FLORA: DUAL-LATENT DISCRETE WORLD MODELS WITH ONE-STEP CONTINUOUS FLOW MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Discrete latent world models have demonstrated remarkable potential for sample-efficient visual reinforcement learning, enabling agents to learn policies from imagined rollouts in compact latent spaces. However, current methods face two key limitations. First, efficient one-step latent predictors typically use factorized sampling, predicting each token of the latent independently and ignoring inter-token dependencies; dependency-aware alternatives such as MaskGIT-based decoding improve coherence but require costly multi-step refinement during imagination. Second, reconstruction-based encoders preserve visual detail but may allocate capacity to task-irrelevant distractors, while reconstruction-free representations can lose fine-grained information needed for precise control. We propose **FLORA**, a discrete latent world model that addresses both limitations in a unified framework. FLORA introduces a *Denoising Latent Predictor (DP)* trained with continuous flow matching and shortcut forcing, enabling it to model inter-token dependencies in a single step with rollout cost comparable to factorized one-step predictors. It also introduces a *Dual-latent Representation (DR)* that pairs a stochastic reconstructive latent with an auxiliary feature-aligned latent, preserving visual detail while capturing task-relevant abstraction. Across Atari 100k, the DeepMind Control Suite, Meta-World, and Crafter, FLORA outperforms or matches strong model-based baselines. On Atari 100k, FLORA achieves a human-normalized mean score of 170% and a median score of 101%, making FLORA the first world model to surpass human-level median performance without decision-time planning.

1 INTRODUCTION

World models (Sutton, 1991; Ha & Schmidhuber, 2018) simulate environment dynamics in a learned latent space and enable reinforcement learning agents to optimize policies through imagined rollouts rather than costly real interaction. By compressing visual observations into compact discrete latents via variational autoencoders (VAEs) (Kingma & Welling, 2014), recent latent-space approaches achieve strong sample efficiency across diverse visual control tasks (e.g., Hafner et al., 2021; Burchi & Timofte, 2025b). Among them, TWISTER (Burchi & Timofte, 2025b) combines a Transformer State-Space Model (TSSM) (Chen et al., 2021) with Action-Conditioned Contrastive Predictive Coding (AC-CPC), and is a particularly strong discrete-latent baseline on Atari 100k (Kaiser et al., 2020) and the DeepMind Control Suite (Tassa et al., 2018).

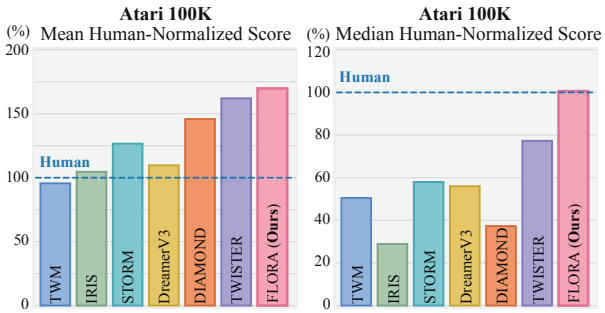

Figure 1: Performance comparison on the Atari 100K benchmark (Kaiser et al., 2020). FLORA reaches human-level median performance among world-model methods without additional decision-time planning.

Despite this progress, two fundamental trade-offs remain unresolved in discrete-latent world models. The first is a *fidelity–relevance trade-off* in representation learning. Pixel-reconstruction objectives (Hafner et al., 2019; 2020; 2021; 2023; Robine et al., 2023; Micheli et al., 2023; Zhang et al., 2023; Seo et al., 2022; Burchi & Timofte, 2025b;a) preserve fine-grained visual detail, but also force

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

the latent to encode task-irrelevant clutter such as backgrounds, lighting variation, and distractors. Reconstruction-free self-supervised objectives (Deng et al., 2022; Burchi & Timofte, 2024; Morihira et al., 2026) improve semantic abstraction, yet often discard spatial detail that remains critical for precise control. The second is an *efficiency–accuracy trade-off* in latent prediction. Fast one-step predictors use factorized sampling (Hafner et al., 2023; Zhang et al., 2023; Burchi & Timofte, 2025b; 2024; Deng et al., 2022; Morihira et al., 2026), predicting each one-hot token of the latent independently in a single forward pass. Although efficient, this parameterization cannot represent inter-token dependencies in the true conditional distribution, so rollout errors compound quickly over imagined trajectories. Dependency-aware alternatives such as autoregressive decoding (Robine et al., 2023; Micheli et al., 2023; 2024) and MaskGIT-based decoding (Chang et al., 2022; Burchi & Timofte, 2025a) recover more coherent predictions, but require multiple sequential refinement steps and therefore slow the imagination loop that dominates actor–critic training. Prior work has attacked these trade-offs separately, but a unified framework that is joint-aware, single-step, and spatially faithful remains missing. Our starting point is that the one-step bottleneck is specific to *discrete* decoding. Under the standard per-token categorical parameterization used by prior one-step world-model predictors, all tokens are sampled in parallel and independently, so the conditional distribution factorizes across token positions. Continuous denoising avoids this restriction: the denoiser generates all tokens jointly from a shared noise sample, so the resulting distribution need not factorize even under one-step inference. The representation trade-off admits a similarly simple resolution. The conflict arises only when a single latent must serve both reconstruction and abstraction. A dual-latent design can instead separate these roles while preserving both spatial fidelity and task relevance.

Building on these observations, we introduce **FLORA** (**FLO**w-based disc**R**ete l**A**tent world model), a discrete-latent world model built on the TWISTER backbone that resolves both trade-offs in a unified architecture. FLORA's *Dual-latent Representation (DR)* pairs a reconstructive encoder with an auxiliary encoder trained under a Barlow Twins objective (Zbontar et al., 2021), so the model retains fine-grained visual detail while acquiring task-relevant abstraction. FLORA's *Denoising Latent Predictor (DP)* replaces factorized sampling with a flow-matching predictor (Lipman et al., 2023) accelerated by shortcut forcing (Frans et al., 2025; Hafner et al., 2025), enabling joint-aware one-step prediction at rollout cost comparable to factorized sampling. We evaluate FLORA on Atari 100k (Kaiser et al., 2020), the DeepMind Control Suite (Tassa et al., 2018), Meta-World (Yu et al., 2020), and Crafter (Hafner, 2022). On Atari 100k, FLORA outperforms TWISTER (Burchi & Timofte, 2025b), DIAMOND (Alonso et al., 2024), DreamerV3 (Hafner et al., 2023), STORM (Zhang et al., 2023), IRIS (Micheli et al., 2023), and TWM (Robine et al., 2023), achieving a human-normalized mean of 170% and a median of 101%. As shown in Fig. 1, this is the first world-model method to surpass human-level median performance without additional decision-time planning (Schrittwieser et al., 2020; Ye et al., 2021; Wang et al., 2024). FLORA also remains competitive across the DeepMind Control Suite, Meta-World, and Crafter, showing that the same design transfers across diverse visual control regimes. Our contributions are summarized as follows:

- We introduce DR, a dual-latent design that complements a reconstructive encoder with an auxiliary encoder, preserving visual detail while improving task-relevant abstraction.
- We introduce DP, a denoising latent predictor that applies continuous flow matching with shortcut forcing to discrete-latent world modeling, capturing inter-token dependencies in one forward pass.
- With DP and DR, FLORA surpasses human-level median performance on Atari 100k and achieves competitive performance on the DeepMind Control Suite, Meta-World, and Crafter.

2 Background

This section reviews the two ingredients FLORA builds on. Section 2.1 reviews latent world models, and Section 2.2 introduces flow matching and shortcut forcing.

2.1 Reinforcement Learning and World Models

We consider reinforcement learning (RL) in a partially observable Markov decision process (POMDP) (Kaelbling et al., 1998), where at each timestep the agent receives an observation o_t, selects an action a_t, and obtains a reward r_t. Model-based RL improves sample efficiency by optimizing policies through imagined trajectories produced by a world model, a learned model that predicts the next observation $\hat{o}_t$, reward $\hat{r}_t$, and episode continuation signal $\hat{c}_t$ from the history of observations and actions.

TWISTER (Burchi & Timofte, 2025b) is a discrete world model that follows DreamerV3 (Hafner et al., 2023). An encoder $q_\phi^z(z_t \mid o_t)$ maps each observation to a stochastic latent z_t composed of categorical tokens, each a one-hot vector sampled from a categorical distribution. A Transformer state-space model (TSSM) (Chen et al., 2021) computes a context vector $h_t = f_\phi(z_{<t}, a_{<t})$ from past latents and actions, and the world-model state $s_t = [z_t, h_t]$ is used to reconstruct o_t and to predict $\hat{r}_t$ and $\hat{c}_t$. A lightweight linear head $p_\phi(\hat{z}_t \mid h_t)$ serves as the dynamics predictor, predicting the latent from the context alone so that the model can roll out trajectories without observations. This predictor outputs an independent categorical distribution for each token, so all tokens are sampled independently in a single forward pass, which we refer to as factorized sampling. TWISTER further introduces action-conditioned contrastive predictive coding (AC-CPC), which makes the world-model state predictive of future latents. A projector q_ϕ^k maps an augmented future latent z'_{t+k} to a target embedding e_t^k, and a predictor p_ϕ^k estimates it as $\hat{e}_t^k = p_\phi^k(s_t, a_{t:t+k})$ from the world-model state and the future actions. The two networks are trained with a contrastive objective over in-batch negatives:

$$\mathcal{L}_{\text{cpc}} = -\frac{1}{K}\sum_{k=0}^{K-1} \log \frac{\exp\big(\text{sim}(z'_{t+k}, s_t)\big)}{\sum_{z'_j \in \mathcal{Z}'} \exp\big(\text{sim}(z'_j, s_t)\big)}, \tag{1}$$

where K is the number of prediction steps, $\mathcal{Z}'$ contains the positive z'_{t+k} and the other latents in the batch as negatives, and $\text{sim}(z'_j, s_t) = q_\phi^k(z'_j)^\top p_\phi^k(s_t, a_{t:t+k})$.

2.2 Flow Matching with Shortcut Forcing

Flow matching (Lipman et al., 2023; Liu et al., 2022) learns a continuous velocity field that transports samples from a simple noise distribution to the data distribution. Given a data sample $\boldsymbol{x}^1 \sim p_{\text{data}}$ and noise $\boldsymbol{x}^0 \sim \mathcal{N}(0, I)$, a linear interpolation defines the path $\boldsymbol{x}^\tau = (1-\tau)\boldsymbol{x}^0 + \tau \boldsymbol{x}^1$ for $\tau \in [0, 1]$, with target velocity $\boldsymbol{x}^1 - \boldsymbol{x}^0$. A velocity network v_ϕ is trained by minimizing the following:

$$\mathcal{L}_{\text{fm}} = \mathbb{E}_{\boldsymbol{x}^0, \boldsymbol{x}^1, \tau}\big[\|v_\phi(\boldsymbol{x}^\tau, \tau) - (\boldsymbol{x}^1 - \boldsymbol{x}^0)\|_2^2\big], \tag{2}$$

and samples are generated at inference by integrating the learned ODE from $\tau = 0$ to $\tau = 1$. However, because flow matching learns only local velocities, accurate generation typically requires many small solver steps; a single Euler step introduces substantial discretization error.

Shortcut models (Frans et al., 2025) address this by conditioning the predictor on a desired step size δ. Rather than predicting an infinitesimal velocity, the shortcut network $v_\phi(\boldsymbol{x}^\tau, \tau, \delta)$ predicts a finite-step direction that advances the sample as $\tilde{\boldsymbol{x}}^{\tau+\delta} = \boldsymbol{x}^\tau + \delta\, v_\phi(\boldsymbol{x}^\tau, \tau, \delta)$. The training objective combines a flow-matching loss at $\delta = 0$ in Eq. (2) with a self-consistency regularization ($\mathcal{L}_{\text{sc}}$) for larger steps:

$$\mathcal{L}_{\text{sc}} = \mathbb{E}_{\boldsymbol{x}^0, \boldsymbol{x}^1, \tau, \delta}\left[\|v_\phi(\boldsymbol{x}^\tau, \tau, 2\delta) - \text{sg}(v_{\text{tgt}})\|_2^2\right], \tag{3}$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operator and $v_{\text{tgt}} = \frac{1}{2}v_\phi(\boldsymbol{x}^\tau, \tau, \delta) + \frac{1}{2}v_\phi(\tilde{\boldsymbol{x}}^{\tau+\delta}, \tau+\delta, \delta)$. A 2δ-step prediction is thus trained to match two consecutive δ-steps, enabling few-step or even one-step generation within a single training run.

Standard flow matching and shortcut models regress a continuous velocity field. When the target $\boldsymbol{x}^1$ is a collection of one-hot categorical tokens, however, the optimal denoiser $D^*(\boldsymbol{x}^\tau, \tau, 0) = \mathbb{E}[\boldsymbol{x}^1 \mid \boldsymbol{x}^\tau]$ yields categorical posterior probabilities lying on the probability simplex, and its extension to $\delta > 0$ remains on the simplex (Lee et al., 2026). The denoiser $D_\phi(\boldsymbol{x}^\tau, \tau, \delta)$ can therefore be parameterized with a per-token softmax, and the velocity is recovered as $v_\phi(\boldsymbol{x}^\tau, \tau, \delta) = \big(D_\phi(\boldsymbol{x}^\tau, \tau, \delta) - \boldsymbol{x}^\tau\big)/(1-\tau)$. This constrains predictions to the simplex while preserving velocity-based shortcut forcing. For the flow-matching base case ($\delta = 0$), the predictor is trained by minimizing cross entropy:

$$\mathcal{L}_{\text{ce}} = \mathbb{E}_{\boldsymbol{x}^0, \boldsymbol{x}^1, \tau}\big[\text{CE}\big(D_\phi(\boldsymbol{x}^\tau, \tau, 0),\, \boldsymbol{x}^1\big)\big], \tag{4}$$

while the self-consistency term retains the squared-error form of Eq. (3). Section 3.3 adapts this formulation to the history-conditioned setting of world modeling.

3 Methodology

Section 3.1 gives an overview of FLORA. Sections 3.2 and 3.3 introduce the Dual-latent Representation (DR) and the Denoising Latent Predictor (DP), which address the fidelity–relevance and efficiency–accuracy trade-offs, respectively. Appendix M covers actor–critic learning in imagination.

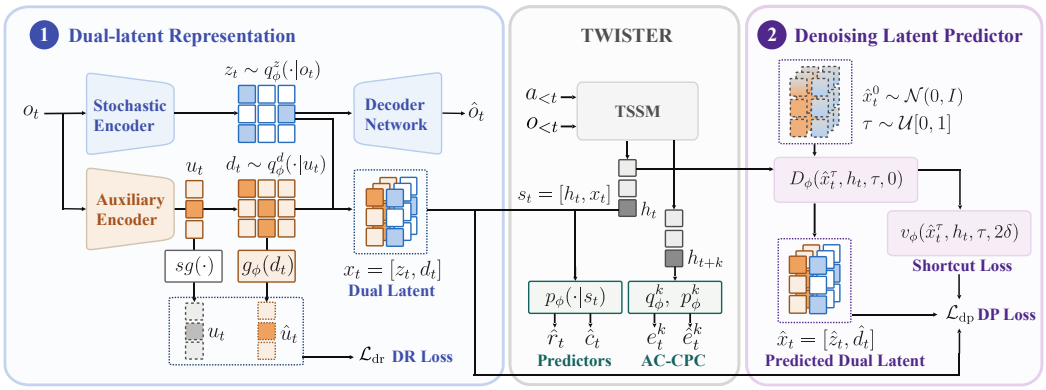

Figure 2: Architecture of FLORA. **(Left, blue)** Dual-latent Representation (DR): each observation is encoded into a reconstructive latent z_t by the stochastic encoder and a feature-aligned latent d_t by the auxiliary encoder. The auxiliary encoder is trained with the Barlow Twins objective $\mathcal{L}_{\text{dr}}$, which aligns the projected latent $g_\phi(d_t)$ with the stop-gradient image features u_t. **(Middle, gray)** TWISTER backbone: a Transformer sequence model processes past latents and actions to produce the context vector h_t, and the world-model state $s_t = [x_t, h_t]$ is used for reward, continuation, and AC-CPC prediction. **(Right, purple)** Denoising Latent Predictor (DP): a flow-matching denoiser with shortcut forcing learns the joint distribution over the dual latent $[\hat{z}_t, \hat{d}_t]$ given h_t and draws samples from it in a single forward pass. It is trained with the flow-matching and self-consistency losses in $\mathcal{L}_{\text{dp}}$.

3.1 Overview of FLORA

FLORA builds upon TWISTER (Burchi & Timofte, 2025b) and modifies how the latent is represented and how it is predicted during imagination. Each observation is represented by a dual latent $x_t = [z_t, d_t]$, which pairs a reconstructive latent z_t with a feature-aligned latent d_t, and the world-model state is defined as $s_t = [x_t, h_t]$. As illustrated in Fig. 2, the world model consists of the components summarized in Table 1.

Table 1: Components of FLORA. † denotes components introduced in this work, and all other components are inherited from TWISTER.

Component	Definition
1. Stochastic Encoder	$z_t \sim q_\phi^z(z_t \mid o_t)$
2. Auxiliary Encoder†	$d_t \sim q_\phi^d(d_t \mid o_t)$
3. Sequence Model	$h_t = f_\phi(x_{<t}, a_{<t})$
4. Denoising Latent Predictor†	$\hat{x}_t \sim p_\phi^{\text{DP}}(\hat{x}_t \mid h_t)$
5. Decoder Network	$\hat{o}_t \sim p_\phi(\hat{o}_t \mid z_t)$
6. Reward Predictor	$\hat{r}_t \sim p_\phi(\hat{r}_t \mid s_t)$
7. Continue Predictor	$\hat{c}_t \sim p_\phi(\hat{c}_t \mid s_t)$

FLORA makes two changes to TWISTER. It adds an auxiliary encoder that produces the feature-aligned latent (DR, Sec. 3.2), and it replaces factorized sampling during imagination with a denoising latent predictor (DP, Sec. 3.3). All other components are inherited from TWISTER, with the sequence model now taking the dual latent x_t as input and the reward and continue predictors taking the world-model state s_t. FLORA also retains the AC-CPC networks q_ϕ^k and p_ϕ^k of TWISTER.

The decoder, reward predictor, and continue predictor are trained by maximum likelihood:

$$\mathcal{L}_{\text{pred}} = -\ln p_\phi(o_t \mid z_t) - \ln p_\phi(r_t \mid s_t) - \ln p_\phi(c_t \mid s_t), \tag{5}$$

where, following TWISTER, the observation term corresponds to a squared error, the reward term uses the symlog two-hot parameterization of DreamerV3 (Hafner et al., 2023), and the continuation term is a Bernoulli likelihood. The complete world-model objective separates the proposed terms from those inherited from TWISTER:

$$\mathcal{L}_{\text{FLORA}} = \underbrace{\mathcal{L}_{\text{dr}} + \mathcal{L}_{\text{dp}}}_{\text{proposed}} + \mathcal{L}_{\text{dyn}} + \mathcal{L}_{\text{pred}} + \mathcal{L}_{\text{cpc}}, \tag{6}$$

where $\mathcal{L}_{\text{dr}}$ and $\mathcal{L}_{\text{dp}}$ train the auxiliary encoder and the denoising latent predictor, respectively (Secs. 3.2 and 3.3). Among the inherited terms, $\mathcal{L}_{\text{pred}}$ is given in Eq. (5), $\mathcal{L}_{\text{cpc}}$ is the AC-CPC objective in Eq. (1) evaluated with the extended world-model state s_t, and $\mathcal{L}_{\text{dyn}}$ is the dynamics loss of TWISTER applied to the dual latent (Sec. 3.2). Each proposed component thus contributes a single objective. The loss weights of TWISTER are retained, and the scales of the two proposed terms are fixed across all benchmarks (Appendix K).

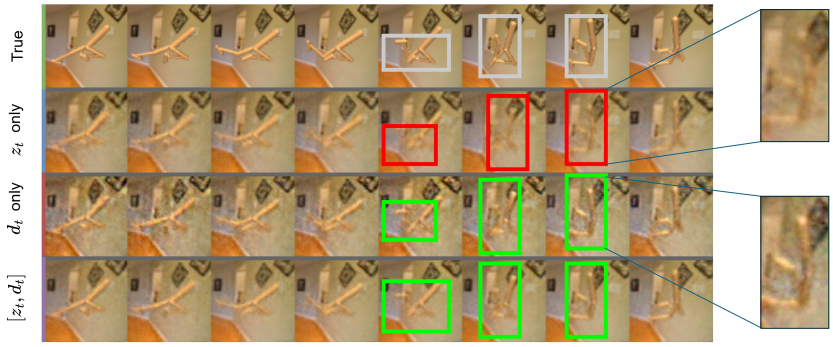

Figure 3: Qualitative comparison of latents on DMC-GB2 (Almuzairee et al., 2024), visualized by a post-hoc decoder trained on frozen representations. Top to bottom: (1) Ground truth. (2) TWISTER's reconstruction-only z_t, which poorly separates the agent from the background. (3) Feature-aligned d_t, which emphasizes the agent but loses scene detail. (4) FLORA's dual latent $[z_t, d_t]$, which recovers both. Boxes mark the agent.

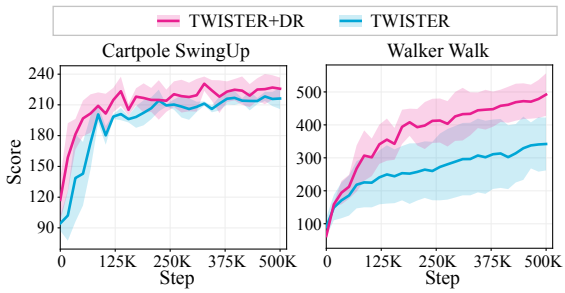

Figure 4: Effect of DR on the DMC-GB2 (Almuzairee et al., 2024) visual distraction tasks *Cartpole Swingup* and *Walker Walk* under the *video-hard* setting.

Table 2: Probing frozen latents of FLORA on *Walker Walk* (DMC-GB2, *video-hard*). Linear and MLP probes are trained to predict the 9-dimensional simulator position vector, and we report the MSE and the Pearson correlation r between predicted and ground-truth values.

	Linear		MLP	
Latent	MSE ↓	r ↑	MSE ↓	r ↑
z_t	0.685	0.445	0.674	0.469
d_t	0.487	0.620	0.459	0.650
$[z_t, d_t]$	**0.483**	**0.622**	**0.442**	**0.661**

3.2 Dual-Latent Representation

In TWISTER, the latent z_t is learned primarily through observation reconstruction, which provides a stable training signal and preserves visual fidelity. When observations contain visually rich but task-irrelevant variation, however, the reconstruction objective may allocate substantial capacity to backgrounds and distractors rather than to control-relevant entities. As shown in the first two rows of Fig. 3 on DMC-GB2 (Almuzairee et al., 2024), frames decoded from z_t fail to separate the agent from the background.

One potential solution is to abandon pixel reconstruction entirely and learn the latent through a self-supervised objective, as in R2-Dreamer (Morihira et al., 2026). As shown in the third row of Fig. 3, the feature-aligned latent d_t, which is likewise trained without reconstruction, captures the agent but discards visual detail. Reconstruction-only latents thus spend capacity on task-irrelevant visual variation, whereas reconstruction-free latents lose details needed for precise control.

DR addresses this fidelity–relevance trade-off by maintaining both latents. The reconstructive latent z_t captures visual detail, while the feature-aligned latent d_t emphasizes task-relevant information. A post-hoc decoder applied to the dual latent $x_t = [z_t, d_t]$ recovers both the scene and the agent (fourth row of Fig. 3). Augmenting TWISTER with DR also improves returns on both DMC-GB2 tasks, with a pronounced gain on *Walker Walk* (Fig. 4). Linear and MLP probes further show that simulator states are more readily decoded from d_t than from z_t, and most accurately from x_t (Table 2).

To obtain d_t, we introduce an auxiliary encoder trained with the DR loss $\mathcal{L}_{\text{dr}}$, and we regularize the dual latent x_t with a dynamics loss $\mathcal{L}_{\text{dyn}}$, as detailed below.

Auxiliary Encoder. The auxiliary encoder $q^d_\phi(d_t \mid o_t)$ consists of a feature extractor f^d_ϕ and a categorical head. Given an observation o_t, the feature extractor computes image features $u_t = f^d_\phi(o_t)$, and the categorical head predicts a categorical distribution over classes for each token, from which d_t is sampled. As with z_t, d_t is thus a discrete latent composed of one-hot categorical tokens, and

the dual latent x_t is formed by concatenating the tokens of both latents. For training, a projector g_ϕ maps the latent back to the feature space as $\hat{u}_t = g_\phi(d_t)$. The projector is used only during training and is therefore omitted from Table 1. Following Morihira et al. (2026), the auxiliary encoder is trained with a Barlow Twins objective (Zbontar et al., 2021) between the projected features and the stop-gradient image features:

$$\mathcal{L}_{\text{dr}} = \sum_i (1 - C_{ii})^2 + \lambda_{\text{bt}} \sum_{i \neq j} C_{ij}^2, \qquad C = \text{CrossCorr}(\text{sg}(u_t), \hat{u}_t)\,, \tag{7}$$

where $\lambda_{\text{bt}} = 5 \times 10^{-4}$. The diagonal term aligns $\hat{u}_t$ with the image features, and the off-diagonal term reduces redundancy across feature dimensions and prevents collapse. We accordingly term d_t the feature-aligned latent, as it is trained to align with image features rather than to reconstruct pixels. Details on the normalization and cross-correlation computation are provided in Appendix J.

Dynamics Loss. To stabilize training, we follow TWISTER and apply its dynamics loss to the dual latent, which encourages both encoders to produce latents that are predictable from the history:

$$\mathcal{L}_{\text{dyn}} = \beta_{\text{dyn}} \text{KL}[\text{sg}(q_\phi(x_t \mid o_t)) \,\|\, p_\phi(\hat{x}_t \mid h_t)] + \beta_{\text{reg}} \text{KL}[q_\phi(x_t \mid o_t) \,\|\, \text{sg}(p_\phi(\hat{x}_t \mid h_t))]\,, \tag{8}$$

where $q_\phi(x_t \mid o_t) = q_\phi^z(z_t \mid o_t)\, q_\phi^d(d_t \mid o_t)$, $p_\phi(\hat{x}_t \mid h_t) = \prod_i p_\phi(\hat{x}_t^{(i)} \mid h_t)$ with $\hat{x}_t^{(i)}$ denoting the i-th token of $\hat{x}_t$, $\beta_{\text{dyn}} = 0.5$, and $\beta_{\text{reg}} = 0.1$. The prior $p_\phi(\hat{x}_t \mid h_t)$ is used only in this loss, whereas latents during imagination are generated by the denoising latent predictor (Sec. 3.3).

3.3 Denoising Latent Predictor

A key component of the TWISTER framework is the dynamics predictor $p_\phi(\hat{z}_t \mid h_t)$, which generates latents during imagined rollouts and therefore governs both their accuracy and their efficiency. Ideally, this distribution should be both accurate and fast to sample from. In TWISTER, however, it is conditionally factorized over tokens, $p_\phi(\hat{z}_t \mid h_t) = \prod_i p_\phi(\hat{z}_t^{(i)} \mid h_t)$, which treats the tokens as conditionally independent given h_t and therefore cannot capture the inter-token dependencies of the encoder distribution $q_\phi^z(z_t \mid o_t)$. We use this z_t-only formulation to motivate the prediction problem before introducing the full FLORA predictor. One approach to address this is EMERALD (Burchi & Timofte, 2025a), which operates over S refinement steps indexed by $s \in \{1, \ldots, S\}$. EMERALD introduces an additional decoder $p_\phi^{\text{iter}}(\hat{z}_t^{s+1} \mid h_t, \hat{z}_t^s)$ using an iterative MaskGIT (Chang et al., 2022) to progressively refine the latent prediction. At each step, the decoder masks a subset of tokens and resamples them conditioned on the previous refinement $\hat{z}_t^s$, iteratively improving the match to the encoder distribution $q_\phi^z(z_t \mid o_t)$. While this approach improves latent accuracy, it incurs an S-fold increase in per-step computation compared with the one-step factorized predictor. Since rollout inference dominates the computational budget in actor–critic learning, this overhead becomes a substantial bottleneck for training efficiency.

To address this trade-off, we propose DP, a denoising latent predictor based on flow matching with shortcut forcing. By generating the entire latent jointly with a flow-matching model, DP captures inter-token dependencies. In addition, the self-consistency loss (Eq. (3)) enables DP to sample in a single forward pass, preserving efficiency.

Fig. 5 presents a controlled comparison that isolates the effectiveness of DP, comparing (1) the factorized one-step baseline (TWISTER), (2) an iterative dependency-aware predictor (TWISTER+MaskGIT), and (3) our joint-aware one-step predictor (TWISTER+DP) in terms of per-token prediction accuracy across the rollout horizon and wall-clock rollout time. DP achieves higher prediction accuracy than the factorized baseline while being substantially faster than the iterative alternative. Section 4.5 further ablates the number of denoising steps.

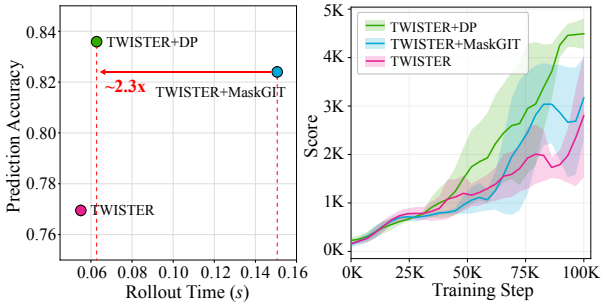

Figure 5: **Left:** Per-token prediction accuracy versus rollout time shows that TWISTER+DP achieves TWISTER+MaskGIT accuracy while being 2.3× faster (0.06s vs 0.14s), outperforming the baseline TWISTER. **Right:** Learning curves on Q*bert suggest that the improved latent prediction quality translates to downstream RL performance.

Implementation and Training Objective. In FLORA, DP operates on the dual latent $\hat{x}_t = [\hat{z}_t, \hat{d}_t]$. The denoiser is parameterized as $D_\phi(\hat{x}_t^\tau, h_t, \tau, \delta)$, where $\hat{x}_t^\tau$ is a linear interpolation between noise and the encoder latent:

$$\hat{x}_t^\tau = (1-\tau)\hat{x}_t^0 + \tau \hat{x}_t^1, \qquad \hat{x}_t^0 \sim \mathcal{N}(0, I), \quad \hat{x}_t^1 = x_t, \quad \tau \sim \mathcal{U}[0,1]. \tag{9}$$

The objective combines the flow-matching loss (Eq. (4)) and the self-consistency loss (Eq. (3)):

$$\mathcal{L}_{\text{dp}} = \mathbb{E}_\tau \left[\text{CE}\big(D_\phi(\hat{x}_t^\tau, h_t, \tau, 0), \hat{x}_t^1\big) \right] + \mathbb{E}_{\tau,\delta} \left[\| v_\phi(\hat{x}_t^\tau, h_t, \tau, 2\delta) - \text{sg}(v_{\text{tgt}}) \|_2^2 \right], \tag{10}$$

where $v_\phi(\hat{x}_t^\tau, h_t, \tau, \delta) = \big(D_\phi(\hat{x}_t^\tau, h_t, \tau, \delta) - \hat{x}_t^\tau\big)/(1-\tau)$ and $v_{\text{tgt}} = \frac{1}{2} v_\phi(\hat{x}_t^\tau, h_t, \tau, \delta) + \frac{1}{2} v_\phi(\tilde{x}_t^{\tau+\delta}, h_t, \tau+\delta, \delta)$ with $\tilde{x}_t^{\tau+\delta} = \hat{x}_t^\tau + \delta\, v_\phi(\hat{x}_t^\tau, h_t, \tau, \delta)$. Following Frans et al. (2025), the shortcut step is drawn uniformly from a dyadic schedule, $\delta \sim \mathcal{U}\{2^{-1}, 2^{-2}, \ldots, 2^{-\kappa}\}$, with $\kappa = 3$ in all experiments.

At inference, DP takes a single shortcut step of size $\delta = 1$ from $\tau = 0$. Since $\hat{x}_t^0 + v_\phi(\hat{x}_t^0, h_t, 0, 1) = D_\phi(\hat{x}_t^0, h_t, 0, 1)$, this step amounts to a single evaluation of the denoiser, which maps the noise directly to a clean latent sample. An $\arg\max$ over classes at each position then produces the one-hot tokens of the predicted latent $\hat{x}_t$, so DP defines

$$p_\phi^{\text{DP}}(\hat{x}_t \mid h_t) = \mathbb{E}_{\hat{x}_t^0 \sim \mathcal{N}(0,I)} \left[\mathbb{I}\big\{ \hat{x}_t = \arg\max D_\phi(\hat{x}_t^0, h_t, 0, 1) \big\} \right], \tag{11}$$

where $\mathbb{I}\{\cdot\}$ denotes the indicator function. The self-consistency loss trains the one-step map $D_\phi(\cdot, h_t, 0, 1)$ to match the endpoint of the flow-matching ODE, which transports $\mathcal{N}(0, I)$ to the joint distribution of x_t given h_t. At the optimum, p_ϕ^{DP} therefore matches this joint distribution rather than its per-token marginals.

In summary, DR and DP modify complementary stages of the same pipeline: DR determines what the latent represents, and DP determines how it is generated during imagination. The controlled comparison in Fig. 5 shows that DP matches the prediction accuracy of MaskGIT at a $2.3\times$ lower rollout time, and Sec. 4 evaluates the contribution of each component to downstream performance.

4 Experimental Results

We evaluate FLORA on four visual control benchmarks chosen to stress different aspects of world-model quality: Atari 100k (Kaiser et al., 2020) for sample-efficient discrete control, the DeepMind Control Suite (Tassa et al., 2018) for continuous-action control from pixels, Meta-World (Yu et al., 2020) for contact-rich robotic manipulation, and Crafter (Hafner, 2022) for long-horizon exploration under partial observability. Together, these settings probe both axes that our method targets: representation quality (Sec. 3.2) and prediction fidelity under tight rollout budgets (Sec. 3.3). We then ablate the contributions of the Dual-latent Representation (DR) and the Denoising Latent Predictor (DP). All results of FLORA are averaged over three seeds, and full experimental details, including environment settings, training budgets, and evaluation protocols, are provided in Appendix D.

4.1 Atari 100k

Table 3 reports per-game scores and aggregate human-normalized metrics on Atari 100k. FLORA achieves the strongest mean and median HNS among the compared world-model baselines and is superhuman on 13 of 26 games. Stratified bootstrap confidence intervals and additional aggregate metrics are reported in Appendix G. Compared with the strongest prior baseline, TWISTER, FLORA raises the mean HNS from 1.62 to 1.70 and the median from 0.77 to 1.01, surpassing human-level median performance. The large gain in median indicates that the improvements are spread broadly across games rather than concentrated on a few outliers, consistent with DR and DP being general-purpose modifications to representation and prediction rather than environment-specific designs.

4.2 DeepMind Control Suite (Vision)

Fig. 6a summarizes aggregate performance on the DeepMind Control Suite, and full per-task scores are reported in Appendix E. FLORA attains the highest mean return (825.0) among the compared methods, outperforms TWISTER on 18 of 20 tasks and ties it on one, and achieves the best score

Table 3: Atari 100k benchmark results. The best score among methods is bolded, and the second-best score is underlined.

Game	Random	Human	SimPLe	TWM	IRIS	STORM	DreamerV3	DIAMOND	EMERALD	TWISTER	FLORA
Alien	227.8	7127.7	616.9	674.6	420	983.6	959	744.1	651	970	**1428.5**
Amidar	5.8	1719.5	74.3	121.8	143	204.8	139	**225.8**	129	184	182.4
Assault	222.4	742	527.2	682.6	1524.4	801	706	**1526.4**	798	721	904.8
Asterix	210	8503.3	1128.3	1116.6	853.6	1028	932	**3698.5**	1045	1306	1172
Bank Heist	14.2	753.1	34.2	466.7	53.1	641.2	649	19.7	927	**942**	924.4
Battle Zone	2360	37187.5	4031.2	5068	13074	13540	12250	4702	**15800**	9920	12380
Boxing	0.1	12.1	7.8	77.8	70.1	79.7	78	86.9	71	88	**94.3**
Breakout	1.7	30.5	16.4	20	83.7	15.9	31	**132.5**	62	35	95.9
Chopper Command	811	7387.8	979.4	1697.4	1565	**1888**	420	1369.8	990	910	1140.7
Crazy Climber	10780.5	35829.4	62583.6	71820.4	59324.2	66776	97190	**99167.8**	75380	81880	88193.3
Demon Attack	152.1	1971	208.1	350.2	**2034.4**	164.6	303	288.1	498	289	393.8
Freeway	0	29.6	16.7	24.3	31.1	**33.5**	0	33.3	31	32	32.4
Frostbite	65.2	4334.7	236.9	**1475.6**	259.1	1316	909	274.1	221	305	323.3
Gopher	257.6	2412.5	596.8	1674.8	2236.1	8239.6	3730	5897.9	14702	**22234**	21178.3
Hero	1027	30826.4	2656.6	7254	7037.4	11044.3	**11161**	5621.8	7655	8773	10918.9
James Bond	29	302.8	100.5	362.4	462.7	509	445	427.4	195	573	**656.8**
Kangaroo	52	3035	51.2	1240	838.2	4208	4098	5382.2	**8780**	6016	5730.7
Krull	1598	2665.5	2204.8	6349.2	6616.4	8412.6	7782	8610.1	7600	**8839**	6597.3
Kung Fu Master	258.5	22736.3	14862.5	24554.6	21759.8	26182	21420	18713.6	22822	23442	**29486.7**
Ms Pacman	307.3	6951.6	1480	1588.4	999.1	**2673.5**	1327	1958.2	1710	2206	2024
Pong	-20.7	14.6	12.8	18.8	14.6	11.3	18	20.4	20	20	**20.6**
Private Eye	24.9	69571.3	35	86.6	100	**7781**	882	114.3	100	1608	100.0
Qbert	163.9	13455	1288.8	3330.8	745.7	4522.5	3405	4499.3	1245	3197	**5207**
Road Runner	11.5	7845	5640.6	9109	9614.6	17564	15565	**20673.2**	6620	17832	18904.5
Seaquest	68.4	42054.7	683.3	**774.4**	661.3	525.2	618	551.2	468	532	513.7
Up N Down	533.4	11693.2	3350.3	**15981.7**	3546.2	7985	7600	3856.3	5227	7068	10768.2
# Superhuman	0	N/A	1	8	10	10	9	11	11	12	**13**
Mean	0	1	0.33	0.96	1.05	1.27	1.12	1.46	1.34	1.62	**1.70**
Median	0	1	0.13	0.51	0.29	0.58	0.49	0.37	0.51	0.77	**1.01**

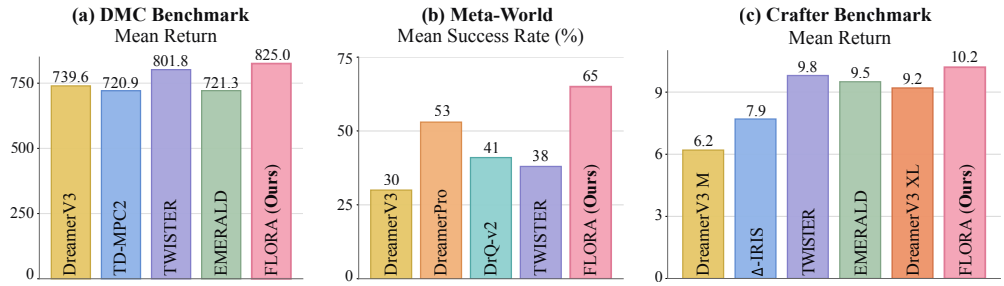

Figure 6: FLORA achieves the highest mean return on DMC, the highest mean success rate on Meta-World, and the highest return on Crafter.

on 12 tasks. The largest gains over TWISTER occur on higher-dimensional locomotion and harder-dynamics tasks, including Quadruped Run (652 $\rightarrow$ 717), Cheetah Run (694 $\rightarrow$ 743), and Reacher Hard (566 $\rightarrow$ 702). These tasks involve richer dynamics, where stronger inter-token dependencies are expected and a factorized predictor is most limited.

4.3 META-WORLD

To evaluate FLORA beyond locomotion and game environments, we consider six vision-based manipulation tasks from Meta-World (Yu et al., 2020): Basketball, Hand Insert, Handle Pull, Soccer, Stick Push, and Sweep. These tasks require precise interaction with small objects under contact-rich dynamics. We compare against DreamerV3, DreamerPro (Deng et al., 2022), the model-free DrQ-v2 (Yarats et al., 2022), and TWISTER, training each method for 1M environment steps per task and reporting the final success rate averaged over three seeds. As shown in Fig. 6b, FLORA achieves the highest mean success rate (65%), ahead of the strongest baseline DreamerPro (53%) and TWISTER (38%). Per-task results are reported in Appendix F.

4.4 CRAFTER

Fig. 6c reports results on Crafter, where FLORA achieves the highest return. Crafter combines visual clutter and partial observability with long-horizon achievement chains, which relate to DR and DP, respectively. FLORA also outperforms DreamerV3 XL, which uses substantially more parameters, suggesting that the gain stems from the representation and predictor changes rather than from scale.

4.5 Ablation Studies

We ablate FLORA on Atari-5 (Aitchison et al., 2023), a five-game subset selected to estimate the 57-game median score to within 10%, which allows us to evaluate multiple configurations at a fraction of the cost of the full benchmark. Table 4 reports human-normalized scores, parameter counts, and training time measured on a single NVIDIA RTX 4090 GPU.

Table 4: Ablations on Atari-5. TWISTER*: capacity-matched TWISTER. FM: Flow Matching; SM: Shortcut Model. Best is bolded and second-best is underlined within each panel.

Method	Params	HNS		Time (min)
		Mean	Median	
(a) DR and DP				
TWISTER	41M	0.39	0.24	310
w/ DR	53M	0.45	<u>0.33</u>	312
w/ DP	43.5M	<u>0.47</u>	0.30	323
TWISTER*	57M	0.42	0.28	313
FLORA	57M	**0.51**	**0.37**	330
(b) DP algorithm and steps				
FM (4-step)	57M	<u>0.58</u>	0.22	357
FM (1-step)	57M	0.12	0.10	**328**
SM (4-step)	57M	**0.64**	<u>0.33</u>	370
SM (1-step)†	57M	0.51	**0.37**	<u>330</u>

† Default configuration of FLORA.

Effect of DR and DP. Both components contribute independently. Adding either DR or DP to TWISTER improves both the mean and the median HNS, with DR yielding the larger gain in median and DP the larger gain in mean (Table 4a). Combining them gives the best results on both metrics, raising the mean from 0.39 to 0.51 and the median from 0.24 to 0.37, which shows that DR and DP are complementary. This improvement requires only 20 additional minutes of training.

Effect of model capacity. The gains do not stem from additional parameters. A capacity-matched TWISTER*, which enlarges the TSSM and CNN of TWISTER to the same 57M parameters as FLORA, reaches only 0.42 mean and 0.28 median HNS, whereas FLORA reaches 0.51 and 0.37. Even TWISTER+DR (53M) and TWISTER+DP (43.5M) outperform TWISTER* on both metrics with fewer parameters.

Latent predictor design. Shortcut forcing enables accurate one-step prediction (Table 4b). One-step flow matching collapses to a mean HNS of 0.12, and flow matching requires four steps to recover a competitive mean. With shortcut forcing, the one-step predictor attains the best median (0.37) at nearly the same training time as one-step flow matching. Four steps further raise the mean to 0.64 but add about 40 minutes of training, so we adopt the one-step shortcut model as the default trade-off.

5 Conclusion

We presented FLORA, a discrete-latent world model that addresses two longstanding trade-offs in model-based reinforcement learning: the fidelity–relevance trade-off in representation and the efficiency–accuracy trade-off in prediction. To resolve the first, the Dual-latent Representation (Sec. 3.2) pairs the reconstructive latent with a feature-aligned latent produced by an auxiliary encoder, retaining visual detail while concentrating capacity on task-relevant abstraction. To resolve the second, the Denoising Latent Predictor (Sec. 3.3) replaces factorized sampling with a continuous flow-matching predictor accelerated by shortcut forcing, capturing inter-token dependencies in a single forward pass. Across Atari 100k, the DeepMind Control Suite, Meta-World, and Crafter, FLORA improves over strong prior baselines: it is the first world-model method to surpass human-level median performance on Atari 100k without decision-time planning, matches or exceeds TWISTER on 19 of 20 DMC tasks, outperforms TWISTER on all six Meta-World tasks, and achieves the highest return on Crafter. Together, these results suggest that resolving both trade-offs in a unified architecture yields a discrete-latent world model that is simultaneously efficient and accurate.

AI use statement

Large language models were used for writing assistance, drafting and revising passages and correcting grammar, and for visualizing results, where they wrote the plotting code that renders the figures from the result files. All technical content, experimental results, and claims were produced by the authors, who reviewed every AI-assisted passage and figure and take full responsibility for the content of the paper.

Reproducibility Statement

Every component and objective of FLORA is specified in Section 3, with the computation of the DR loss detailed in Appendix J and actor–critic learning in Appendix M. The hyperparameters are

listed in Appendix K, and the environment settings, training budgets, and evaluation protocols of every benchmark are described in Appendix D. All results of FLORA are averaged over three seeds, per-task results are reported in Appendices E and F, and all wall-clock times are measured on a single NVIDIA RTX 4090 GPU. We will release the code and the configuration file of every benchmark upon publication.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29304–29320, 2021.

Matthew Aitchison, Penny Sweetser, and Marcus Hutter. Atari-5: Distilling the arcade learning environment down to five games. In *International Conference on Machine Learning*, pp. 421–438. PMLR, 2023.

Abdulaziz Almuzairee, Nicklas Hansen, and Henrik I. Christensen. A recipe for unbounded data augmentation in visual reinforcement learning. *Reinforcement Learning Journal*, 1:130–157, 2024.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos J. Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in Atari. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:58757–58791, 2024.

Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47: 253–279, 2013.

Maxime Burchi and Radu Timofte. MuDreamer: Learning predictive world models without reconstruction. *arXiv preprint arXiv:2405.15083*, 2024.

Maxime Burchi and Radu Timofte. Accurate and efficient world modeling with masked latent transformers. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 5894–5912. PMLR, 2025a.

Maxime Burchi and Radu Timofte. Learning transformer-based world models with contrastive predictive coding. In *International Conference on Learning Representations (ICLR)*, 2025b. arXiv:2503.04416.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. MaskGIT: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11315–11325, 2022.

Chang Chen, Jaesik Yoon, Yi-Fu Wu, and Sungjin Ahn. TransDreamer: Reinforcement learning with transformer world models. In *Deep RL Workshop, NeurIPS*, 2021.

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2208.04202 (2022).

Fei Deng, Ingook Jang, and Sungjin Ahn. DreamerPro: Reconstruction-free model-based reinforcement learning with prototypical representations. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 4956–4975. PMLR, 2022.

Lijie Fan, Tianhong Li, Siyang Qin, Yuanzhen Li, Chen Sun, Michael Rubinstein, Deqing Sun, Kaiming He, and Yonglong Tian. Fluid: Scaling autoregressive text-to-image generative models with continuous tokens, 2024. URL `https://arxiv.org/abs/2410.13863`.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.12557 (2024).

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in Neural Information Processing Systems*, 31, 2018.

Danijar Hafner. Benchmarking the spectrum of agent capabilities. In *International Conference on Learning Representations (ICLR)*, 2022.

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 (ICML)*, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations (ICLR)*, 2020.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering Atari with discrete world models. In *International Conference on Learning Representations (ICLR)*, 2021.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models. *arXiv preprint arXiv:2509.24527*, 2025.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. arXiv:2310.16828.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1):99–134, 1998. doi: 10.1016/S0004-3702(98)00023-X.

Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H. Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. Model-based reinforcement learning for Atari. In *International Conference on Learning Representations (ICLR)*, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Chanhyuk Lee, Jaehoon Yoo, Manan Agarwal, Sheel Shah, Jerry Huang, Aditi Raghunathan, Seunghoon Hong, Nicholas M. Boffi, and Jinwoo Kim. Flow map language models: One-step language modeling via continuous denoising, 2026. URL https://arxiv.org/abs/2602.16813.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2210.02747.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. URL https://arxiv.org/abs/2209.03003.

Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Q. Weinberger. Latent diffusion for language generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 56998–57025, 2023.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *International Conference on Learning Representations (ICLR)*, 2023.

Vincent Micheli, Eloi Alonso, and François Fleuret. Efficient world models with context-aware tokenization. In *International Conference on Machine Learning (ICML)*, pp. 35623–35638, 2024.

Naoki Morihira, Amal Nahar, Kartik Bharadwaj, Yasuhiro Kato, Akinobu Hayashi, and Tatsuya Harada. R2-Dreamer: Redundancy-reduced world models without decoders or augmentation. In *International Conference on Learning Representations (ICLR)*, 2026.

Jan Robine, Marc Höftmann, Tobias Uelwer, and Stefan Harmeling. Transformer-based world models are happy with 100k interactions. In *International Conference on Learning Representations (ICLR)*, 2023.

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

Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control. In *Conference on Robot Learning (CoRL)*, 2022. arXiv:2206.14244.

Robin Strudel, Corentin Tallec, Florent Altché, Yilun Du, Yaroslav Ganin, Arthur Mensch, Will Grathwohl, Nikolay Savinov, Sander Dieleman, Laurent Sifre, and Rémi Leblond. Self-conditioned embedding diffusion for text generation, 2022. URL https://arxiv.org/abs/2211.04236.

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

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. DeepMind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Shengjie Wang, Shaohuai Liu, Weirui Ye, Jiacheng You, and Yang Gao. EfficientZero V2: Mastering discrete and continuous control with limited data. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 51041–51062. PMLR, 2024.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. Mastering atari games with limited data. In *Advances in Neural Information Processing Systems*, volume 34, pp. 25476–25488, 2021.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1094–1100. PMLR, 2020.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow Twins: Self-supervised learning via redundancy reduction. *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

Weipu Zhang, Gang Wang, Jian Sun, Yetian Yuan, and Gao Huang. STORM: Efficient stochastic transformer based world models for reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2023.

A Impact Statement

FLORA addresses two fundamental limitations in discrete-latent world models: the fidelity–relevance trade-off in representation learning and the efficiency–accuracy trade-off in latent prediction. Our work contributes to sample-efficient reinforcement learning by improving the quality of imagined rollouts. Specifically, FLORA introduces a Dual-latent Representation that pairs a reconstructive latent with a feature-aligned latent, combining visual fidelity with task-relevant abstraction, and a Denoising Latent Predictor that captures inter-token dependencies in a single forward pass through continuous flow matching and shortcut forcing. The empirical results across Atari 100k, the DeepMind Control Suite, Meta-World, and Crafter suggest that improving both representation quality and latent prediction fidelity can lead to more effective world-model learning across diverse visual control domains.

World models are a promising direction for improving sample efficiency in reinforcement learning by reducing the need for direct environment interaction, which is particularly costly in domains such as robotic manipulation. While FLORA advances this direction, learned world models may still contain prediction errors, which can lead to suboptimal agent behavior during imagination-based policy learning. We believe that improving representation quality and latent prediction fidelity, as done in FLORA, is a useful step toward more reliable and efficient world-model-based agents.

B Limitations

FLORA improves discrete-latent world modeling by introducing two additional components on top of the TWISTER backbone: the Dual-latent Representation (DR) and the Denoising Latent Predictor (DP). While these components improve representation quality and latent prediction fidelity, they also increase model size and training cost compared with the original TWISTER architecture. In particular, DR adds an auxiliary encoder, and DP replaces the lightweight factorized latent predictor with a denoising predictor trained with flow matching and shortcut forcing.

Our profiling in Appendix H shows that this overhead remains modest in practice. On a single Atari 100k task, FLORA increases the total per-update time from 181.74 ms to 195.56 ms, corresponding to a 7.61% increase, and the full training time from 5.05 to 5.43 hours on a single NVIDIA RTX 4090 GPU. Nevertheless, FLORA is not as lightweight as the original factorized TWISTER predictor. Reducing the parameter and computation overhead of the auxiliary encoder and denoising predictor is an important direction for future work, especially for settings with strict compute budgets or larger-scale benchmarks.

C Related Work

C.1 Model-Based Reinforcement Learning

Model-based reinforcement learning (MBRL) improves sample efficiency by learning a predictive model of the environment and using it for planning or policy optimization (Sutton, 1991; Ha & Schmidhuber, 2018; Hafner et al., 2019; 2020). In visual control, latent world models have become a dominant paradigm: instead of modeling high-dimensional observations directly in pixel space, they compress observations into compact latent states and perform imagination in the learned latent space. The Dreamer series (Hafner et al., 2020; 2021; 2023) established this approach as a practical and scalable framework by jointly learning a latent dynamics model, reconstruction and reward predictors, and actor–critic policies from imagined trajectories. In particular, DreamerV3 (Hafner et al., 2023) showed that a single set of hyperparameters can achieve strong performance across diverse domains, making it a widely adopted foundation for subsequent world-model agents.

Recent work has extended Dreamer-style latent world models along two major directions. One direction improves the sequence model used for latent dynamics. Transformer-based world models such as TransDreamer (Chen et al., 2021), TWM (Robine et al., 2023), STORM (Zhang et al., 2023), and TWISTER (Burchi & Timofte, 2025b) replace recurrent dynamics with masked self-attention to better capture temporal dependencies. TWISTER (Burchi & Timofte, 2025b) is the closest reference to FLORA, providing a strong Transformer-based discrete-latent backbone while retaining a factorized one-step latent predictor.

Another direction improves the quality of latent prediction or representation learning. On the generative side, methods such as IRIS (Micheli et al., 2023), Δ-IRIS (Micheli et al., 2024), and EMERALD (Burchi & Timofte, 2025a) model token dependencies using autoregressive or MaskGIT-style decoding (Chang et al., 2022). These approaches improve prediction coherence but require multiple sequential decoding steps, increasing imagination rollout cost. On the representation side, reconstruction-free methods such as DreamerPro (Deng et al., 2022), MuDreamer (Burchi & Timofte, 2024), and R2-Dreamer (Morihira et al., 2026) replace pixel reconstruction with self-supervised or predictive objectives to reduce redundant visual information and improve computational efficiency. However, removing the reconstruction signal can also weaken the model's ability to preserve fine-grained visual details needed for precise control.

C.2 Continuous Generative Modeling for Discrete Distributions

Continuous generative models have recently been explored as an alternative to purely discrete generation. One line of work maps discrete variables into a continuous space before applying diffusion or flow-based modeling. Analog Bits (Chen et al., 2023) represents discrete data as real-valued binary codes and trains a continuous diffusion model over these analog bits, while embedding diffusion methods apply continuous denoising directly in token-embedding space (Strudel et al., 2022; Lovelace et al., 2023). More recently, Flow Map Language Models (Lee et al., 2026) show that flow-based continuous denoising over one-hot token encodings can outperform discrete diffusion in few-step and one-step language generation. In vision, Fluid (Fan et al., 2024) further suggests that continuous token parameterizations can scale more favorably than discrete tokens for autoregressive text-to-image generation.

These works support the broader idea that discrete data need not be generated by discrete samplers alone. However, they primarily study language or image generation, where sampling cost affects generation latency. FLORA instead applies continuous denoising to the latent predictor of a model-based RL world model, where the predictor is queried repeatedly inside imagined actor–critic rollouts. This setting makes one-step generation not only desirable but necessary for efficient policy learning. Unlike prior continuous-token generative models, FLORA targets discrete latent world modeling: it predicts discrete latents with a shortcut-forced flow model, capturing inter-token dependencies while retaining the rollout cost of factorized sampling.

D EXPERIMENTAL SETUP

Unless otherwise specified, all experiments are conducted with three random seeds per task, and we report seed-aggregated results.

Baselines. We compare FLORA with representative model-based reinforcement learning baselines across each benchmark. On Atari 100k, we include both latent-space and pixel-space world models. SimPLe (Kaiser et al., 2020) is an early pixel-level model-based Atari agent, while TWM (Robine et al., 2023), IRIS (Micheli et al., 2023), STORM (Zhang et al., 2023), DreamerV3 (Hafner et al., 2023), EMERALD (Burchi & Timofte, 2025a), and TWISTER (Burchi & Timofte, 2025b) represent recent latent-space world models with recurrent or Transformer-based dynamics. DIAMOND (Alonso et al., 2024) is a diffusion-based pixel-space world model that achieves strong Atari 100k performance by modeling visual details directly in observation space. These baselines cover the main design choices in sample-efficient Atari world modeling: pixel-space versus latent-space prediction, recurrent versus Transformer dynamics, and factorized versus dependency-aware latent generation.

On the DeepMind Control Suite Vision benchmark, we compare against strong visual continuous-control baselines. DreamerV3 (Hafner et al., 2023) is a robust recurrent latent world model and serves as a widely used reference method. TD-MPC2 (Hansen et al., 2024) is a strong model-based baseline specifically designed for scalable continuous control with latent-space planning. EMERALD (Burchi & Timofte, 2025a) and TWISTER (Burchi & Timofte, 2025b) are recent Transformer-based world models that provide the closest comparison to FLORA in terms of architecture and latent-space imagination.

On Crafter, we compare against DreamerV3, Δ-IRIS (Micheli et al., 2024), EMERALD, and TWISTER. Crafter emphasizes long-horizon exploration, memory, and partial observability, making these world-model baselines useful references for evaluating whether FLORA's improvements in latent prediction and representation quality translate to memory-demanding visual control.

On Meta-World, we compare against DreamerV3, DreamerPro (Deng et al., 2022), DrQ-v2 (Yarats et al., 2022), and TWISTER. DreamerPro is a reconstruction-free world model with prototypical representations, and DrQ-v2 is a strong model-free baseline that learns directly from pixels with data augmentation. These baselines test whether the improvements of FLORA extend to contact-rich manipulation.

Atari 100k. We evaluate on the Atari 100k benchmark (Kaiser et al., 2020; Bellemare et al., 2013), which consists of 26 Atari games under a limited data budget. Following the standard protocol, each agent is trained for 100k environment interaction steps with a frame skip of 4, corresponding to 400k environment frames. We report the human-normalized score (HNS): $\mathrm{HNS} = \frac{\mathrm{score}_{\mathrm{agent}} - \mathrm{score}_{\mathrm{random}}}{\mathrm{score}_{\mathrm{human}} - \mathrm{score}_{\mathrm{random}}}$, and summarize performance using the mean and median HNS across games, as well as the number of games where the agent exceeds human-level performance.

DeepMind Control Suite. We evaluate visual continuous-control performance on the DeepMind Control Suite (Tassa et al., 2018). Agents learn from pixel observations and are trained for 500k environment interaction steps with a frame skip of 2, corresponding to 1M environment frames. We report the mean return across tasks.

DMC-GB2. We use DMC-GB2 (Almuzairee et al., 2024) to evaluate robustness under visual distractions. Following the DeepMind Control setting, agents are trained for 500k environment interaction steps with a frame skip of 2, corresponding to 1M environment frames. We use the *video-hard* setting in our fidelity–relevance analysis to stress-test whether the representation can separate task-relevant foreground dynamics from dynamic background distractors.

Meta-World. We evaluate vision-based robotic manipulation on six tasks from Meta-World (Yu et al., 2020): Basketball, Hand Insert, Handle Pull, Soccer, Stick Push, and Sweep. These tasks require precise interaction with small objects under contact-rich dynamics. Agents are trained for 1M environment steps per task, and we report the final success rate averaged over three seeds.

Crafter. We evaluate long-horizon exploration and memory on Crafter (Hafner, 2022), a procedurally generated survival environment with visual observations, discrete actions, and partial observability. Agents are trained for 1M environment interaction steps. We report the average return following prior model-based reinforcement learning evaluations.

E DeepMind Control Suite Results

This section provides the complete DeepMind Control Suite results from visual observations. Table 5 reports per-task returns, while Fig. 7 shows the corresponding training curves. All methods are evaluated under the same pixel-based control setting and the same interaction budget described in Appendix D.

Table 5: Results on the DeepMind Control Suite. The best result is bolded, and the second-best result is underlined.

Task	DreamerV3	TD-MPC2	TWISTER	EMERALD	FLORA (ours)
Acrobot Swingup	210	216	<u>239.4</u>	42.9	**263.9**
Ball In Cup Catch	957.1	717.2	<u>966.8</u>	963.2	**972.7**
Cartpole Balance	996.4	931.1	<u>997.9</u>	997.6	**999.5**
Cartpole Balance Sparse	**1000**	**1000**	**1000**	**1000**	**1000**
Cartpole Swingup	819.1	808.1	819.2	<u>855.6</u>	**868.9**
Cartpole Swingup Sparse	**792.9**	739	735	735.5	<u>766.5</u>
Cheetah Run	<u>728.7</u>	550.3	694	670.8	**742.8**
Finger Spin	818.5	**986**	976.3	945	<u>985.9</u>
Finger Turn Easy	787.7	788.9	923.7	**988**	<u>944.9</u>
Finger Turn Hard	810.8	871.8	<u>910.3</u>	879.8	**918.7**
Hopper Hop	**369.6**	211.1	313.9	306	<u>329.0</u>
Hopper Stand	900.6	803	<u>932</u>	855.6	**941.0**
Pendulum Swingup	806.3	743.2	832.1	<u>843.8</u>	**848.4**
Quadruped Run	352.3	361.9	<u>652.1</u>	170.2	**716.8**
Quadruped Walk	352.6	252.6	**904.9**	421.2	<u>867.0</u>
Reacher Easy	898.9	<u>971</u>	933.1	964.9	**978.2**
Reacher Hard	499.2	**876.9**	565.8	430.4	<u>701.6</u>
Walker Run	**757.8**	<u>728.1</u>	711.2	421.6	719.0
Walker Stand	976.7	915.8	<u>976.9</u>	<u>976.9</u>	**980.8**
Walker Walk	<u>955.8</u>	945.1	951.3	**957.2**	954.4
Mean	739.6	720.9	<u>801.8</u>	721.3	**825.0**

As shown in Table 5, FLORA achieves the highest mean return among the compared methods (825.0) and matches or exceeds TWISTER on 19 of 20 tasks. At the per-task level, it obtains the best score on 12 tasks and ranks in the top two on 18 tasks, indicating that the gains are broadly distributed across continuous-control environments rather than concentrated on a few tasks. The largest improvements over TWISTER occur on tasks with higher-dimensional or harder dynamics, including Quadruped Run ($652.1 \rightarrow 716.8$), Cheetah Run ($694.0 \rightarrow 742.8$), and Reacher Hard ($565.8 \rightarrow 701.6$). These results suggest that the benefits of FLORA arise from the combination of DR and DP: DR improves the visual state representation, and DP improves the fidelity of imagined latent transitions.

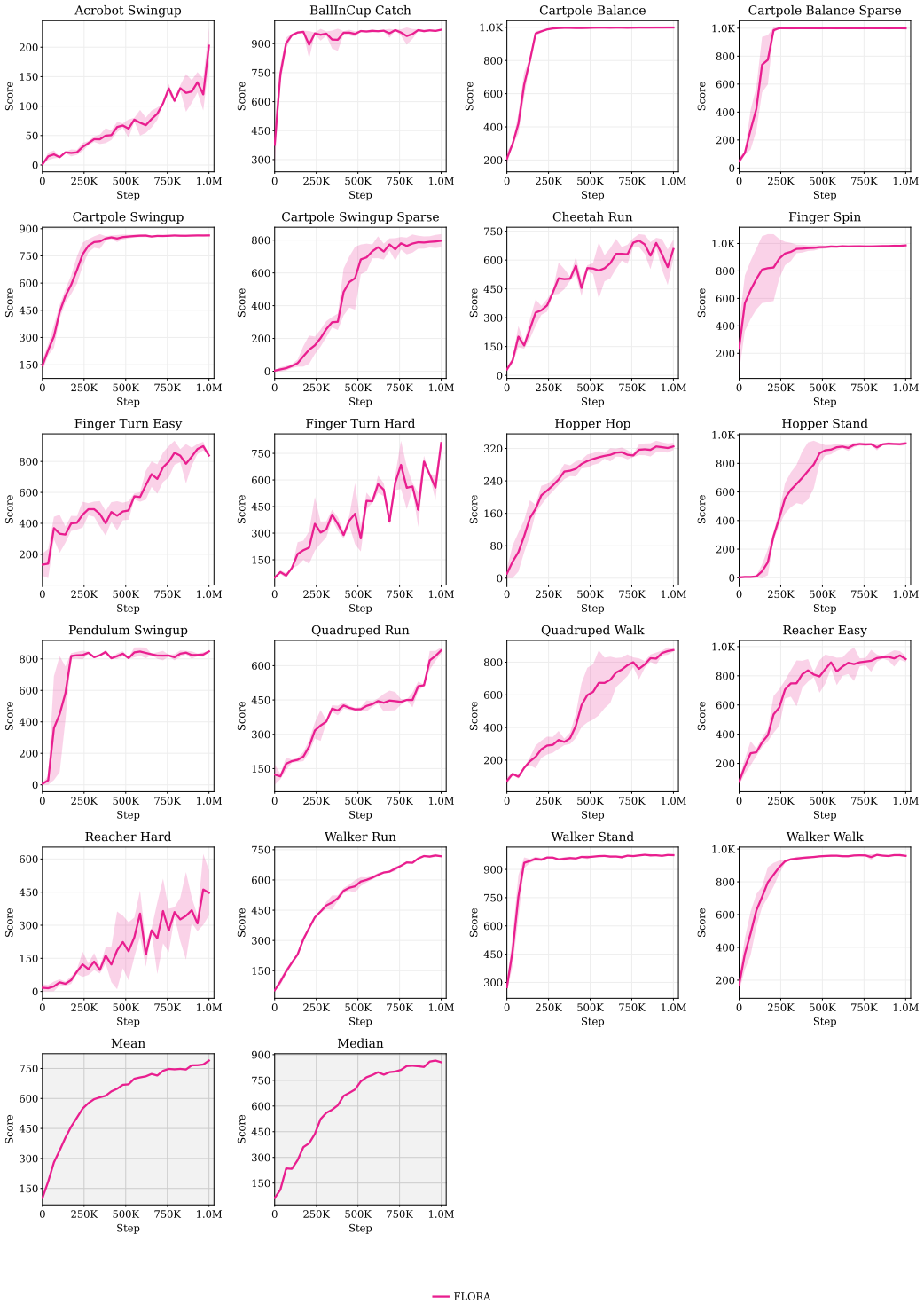

Figure 7: Training curves on the DeepMind Control Suite from visual observations. Solid lines denote the mean return across seeds, and shaded regions denote standard deviation.

F META-WORLD RESULTS

We evaluate on six vision-based manipulation tasks from Meta-World (Yu et al., 2020): Basketball, Hand Insert, Handle Pull, Soccer, Stick Push, and Sweep. These tasks involve precise object interaction under contact-rich dynamics. Each method is trained for 1M environment steps per task, and we report the final success rate averaged over three seeds. We compare FLORA with the model-based baselines DreamerV3 (Hafner et al., 2023), DreamerPro (Deng et al., 2022), and TWISTER (Burchi & Timofte, 2025b), as well as the model-free baseline DrQ-v2 (Yarats et al., 2022). FLORA uses the same hyperparameters as in the other benchmarks (Appendix K).

Table 6 reports the per-task success rates. FLORA achieves the highest mean success rate, outperforms TWISTER on all six tasks, and achieves the best success rate on four of them. DreamerPro and DrQ-v2 attain the best results on Handle Pull and Sweep, respectively, where FLORA ranks second.

Table 6: Success rates (%) on six vision-based Meta-World (Yu et al., 2020) manipulation tasks after 1M environment steps, averaged over three seeds. The best result is bolded, and the second-best result is underlined.

Task	DreamerV3	DreamerPro	DrQ-v2	TWISTER	FLORA (ours)
Basketball	0	<u>33</u>	7	10	**47**
Hand Insert	23	<u>47</u>	46	37	**70**
Handle Pull	60	**100**	0	60	<u>67</u>
Soccer	**47**	15	33	<u>37</u>	**47**
Stick Push	0	50	<u>58</u>	33	**67**
Sweep	50	76	**100**	53	<u>90</u>
Mean	30	<u>53</u>	41	38	**65**

G Additional Results on Atari 100k

This section provides additional analyses of FLORA on the Atari 100k benchmark. Beyond the per-game scores reported in Table 3, we evaluate aggregate reliability metrics, optimality gap, performance profiles, probability of improvement, and per-game training curves. These analyses follow the recommendations of Agarwal et al. (2021), which emphasize uncertainty-aware comparisons rather than relying only on point estimates.

Aggregate human-normalized metrics. Fig. 8 reports mean, median, and interquartile mean (IQM) human-normalized scores with stratified bootstrap confidence intervals. FLORA achieves the strongest performance across all three aggregate metrics. The improvement in median and IQM is particularly important because these metrics are less dominated by a small number of outlier games than the mean. This indicates that FLORA's gains are broadly distributed across the Atari 100k suite rather than being concentrated in only a few high-scoring environments.

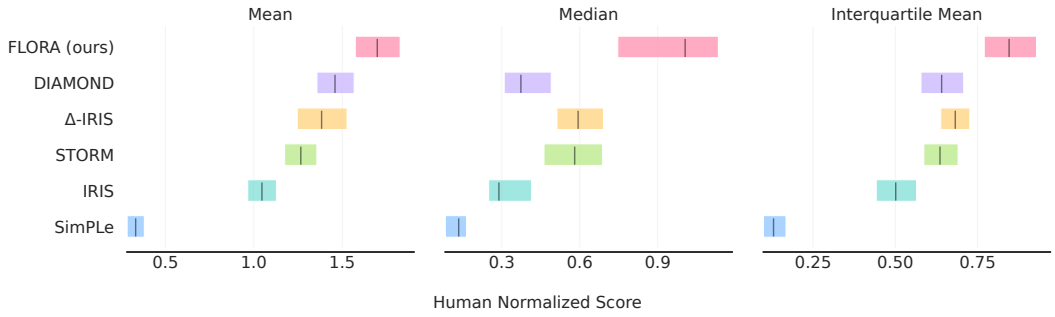

Figure 8: Aggregate Atari 100k results with stratified bootstrap confidence intervals. We report mean, median, and interquartile mean (IQM) human-normalized scores. FLORA obtains the strongest aggregate performance across all metrics.

Optimality gap. Fig. 9 reports the optimality gap, where lower values indicate better performance. FLORA achieves the smallest optimality gap among the compared methods, further confirming that its improvement is not limited to average score but also reflects stronger coverage across games. This complements the median and IQM results, since the optimality gap penalizes methods that perform poorly on a subset of tasks.

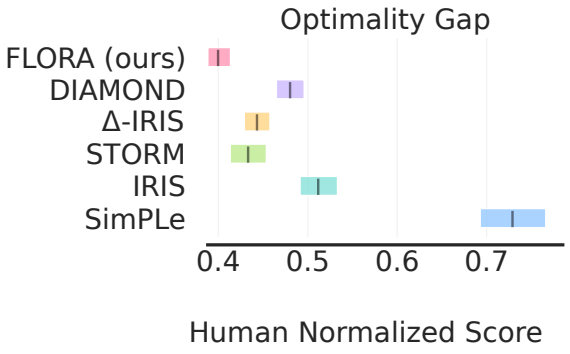

Figure 9: Optimality gap on Atari 100k with stratified bootstrap confidence intervals. Lower is better. FLORA achieves the smallest optimality gap among the compared methods.

Performance profiles. Fig. 10 shows performance profiles, which plot the fraction of runs whose human-normalized score exceeds a threshold τ. FLORA maintains a higher fraction of successful runs over a broad range of thresholds and preserves a stronger high-score tail. This suggests that FLORA improves not only typical performance but also the probability of obtaining strong policies across different games and seeds.

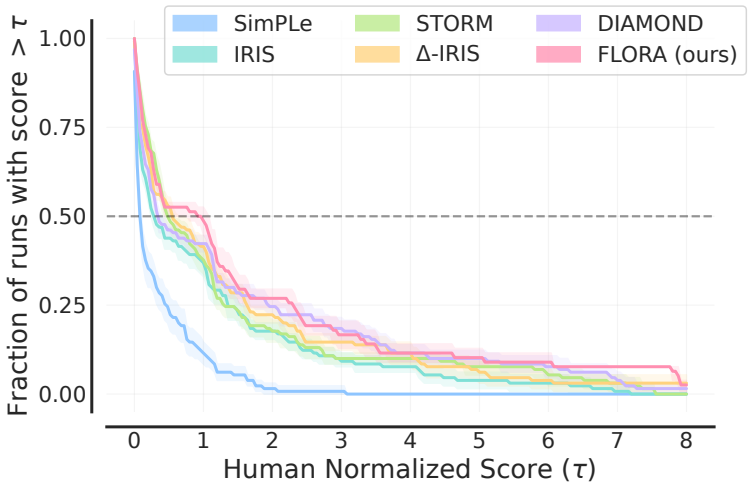

Figure 10: Performance profiles on Atari 100k. Each curve shows the fraction of runs whose human-normalized score exceeds the threshold τ. FLORA maintains a stronger profile across a broad range of score thresholds.

Probability of improvement. Fig. 11 reports the probability that FLORA outperforms each baseline on a randomly selected task. The probability is above 0.5 for all compared baselines, indicating that FLORA is more likely to outperform each method than vice versa. This pairwise analysis provides an additional robustness check beyond aggregate metrics, since it directly measures the likelihood of improvement across the benchmark.

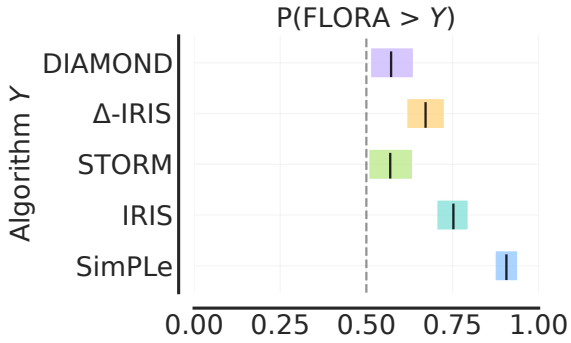

Figure 11: Probability of improvement on Atari 100k. Each row estimates the probability that FLORA outperforms the corresponding baseline on a randomly selected task, with stratified bootstrap confidence intervals. Values above 0.5 indicate that FLORA is more likely to outperform the baseline than underperform it.

Training curves. Fig. 12 shows per-game training curves for FLORA. The curves demonstrate stable learning across most Atari games and show consistent improvement in both aggregate human-normalized mean and median throughout training. In particular, the median curve rises steadily and exceeds human-level performance by the end of training, matching the final aggregate results reported in the main text.

Overall, these additional analyses support the conclusion that FLORA provides robust improvements on Atari 100k. The aggregate metrics show strong mean, median, and IQM performance; the optimality gap and performance profiles indicate better coverage across games; and the probability-of-improvement analysis confirms that FLORA is more likely to outperform each baseline on a randomly selected task. Together with the per-game training curves, these results reinforce that FLORA's gains arise from broadly improved world-model learning rather than from isolated outlier performance.

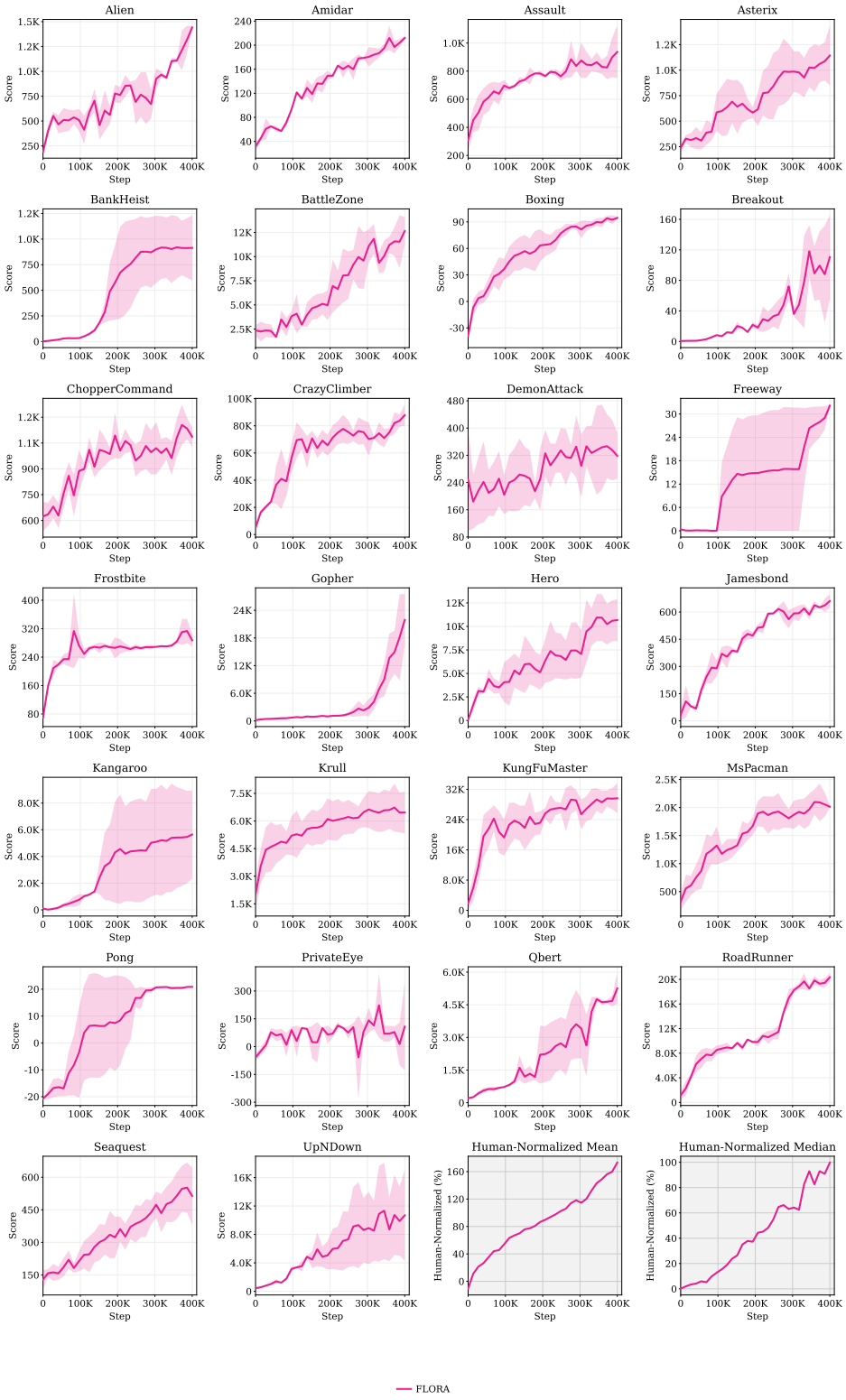

Figure 12: FLORA training curves on all 26 Atari 100k games. Solid lines denote the mean across seeds, and shaded regions denote standard deviation. The final two panels report aggregate human-normalized mean and median scores over training.

H TRAINING TIME PROFILE

We profile the computational cost of FLORA on a single Atari 100k task to quantify the overhead introduced by the proposed components. All measurements are obtained on a single NVIDIA RTX 4090 GPU. Table 7 decomposes the wall-clock time of a single training update into world-model learning, actor–critic learning, environment interaction, and auxiliary updates.

Table 7: FLORA training-time profile with per-stage decomposition.

Single update	**Time (ms)**
Total	195.56
World model update	109.19
Actor–Critic model update	79.64
Imagination step ($\times 15$)	60.67
Reward prediction	1.58
Value prediction	1.57
Continuation prediction	1.43
Action prediction	1.58
Other (loss/backward, etc.)	6.02
Env step	4.91
Target network update	0.16

The profile shows that most of the training time is spent on world-model and actor–critic updates. In particular, imagined rollout generation accounts for the dominant portion of the actor–critic update, taking 60.67 ms out of 79.64 ms. This confirms that the efficiency of latent rollout is critical for practical world-model training, and motivates our design choice of using a one-step denoising predictor rather than an iterative refinement procedure.

Table 8: Computational efficiency comparison between TWISTER and FLORA.

Module	**TWISTER (ms/hours)**	**FLORA (ms/hours)**	**Difference (%)**
Single Update			
Total	181.74 ms	195.56 ms	+7.61%
World Model Update	108.98 ms	109.19 ms	+0.19%
Actor-Critic Model Update	66.87 ms	79.64 ms	+19.09%
Full Run (100,000 updates)			
Training time	5.05 hours	5.43 hours	+7.61%

Table 8 compares the training efficiency of FLORA against the TWISTER backbone. FLORA increases the total per-update time from 181.74 ms to 195.56 ms, corresponding to a modest 7.61% overhead. The world-model update time remains nearly unchanged ($+0.19\%$), indicating that the additional auxiliary encoder and denoising objective do not substantially increase world-model optimization cost. The main overhead appears in the actor–critic update, where imagined rollouts query the denoising latent predictor during policy learning. Nevertheless, because DP performs one-step prediction, the overall training time increases only from 5.05 to 5.43 hours for a full $100{,}000$-update run. These results show that FLORA improves prediction and representation quality while preserving training efficiency close to the TWISTER baseline.

I CRAFTER TRAINING CURVES

This section provides the training curve of FLORA on Crafter over the 1M environment-step budget. Crafter evaluates long-horizon exploration, partial observability, and memory in a procedurally generated visual environment. In addition to the final return reported in the main text, the training curve shows how performance evolves throughout learning.

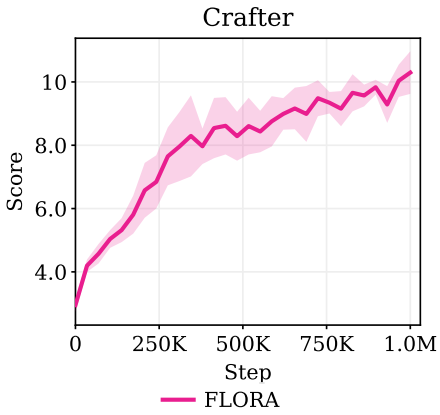

Figure 13: Training curve on Crafter over 1M environment steps. The solid line denotes the mean return across seeds, and the shaded region denotes standard deviation. FLORA steadily improves throughout training, indicating stable learning in a long-horizon partially observable environment.

As shown in Fig. 13, FLORA exhibits consistent improvement over the full training budget and reaches an average return 10.2 by the end of training. This trend supports the aggregate Crafter result reported in Sec. 4, suggesting that the combination of DR and DP provides a stable world-model learning signal for long-horizon visual control.

J Pseudocode for the Dual-latent Representation Loss

Algorithm 1 details the computation of $\mathcal{L}_{\mathrm{dr}}$ in Eq. (7), including the normalization and cross-correlation steps. Unlike R2-Dreamer (Morihira et al., 2026), which computes the cross-correlation between a projected world-model state and an image embedding, FLORA computes it between the projected feature-aligned latent $\hat{u}_t$ and the image features $\mathrm{sg}(u_t)$ extracted from the same observation.

Algorithm 1 Computation of $\mathcal{L}_{\mathrm{dr}}$ (PyTorch-style pseudocode)

Require: Observations $o \in \mathbb{R}^{B\times T\times C\times H\times W}$; feature extractor f_ϕ^d; categorical head q_ϕ^d (N tokens, K classes each); projector g_ϕ; redundancy weight λ_{bt}; $\epsilon = 10^{-8}$
Ensure: Dual-latent representation loss $\mathcal{L}_{\mathrm{dr}}$

$M \leftarrow B \cdot T$; $\quad o \leftarrow o.\mathrm{reshape}(M, C, H, W)$ ▷ all timesteps form one batch
$u \leftarrow f_\phi^d(o)$ ▷ $[M, D]$ image features $u_t = f_\phi^d(o_t)$
$\mathrm{logits} \leftarrow q_\phi^d(u)$ ▷ $[M, N, K]$ per-token logits
$d \leftarrow \mathrm{OneHotDist}(\mathrm{logits}).\mathrm{rsample}()$ ▷ $d_t \sim q_\phi^d(\cdot \mid u_t)$, straight-through
$\hat{u} \leftarrow g_\phi(d.\mathrm{flatten}(-2, -1))$ ▷ $[M, D]$ projected latent $\hat{u}_t = g_\phi(d_t)$
$C \leftarrow \textsc{CrossCorr}(u.\mathrm{detach}(), \hat{u})$ ▷ $[D, D]$, $\mathrm{CrossCorr}(\mathrm{sg}(u_t), \hat{u}_t)$
$\mathcal{L}_{\mathrm{dr}} \leftarrow \sum_i (1 - C_{ii})^2 + \lambda_{\mathrm{bt}} \sum_{i\neq j} C_{ij}^2$ ▷ Eq. (7)
return $\mathcal{L}_{\mathrm{dr}}$

function $\textsc{CrossCorr}(a, b)$ ▷ $a, b \in \mathbb{R}^{M\times D}$
  $a^{\mathrm{norm}} \leftarrow (a - a.\mathrm{mean}(0))/(a.\mathrm{std}(0) + \epsilon)$ ▷ standardize each dim over the batch
  $b^{\mathrm{norm}} \leftarrow (b - b.\mathrm{mean}(0))/(b.\mathrm{std}(0) + \epsilon)$
  return $a^{\mathrm{norm}\top} b^{\mathrm{norm}}/M$
end function

K HYPERPARAMETERS

Table 9 lists the hyperparameters used in our implementation.

Table 9: FLORA hyperparameters.

Parameter	Symbol	Setting
General		
Batch size	B	16
Sequence length	T	64
Optimizer	–	Adam (Kingma & Ba, 2015)
Image resolution	–	64×64 (RGB)
Replay capacity	–	10^6
World Model		
Number of tokens in z_t	–	32
Classes per token in z_t	–	32
KL prior loss scale	β_{dyn}	0.5
KL posterior loss scale	β_{reg}	0.1
World model learning rate	α	1×10^{-4}
Adam betas	β_1, β_2	$(0.9, 0.999)$
Adam epsilon	ϵ	1×10^{-8}
Gradient clipping	–	1000
AC-CPC		
Contrastive steps	K	10
Contrastive loss scale	–	1.0
Random crop resize scale	–	$(0.25, 1.0)$
Dual-latent Representation (DR)		
Number of tokens in d_t	–	32
Classes per token in d_t	–	32
Projector output dimension	–	1024
BT redundancy weight	λ_{bt}	5×10^{-4}
DR loss scale	–	0.05
Denoising Latent Predictor (DP)		
DP loss scale	–	0.5
Number of denoising steps	–	1
Actor–Critic		
Imagination horizon	H	15
Discount factor	γ	0.997
TD-λ	λ	0.95
Actor entropy scale	η	3×10^{-4}
Actor learning rate	α	3×10^{-5}
Critic learning rate	α	3×10^{-5}
Adam betas	β_1, β_2	$(0.9, 0.999)$
Adam epsilon	ϵ	1×10^{-5}
Gradient clipping	–	100

L DMC-GB2

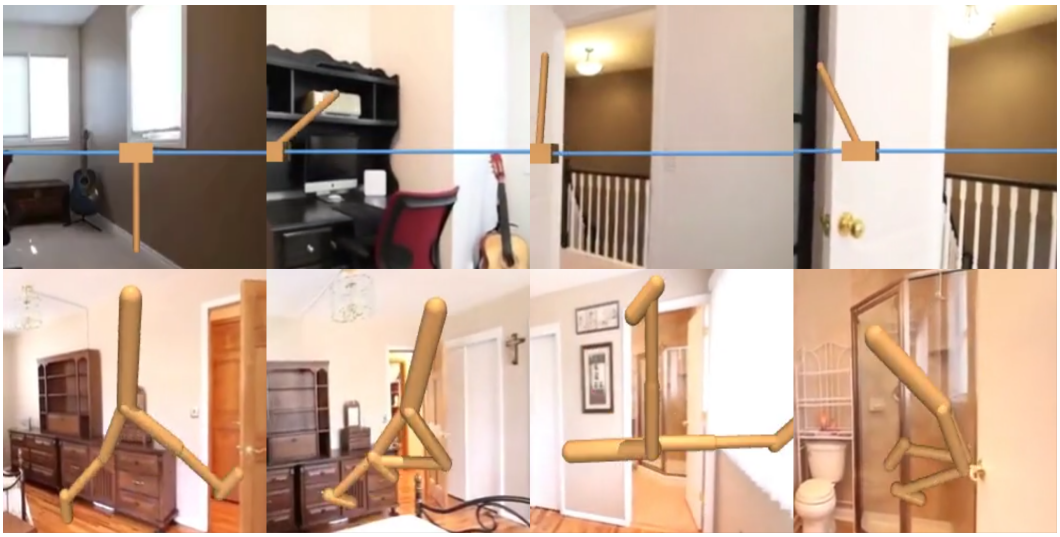

Figure 14: Example observations from the DMC-GB2 (Almuzairee et al., 2024) *Video Hard* setting used in our fidelity–relevance analysis. The top row shows *Cartpole Swingup*, and the bottom row shows *Walker Walk*. In this setting, the original background is replaced by natural video distractors while the underlying control dynamics remain unchanged.

DMC-GB2 (Almuzairee et al., 2024) is used in this work as a motivational benchmark for analyzing the fidelity–relevance trade-off discussed in Sec. 3.2. The benchmark extends the visual generalization setting of the DeepMind Control Suite by evaluating agents under visually shifted environments. This setup is useful for our analysis because it isolates a failure mode of reconstruction-heavy latent representations: when the observation contains visually rich but task-irrelevant variation, a reconstruction objective may allocate capacity to background appearance rather than to control-relevant foreground entities.

We focus on the *Video Hard* condition for two representative tasks, *Cartpole Swingup* and *Walker Walk*, as illustrated in Fig. 14. In this condition, the static background is replaced by natural video sequences, creating strong non-stationary visual distractors while leaving the underlying control task unchanged. This makes *Video Hard* a direct stress test for whether the learned representation tracks task-relevant dynamics or instead entangles them with unstable background appearance.

We use DMC-GB2 as a compact motivational study rather than as a main benchmark with a full quantitative comparison. The goal is to illustrate why DR is needed. Under severe visual distractions, TWISTER's reconstructive latent z_t tends to entangle the foreground agent with the changing background, causing task-relevant objects to become blurred or underrepresented in reconstruction. In contrast, FLORA's feature-aligned latent d_t provides a complementary representation that better captures the dynamic objects relevant for control. This qualitative behavior is visualized in Fig. 3, and the corresponding learning curves in Fig. 4 show that augmenting TWISTER with DR improves robustness on representative DMC-GB2 tasks. Together, these results support our use of DMC-GB2 as evidence that task-relevant abstraction is valuable when visual fidelity and control relevance pull the representation in different directions.

M AGENT BEHAVIOR LEARNING

Following prior world-model agents (Hafner et al., 2020; 2021; 2023; Zhang et al., 2023; Burchi & Timofte, 2025a;b), FLORA trains its actor $\pi_\theta(a_t \mid s_t)$ and critic $v_\psi(s_t)$ entirely on imagined trajectories. We keep this procedure identical to TWISTER so that performance differences can be attributed to the world model. Imagination starts from world-model states inferred on replayed sequences and unrolls the world model for $H = 15$ steps under the current policy. Each imagined latent is generated by the denoising latent predictor (Eq. (11)), and rewards and continuation flags are predicted from the imagined world-model states $\hat{s}_t$.

The critic is trained by maximum likelihood to predict λ-returns,

$$\mathcal{L}(\psi) = -\mathbb{E}\Big[\sum_{t=1}^{H} \ln p_\psi\big(\mathrm{sg}(R_t^\lambda) \mid \hat{s}_t\big)\Big], \qquad R_t^\lambda = \hat{r}_{t+1} + \gamma\, \hat{c}_{t+1}\big((1-\lambda)\, v_\psi(\hat{s}_{t+1}) + \lambda R_{t+1}^\lambda\big), \tag{12}$$

with $\gamma = 0.997$ and $\lambda = 0.95$, and is regularized toward the predictions of an exponential moving average of its own parameters. The actor is optimized with REINFORCE (Williams, 1992) and an entropy regularizer:

$$\mathcal{L}(\theta) = -\mathbb{E}\Big[\sum_{t=1}^{H} \mathrm{sg}\Big(\frac{R_t^\lambda - v_\psi(\hat{s}_t)}{\max(1, S)}\Big) \ln \pi_\theta(a_t \mid \hat{s}_t) + \eta\, \mathcal{H}\big[\pi_\theta(\cdot \mid \hat{s}_t)\big]\Big], \tag{13}$$

where $\eta = 3 \times 10^{-4}$ and S is an exponential moving average of the range between the 5th and 95th percentiles of the returns. The remaining hyperparameters are listed in Appendix K.