z_t is a more informative predictor of future dynamics than the current visual observation o_t.

Latent beats the future image. Remarkably in Table 1, DILLO-4B's T2O score also exceeds the CAP baseline on `Drawer-Open` (**0.705** vs. **0.609**), where CAP has access to the actual future observation o_{t+k} and 3 in-context examples of (o_t,o_{t+k}, y_{t+k}), which allow for pure captioning without having to predict the future. This indicates that z_t is not merely sufficient but a *superior* predictive statistic: as a task-optimized embedding, it is less susceptible to the noise and perceptual ambiguity inherent in raw pixel data. This directly justifies the Zero-Visual-Overhead architecture.

Semantically richer descriptions. The LLM-judged scores in Table 1 show that both DILLO-1B and DILLO-4B outperform all baselines across the vast majority of tasks on both benchmarks. On MetaWorld `Drawer-Open`, DILLO-4B achieves **0.628** against **0.497** for FS and **0.252** for CAP. This strong performance extends to LIBERO, with two notable exceptions. On LIBERO-90, which spans 90 distinct tasks, performance relative to CAP suggests that larger-scale data collection is needed to handle the increased task diversity. On LIBERO-Spatial, significant variability in object placements, combined with the absence of explicit goal position conditioning in the current DILLO formulation, limits generalization to spatially complex configurations. These are targeted failure modes rather than systemic weaknesses, and both suggest clear directions for future work.

Figure 4 provides qualitative validation on LIBERO. In the POSITIVE outcome (left), the predicted description $\hat{d}$ accurately articulates the arm's successful engagement with the object. In the NEGATIVE outcome (right), DILLO correctly anticipates the impending failure, the arm disengages from the target, and issues

Table 3: Steering Success Rates and Verdict Accuracy. We report Steering Success Rates (SR) and Safety Verdict Accuracy (%). High success rates and accuracy confirm that the model effectively steers rollouts while discriminating successful dynamics from failures.

		Success Rate			Verdict Accuracy	
	Task	Base	DILLO-1B	DILLO-4B	DILLO-1B	DILLO-4B
MetaWorld	Soccer	65.0%	75.0%	**80.0%**	82.7%	**87.4%**
	Sweep-Into	80.0%	**90.0%**	85.0%	**97.7%**	92.1%
	Drawer-Open	75.0%	70.0%	**90.0%**	80.6%	**94.7%**
LIBERO	Goal	86.0%	87.5%	**89.0%**	80.0%	**87.5%**
	Object	82.0%	86.5%	**88.0%**	79.8%	**81.2%**
	Spatial	70.0%	**76.0%**	73.0%	77.8%	**86.7%**
	10	64.0%	71.5%	**74.0%**	88.6%	**92.5%**
	90	**71.4%**	68.6%	70.5%	**86.9%**	85.7%

a NEGATIVE verdict. Both the description and the verdict are generated solely from z_t and $a_{t:t+k}$, with no access to visual observations at inference time.

4.4 Proactive Policy Steering and Inference Latency

Proactive Policy Steering. We employ DILLO as a rejection sampling controller (N=5) to quantify steering gains over 20 episodes per task. Table 3 reports success rates across both benchmarks.

On MetaWorld, DILLO consistently improves over the base policy. On `Soccer`, DILLO-4B raises the success rate from 65.0% to **80.0%**, and even the smaller DILLO-1B achieves 75.0%. On `Sweep-Into`, DILLO-1B reaches **90.0%**, the highest score across all model variants on that task.

On LIBERO, gains are consistent across the four main suites. DILLO-4B improves LIBERO-Goal from 86.0% to **89.0%** and LIBERO-Object from 82.0% to **88.0%**, confirming that the distilled verdict generalizes effectively to multi-task, language-conditioned settings. The exception is LIBERO-90, where the base policy (71.4%) marginally outperforms both DILLO variants (68.6% and 70.5%). We analyze 900 LIBERO-90 episodes under varying rejection thresholds to isolate the source of failures. Moderate gating yields a net positive effect, improving success by +1.7% overall and by +17.5% on tasks below 70% baseline success. Heavier gating, however, over-rejects candidate actions and reduces success by −4.4%, indicating that broad multi-task deployment requires task-conditioned calibration rather than a single global rejection threshold.

Generalization across LIBERO suites. To test cross-suite transfer, we steer the LIBERO-10 ACT policy with our LIBERO-90-trained explainer. DILLO raises policy success from 64.0% to 69.5%, approaching the 71.5% of same-suite steering. Meaning that the learned semantic evaluator does not simply memorize a single suite but can generalize to similar scenarios.

Positive/Negative Classification. To understand the source of this effectiveness, we analyze the binary classification accuracy of the verdict token $\hat{c}$. Table 3 shows that DILLO-4B achieves an average accuracy of **91.4%** across tasks, without access to any visual observation at inference time. A human validation study (see supplementary material) further corroborates these results. On MetaWorld, DILLO agrees with human positive assessments on 84.1% of 60 episodes. On LIBERO, DILLO's verdict agrees with ground truth at 72.1%, exceeding the human-vs-ground-truth agreement of 69.5%, while human evaluators rate its descriptions at 3.62/5.0, confirming interpretability to non-expert observers.
Inference Latency. Per decision step (one action chunk of 20 simulation steps), the base policy accounts for $T_{\text{sample}} \approx 10\,\text{ms}$, and DILLO adds 0.26 s (1B) and 0.38 s (4B) per step. This is a $\sim$**14**$\times$ and $\sim$**10**$\times$ speedup over Forewarn [31] (3.70 s), measured on an NVIDIA RTX 3090 against Forewarn's more powerful RTX A6000. Additionally on LIBERO-10, DILLO-1B reduces average episode length by 36% (390 $\rightarrow$ 250 simulation steps, 19.8 $\rightarrow$ 13.0 decision steps), yielding a net episode-level overhead of only $\sim$3 s over the baseline.

5 Limitations

DILLO requires re-alignment of the projectors for each new policy architecture, as the latent manifold z_t is architecture-specific. The training pipeline itself is environment- and action-representation-agnostic, enabling direct deployment on physical hardware by having the teacher annotate real trajectories. The privileged information (ground-truth 6DoF poses) is required only for the teacher during offline annotation, never at inference; in practice, it can be substituted by estimated poses using a pose estimation algorithm [29], though perception noise may degrade distillation quality.

6 Conclusion

We introduce DILLO, a Distilled Language-Action World Model that bypasses the latency bottleneck of visual simulation in proactive agent steering. By distilling the semantic foresight of a simulator-privileged teacher into a fast, latent-conditioned student, we demonstrate that a policy's internal representations already encode the necessary dynamics to anticipate action outcomes. This Latent Sufficiency Hypothesis allows us to shift the paradigm of world modeling from computationally expensive "simulate-then-act" pipelines to a streamlined "describe-then-act" approach.

Our extensive evaluation across both single-task (MetaWorld) and multi-task (LIBERO) environments confirms that DILLO produces high-fidelity outcome descriptions and failure verdicts directly from the policy latent z_t. This architecture achieves a $\sim$14$\times$ inference speedup over visual world models while effectively steering policies to high success rates. Ultimately, DILLO establishes that real-time semantic foresight does not require the expensive rendering of future states, offering a fast, interpretable, reliability layer for embodied agents.

Acknowledgments

I.S. gratefully acknowledges the NVIDIA Academic Grant Program for the donation of a DGX Spark. This work was further supported by Panasonic and by Sapienza grant RG123188B3EF6A80 (CENTS). We also thank CINECA for the allocation of computational resources.

References

1. Agia, C., Sinha, R., Yang, J., Cao, Z., Antonova, R., Pavone, M., Bohg, J.: Unpacking failure modes of generative policies: Runtime monitoring of consistency and progress. In: Agrawal, P., Kroemer, O., Burgard, W. (eds.) Conference on Robot Learning, 6-9 November 2024, Munich, Germany. Proceedings of Machine Learning Research, vol. 270, pp. 689–723. PMLR (2024), https://proceedings.mlr.press/v270/agia25a.html
2. Duan, J., Pumacay, W., Kumar, N., Wang, Y.R., Tian, S., Yuan, W., Krishna, R., Fox, D., Mandlekar, A., Guo, Y.: AHA: A vision-language-model for detecting and reasoning over failures in robotic manipulation. In: The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025 (2025), https://openreview.net/forum?id=JVkdSi7Ekg
3. Ebert, F., Finn, C., Dasari, S., Xie, A., Lee, A.X., Levine, S.: Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. CoRR **abs/1812.00568** (2018), http://arxiv.org/abs/1812.00568
4. Finn, C., Goodfellow, I.J., Levine, S.: Unsupervised learning for physical interaction through video prediction. In: Lee, D.D., Sugiyama, M., von Luxburg, U., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain. pp. 64–72 (2016), https://proceedings.neurips.cc/paper/2016/hash/d9d4f495e875a2e075a1a4a6e1b9770f-Abstract.html
5. Finn, C., Levine, S.: Deep visual foresight for planning robot motion. In: 2017 IEEE International Conference on Robotics and Automation, ICRA 2017, Singapore, Singapore, May 29 - June 3, 2017. pp. 2786–2793. IEEE (2017). https://doi.org/10.1109/ICRA.2017.7989324, https://doi.org/10.1109/ICRA.2017.7989324
6. García, C.E., Prett, D.M., Morari, M.: Model predictive control: Theory and practice—a survey. Automatica **25**(3), 335–348 (1989). https://doi.org/https://doi.org/10.1016/0005-1098(89)90002-2, https://www.sciencedirect.com/science/article/pii/0005109889900022
7. Gu, Q., Ju, Y., Sun, S., Gilitschenski, I., Nishimura, H., Itkina, M., Shkurti, F.: SAFE: Multitask failure detection for vision-language-action models. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025), https://openreview.net/forum?id=XPyAukgsFf
8. Guan, L., Zhou, Y., Liu, D., Zha, Y., Amor, H.B., Kambhampati, S.: "task success" is not enough: Investigating the use of video-language models as behavior critics for catching undesirable agent behaviors. CoRR **abs/2402.04210** (2024). https://doi.org/10.48550/ARXIV.2402.04210, https://doi.org/10.48550/arXiv.2402.04210
9. Ha, D., Schmidhuber, J.: Recurrent world models facilitate policy evolution. In: Bengio, S., Wallach, H.M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett,

R. (eds.) Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada. pp. 2455–2467 (2018), https://proceedings.neurips.cc/paper/2018/hash/2de5d16682c3c35007e4e92982f1a2ba-Abstract.html

10. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: Dy, J.G., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018. Proceedings of Machine Learning Research, vol. 80, pp. 1856–1865. PMLR (2018), http://proceedings.mlr.press/v80/haarnoja18b.html
11. Hafner, D., Lillicrap, T.P., Ba, J., Norouzi, M.: Dream to control: Learning behaviors by latent imagination. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net (2020), https://openreview.net/forum?id=S1lOTC4tDS
12. Hafner, D., Lillicrap, T.P., Norouzi, M., Ba, J.: Mastering atari with discrete world models. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net (2021), https://openreview.net/forum?id=0oabwyZbOu
13. Hafner, D., Pasukonis, J., Ba, J., Lillicrap, T.P.: Mastering diverse domains through world models. CoRR **abs/2301.04104** (2023). https://doi.org/10.48550/ARXIV.2301.04104, https://doi.org/10.48550/arXiv.2301.04104
14. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network (2015), https://arxiv.org/abs/1503.02531
15. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. In: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net (2022), https://openreview.net/forum?id=nZeVKeeFYf9
16. Li, C., Krause, A., Hutter, M.: Robotic world model: A neural network simulator for robust policy optimization in robotics. CoRR **abs/2501.10100** (2025). https://doi.org/10.48550/ARXIV.2501.10100, https://doi.org/10.48550/arXiv.2501.10100
17. Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out. pp. 74–81. Association for Computational Linguistics, Barcelona, Spain (Jul 2004), https://aclanthology.org/W04-1013/
18. Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., Stone, P.: Libero: Benchmarking knowledge transfer for lifelong robot learning. Advances in Neural Information Processing Systems **36**, 44776–44791 (2023)
19. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023 (2023), http://papers.nips.cc/paper_files/paper/2023/hash/6dcf277ea32ce3288914faf369fe6de0-Abstract-Conference.html
20. Liu, Y., Hamid, J.I., Xie, A., Lee, Y., Du, M., Finn, C.: Bidirectional decoding: Improving action chunking via closed-loop resampling. CoRR **abs/2408.17355** (2024). https://doi.org/10.48550/ARXIV.2408.17355, https://doi.org/10.48550/arXiv.2408.17355
21. Liu, Z., Bahety, A., Song, S.: REFLECT: summarizing robot experiences for failure explanation and correction. In: Tan, J., Toussaint, M., Darvish, K. (eds.) Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA.

Proceedings of Machine Learning Research, vol. 229, pp. 3468–3484. PMLR (2023), https://proceedings.mlr.press/v229/liu23g.html
22. Milani, S., Topin, N., Veloso, M., Fang, F.: Explainable reinforcement learning: A survey and comparative review. ACM Comput. Surv. **56**(7), 168:1–168:36 (2024). https://doi.org/10.1145/3616864, https://doi.org/10.1145/3616864
23. Papineni, K., Roukos, S., Ward, T., Zhu, W.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA. pp. 311–318. ACL (2002). https://doi.org/10.3115/1073083.1073135, https://aclanthology.org/P02-1040/
24. Sammani, F., Mukherjee, T., Deligiannis, N.: NLX-GPT: A model for natural language explanations in vision and vision-language tasks. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022. pp. 8312–8322. IEEE (2022). https://doi.org/10.1109/CVPR52688.2022.00814, https://doi.org/10.1109/CVPR52688.2022.00814
25. Sinha, R., Elhafsi, A., Agia, C., Foutter, M., Schmerling, E., Pavone, M.: Real-time anomaly detection and reactive planning with large language models. In: Kulic, D., Venture, G., Bekris, K.E., Coronado, E. (eds.) Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024 (2024). https://doi.org/10.15607/RSS.2024.XX.114, https://doi.org/10.15607/RSS.2024.XX.114
26. Sutton, R.S.: Dyna, an integrated architecture for learning, planning, and reacting. SIGART Bull. **2**(4), 160–163 (1991). https://doi.org/10.1145/122344.122377, https://doi.org/10.1145/122344.122377
27. Team, G.: Gemma 3 technical report. CoRR **abs/2503.19786** (2025). https://doi.org/10.48550/ARXIV.2503.19786, https://doi.org/10.48550/arXiv.2503.19786
28. Wang, Z., Liang, B., Dhat, V., Brumbaugh, Z., Walker, N., Krishna, R., Cakmak, M.: I can tell what I am doing: Toward real-world natural language grounding of robot experiences. In: Agrawal, P., Kroemer, O., Burgard, W. (eds.) Conference on Robot Learning, 6-9 November 2024, Munich, Germany. Proceedings of Machine Learning Research, vol. 270, pp. 1863–1890. PMLR (2024), https://proceedings.mlr.press/v270/wang25g.html
29. Wen, B., Yang, W., Kautz, J., Birchfield, S.: Foundationpose: Unified 6d pose estimation and tracking of novel objects. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024. pp. 17868–17879. IEEE (2024). https://doi.org/10.1109/CVPR52733.2024.01692, https://doi.org/10.1109/CVPR52733.2024.01692
30. Wojciechowski, A., Lango, M., Dusek, O.: Faithful and plausible natural language explanations for image classification: A pipeline approach. In: Al-Onaizan, Y., Bansal, M., Chen, Y. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024. pp. 2340–2351. Association for Computational Linguistics (2024). https://doi.org/10.18653/V1/2024.FINDINGS-EMNLP.130, https://doi.org/10.18653/v1/2024.findings-emnlp.130
31. Wu, Y., Tian, R., Swamy, G., Bajcsy, A.: From foresight to forethought: Vlm-in-the-loop policy steering via latent alignment. In: ICLR 2025 Workshop on World Models: Understanding, Modelling and Scaling
32. Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu,

Q., Men, R., Lin, R., Li, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., Qiu, Z.: Qwen2.5 technical report (2024). https://doi.org/10.48550/ARXIV.2412.15115, https://doi.org/10.48550/arXiv.2412.15115

33. Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., Levine, S.: Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In: Kaelbling, L.P., Kragic, D., Sugiura, K. (eds.) 3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings. Proceedings of Machine Learning Research, vol. 100, pp. 1094–1100. PMLR (2019), http://proceedings.mlr.press/v100/yu20a.html
34. Zawalski, M., Chen, W., Pertsch, K., Mees, O., Finn, C., Levine, S.: Robotic control via embodied chain-of-thought reasoning. In: Agrawal, P., Kroemer, O., Burgard, W. (eds.) Conference on Robot Learning, 6-9 November 2024, Munich, Germany. Proceedings of Machine Learning Research, vol. 270, pp. 3157–3181. PMLR (2024), https://proceedings.mlr.press/v270/zawalski25a.html
35. Zhao, T.Z., Kumar, V., Levine, S., Finn, C.: Learning fine-grained bimanual manipulation with low-cost hardware. In: Proceedings of Robotics: Science and Systems (RSS) (2023)

Supplementary Material

We supplement the main paper with additional details, results, and qualitative examples. To complement the Fidelity Text-to-Observation results available in the main manuscript, we present Fidelity Text-to-Text results (Sec. A). We also provide the input prompts used for the baselines (Sec. B), and additional qualitative examples (Sec. C). Furthermore, we describe and show results from an AI-Based Study (Sec. D) and a Human Validation Study (Sec. E). Finally, we explore alternative solutions for steering (Sec. F).

A Fidelity Text-to-Text

To complement our physical grounding reported in Sec.4, we introduce Fidelity Text-to-Text (T2T). Unlike the Text-to-Observation (T2O) metric, which compares predictions against physical state transitions, T2T evaluates the alignment between the predicted description and the ground truth reference text. We apply the parser function g_{parse} to both the generated and the ground-truth descriptions, projecting both into the shared space of canonical directional labels. The final Fidelity T2T score is computed as the accuracy ratio of matching labels for every example. We note that this is a proxy for textual faithfulness rather than physical grounding; this metric effectively quantifies how closely the model reproduces the linguistic patterns of the reference descriptions. The Fidelity Text-to-Text results, presented in Table 4, demonstrate that both the DILLO-1b and DILLO-4b variants substantially outperform their respective baselines.

Table 4: Fidelity Text-to-Text Results. We compare DILLO against predictive baselines (Few-Shot and Zero-Shot) across two model scales (1B and 4B parameters) on both the MetaWorld and LIBERO suites. Unlike the predictive baselines, the captioning model (CAP) has privileged access to past and future information. DILLO consistently outperforms the fair predictive baselines, validating that the latent state is a sufficient statistic for semantic prediction.

		1B Parameter Models				**4B Parameter Models**			
Suite	**Task**	**DILLO-1B**	FS	ZS	CAP	**DILLO-4B**	FS	ZS	CAP
MetaWorld	Soccer	**0.523**	0.451	0.289	0.475	**0.655**	0.495	0.519	0.617
	Sweep-Into	**0.693**	0.494	0.202	0.140	**0.649**	0.592	0.567	0.594
	Drawer-Open	0.531	**0.578**	0.245	0.570	**0.735**	0.685	0.245	0.621
LIBERO	Goal	**0.727**	0.262	0.287	0.723	0.710	0.422	0.241	**0.905**
	Object	0.622	0.388	0.204	**0.820**	0.587	0.546	0.225	**0.877**
	Spatial	0.668	0.546	0.341	**0.794**	0.656	0.563	0.194	**0.902**
	10	**0.648**	0.214	0.199	0.630	0.686	0.449	0.265	**0.867**
	90	**0.708**	0.258	0.206	0.640	0.650	0.278	0.258	**0.713**

A.1 Parser Definition

Both T2O and T2T rely on a shared deterministic parser g_{parse}, which projects a free-form natural language description into a fixed set of canonical semantic

slots. Given a textual description d, the parser $g_{\text{parse}}(d)$ returns a structured representation with two entity scopes, *robot* and *object*, each comprising directional and relational slots:

- **Robot (arm/gripper):** three directional slots for each spatial axis (`x` $\in$ {*left*, *right*, *no change*}, `y` $\in$ {*forward*, *backward*, *no change*}, `z` $\in$ {*up*, *down*, *no change*}) and one relational slot (`approach` $\in$ {*approaching object*, *moving away from object*, *no arm-to-object change*}).
- **Object:** three directional slots (`x`, `y`, `z`), using the same directional labels plus *stationary* for no movement, and one task-progress slot (`target` $\in$ {*on target*, *closer to target*, *farther from target*}, or task-specific labels such as *opening drawer*, *closed drawer*).

Slots that are not mentioned in d are left as `null` and excluded from scoring.

Parsing Procedure. Given a textual description d, the parser operates as follows:

1. **Preprocessing:** All XML/HTML tags are stripped and the text is lowercased.
2. **Sentence splitting:** The text is split into individual sentences using punctuation delimiters (`.`, `;`, `!`, `?`).
3. **Entity-scoped extraction:** For each sentence, the parser determines whether it mentions the *robot* (via aliases such as "gripper", "arm", "end-effector", etc.), the *object* (via aliases such as "object", "ball", "cube", etc.), or both, using regular expression matching. Directional keywords are then attributed to the appropriate entity:
   - If only the robot is mentioned, directional and approach cues are assigned to the robot slots.
   - If only the object is mentioned, directional and target-progress cues are assigned to the object slots.
   - If both are mentioned in the same sentence, the sentence is split on the object noun, and the first clause is attributed to the robot while the second is attributed to the object.
4. **Directional matching:** Within each clause, axis-aligned movement labels are extracted by matching against curated regular expression patterns. For example, the x-axis is labeled *left* if patterns such as "*left*", "*to the left*", or "$-x$" are matched and no contradictory rightward pattern is present. Analogous rules apply to the y- and z-axes.
5. **Global cues:** If stationarity keywords (*e.g.*, "stationary", "motionless", "remains in place") appear anywhere in the description, all object directional slots are set to *stationary*.

B Baselines Prompts

All baselines utilize a shared base prompt, identical to the one employed for the ZeroShot baseline, in which we explicitly define the input schema and instruct the

```
You are given several consecutive observations from a robotic manipulation episode. The robot must interact
      with an object to achieve a task.
Object: drawer
Task: The task is to open the drawer
Task goal: The drawer should be opened in such a way that it reaches the target position
Target position (xyz): [0., 0.539, 0.09]

Observation schema
- Gripper: [x, y, z, openness]
    * x, y, z (meters in a fixed world frame)
    * openness, with values in [0, 1] where 0=closed, 1=fully open
- Object: [x, y, z]
    * x, y, z (same frame as gripper)

Oracular schema (future hint)
- You will be given an Oracular block with auxiliary information about the immediate next observation (i.e.,
       what happens right after the last observation).
- How to use it:
    * Treat the oracular information as privileged signal to resolve ambiguity and correct noise from the
          observation trend or action intent.
    * Never copy numbers or restate raw values; use only qualitative cues for a concise, natural-language
          description.
    * The oracle only refers to the *imminent* next step; do not speculate beyond it.

Describe what will happen next, the imminent behavior after the last given observation. Focus strictly on
      the robot object interaction.

What to cover (in order)
1) Proximity/intent: Is the gripper moving toward or away from the object?
2) Mode of interaction: preparing to interact, aligning, executing a grasp, positioning to push/slide/
      deflect, maintaining or breaking contact.
3) Object reaction: is the object starting to move (and in which direction)? If the robot is not in contact,
       note residual motion (e.g., momentum transfer).
4) Gripper openness change: closing to grasp intent; opening to release/avoid; steady to maintain/wait.

Direction words
- Left/Right: x decrease/increase
- Backward/Forward: y decrease/increase
- Down/Up: z decrease/increase
Never say "along the X-axis/Y-axis/Z-axis"; use the words above.

Style constraints
- First describe the robotic arm, then the object.
- Keep it 1-3 sentences, concise, action-focused.
- Avoid hedging (e.g., "maybe", "might") unless evidence is weak.
- If the arm moves away and the object is not moving (or moving away from the target), point out potential
      misbehavior.
- Do not include numbers, calculations, or restate the raw observations.
- Output only the description text.

Example 1:
- Obs 1 Gripper: [-0.158, 0.851, 0.402, 0.851] Object: [-0., 0.728, 0.09]
- Oracular Info Gripper: [-0.138, 0.852, 0.415, 0.718] Object: [-0., 0.728, 0.09]
- Description: The gripper moves slightly right, forward, and up while closing, suggesting it's attempting
      to align with and grasp the object. The object remains stationary.

Now produce the description for the following input:
- Obs 1 Gripper: [0.001, 0.736, 0.154, 1.] Object: [0., 0.701, 0.09]
- Oracular Info Gripper: [-0.006, 0.696, 0.146, 0.979] Object: [0., 0.662, 0.09]

The description is:
```

Fig. 5: Gemma-3-1b and Gemma-3-4b prompts used in the Fewshot-Captioning configuration. For Gemma-3-4b, we also interleave images with the given observations.

model to generate a description. This schema ensures the model correctly interprets the numerical observation data by specifying the following task-dependent elements:

- **Object Type:** The entity involved in the interaction.
- **Task Description:** A high-level definition of the objective, such as minimizing the distance of an object to a target position, opening a drawer, or moving a ball into a goal.
- **Task Goal:** The specific success condition (*e.g.*, "The drawer should be opened to reach the target position").
- **Target Position:** The coordinates required to enable reasoning about object-to-target relationships.

To adapt this for the FewShot baseline, we extend the prompt by appending three input-output demonstrations. Finally, for the FewShot-Captioning baseline, we supplement both demonstrations and inputs with information of the future. We introduce an explicit schema to instruct the model on how to utilize this privileged data, defined as the observation of the environment state following the execution of an action. Figure 5 illustrates the complete prompt used for the Fewshot-Captioning baseline; note that only a single demonstration is displayed due to space constraints. Additionally, for the 4b baselines, we modify the input prompt by interleaving RGB images with their corresponding observations.

C Additional Qualitative Examples

We provide comprehensive qualitative examples covering the three tasks utilized from the MetaWorld simulator in Figure 6. Note that the gripper and object directional changes in the descriptions must be interpreted from the robotic arm's point of view. Our model, DILLO, successfully describes the motion of both the gripper and the object of interest, while also accurately accounting for crucial gripper-object and object-target relationships (*e.g.*, noting that the object is "further from the target" in the negative example for the Drawer-Open task). DILLO, correctly predicts the verdict (indicated by green and red symbols in the figure). A verdict of NEGATIVE is specifically triggered when the robotic arm moves away from the target and object positions.

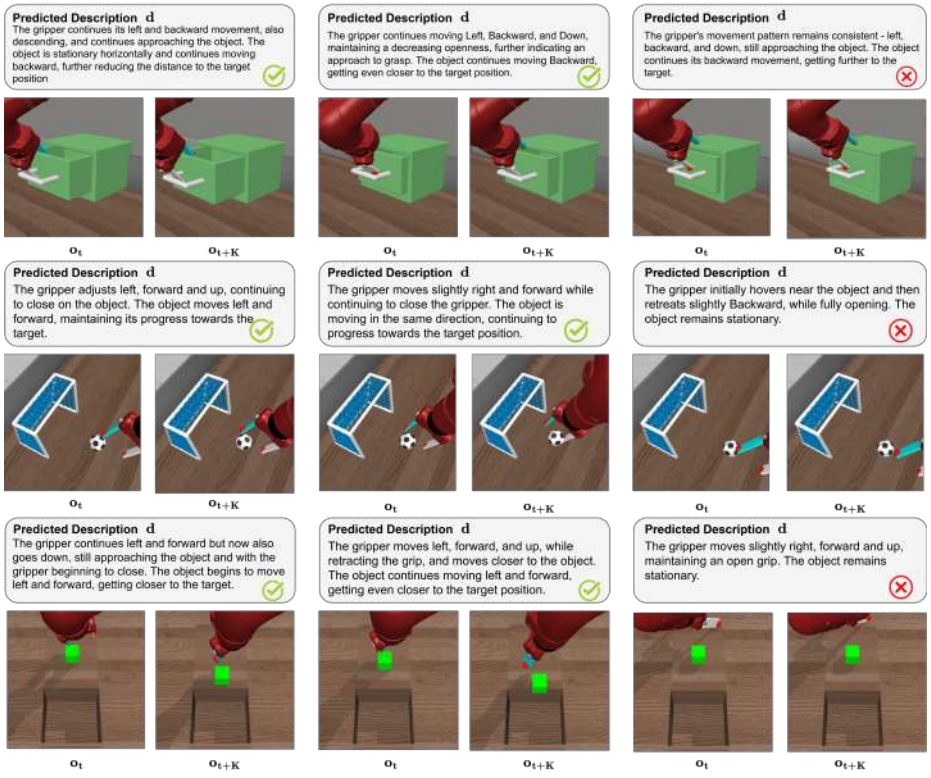

Fig. 6: Qualitative examples of DILLO for Drawer-Open (Top), Soccer (Middle), and Sweep-Into (Bottom) tasks.

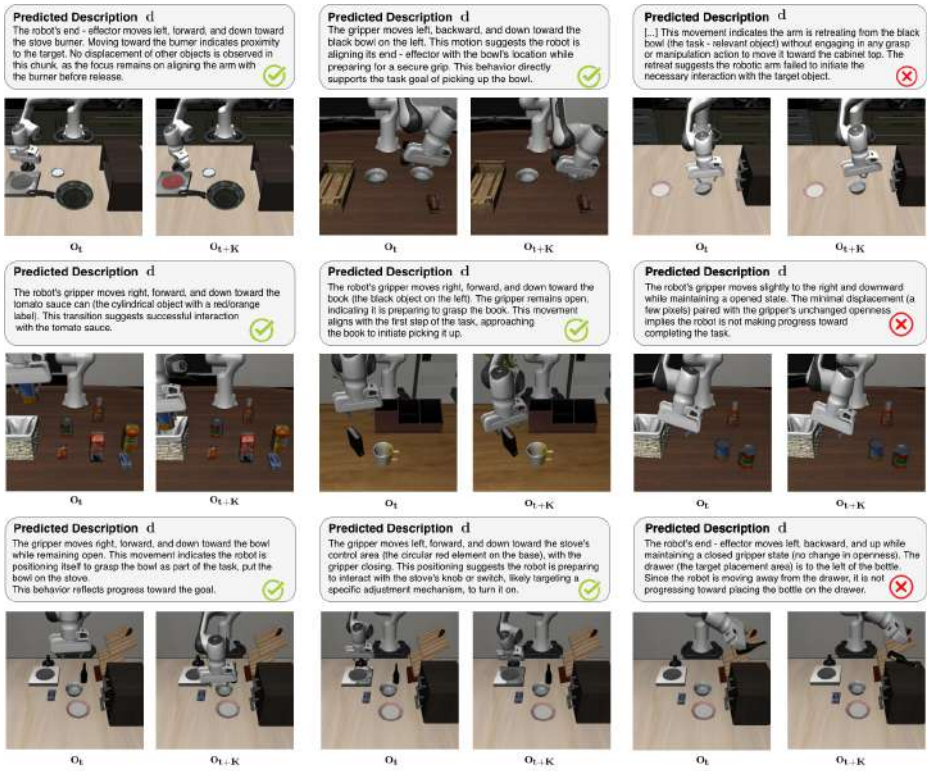

Fig. 7: Qualitative examples of DILLO for LIBERO-90 (Top), LIBERO-10 (Middle), and LIBERO-Goal (Bottom) tasks. Descriptions are trimmed for readability.

In Figure 7, we report additional qualitative rollouts on LIBERO-90, LIBERO-10, and LIBERO-Goal. Compared to MetaWorld, these tasks involve more objects, longer horizons, and language-conditioned goals; as a result, the LIBERO-trained DILLO naturally produces more detailed and fine-grained descriptions that capture multi-step intent and object–object relations. For readability, we truncate these descriptions in the figure—for instance, the middle POSITIVE example in LIBERO-Goal would read:

"The gripper moves left, forward, and down toward the stove's control area (the circular red element on the base), with the gripper closing. This positioning suggests the robot is preparing to interact with the stove's knob or switch, likely targeting a specific adjustment mechanism, to turn it on. The small movement in the gripper's path of least resistance aligns with approaching the task-relevant object (stove control), indicating intent to engage with the system for activating the stove. No visible displacement of the stove itself occurs, but the directional

shift indicates preparation for direct interaction with the ignition component.",

which is shortened to a concise summary in the visualization.

D AI-Based Study

To evaluate the quality and coherence of the generated descriptions across our model and baselines, we conducted an AI-based preference study using an LLM as an automated judge. This approach allows us to measure the perceived accuracy of the linguistic outputs at scale, mitigating the expense and variability of traditional human studies. The study relies on a specific, structured prompt designed to instruct the judging LLM (Qwen2.5-32B-Instruct) on the preference task. Each query includes the following inputs:

- Observations (o_t, o_{t+k}): Two consecutive observations (current state and subsequent state) representing the physical transition.
- Candidate Descriptions: A set of $K+1$ descriptions, consisting of one Ground Truth (GT) description and K descriptions generated by the baselines and DILLO. Each candidate is assigned a unique, numeric ID (from 1 to K).
- Instructions: Guidance on the reference example and constraints for the output format (returning only the preferred ID).

The employed query prompt is shown in Figure 8. To automate the evaluation process, we define a function responsible for prompt formatting, querying the LLM judge, and extracting the preference ID. To mitigate positional bias (*i.e.*, preventing the model from systematically selecting the first ID), the candidate IDs and their corresponding descriptions are shuffled prior to formatting. The results of this AI-based study are detailed in Figure 9. As can be seen, both the DILLO-1b and DILLO-4b models achieve a higher judge preference compared to the baselines.

```
You are a meticulous evaluator. You are given a REFERENCE (ground-truth
    description)
and several MODEL RESPONSES describing the same event. Compare them carefully
     and decide
which numbered response best matches the REFERENCE at the level of atomic
    actions
(motion, direction, gripper state, object behavior, etc.).

## Rules
- Focus on how accurately each response reflects the REFERENCE.
- Ignore harmless extra details.
- Penalize contradictions (e.g., left vs right, open vs close).
- Choose exactly ONE winner (the best match).
- If two are equally good, choose the more precise one.

## Output Format
Return ONLY valid JSON in this exact format:

{{
  "example_id": "<EXAMPLE_ID>",
  "winner_index": <int>,  // index of the best response (1-based)
  "reason": "<short justification>"
}}

Now evaluate this case.

EXAMPLE_ID: {example_id}

REFERENCE:
{reference}

RESPONSES:
{responses}
```

Fig. 8: The input prompt utilized for querying Qwen2.5-32B-Instruct in our AI-based study

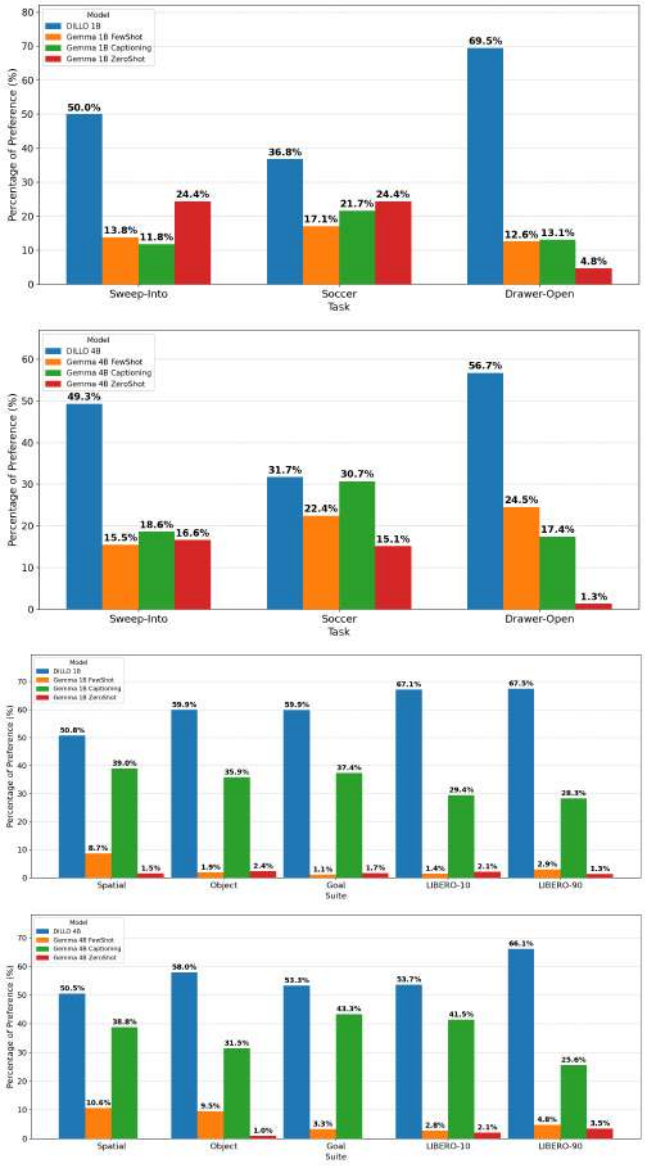

Fig. 9: AI judge preference percentage results for the 1B and 4B models across the three MetaWorld tasks (Sweep-Into, Soccer, and Drawer-Open) and the five LIBERO suites (Spatial, Object, Goal, LIBERO-10, and LIBERO-90).

E Human Validation Study

To complement the automated evaluation, we conducted a human validation study on 120 randomly sampled episodes: 60 from MetaWorld (20 per task, balanced between POSITIVE and NEGATIVE outcomes) and 60 from LIBERO. **Annotators.** Ten annotators with a background in robotics and computer vision (PhD-level) participated in the study. Prior to participation, all annotators were provided with an informed consent form detailing the study's purpose, task structure, and data usage policy, and confirmed their agreement before proceeding. **Protocol.** For each episode, annotators were shown: (i) the natural language task instruction, (ii) DILLO's generated description d_{t+k}, and (iii) the RGB observations before and after action execution (o_t and o_{t+k}). Annotators were then asked to:

1. Rate the accuracy of d_{t+k} on a 1–5 Likert scale, where 1 = *Completely Inaccurate* and 5 = *Perfectly Accurate*.
2. Classify the observed outcome as POSITIVE (effective, advances the task) or NEGATIVE (ineffective, useless, or detrimental to the goal).

Human verdict labels were then compared against DILLO's predicted verdict $\hat{c}$ and the ground-truth labels c_T. Formally, given per-episode labels $\hat{c}_i$ from DILLO and c_i^{human} from the aggregated human majority vote, we define agreement as

$$\text{Agree}(\hat{c}, c^{\text{human}}) \;=\; \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\big[\hat{c}_i = c_i^{\text{human}}\big],$$

with analogous definitions for $\text{Agree}(c_T, c^{\text{human}})$ and $\text{Agree}(\hat{c}, c_T)$. **Results.** On MetaWorld, DILLO's predicted verdict $\hat{c}$ agrees with human assessments on **84.1%** of episodes. On LIBERO, DILLO achieves a verdict agreement with ground truth c_T of **72.1%**, exceeding the human-vs-ground-truth agreement of 69.5%, indicating that DILLO's latent-conditioned predictions are a more reliable safety signal than human visual inspection alone. DILLO-vs-human agreement on LIBERO stands at 68.8%. Regarding description quality, human annotators assigned an average accuracy rating of $\mathbf{3.62/5.0} \pm 1.09$, confirming that the distilled natural language foresight is both accurate and interpretable to domain-expert observers.

F Ablation Studies

We investigate which components of DILLO contribute to effective policy steering by comparing the complete multistage training procedure against three alternatives. *Stage 3 only* directly optimizes the final description-and-verdict objective, omitting the projector-alignment and description-distillation stages. *Verdict only* removes semantic description generation and trains the model to predict only the binary POSITIVE/NEGATIVE verdict. Finally, the *MLP critic* directly predicts the binary verdict from the policy latent and candidate action chunk,

Table 5: Steering ablation on LIBERO validation suites. Episode success rates (%). Multistage denotes the complete DILLO training procedure.

		DILLO-1B			DILLO-4B			
Suite	Base	Multistage	Stage 3 only	Verdict only	Multistage	Stage 3 only	Verdict only	MLP
LIBERO-Goal	86.0	87.5	79.0	85.0	89.0	**92.0**	83.0	86.0
LIBERO-Object	82.0	86.5	80.0	83.0	**88.0**	81.0	81.0	80.5
LIBERO-Spatial	70.0	**76.0**	69.0	71.0	73.0	54.0	64.0	70.5
LIBERO-10	64.0	71.5	59.0	65.0	**74.0**	67.0	63.0	64.0
LIBERO-90	71.4	68.6	70.8	71.5	70.5	**73.6**	70.1	70.2

providing a simpler learned classifier without language generation. All variants are evaluated using the same rejection-sampling protocol and candidate budget (N=5).

Table 5 shows that the complete multistage procedure achieves the strongest overall steering performance for both model sizes. Averaged across the five LIBERO suites, DILLO-1B and DILLO-4B obtain success rates of (78.0%) and (78.9%), respectively, compared with (74.7%) for the base policy. Removing the training curriculum substantially reduces performance: the Stage-3-only variants achieve (71.6%) and (73.5%), corresponding to decreases of (6.4) and (5.4) percentage points relative to the complete 1B and 4B models. Although Stage-3-only training performs strongly on individual suites, such as LIBERO-Goal and LIBERO-90 for the 4B model, its performance is less consistent across tasks. This indicates that directly optimizing the final objective can produce effective task-specific steering, but does not provide the same robustness as the progressive alignment and distillation procedure.

The verdict-only ablation further isolates the contribution of semantic description generation. Without the description objective, average success decreases from (78.0%) to (75.1%) for the 1B model and from (78.9%) to (72.2%) for the 4B model. The effect is particularly pronounced for DILLO-4B, suggesting that binary supervision alone does not reliably exploit the capacity of the larger language model. Generating a description of the predicted physical outcome provides a denser intermediate supervision signal, encouraging the model to ground its verdict in the expected consequences of the proposed action rather than learning only a binary decision boundary.

The latent-conditioned MLP critic also fails to improve the base policy on average, obtaining (74.2%) success compared with the base policy's (74.7%). It leaves performance unchanged on LIBERO-10 and LIBERO-Goal, produces only a marginal improvement on LIBERO-Spatial, and decreases performance on LIBERO-Object and LIBERO-90. Thus, access to the policy latent, candidate actions, and binary teacher labels is not sufficient to produce reliable steering with the tested classifier. Overall, these ablations show that both the multistage curriculum and semantic outcome-generation objective contribute to DILLO's

steering performance. While they do not establish that natural-language generation is strictly necessary, they indicate that semantic description generation provides a useful intermediate learning signal that leads to more consistent and effective action selection than direct binary classification.