World–Value–Action Model: Implicit Planning for Vision–Language–Action Systems

Runze Li[*]
Westlake University
Hangzhou, China

Hongyin Zhang[*]
Westlake University
Hangzhou, China

Junxi Jin
Westlake University
Hangzhou, China

Qixin Zeng
Westlake University
Hangzhou, China

Zifeng Zhuang
Westlake University
Hangzhou, China

Yiqi Tang
Westlake University
Hangzhou, China

Shangke Lyu[†]
Nanjing University
Suzhou, China

Donglin Wang[†]
Westlake University
Hangzhou, China

Abstract

Vision–Language–Action (VLA) models have emerged as a promising paradigm for building embodied agents that ground perception and language into action. However, most existing approaches rely on direct action prediction, lacking the ability to reason over long-horizon trajectories and evaluate their consequences, which limits performance in complex decision-making tasks. In this work, we introduce **World–Value–Action (WAV)** model, a unified framework that enables implicit planning in VLA systems. Rather than performing explicit trajectory optimization, WAV model learn a structured latent representation of future trajectories conditioned on visual observations and language instructions. A learned world model predicts future states, while a trajectory value function evaluates their long-horizon utility. Action generation is then formulated as inference in this latent space, where the model progressively concentrates probability mass on high-value and dynamically feasible trajectories. We provide a theoretical perspective showing that planning directly in action space suffers from an exponential decay in the probability of feasible trajectories as the horizon increases. In contrast, latent-space inference reshapes the search distribution toward feasible regions, enabling efficient long-horizon decision making. Extensive simulations and real-world experiments demonstrate that the WAV model consistently outperforms state-of-the-art methods, achieving significant improvements in task success rate, generalization ability, and robustness, especially in long-horizon and compositional scenarios.

1 Introduction

Building embodied agents that can robustly perceive, reason, and act in complex physical environments remains a central challenge in artificial intelligence [Mason, 2001]. Vision–Language–Action (VLA) models have recently made significant progress by leveraging large-scale pre-trained vision–language models to map multimodal observations and natural language instructions to robot actions, enabling strong semantic generalization [Brohan et al., 2022, Zitkovich et al., 2023, Kim et al., 2024,

[*]Equal contribution.

[†]Corresponding authors: shangke_lyu@nju.edu.cn wangdonglin@westlake.edu.cn

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Black et al., 2024]. However, most existing VLA approaches rely on direct action prediction, treating each decision step independently and lacking mechanisms to reason over long-horizon trajectories or evaluate their outcomes. This limitation hinders their ability to perform coherent multi-step decision making in complex environments.

In parallel, world models aim to capture environment dynamics by predicting future observations, providing a structured representation of how the world evolves over time [Wu et al., 2024, Agarwal et al., 2025]. While such models enable reasoning about future states, they typically do not provide a principled mechanism for evaluating trajectories or selecting actions. Recent works attempt to unify world modeling and policy learning within a single architecture [Cen et al., 2025b, Liao et al., 2025, Bi et al., 2025a, Ye et al., 2026, Li et al., 2026], but largely rely on supervised learning over static datasets, resulting in limited ability to capture fine-grained dynamics and assess long-horizon trajectory quality. Another line of research leverages learned world models as simulators for data augmentation in RL [Guo et al., 2025, Li et al., 2025, Zhu et al., 2025]. However, this paradigm is fundamentally constrained by the generalization capability of the world model, leading to compounding errors and unreliable simulated data in complex or out-of-distribution scenarios. In contrast, model-based RL suggests that effective long-horizon decision making arises from the combination of predictive modeling and trajectory evaluation, where planning can be viewed as iteratively refining candidate futures based on their expected value [Hansen et al., 2022, 2023, 2025]. This perspective raises a fundamental question: *can planning be learned as an implicit inference process, rather than implemented as an explicit optimization module?*

In this work, we answer this question affirmatively by proposing **World–Value–Action (WAV**[2]**)** model, a unified framework that endows VLA systems with implicit planning capabilities. Instead of performing explicit search in the high-dimensional action space, the WAV learns a latent distribution over future trajectories conditioned on visual observations and language instructions. A learned world model predicts future states, while a trajectory value function evaluates their long-horizon utility. Crucially, action generation is formulated as inference in this latent space, where the model progressively concentrates probability mass on trajectories that are both dynamically feasible and high-value. We further provide a theoretical analysis showing that direct planning in action space suffers from an exponential decay in the probability of feasible trajectories as the planning horizon increases. In contrast, latent-space inference reshapes the search distribution toward feasible regions, while iterative refinement enables efficient optimization within this restricted domain. This perspective explains how planning can emerge implicitly from the joint learning of world dynamics and value estimation. Our results demonstrate consistent improvements over strong baselines in task success rate, generalization, and robustness, with particularly large gains in long-horizon and compositional scenarios. Our contributions are summarized as follows:

- We introduce **World–Value–Action (WAV)** model, a unified framework that enables implicit planning in VLA systems through joint modeling of dynamics and trajectory value.
- We provide a theoretical perspective showing that latent planning mitigates the exponential decay of feasible trajectory probability inherent in action-space planning.
- The WAV demonstrates state-of-the-art performance in simulated benchmarks and real-world scenarios, achieving significant improvements in long-horizon reasoning and generalization.

2 Related Work

2.1 World Models for VLA

World models have emerged as a core paradigm for equipping VLA models with predictive imagination and physical reasoning capabilities for robotic manipulation tasks. A large body of work focuses on unified architecture design, enabling joint modeling of policy execution and future state prediction within a shared transformer backbone, thereby bridging the gap between visual dynamics modeling and end-to-end robotic control in VLA systems Bi et al. [2025b], Cen et al. [2025a], Bu et al. [2025a], Cen et al. [2025c], Zhang et al. [2025c,a], Liao et al. [2025]. Subsequent efforts further improve the modeling capacity and policy–dynamics synergy of this paradigm, such as DreamVLA [Zhang et al., 2025c], GEVRM [Zhang et al., 2025a], and Bu et al. [Bu et al., 2025a], which enhance

[2]We refer to this model as WAV, emphasizing the WAVe-like propagation of value through predicted futures.

long-horizon prediction accuracy and manipulation performance within integrated VLA–world model systems. Beyond architectural unification, another line of work explores world model-driven virtual environments for VLA post-training. NORA [Hung et al., 2025] and RoboScape [Tang et al., 2025] develop scalable pipelines for policy learning in simulation, while Xiao et al. [Xiao et al., 2025] and Li et al. [Li et al., 2025] leverage world models to generate synthetic rewards for RL or preference optimization, enabling further policy refinement. Rather than scaling model capacity or relying on external simulation, our work introduces implicit planning as an internal inference mechanism, offering a fundamentally different approach to long-horizon decision making in VLA systems.

2.2 Model-Based Reinforcement Learning

Model-based RL improves sample efficiency and long-horizon decision making by learning environment models for planning or policy optimization. Early work follows the Dyna paradigm and probabilistic dynamics modeling [Sutton, 1991, Deisenroth and Rasmussen, 2011], and is later extended with neural dynamics, uncertainty-aware model predictive control, and hybrid model-based (or model-free) optimization [Chua et al., 2018, Williams et al., 2018, Janner et al., 2019]. Latent world models further enable imagined rollouts from high-dimensional observations, as demonstrated in PlaNet, Dreamer, and MuZero [Hafner et al., 2019b,a, Schrittwieser et al., 2020]. More recent approaches emphasize scalability and task generalization, where methods such as TD-MPC and TD-MPC2 [Hansen et al., 2022, 2023] integrate latent planning with value learning, and subsequent work [Hansen et al., 2025] shows that large-scale world models can generalize across diverse tasks and dynamics. Rather than relying on explicit planning or iterative policy optimization, our approach treats planning as an emergent property of inference in a structured generative model, offering a fundamentally different perspective on long-horizon control.

3 Preliminaries

Vision–Language–Action (VLA) Models. We consider language-conditioned robotic manipulation, where at each timestep the model receives a visual observation $o_t \in \mathcal{O}$, a language instruction $g \in \mathcal{G}$, and optionally proprioceptive state $p_t \in \mathcal{P}$, and outputs an action $a_t \in \mathcal{A}$. Given an expert dataset $\mathcal{D}_{\text{expert}} = \{(g, \tau)\}$, where $\tau = \{(o_1, p_1, a_1), \ldots, (o_T, p_T, a_T)\}$, a VLA model π_θ is modeled as a conditional distribution over actions, i.e., $\pi_\theta(a_{1:T} \mid o_{1:T}, p_{1:T}, g) = \prod_{t=1}^{T} \pi_\theta(a_t \mid o_{1:t}, p_{1:t}, a_{1:t-1}, g)$. In practice, the VLA model predicts short action sequences conditioned on the current context, $\pi_\theta(a_{t:t+H} \mid o_t, p_t, g)$, and is trained via maximum likelihood on $\mathcal{D}_{\text{expert}}$.

Model Predictive Control (MPC). MPC [Garcia et al., 1989] is a general paradigm for model-based decision making, where actions are selected by optimizing predicted future trajectories over a finite horizon. At each decision step t, MPC solves: $a^{\star}_{t:t+H} = \arg\max_{a_{t:t+H}} \mathbb{E}\left[\sum_{i=0}^{H} \gamma^i R(s_{t+i}, a_{t+i})\right]$, where $\gamma \in (0, 1)$ is a discount factor and $R(\cdot)$ denotes the reward function. Candidate trajectories are evaluated by rolling out a learned world model. By explicitly reasoning over future trajectories, MPC achieves improved robustness and sample efficiency compared to purely reactive, model-free policies.

4 Methodology

In this work, we propose **WAV**, an end-to-end modular model for long-horizon reasoning and decision making via latent planning. We establish its theoretical foundation, instantiate it as a unified architecture, and realize it through iterative inference.

4.1 Theoretical Analysis of Latent Planning

Action-Space Planning and the Curse of Feasibility. Long-horizon decision making in VLA tasks poses a fundamental challenge for planning-based methods. An intuitive idea of model-based planning seems to be to sample candidate action sequences and generalize them through a dynamic process. However, as the planning horizon increases, this approach quickly becomes ineffective. The reason is not merely that the action space is high-dimensional, but that the set of feasible trajectories occupies an increasingly negligible fraction of the ambient trajectory space. Formally, a horizon-H trajectory resides in the space $\mathcal{X} = \mathcal{S}^H \times \mathcal{A}^H$, whose dimensionality grows linearly with H. In

VLA settings, visual constraints (e.g., scene geometry and object states), physical dynamics (e.g., contact and kinematics), and task-level semantic requirements (e.g., instruction grounding) jointly restrict valid behaviors to a thin manifold $\mathcal{M}_{\text{traj}} \subset \mathcal{X}$ with substantially lower intrinsic dimension. Lemma 4.1 formalizes the consequence of this mismatch: under uniform exploration in trajectory space, the probability of sampling even approximately feasible trajectories decays exponentially with the planning horizon. This result reveals a fundamental limitation of direct action-space search in VLA systems: the vast majority of sampled trajectories are infeasible by construction, irrespective of the planner's sophistication or the fidelity of the underlying models.

Lemma 4.1 (Vanishing Feasible Mass in Trajectory Space)**.** *Let $\mathcal{X} = \mathcal{S}^H \times \mathcal{A}^H$ be a bounded trajectory space with ambient dimension $D = H(\dim \mathcal{S} + \dim \mathcal{A})$, equipped with a reference measure μ. Let $\mathcal{M}_{\text{traj}} \subset \mathcal{X}$ be a compact subset with intrinsic dimension $d \ll D$. Then, for any fixed tolerance $\epsilon \in (0, 1)$, there exists a constant $c > 0$ such that*

$$\frac{\mu\big(\mathcal{N}_\epsilon(\mathcal{M}_{\text{traj}})\big)}{\mu(\mathcal{X})} \leq \exp(-cH), \tag{1}$$

where $\mathcal{N}_\epsilon(\mathcal{M}_{\text{traj}})$ denotes the ϵ-neighborhood of $\mathcal{M}_{\text{traj}}$.

Latent Planning as Probability Mass Reweighting. Latent planning alleviates the feasibility bottleneck in VLA tasks by reshaping the sampling distribution over trajectories. Rather than sampling directly in the ambient action space, it samples a latent variable and decodes it into a full trajectory via a learned generative model. When this generator captures the manifold of perceptually grounded and executable behaviors, latent sampling induces a distribution that concentrates probability mass on trajectories consistent with scene geometry, task semantics, and action affordances. Proposition 4.2 shows that, compared to uniform action-space sampling, this leads to exponentially higher probability of feasible trajectories as the horizon grows. This advantage arises not from dimensionality reduction, but from reweighting the search toward dynamically and semantically valid regions, effectively acting as a probabilistic projection onto the feasible trajectory manifold.

Proposition 4.2 (Latent Reweighting of the Search Distribution)**.** *Let $\mathcal{M}_{\text{traj}} \subset \mathcal{X}$ denote the feasible trajectory set. Consider a latent generator $\mathcal{W}_\theta : \mathbb{R}^{d_z} \to \mathcal{X}$ and a state-conditioned latent distribution $f_\theta(s_t)$. Assume that the learned model approximately parameterizes feasible trajectories in the sense that there exists $\delta \in (0, 1)$ such that $P_{\text{latent}}(\mathcal{M}_{\text{traj}}) = \Pr_{z\sim f_\theta(s_t)}\big[\mathcal{W}_\theta(z) \in \mathcal{M}_{\text{traj}}\big] \geq 1 - \delta$. Let $\Phi : \mathcal{A}^H \to \mathcal{X}$ denote the trajectory rollout map induced by the system dynamics. Then the ratio of feasible-trajectory probabilities satisfies*

$$\frac{\Pr_{z\sim f_\theta(s_t)}\big[\mathcal{W}_\theta(z) \in \mathcal{M}_{\text{traj}}\big]}{\Pr_{a_{t:t+H}\sim \text{Unif}(\mathcal{A}^H)}\big[\Phi(a_{t:t+H}) \in \mathcal{M}_{\text{traj}}\big]} \geq \exp(cH)\,(1 - \delta). \tag{2}$$

Consequently, latent planning assigns exponentially larger probability mass to feasible trajectories relative to direct action-space sampling as the horizon grows.

Why Feasibility Is Not Enough. While Proposition A.2 explains why latent planning avoids the exponential infeasibility of direct action-space search in VLA tasks, feasibility alone is insufficient for high-quality decision making. The feasible set $\mathcal{M}_{\text{traj}}$ still contains trajectories with widely varying task returns, from minimally valid behaviors to those that correctly satisfy language intent under visual constraints. Thus, merely assigning non-vanishing probability mass to perceptually grounded and executable trajectories does not ensure that high-value solutions will be found with a practical number of samples. Corollary 4.3 formalizes this limitation by showing that one-shot latent sampling fails under any subexponential sample budget. Instead, effective VLA planning requires iterative inference that progressively concentrates probability mass within the feasible set toward trajectories that are both semantically aligned and high-value.

Corollary 4.3 (Necessity of Iterative Latent Planning)**.** *Let $\mathcal{M}_{\text{traj}} \subset \mathcal{X}$ denote the feasible trajectory set, and let $V(\tau) = \sum_{h=0}^{H-1} \gamma^h r(s_{t+h}, a_{t+h})$ denote the cumulative discounted return of a trajectory τ, where r is a bounded per-step reward and $\gamma \in (0, 1]$. Let $\mathcal{W}_\theta : \mathbb{R}^{d_z} \to \mathcal{X}$ be a learned latent trajectory generator inducing the distribution $P_{\text{latent}} = (\mathcal{W}_\theta)_{\#} f_\theta$. Assume that $P_{\text{latent}}(\mathcal{M}_{\text{traj}}) \geq 1 - \delta$ for some $\delta \in (0, 1)$. Then, for any fixed sample budget N that grows subexponentially with the planning horizon H, one-shot sampling from P_{latent} does not, in general, guarantee with high probability the discovery of a near-optimal trajectory. Iterative inference procedures that adaptively*

reweight the latent distribution are therefore necessary to concentrate probability mass on high-value feasible trajectories.

Taken together, these results motivate the structure of proposed WAV model. Latent planning resolves the feasibility bottleneck by reshaping the search distribution, while iterative inference resolves the remaining optimization challenge within the feasible manifold. Therefore, we first construct a modular end-to-end model architecture, and then explicitly combine these two components on this basis, thereby enabling efficient long-horizon planning in complex VLA tasks, as shown in Fig. 1. The theoretical analysis details are given in Appendix Sec. A.

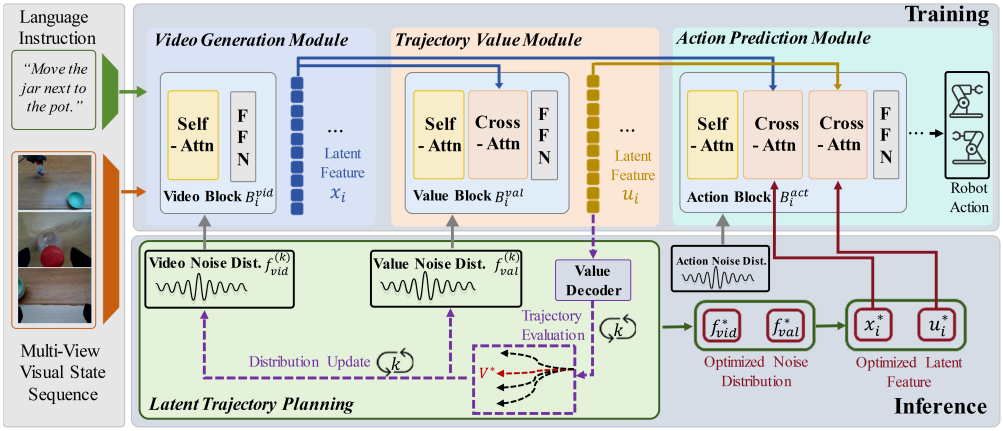

Figure 1: The proposed WAV model decomposes planning and control into three tightly coupled components: **1)** a language-conditioned video generation module for predicting future visual trajectories; **2)** a trajectory value module for long-horizon evaluation; and **3)** an action module for generating executable robot actions. Together, these components enable implicit planning in latent space, where action generation arises from inference over predicted futures guided by trajectory value.

4.2 Model Architecture and Training Objectives

Language-Conditioned Video Generation Module. The video generation module predicts future visual trajectories conditioned on both past observations and task-level language instructions. Given an instruction g, a frozen T5-XXL encoder [Raffel et al., 2020] produces a sequence of language embeddings $\mathcal{T}(g) \in \mathbb{R}^{L_g \times d_t}$, which are injected into a Diffusion Transformer (DiT) [Peebles and Xie, 2023] backbone via cross-attention. Let $i \in \{h, l, r\}$ index camera views (e.g., head, left, right). We denote by $v_0^{(i)}$ the encoded initial observation, $v_{\hat{t}}^{(i)}$ a sparse visual memory at time $\hat{t}$, and $z^{(i)} \sim \mathcal{N}(0, I)$ a view-specific latent noise variable. The video generation module $\mathcal{W}$ predicts the next visual segment as $\hat{x}_{t:t+N} = \mathcal{W}\big(\{v_0^{(i)}, v_{\hat{t}}^{(i)}, z^{(i)}\}_i, \mathcal{T}(g)\big)$. By jointly conditioning on multi-view visual history, stochastic latent variables, and language semantics, $\mathcal{W}$ captures both task intent and spatiotemporal dynamics. Video generation proceeds autoregressively, yielding a temporally coherent visual rollout that represents the policy's future execution. Here, with a slight abuse of notation, we use $\mathcal{W}$ to denote the visual rollout component of the latent trajectory generator $\mathcal{W}_\theta$ introduced in Section 4.1, which produces the visual subspace of a full state–action trajectory.

Trajectory Value and Action Decoding Modules. Both the trajectory value and action decoding modules adopt a DiT-based autoregressive architecture operating on latent trajectory representations. Visual features are first transformed into temporally structured tokens, which are then evaluated and decoded into actions. At DiT block i, video tokens are processed as $\mathbf{x}_i = \mathcal{B}_i^{\text{vid}}(\mathbf{x}_{\text{in}}, \mathcal{T}(g))$, where $\mathcal{B}_i^{\text{vid}}$ denotes the ith visual transformer block. The trajectory value module initializes a latent value token $\mathbf{z}_{\text{val}} \sim \mathcal{N}(0, I)$ and refines it through cross-attention with video features: $\mathbf{u}_i = \mathcal{B}_i^{\text{val}}\big(\mathbf{z}_{\text{val}}, \text{CrossAttn}(\mathbf{z}_{\text{val}}, \mathbf{x}_i)\big)$, where $\mathbf{u}_i$ represents the latent trajectory value embedding. The action decoder initializes latent action tokens $\mathbf{z}_{\text{act}}$ and sequentially integrates video and value feature:

$$\mathbf{z}_{\text{act}}^{(i)} = \mathcal{B}_i^{\text{act}}\Big(\mathbf{z}_{\text{act}}^{(i-1)}, \text{CrossAttn}(\mathbf{z}_{\text{act}}^{(i-1)}, \mathbf{x}_i)\Big), \text{and } \mathbf{a}_i = \mathcal{B}_i^{\text{act}}\Big(\mathbf{z}_{\text{act}}^{(i)}, \text{CrossAttn}(\mathbf{z}_{\text{act}}^{(i)}, \mathbf{u}_i)\Big). \quad (3)$$

Training via Flow Matching. We adopt a three-stage training strategy based on flow matching. Following standard flow matching formulations, we assume that x^0, v^0, a^0 are samples drawn from the corresponding base noise distributions for video, trajectory return, and robot actions, respectively, while x^1, v^1, a^1 denote samples from the target data distributions, including the target video trajectories, the return $v^1 = \sum_{i=0}^{H} \gamma^i R(s_{t+i}, a_{t+i})$, and the target robot action distribution. Here we use a rule-based dense reward function similar to that used in previous work– ReinboT [Zhang et al., 2025b] (details in Appendix Sec. C). First, the video generation module is trained using a video flow loss: $\mathcal{L}_{\text{vid}} = \mathbb{E}\big[\|v_\theta(t, l, o, x^t) - (x^1 - x^0)\|_2^2\big]$. The video model is then frozen, and a trajectory value module is trained to predict cumulative discounted rewards: $\mathcal{L}_{\text{val}} = \mathbb{E}\big[\|v_\theta(t, l, o, z_{\text{vid}}, v^t) - (v^1 - v^0)\|_2^2\big]$. Finally, all modules are jointly trained with an action flow loss: $\mathcal{L}_{\text{act}} = \mathbb{E}\big[\|v_\theta(t, l, o, z_{\text{vid}}, z_{\text{val}}, a^t) - (a^1 - a^0)\|_2^2\big]$.

4.3 Latent Planning and Iterative Inference

Inspired by Model Predictive Path Integral (MPPI) [Williams et al., 2015] control, we perform iterative latent-space trajectory optimization during inference. Instead of fixing latent priors to standard Gaussians, we adaptively update the noise distributions of the video and value modules based on trajectory evaluations. We maintain two independent Gaussian noise distributions: a video flow noise distribution $f_{\text{vid}}^{(k)} = \mathcal{N}\big(\boldsymbol{\mu}_{\text{vid}}^{(k)}, \text{diag}\big((\boldsymbol{\sigma}_{\text{vid}}^{(k)})^2\big)\big)$, and a trajectory value flow noise distribution $f_{\text{val}}^{(k)} = \mathcal{N}\big(\boldsymbol{\mu}_{\text{val}}^{(k)}, \text{diag}\big((\boldsymbol{\sigma}_{\text{val}}^{(k)})^2\big)\big)$, where $k \in \{1, \ldots, K\}$ denotes the iteration index.

Algorithm 1 WAV: Latent Trajectory Planning

```
Video generator module 𝒲, trajectory value module 𝒱, initial flow noise distributions f_vid^(0), f_val^(0), samples (M, N), elites (K_1, K_2).
for iteration k = 1, ..., K do
    Sample video latent noise z_vid^(m) ~ f_vid^(k-1)
    Generate video feature x^(m)
    for each video feature x^(m) do
        Sample value latent noise z_val^(m,n) ~ f_val^(k-1)
        Evaluate values U^(m,n)
        Compute score φ^(m)
    end for
    Update f_vid^(k) using top-K_1 samples by Eq. 4
    Update f_val^(k) using top-K_2 samples by Eq. 5
end for
Sample optimized latent noise z*_vid, z*_val
Predict action by Eq. 3
```

At iteration k, we first sample M groups of video latent noise: $\big\{\mathbf{z}_{\text{vid}}^{(m)}\big\}_{m=1}^{M} \sim f_{\text{vid}}^{(k-1)}$. Each latent sample is denoised by the video generation module $\mathcal{W}$ to produce video feature chunking: $\mathbf{x}_{1:H_1}^{(m)} = \mathcal{W}\big(\mathbf{z}_{\text{vid}}^{(m)}\big)$, where $\mathbf{x}^{(m)} \in \mathbb{R}^{H_1 \times d_v}$ represents a predicted future visual trajectory feature. For each video feature chunking $\mathbf{x}^{(m)}$, we sample N groups of trajectory value flow noise: $\big\{\mathbf{z}_{\text{val}}^{(m,n)}\big\}_{n=1}^{N} \sim f_{\text{val}}^{(k-1)}$, and denoise them using the trajectory value module $\mathcal{V}$ conditioned on the same video features: $\mathbf{v}^{(m,n)} = \mathcal{V}\big(\mathbf{x}^{(m)}, \mathbf{z}_{\text{val}}^{(m,n)}\big)$, where $\mathbf{v}^{(m,n)}$ denotes a predicted state-value.

To obtain a stable assessment of future trajectory quality, we compute the Signal-to-Noise Ratio (SNR) of the value predictions for each exploration: $\text{SNR}^{(m,n)} = \mathbb{E}\big[\mathbf{v}^{(m,n)}\big] / \big(\text{Std}\big[\mathbf{v}^{(m,n)}\big] + \epsilon\big)$ where ϵ is a small constant for numerical stability. The final score of the mth video exploration is defined as the most reliable evaluation among its N value estimates: $\phi^{(m)} = \max_{n \in \{1, \ldots, N\}} \text{SNR}^{(m,n)}$. Based on the scores $\{\phi^{(m)}\}_{m=1}^{M}$, we select the top K_1 elite video explorations. The video noise distribution is updated using the empirical mean and standard deviation of their corresponding latent variables:

$$\boldsymbol{\mu}_{\text{vid}}^{(k)} = \frac{1}{K_1} \sum_{m \in \mathcal{E}_{\text{vid}}} \mathbf{z}_{\text{vid}}^{(m)}, \text{ and } \boldsymbol{\sigma}_{\text{vid}}^{(k)} = \sqrt{\frac{1}{K_1} \sum_{m \in \mathcal{E}_{\text{vid}}} \Big(\mathbf{z}_{\text{vid}}^{(m)} - \boldsymbol{\mu}_{\text{vid}}^{(k)}\Big)^2}. \tag{4}$$

Similarly, from all $M \times N$ value evaluations, we select the top K_2 samples with the highest SNR and update the value noise distribution:

$$\boldsymbol{\mu}_{\text{val}}^{(k)} = \frac{1}{K_2} \sum_{(m,n) \in \mathcal{E}_{\text{val}}} \mathbf{z}_{\text{val}}^{(m,n)}, \text{ and } \boldsymbol{\sigma}_{\text{val}}^{(k)} = \sqrt{\frac{1}{K_2} \sum_{(m,n) \in \mathcal{E}_{\text{val}}} \Big(\mathbf{z}_{\text{val}}^{(m,n)} - \boldsymbol{\mu}_{\text{val}}^{(k)}\Big)^2}. \tag{5}$$

To prevent distribution collapse and improve iteration stability, we apply variance decay and exponential smoothing: $\boldsymbol{\mu}^{(k)} \leftarrow \alpha \boldsymbol{\mu}^{(k)} + (1-\alpha)\boldsymbol{\mu}^{(k-1)}$, $\boldsymbol{\sigma}^{(k)} \leftarrow \beta \boldsymbol{\sigma}^{(k)} + (1-\beta)\boldsymbol{\sigma}^{(k-1)}$ with smoothing parameters

$\alpha, \beta \in (0, 1)$. After K iterations, we sample latent variables from the optimized distributions $f_{\text{vid}}^{(K)}$ and $f_{\text{val}}^{(K)}$, and obtain the corresponding video and value features via flow denoising. These features are finally fused by the action decoding module to predict decision-making actions. Algo. 1 summarizes the proposed latent planning procedure.

5 Experiments

In this section, we comprehensively explore how the proposed WAV model effectively implements latent trajectory planning, thereby improving the performance of visual language manipulation in general scenarios. To this end, our experiments aim to investigate the following questions: **1)** Compared with baseline algorithms, does WAV exhibit better generalization ability and higher success rate when performing robot manipulation tasks? **2)** Can WAV effectively generalize to complex long-horizon real-world manipulation tasks? **3)** How important is latent trajectory planning based on the world model to the overall generalization performance of WAV model? **4)** What are the inference efficiency and memory overhead of WAV model?

5.1 Generalization Comparison in Simulation

Simulation setup. We evaluate the WAV on the LIBERO benchmark [Liu et al., 2023], a comprehensive suite for multi-task and lifelong robotic manipulation. LIBERO consists of four task suites designed to probe different aspects of generalization. **Spatial** evaluates spatial reasoning by varying scene layouts while keeping object identities fixed. **Object** tests object-level generalization by changing object instances within a fixed environment. **Goal** assesses goal-conditioned behavior by varying task goals under the same scene configuration. Finally, **Long** includes long-horizon compositional tasks involving diverse objects, layouts, and goals, posing significant challenges for temporal consistency and multi-step reasoning. A single WAV model is deployed across four suites.

Table 1: Performance comparison on the LIBERO benchmark.

Models	Params	LIBERO				Avg.
		Spatial	Object	Goal	Long	
Diffusion Policy Chi et al. [2023]	-	78.3	92.5	68.3	50.5	72.4
Octo Team et al. [2024]	-	78.9	85.7	84.6	51.1	75.1
OpenVLA Kim et al. [2024]	7b	84.9	88.4	79.2	53.7	76.5
SpatialVLA Qu et al. [2025]	4b	88.2	89.9	78.6	55.5	78.1
DiT Policy Hou et al. [2024]	-	84.2	96.3	85.4	63.8	82.4
π_0-FAST Pertsch et al. [2025]	3b	96.4	96.8	88.6	60.2	85.5
GR00T-N1 Bjorck et al. [2025]	2b	94.4	97.6	93.0	90.6	93.9
π_0 Black et al. [2024]	3b	96.8	98.8	95.8	85.2	94.2
VLA-0 Goyal et al. [2025]	3b	97.0	97.8	96.2	87.6	94.7
Discrete Diffusion VLA Liang et al. [2025]	7b	97.2	98.6	97.4	92.0	96.3
MemoryVLA Shi et al. [2025]	7b	98.4	98.4	96.4	93.4	96.5
$\pi_{0.5}$ Intelligence et al. [2025]	3b	98.8	98.2	98.0	92.4	96.8
OpenVLA-OFT Kim et al. [2025]	7b	97.6	98.4	97.9	94.5	97.1
VLA-Adapter Wang et al. [2025]	0.5b	97.8	99.2	97.2	**95.0**	97.3
World Model						
WorldVLA Cen et al. [2025c]	7b	85.6	89.0	82.6	59.0	79.1
CoT-VLA Zhao et al. [2025]	7b	87.5	91.6	87.6	69.0	81.1
FlowVLA Zhong et al. [2025]	8.5b	93.2	95.0	91.6	72.6	88.1
DreamVLA Zhang et al. [2025c]	3b	97.5	94.0	89.5	89.5	92.6
UD-VLA Chen et al. [2025]	-	94.1	95.7	91.2	89.6	92.7
RynnVLA-002-Discrete Cen et al. [2025a]	7b	94.2	96.8	94.6	87.6	93.3
UniVLA Bu et al. [2025b]	7b	95.4	98.8	93.5	94.0	95.5
$\mathcal{F}_1$Lv et al. [2025]	4.5b	98.2	97.8	95.4	91.3	95.7
GE-ACT Liao et al. [2025]	2b	98.2	97.6	95.8	94.4	96.5
WAV (Ours)	2.2b	**99.6**	**100.0**	**98.6**	94.4	**98.1**
W/o Latent Trajectory Planning		99.0	99.6	95.0	91.8	96.4

Results. Tab. 1 reports results on the LIBERO benchmark. Our method achieves state-of-the-art (or competitive) performance across all task suites, with an overall average score of 98.1. These results highlight its effectiveness in handling long-horizon and compositional tasks, where efficient trajectory-level inference is critical. To assess the contribution of implicit planning, we perform an

ablation by removing latent trajectory planning. This leads to an average performance drop of 1.7 points, with the largest degradation on the **Long** suite, decreasing from 94.4 to 91.8. This trend underscores the importance of trajectory-level planning for robust and generalizable task execution. Notably, the performance gap widens with increasing task horizon, suggesting that implicit planning is particularly beneficial for mitigating compounding errors in long-horizon scenarios.

5.2 Real-World Evaluation

Real-world setup. We evaluate WAV on a real-world dual-arm robotic platform, Piper, across a set of challenging manipulation tasks designed to probe key policy capabilities, including spatial reasoning, deformable object manipulation, visual grounding, and long-horizon planning. Representative tasks include **bowl organization**, **towel flattening**, and a **long-horizon drawer task** that requires opening a drawer, placing an object inside, and closing it. For a fair comparison, we select GE-ACT [Liao et al., 2025] as the baseline model, as it uses the same video pre-trained model and has a similar network architecture and parameter count. Following standard practice, we adopt a strict binary success metric: a trial is considered successful only if the task is fully completed, with no partial credit. More details are in Appendix Sec. C.

Results. As shown in Fig. 2, WAV achieved a significantly higher success rate across all tasks than the baseline, with the average success rate increasing from 35.6% to 75.6%. Qualitative results (Fig. 3) reveal distinct failure modes of the baseline. GE-ACT frequently exhibits inaccurate action execution and weak spatial grounding, such as misaligning with the drawer handle or failing to grasp objects reliably. These errors accumulate over multiple steps, leading to cascading failures in long-horizon tasks. In contrast, WAV produces more consistent and coherent multi-step behaviors. By implicitly planning over future trajectories, it anticipates downstream consequences of actions, resulting in more accurate interactions and improved robustness. This advantage is most pronounced in tasks requiring multi-step coordination, highlighting the role of implicit planning in mitigating error accumulation in real-world settings.

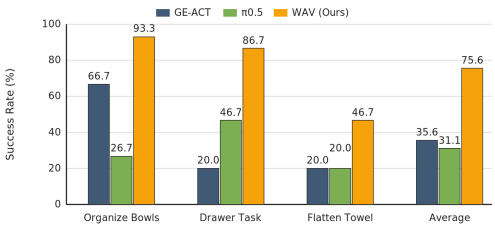

Figure 2: Quantitative comparison of WAV with GE-ACT and $\pi_{0.5}$ on real-world tasks. Each result is averaged over 15 trials.

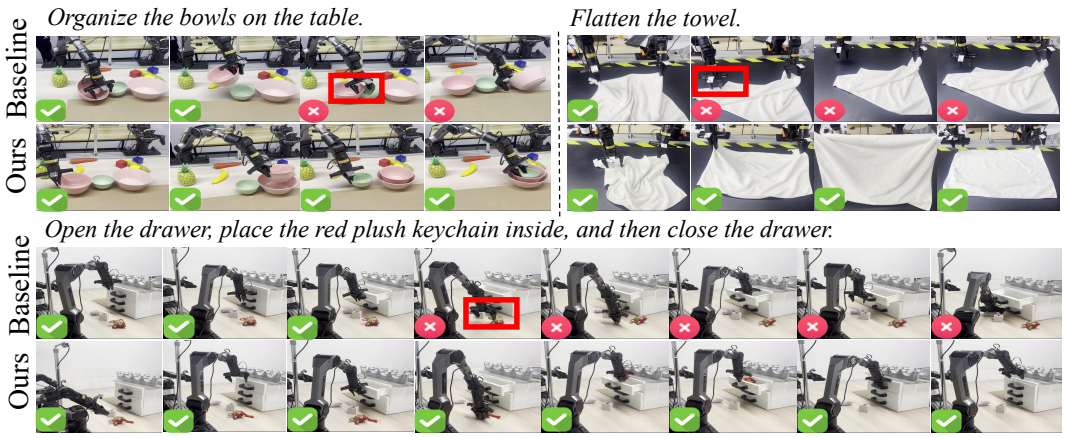

Figure 3: Qualitative comparison between WAV model (Ours) and the GE-ACT (Baseline) across different real-world manipulation tasks.

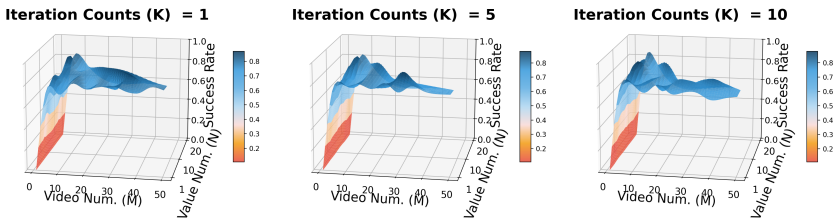

Figure 4: WAV performance under varying iteration counts K, video latent noise sampling numbers M, and trajectory value noise sampling numbers N.

5.3 Ablation Study

Effect of Iteration Counts K and Sampling Sizes M, N. As shown in Fig. 4, increasing K from 1 to 5 leads to a clear improvement in success rate, indicating that iterative latent planning effectively refines the trajectory distribution toward high-value regions. However, further increasing K to 10 yields only marginal gains, suggesting diminishing returns from excessive iterations. In contrast, the performance is highly sensitive to M, where increasing the number of video samples results in a steep improvement, especially in the low-sample regime. This highlights the importance of sufficiently exploring future trajectory hypotheses. The influence of N is comparatively milder, with performance saturating earlier, indicating that dense value evaluation provides limited additional benefit once a reasonable estimation is achieved. More results are given in Appendix Fig. 7. Appendix Fig. 8 and 9 compare trajectory value and video predictions to ground truth, respectively.

Effect of Smoothing Parameters (α, β). Fig. 5 (**Left**) shows that when α is small, the success rate drops sharply, indicating that insufficient smoothing leads to unstable distribution updates and poor convergence. As α increases, performance improves rapidly and then plateaus, showing a clear steep region followed by saturation. A similar but slightly milder trend is observed for β. This suggests that stabilizing the update of the latent distribution is crucial, but overly strong smoothing provides limited additional benefit once convergence is achieved.

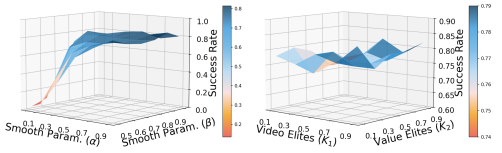

Figure 5: **Left:** WAV sensitivity to smoothing parameters α and β. **Right:** WAV sensitivity to elite counts K_1 and K_2.

Effect of Elite Selection (K_1, K_2). Fig. 5 (**Right**) shows that the performance is relatively stable across different choices of K_1 and K_2, indicating that the WAV is not highly sensitive to these parameters. However, extremely small values lead to noticeable performance degradation, suggesting that a sufficient number of elite samples is necessary for stable distribution updates.

Performance–Efficiency Trade-off. Fig. 6 shows that increasing the number of iterations from 0 to 3 consistently improves performance. Beyond 3 gains saturate while computational cost, particularly inference time, continues to grow. Therefore, $K = 3$ provides a favorable balance between effectiveness and efficiency, achieving strong performance with moderate computational overhead. This behavior reflects the diminishing returns of iterative refinement, indicating that a small number of inference steps is sufficient to capture most of the planning benefits.

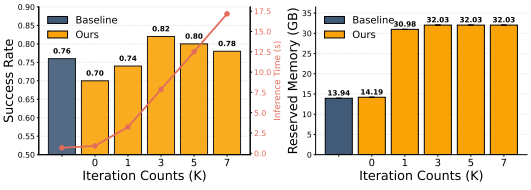

Figure 6: Performance and efficiency trade-off of WAV across K: success rate and inference time (**Left**), and GPU memory usage (**Right**). Baseline: GE-ACT. Ours: WAV.

6 Conclusion

In this work, we introduced **WAV**, a unified framework for integrating perception, reasoning, and action in embodied agents. The key innovation lies in latent trajectory planning, which maintains a compact representation of plausible futures and reshapes the search distribution toward high-value, low-uncertainty regions, thereby significantly reducing the complexity of long-horizon planning. Empirical results demonstrate that WAV consistently improves task success, generalization, and robustness over previous baselines across simulation and real-world settings. The main limitation of current work is the time and storage overhead required for model deployment. Future directions include extending the framework to richer multi-modal instructions and enabling real-time closed-loop deployment on physical robotic systems. More broadly, our results suggest that planning can be naturally realized as an inference process within structured generative models, offering a promising direction for unifying prediction, evaluation, and control.

Acknowledgments

This work was supported by the Brain Science and Brain-like Intelligence Technology — National Science and Technology Major Project (Grant No. 2022ZD0208800).

References

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model, 2025a. URL `https://arxiv.org/abs/2512.13030`.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. *arXiv preprint arXiv:2512.13030*, 2025b.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al.
pi_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Learning to Act Anywhere with Task-centric Latent Actions. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025a. doi: 10.15607/RSS.2025.XXI.014.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025b.

Jun Cen, Siteng Huang, Yuqian Yuan, Hangjie Yuan, Chaohui Yu, Yuming Jiang, Jiayan Guo, Kehan Li, Hao Luo, Fan Wang, et al. Rynnvla-002: A unified vision-language-action and world model. *arXiv preprint arXiv:2511.17502*, 2025a.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. Worldvla: Towards autoregressive action world model, 2025b. URL https://arxiv.org/abs/2506.21539.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025c.

Jiayi Chen, Wenxuan Song, Pengxiang Ding, Ziyang Zhou, Han Zhao, Feilong Tang, Donglin Wang, and Haoang Li. Unified diffusion vla: Vision-language-action model via joint discrete denoising diffusion process. *arXiv preprint arXiv:2511.01718*, 2025.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

Marc Deisenroth and Carl E Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472, 2011.

Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335–348, 1989.

Ankit Goyal, Hugo Hadfield, Xuning Yang, Valts Blukis, and Fabio Ramos. Vla-0: Building state-of-the-art vlas with zero modification. *arXiv preprint arXiv:2510.13054*, 2025.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. *arXiv preprint arXiv:2510.10125*, 2025.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019a.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR, 2019b.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. *arXiv preprint arXiv:2203.04955*, 2022.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. *arXiv preprint arXiv:2310.16828*, 2023.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Learning massively multitask world models for continuous control. *arXiv preprint arXiv:2511.19584*, 2025.

Zhi Hou, Tianyi Zhang, Yuwen Xiong, Hengjun Pu, Chengyang Zhao, Ronglei Tong, Yu Qiao, Jifeng Dai, and Yuntao Chen. Diffusion transformer policy. *arXiv preprint arXiv:2410.15959*, 2024.

Chia-Yu Hung, Navonil Majumder, Haoyuan Deng, Liu Renhang, Yankang Ang, Amir Zadeh, Chuan Li, Dorien Herremans, Ziwei Wang, and Soujanya Poria. Nora-1.5: A vision-language-action model trained using world model-and action-based preference rewards. *arXiv preprint arXiv:2511.14659*, 2025.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. *arXiv preprint arXiv:2504.16054*, 2025.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32, 2019.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Hengtao Li, Pengxiang Ding, Runze Suo, Yihao Wang, Zirui Ge, Dongyuan Zang, Kexian Yu, Mingyang Sun, Hongyin Zhang, Donglin Wang, et al. Vla-rft: Vision-language-action reinforcement fine-tuning with verified rewards in world simulators. *arXiv preprint arXiv:2510.00406*, 2025.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

Zhixuan Liang, Yizhuo Li, Tianshuo Yang, Chengyue Wu, Sitong Mao, Tian Nian, Liuao Pei, Shunbo Zhou, Xiaokang Yang, Jiangmiao Pang, et al. Discrete diffusion vla: Bringing discrete diffusion to action decoding in vision-language-action policies. *arXiv preprint arXiv:2508.20072*, 2025.

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Jingbin Cai, Si Liu, Jianlan Luo, et al. Genie envisioner: A unified world foundation platform for robotic manipulation. *arXiv preprint arXiv:2508.05635*, 2025.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Qi Lv, Weijie Kong, Hao Li, Jia Zeng, Zherui Qiu, Delin Qu, Haoming Song, Qizhi Chen, Xiang Deng, and Jiangmiao Pang. F1: A vision-language-action model bridging understanding and generation to actions. *arXiv preprint arXiv:2509.06951*, 2025.

Matthew T Mason. *Mechanics of robotic manipulation*. MIT press, 2001.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2508.19236*, 2025.

Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

Yinzhou Tang, Yu Shang, Yinuo Chen, Bingwen Wei, Xin Zhang, Shu'ang Yu, Liangzhi Shi, Chao Yu, Chen Gao, Wei Wu, et al. Roboscape-r: Unified reward-observation world models for generalizable robotics training via rl. *arXiv preprint arXiv:2512.03556*, 2025.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Yihao Wang, Pengxiang Ding, Lingxiao Li, Can Cui, Zirui Ge, Xinyang Tong, Wenxuan Song, Han Zhao, Wei Zhao, Pengxu Hou, et al. Vla-adapter: An effective paradigm for tiny-scale vision-language-action model. *arXiv preprint arXiv:2509.09372*, 2025.

Grady Williams, Andrew Aldrich, and Evangelos Theodorou. Model predictive path integral control using covariance variable importance sampling. *arXiv preprint arXiv:1509.01149*, 2015.

Grady Williams, Paul Drews, Brian Goldfain, James M Rehg, and Evangelos A Theodorou. Information-theoretic model predictive control: Theory and applications to autonomous driving. *IEEE Transactions on Robotics*, 34(6):1603–1622, 2018.

Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. ivideogpt: Interactive videogpts are scalable world models. *Advances in Neural Information Processing Systems*, 37:68082–68119, 2024.

Junjin Xiao, Yandan Yang, Xinyuan Chang, Ronghan Chen, Feng Xiong, Mu Xu, Wei-Shi Zheng, and Qing Zhang. World-env: Leveraging world model as a virtual environment for vla post-training. *arXiv preprint arXiv:2509.24948*, 2025.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Hongyin Zhang, Pengxiang Ding, Shangke Lyu, Ying Peng, and Donglin Wang. Gevrm: Goal-expressive video generation model for robust visual manipulation. *arXiv preprint arXiv:2502.09268*, 2025a.

Hongyin Zhang, Zifeng Zhuang, Han Zhao, Pengxiang Ding, Hongchao Lu, and Donglin Wang. Reinbot: Amplifying robot visual-language manipulation with reinforcement learning. *arXiv preprint arXiv:2505.07395*, 2025b.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, Fan Lu, He Wang, et al. Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. *arXiv preprint arXiv:2507.04447*, 2025c.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1702–1713, 2025.

Zhide Zhong, Haodong Yan, Junfeng Li, Xiangchen Liu, Xin Gong, Tianran Zhang, Wenxuan Song, Jiayi Chen, Xinhu Zheng, Hesheng Wang, et al. Flowvla: Visual chain of thought-based motion reasoning for vision-language-action models. *arXiv preprint arXiv:2508.18269*, 2025.

Fangqi Zhu, Zhengyang Yan, Zicong Hong, Quanxin Shou, Xiao Ma, and Song Guo. Wmpo: World model-based policy optimization for vision-language-action models. *arXiv preprint arXiv:2511.09515*, 2025.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.

A Theoretical Analysis of Latent Planning

In this section, we provide a principled explanation for why latent planning is substantially more effective than direct action-space sampling in long-horizon Vision–Language–Action (VLA) tasks. Our analysis is not intended as a tight optimality guarantee, but rather as a geometric and probabilistic characterization of the search difficulty inherent to high-dimensional trajectory spaces, and how latent planning alleviates it by reweighting probability mass toward feasible trajectories.

A.1 Setup

Let the environment be a Markov decision process $(\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma)$. A planning horizon-$H$ trajectory lies in the space

$$\mathcal{X} = \mathcal{S}^H \times \mathcal{A}^H,$$

whose ambient dimension grows linearly with H. We denote by $\mathcal{M}_{\text{traj}} \subset \mathcal{X}$ the set of trajectories that satisfy physical dynamics, contact constraints, and semantic task requirements. Throughout, we treat $\mathcal{M}_{\text{traj}}$ as a thin subset of $\mathcal{X}$ with low intrinsic dimension, consistent with the manifold hypothesis commonly assumed in robotics and control.

We consider a learned latent trajectory generator $\mathcal{W}_\theta : \mathbb{R}^{d_z} \to \mathcal{X}$, which maps a latent variable z to a full trajectory,

$$\tau_{t:t+H} = \mathcal{W}_\theta(z), \qquad z \sim f_\theta(s_t),$$

where $f_\theta(s_t)$ is a state-conditioned latent distribution. The induced trajectory distribution is the pushforward measure

$$P_{\text{latent}} = (\mathcal{W}_\theta)_{\#} f_\theta.$$

A.2 Vanishing Feasible Mass in Trajectory Space

We first formalize the difficulty of direct action-space sampling by characterizing how the probability mass of feasible trajectories shrinks with the planning horizon.

Lemma A.1 (Vanishing Feasible Mass in Trajectory Space)**.** *Let $\mathcal{X} = \mathcal{S}^H \times \mathcal{A}^H$ be a bounded trajectory space with ambient dimension*

$$D = H(\dim \mathcal{S} + \dim \mathcal{A}),$$

equipped with a reference measure μ. Let $\mathcal{M}_{\text{traj}} \subset \mathcal{X}$ be a compact subset with intrinsic dimension $d \ll D$. Then, for any fixed tolerance $\epsilon \in (0, 1)$, there exists a constant $c > 0$ such that

$$\frac{\mu\big(\mathcal{N}_\epsilon(\mathcal{M}_{\text{traj}})\big)}{\mu(\mathcal{X})} \leq \exp(-cH), \tag{6}$$

where $\mathcal{N}_\epsilon(\mathcal{M}_{\text{traj}})$ denotes the ϵ-neighborhood of $\mathcal{M}_{\text{traj}}$.

Proof. We work in the bounded trajectory space $\mathcal{X} \subset \mathbb{R}^D$ equipped with the Lebesgue measure μ. Let $\mathcal{M}_{\text{traj}} \subset \mathcal{X}$ be compact with intrinsic dimension d.

Fix any $\epsilon \in (0, 1)$ and define

$$\mathcal{N}_\epsilon(\mathcal{M}_{\text{traj}}) = \left\{ x \in \mathcal{X} \mid \inf_{y \in \mathcal{M}_{\text{traj}}} \|x - y\|_2 \leq \epsilon \right\}.$$

By standard covering number arguments for compact sets, $\mathcal{M}_{\text{traj}}$ can be covered by at most $N_\epsilon \leq C_1 \epsilon^{-d}$ Euclidean balls of radius ϵ, where $C_1 > 0$ is independent of H. Thus,

$$\mathcal{N}_\epsilon(\mathcal{M}_{\text{traj}}) \subset \bigcup_{i=1}^{N_\epsilon} B(y_i, 2\epsilon).$$

Each such ball has D-dimensional volume proportional to ϵ^D. Up to multiplicative factors that grow at most subexponentially with D, we obtain

$$\mu(\mathcal{N}_\epsilon(\mathcal{M}_{\text{traj}})) \lesssim \epsilon^{D-d}.$$

Since $\mathcal{X}$ is bounded, $\mu(\mathcal{X}) < \infty$. Moreover, the ambient dimension grows linearly with the horizon,

$$D = H(\dim \mathcal{S} + \dim \mathcal{A}),$$

while the intrinsic dimension of feasible trajectories is governed by physical and semantic constraints. We assume that

$$d \leq \lambda H \quad \text{for some } \lambda < \dim \mathcal{S} + \dim \mathcal{A},$$

so that there exists $\kappa > 0$ satisfying $D - d \geq \kappa H$.

Since $\log \epsilon < 0$, it follows that

$$\epsilon^{D-d} = \exp\big((D-d)\log \epsilon\big) \leq \exp\big(-\kappa H |\log \epsilon|\big).$$

Absorbing constant and subexponential terms into the exponent, there exists $c > 0$ such that

$$\frac{\mu(\mathcal{N}_\epsilon(\mathcal{M}_{\text{traj}}))}{\mu(\mathcal{X})} \leq \exp(-cH),$$

which completes the proof. □

Discussion. Lemma A.1 states that, under uniform exploration of the trajectory space, the probability of encountering even an approximately feasible trajectory decays exponentially with the planning horizon. This phenomenon arises from the mismatch between the rapidly growing ambient dimension and the low intrinsic dimensionality of feasible behaviors, and captures a fundamental scaling challenge faced by long-horizon action-space planning.

A.3 Latent Planning as Probability Mass Reweighting

We now contrast action-space sampling with latent-space planning. Rather than comparing absolute volumes, we compare the probability mass assigned to feasible trajectories under different induced distributions.

Proposition A.2 (Latent Reweighting of the Search Distribution)**.** *Let $\mathcal{M}_{\text{traj}} \subset \mathcal{X}$ denote the feasible trajectory set. Consider a latent generator $\mathcal{W}_\theta : \mathbb{R}^{d_z} \to \mathcal{X}$ and a state-conditioned latent distribution $f_\theta(s_t)$. Assume that the learned model approximately parameterizes feasible trajectories in the sense that there exists $\delta \in (0,1)$ such that*

$$P_{\text{latent}}(\mathcal{M}_{\text{traj}}) = \Pr_{z \sim f_\theta(s_t)}[\mathcal{W}_\theta(z) \in \mathcal{M}_{\text{traj}}] \geq 1 - \delta. \tag{7}$$

Let $\Phi : \mathcal{A}^H \to \mathcal{X}$ denote the trajectory rollout map induced by the system dynamics. Then the ratio of feasible-trajectory probabilities satisfies

$$\frac{\Pr_{z \sim f_\theta(s_t)}[\mathcal{W}_\theta(z) \in \mathcal{M}_{\text{traj}}]}{\Pr_{a_{t:t+H} \sim \text{Unif}(\mathcal{A}^H)}[\Phi(a_{t:t+H}) \in \mathcal{M}_{\text{traj}}]} \geq \exp(cH)(1-\delta), \tag{8}$$

where $c > 0$ is the constant from Lemma A.1. Consequently, latent planning assigns exponentially larger probability mass to feasible trajectories relative to direct action-space sampling as the horizon grows.

Proof. Uniform sampling over $\mathcal{A}^H$ induces a trajectory distribution via the rollout map Φ. By Lemma A.1, the probability of sampling an ϵ-feasible trajectory satisfies

$$\Pr_{a_{t:t+H} \sim \text{Unif}(\mathcal{A}^H)}[\Phi(a_{t:t+H}) \in \mathcal{M}_{\text{traj}}] \leq \exp(-cH).$$

On the other hand, latent sampling induces the pushforward distribution

$$P_{\text{latent}} = (\mathcal{W}_\theta)_{\#} f_\theta.$$

By assumption,

$$\Pr_{z \sim f_\theta(s_t)}[\mathcal{W}_\theta(z) \in \mathcal{M}_{\text{traj}}] \geq 1 - \delta.$$

Taking the ratio of the two probabilities yields the stated result. □

Discussion. Proposition A.2 is a conditional comparison: given a learned latent representation that approximately captures the feasible trajectory set, latent planning avoids the exponential decay of probability mass that plagues action-space search. This advantage arises from reweighting the search distribution toward dynamically and semantically valid regions, rather than from exhaustive exploration of a lower-dimensional space.

Proposition A.2 establishes that latent planning reweights the search distribution toward the feasible trajectory manifold, thereby mitigating the exponential infeasibility of action-space exploration. Nevertheless, feasibility alone is insufficient for long-horizon planning, as the feasible set still contains trajectories with widely varying returns. The following corollary formalizes the necessity of iterative latent planning.

A.4 Necessity of Iterative Latent Planning

Corollary A.3 (Necessity of Iterative Latent Planning)**.** *Let $\mathcal{M}_{\mathrm{traj}} \subset \mathcal{X}$ denote the feasible trajectory set, and let*

$$V(\tau) = \sum_{h=0}^{H-1} \gamma^h r(s_{t+h}, a_{t+h})$$

denote the cumulative discounted return of a trajectory τ, where r is a bounded per-step reward and $\gamma \in (0, 1]$. Let $\mathcal{W}_\theta : \mathbb{R}^{d_z} \to \mathcal{X}$ be a learned latent trajectory generator inducing the distribution $P_{\mathrm{latent}} = (\mathcal{W}_\theta)_{\#} f_\theta$.

Assume that

$$P_{\mathrm{latent}}(\mathcal{M}_{\mathrm{traj}}) \geq 1 - \delta \quad \textit{for some } \delta \in (0, 1).$$

Then, for any fixed sample budget N that grows subexponentially with the planning horizon H, one-shot sampling from P_{latent} does not, in general, guarantee with high probability the discovery of a near-optimal trajectory. Iterative inference procedures that adaptively reweight the latent distribution are therefore necessary to concentrate probability mass on high-value feasible trajectories.

Proof. Define the optimal feasible trajectory value

$$V^\star = \sup_{\tau \in \mathcal{M}_{\mathrm{traj}}} V(\tau).$$

For any $\varepsilon > 0$, define the ε-optimal feasible set

$$\mathcal{M}_\varepsilon = \{\tau \in \mathcal{M}_{\mathrm{traj}} \mid V(\tau) \geq V^\star - \varepsilon\}.$$

By construction, $\mathcal{M}_\varepsilon$ is a measurable subset of $\mathcal{M}_{\mathrm{traj}}$. While the latent distribution assigns non-vanishing probability mass to the feasible set $\mathcal{M}_{\mathrm{traj}}$, this assumption alone does not imply any uniform lower bound on $P_{\mathrm{latent}}(\mathcal{M}_\varepsilon)$. In particular, the probability mass of P_{latent} may be dispersed across trajectories with widely varying values.

Let $\tau^{(1)}, \ldots, \tau^{(N)}$ be N independent samples drawn from P_{latent}. The probability of observing at least one ε-optimal trajectory is

$$\Pr\left[\exists i \leq N : \tau^{(i)} \in \mathcal{M}_\varepsilon\right] = 1 - (1 - P_{\mathrm{latent}}(\mathcal{M}_\varepsilon))^N.$$

If N grows subexponentially with H, then unless $P_{\mathrm{latent}}(\mathcal{M}_\varepsilon)$ is bounded below by a constant independent of H, the above probability cannot be guaranteed to remain bounded away from zero as the planning horizon increases. Therefore, without additional mechanisms that adaptively reweight the latent distribution toward higher-value regions, one-shot latent sampling does not provide a reliable strategy for long-horizon optimization.

Iterative inference methods, such as importance-weighted resampling, cross-entropy optimization, or gradient-based refinement in latent space, explicitly modify the sampling distribution based on observed returns, thereby increasing the probability mass assigned to $\mathcal{M}_\varepsilon$ over successive iterations. This establishes the necessity of iterative latent planning. □

B Additional Experimental Results

This section provides additional experiments that further analyze the source of the performance gains, the robustness of the value-guided planning mechanism, and the inference-time computation–performance trade-off.

B.1 Attribution of Performance Gains

To isolate the source of the performance gains, we conduct a controlled attribution study on LIBERO. All variants use the same frozen video, value, and action modules, and differ only in their inference-time strategies. Each adjacent comparison changes a single component, allowing us to separately evaluate the effects of additional sampling, value-based selection, and iterative refinement.

Inference strategy	Value	Iter.	Fwd.	Spatial	Object	Goal	Long	Avg.
One-shot sampling ($K = 0$)	–	–	$1\times$	99.0	99.6	95.0	91.8	96.4
Random selection	–	–	$M\times$	99.0	99.0	96.8	92.2	96.7
Best-of-N, mean-value selection	✓	–	$M\times$	99.4	99.4	96.4	93.0	97.0
Best-of-N, max-SNR selection	✓	–	$M\times$	99.0	98.8	97.6	93.6	97.2
Compute-matched Best-of-N	✓	–	$KM\times$	99.0	98.8	97.6	93.6	97.2
WAV w/o value conditioning	–	✓	$KM\times$	99.6	99.0	96.0	93.8	97.1
WAV (full, $K = 3$)	✓	✓	$KM\times$	**99.6**	**100.0**	**98.6**	**94.4**	**98.1**

Table 2: Attribution study of WAV on LIBERO. All variants use identical frozen video, value, and action modules; only the inference strategy differs.

The attribution ladder reveals four observations. First, additional sampling alone provides little benefit: random selection improves the Long suite by only 0.4 points over one-shot sampling (92.2 vs. 91.8). Second, learned trajectory values provide a meaningful selection signal, with both mean-value and max-SNR selection outperforming random selection. The max-SNR criterion further improves Long and Goal over mean-value selection.

Third, the gain of WAV cannot be explained by Best-of-N or additional sampling alone. Under the same number of forward passes, WAV improves over compute-matched Best-of-N by 0.8 points on Long and 1.0 points on Goal. Moreover, increasing the one-shot candidate pool from M to KM does not improve performance.

Finally, removing value conditioning from iterative refinement reduces Goal performance from 98.6 to 96.0 while retaining the iterative procedure. This isolates the contribution of value guidance from the iteration mechanism itself. Overall, WAV improves the average score from 96.4 to 98.1, with a larger gain of 2.6 points on the Long suite.

B.2 Robustness to Reward Design

We further investigate whether WAV depends critically on the particular dense reward formulation used to train the value module. We retrain only the value module using a sparse binary success reward while keeping all other components unchanged. The dense reward uses only RGB observations and proprioceptive signals available on the robot, as described in Sec. C.2.

Value training reward	Signal source	Spatial	Object	Goal	Long	Avg.
Rule-based dense	RGB + proprioception	99.6	100.0	98.6	94.4	98.1
Sparse success only	Task outcome	98.8	99.4	98.0	97.4	98.4
w/o planning	–	99.0	99.6	95.0	91.8	96.4

Table 3: Robustness to reward design on LIBERO. Only the value module is retrained; all other components remain unchanged.

The sparse-reward variant achieves 98.4 average performance, comparable to the dense-reward variant (98.1) and above the no-planning reference (96.4). This indicates that the effectiveness of value-guided iterative planning is not tied to the specific dense reward formulation. The dense reward provides higher performance on Spatial and Object, whereas the sparse variant performs better on Long. Overall, both reward designs retain a clear advantage over the no-planning reference.

B.3 Inference Cost and Accuracy–Compute Trade-off

WAV exposes K as a test-time computation–performance trade-off. For each planning call, the model predicts an action chunk of eight steps and executes the chunk before the next planning call. Thus, the effective control frequency is $8/T$, where T is the planning latency.

K	M	N	Mean Lat. (s)	P95 Lat. (s)	Freq. (Hz)	Peak Mem. (GB)	LIBERO-10
0	1	1	0.909	0.970	8.80	14.2	70
1	30	5	3.237	3.274	2.47	31.0	74
3	30	5	7.862	7.888	1.02	32.0	**82**
5	30	5	12.500	12.526	0.64	32.0	80

Table 4: Inference latency, memory, effective control frequency, and accuracy under different numbers of iterative updates K on a single NVIDIA A100 80 GB GPU. Each planning call predicts and executes an eight-step action chunk.

For example, at $K = 3$, the measured latency is 7.86 s, corresponding to an effective control frequency of approximately 1.02 Hz. On the harder LIBERO-10 subset, increasing K from 0 to 3 improves the average performance from 70 to 82, while increasing K further to 5 reduces performance to 80. This demonstrates that iterative refinement is not monotonically beneficial and that K serves as a test-time scaling parameter rather than a fixed computational cost.

The measured peak GPU memory remains approximately 32 GB for $K \geq 1$, while latency increases with the number of iterations. These results characterize the practical regime targeted by WAV: quasi-static, long-horizon manipulation rather than high-frequency reactive control.

B.4 Sensitivity to Inference Hyperparameters

We further report the sensitivity of WAV to the number of iterative updates K, video latent samples M, and trajectory value noise samples N. The results characterize the interaction between sampling and iterative refinement and provide additional guidance for selecting the inference-time configuration.

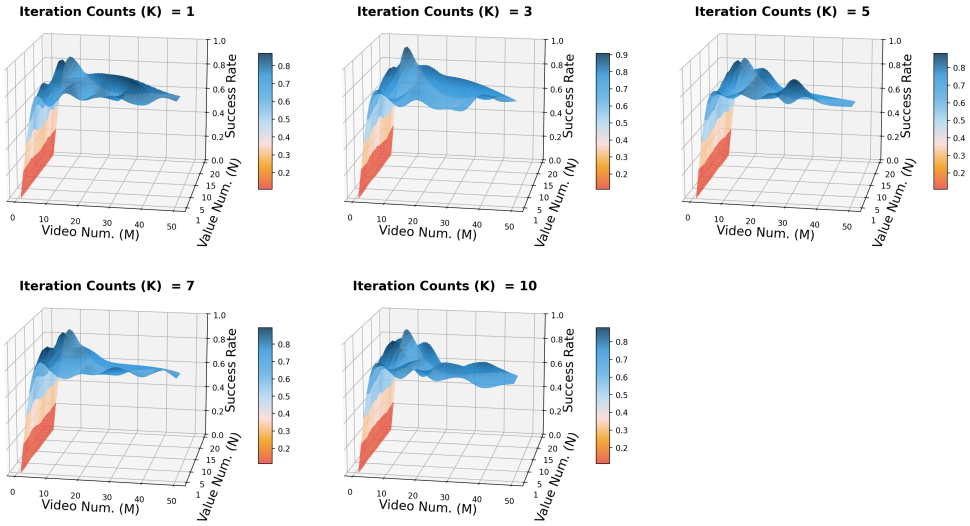

Figure 7: Performance variation trends of the WAV model under different iteration counts K, video latent noise sampling numbers M, and trajectory value noise sampling numbers N.

B.5 Value Prediction

We compare the inferred state-value trajectories with their corresponding ground-truth values in both real-world robot experiments and LIBERO. The value estimates capture the overall temporal trend of trajectory quality across both environments.

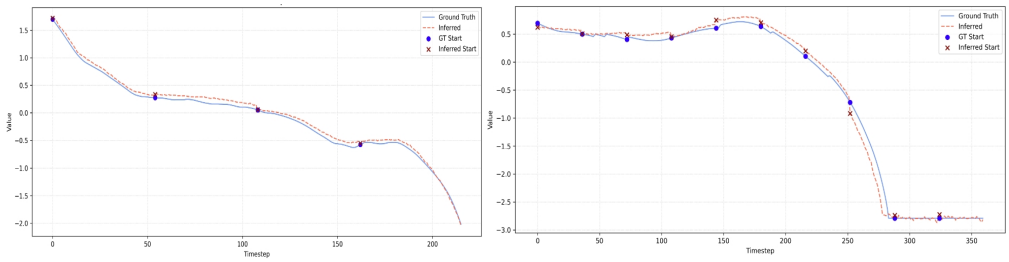

Figure 8: Comparison between predicted and ground-truth state-value trajectories in real-world robot experiments (left) and LIBERO (right).

B.6 Qualitative Video Comparison

We additionally compare the videos generated by the learned video model with the corresponding ground-truth observations on representative LIBERO tasks. The results are shown in Figure 9

C Implementation Details

C.1 Real-world Data Collection

To evaluate the WAV model in real-world settings, we collected approximately 300 successful trajectories each for the drawer-opening and bowl-organization tasks, and 2,000 successful trajectories for the towel-flattening task, using a real-world Piper robot.

Each trajectory lasts on the order of tens of seconds. Each trajectory contains RGB observations from two wrist-mounted cameras with resolution $240 \times 320 \times 3$ and one third-person top camera with resolution $240 \times 424 \times 3$, together with 14-dimensional robot joint states.

We train one task-specific model per task and evaluate each model on the corresponding real-world scenario. This setup enables a controlled and systematic evaluation across diverse manipulation challenges, while highlighting the robustness of the proposed WAV model in real-world deployment.

C.2 Reward Construction

For dense reward design, we follow ReinboT [Zhang et al., 2025b], which decomposes dense rewards into four aspects: sub-goal achievement, task progress, behavior smoothness, and task completion. In this work, we omit the sub-goal achievement term and retain the remaining components.

Concretely, our reward consists of nine reward terms, corresponding to 16 scalar components in the bimanual setting. The definitions and weights are listed in Tab. 5. The dataset we use will be publicly available upon publication.

Reward term	Definition	Weight
Wrist-view MSE reward	$c^b_{1,t} = \exp\Big(-0.01 \cdot \mathrm{MSE}(I^b_t, I^b_T)\Big)$	$+\frac{1}{16}$ each
Wrist-view SSIM reward	$c^b_{2,t} = \exp\Big(\mathrm{SSIM}(I^b_t, I^b_T) - 1\Big)$	$+\frac{1}{16}$ each
Top-view MSE reward	$c_{3,t} = \exp\Big(-0.01 \cdot \mathrm{MSE}(I^{\mathrm{top}}_t, I^{\mathrm{top}}_T)\Big)$	$+\frac{1}{16}$
Top-view SSIM reward	$c_{4,t} = \exp\Big(\mathrm{SSIM}(I^{\mathrm{top}}_t, I^{\mathrm{top}}_T) - 1\Big)$	$+\frac{1}{16}$
Joint-state proximity reward	$c^b_{5,t} = \exp\Big(-\lVert s^b_t - s^b_T \rVert_2\Big)$	$+\frac{1}{16}$ each
Joint velocity penalty	$c^b_{6,t} = \sum_j \lvert \Delta s^b_{t,j} \rvert$	$-\frac{1}{16}$ each
Joint acceleration penalty	$c^b_{7,t} = \sum_j \lvert \Delta^2 s^b_{t,j} \rvert$	$-\frac{1}{16}$ each
Action velocity penalty	$c^b_{8,t} = \sum_j \lvert \Delta a^b_{t,j} \rvert$	$-\frac{0.1}{16}$ each
Action acceleration penalty	$c^b_{9,t} = \sum_j \lvert \Delta^2 a^b_{t,j} \rvert$	$-\frac{0.1}{16}$ each

Table 5: Compact formulation of the dense reward terms in the bimanual setting, where $b \in \{L, R\}$ denotes the left and right arms.

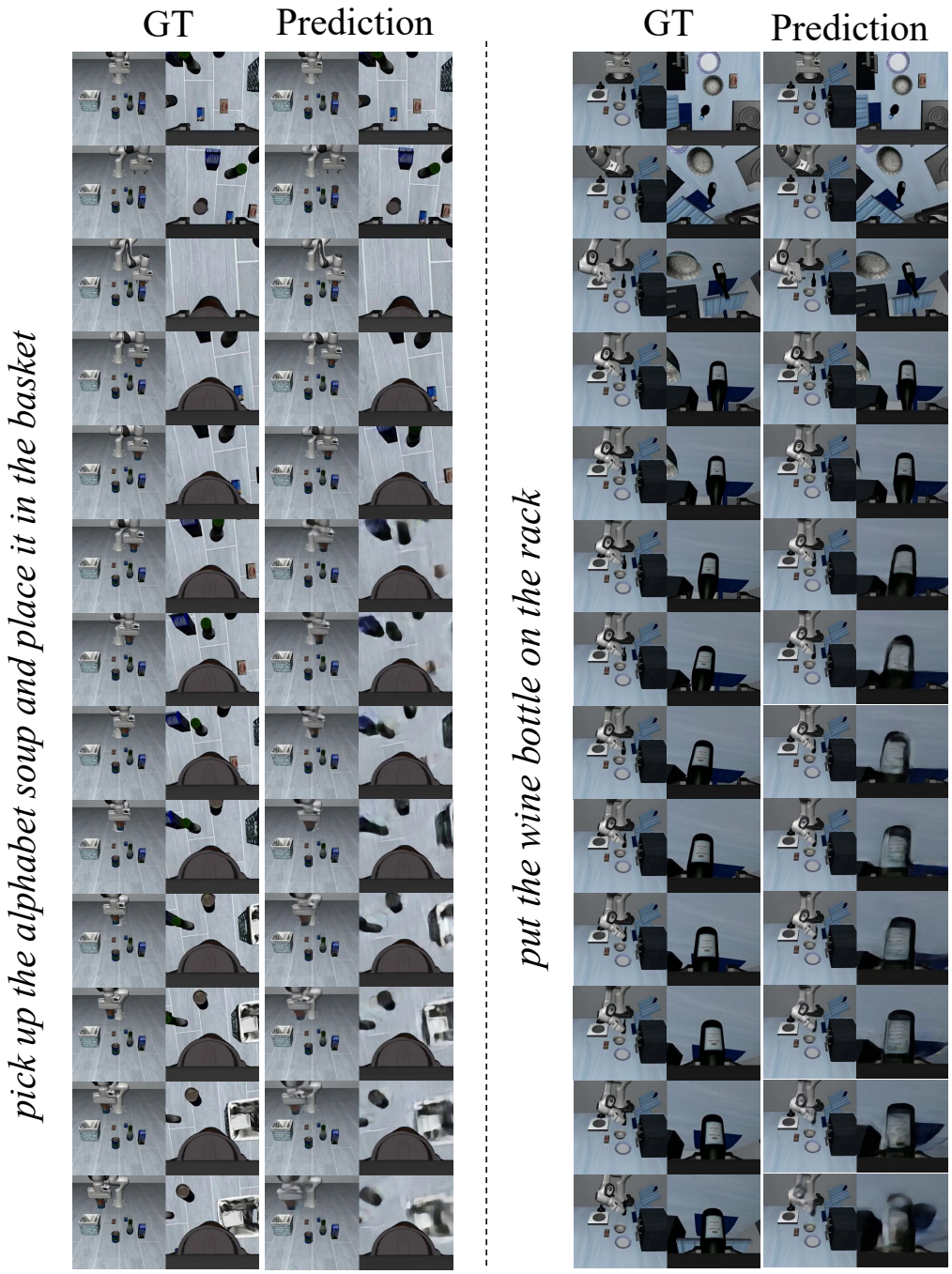

Figure 9: Qualitative comparison between predicted videos and ground-truth observations on two LIBERO tasks.

C.3 Training Configuration

The hyperparameter configurations for video training and value/action training are summarized in Tables 6 and 7.

Table 6: Video training hyperparameter configuration.

Parameter	Value
Gradient clip	1.0
Steps	40000
Warm-up steps	1000
Batch size	128
Learning rate	$3e-4$
Weight decay	$1e-5$
Caption Dropout	0.06
Optimizer	Adam ($\beta_1 = 0.9,\ \beta_2 = 0.95,\ \beta_3 = 0.999$)

Table 7: Value and action training hyperparameter configuration.

Parameter	Value
Gradient clip	1.0
Steps	30000
Warm-up steps	1000
Batch size	128
Learning rate	$5e-5$
Weight decay	$1e-5$
Caption Dropout	0
Optimizer	Adam ($\beta_1 = 0.9,\ \beta_2 = 0.95,\ \beta_3 = 0.999$)

C.4 Computational Resources

Both our method and the reproduced baseline are trained using 8 NVIDIA A100-SXM4-80GB GPUs on a server equipped with Intel Xeon Platinum 8358 CPUs @ 2.60GHz.

For the four LIBERO tasks, we perform full-parameter fine-tuning for 30,000 training steps on video and 40,000 training steps on value/action, which requires approximately 5 days.

For the real-world Piper manipulation data, we train one model per task for 30,000 training steps on video and 30,000 training steps on value/action, which requires approximately 3 days.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction accurately state the core contributions (WAV framework for implicit planning, theoretical proof, SOTA performance on simulation and real-world robot tasks), which are fully supported by the theoretical analysis (Section 4, Appendix A), methodology (Section 4), and experimental results (Section 5) in the paper.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section 6 explicitly discusses the main limitations, including time and storage overhead for model deployment

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: All theoretical results (Lemma 4.1, Proposition 4.2, Corollary 4.3) are equipped with complete assumptions and formal proofs. Assumptions are clearly stated in the main text, and full proofs are provided in Appendix A.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: The paper fully discloses all details for reproducing main results: experimental setups (LIBERO suites, real-world Piper platform), model architecture, training pipeline, evaluation metrics, and hyperparameters (Appendix B).Code is also provided

Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The paper provides open-source code. The LIBERO benchmark is public, and the real-world robot dataset will be released upon publication.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification:All training/test details are specified: data sources, splits, hyperparameters (Tables 3–4), optimizer (Adam), batch size, training steps, and hardware setup. Full details are in Appendix B.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: All experiments report mean results over multiple independent runs (15 trials for real-world tasks,50 trials for simulation environment).On the right side of Figure 6, the error bars represent the standard deviation from 50 trials.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The paper specifies compute hardware (8×NVIDIA A100-SXM4-80GB GPUs, Intel Xeon Platinum 8358 CPU), training time (5 days for LIBERO, 3 days for real-world tasks), and inference time/memory usage. Full details are in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research focuses on algorithmic development for robotic manipulation, with no human subjects, privacy data, or unethical applications. All datasets and code comply with open licenses and NeurIPS Ethics Guidelines.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [No]

    Justification: The paper focuses on technical contributions and does not include a dedicated section on positive/negative societal impacts. While the method can benefit robotics applications, potential impacts are not explicitly discussed.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The proposed model and datasets have no high risk of misuse. No special safeguards are required beyond standard open-source practices.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [No]

Justification:All existing assets (baseline models, LIBERO dataset) are properly cited, but their specific licenses, versions, and URLs are not explicitly mentioned in the paper.

Guidelines:

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

13. **New assets**

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

Answer: [Yes]

Justification: The new code and real-world dataset are well-documented in the paper and supplementary materials. Documentation (architecture, training, usage) is provided alongside the assets, with anonymized release for submission.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The research does not involve crowdsourcing or human subjects.

Guidelines:

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

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

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

Answer: [N/A]

Justification: No human subjects are involved, so IRB approval is not applicable.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLMs are only used for writing/editing, not as a core component of the method. No declaration is needed.

Guidelines:

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