PRESHORT: EFFICIENT WORLD MODELING WITH PRETRAINED VISUAL REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Model-based reinforcement learning (MBRL) leverages learned world models for sample-efficient control, provided that their representations retain the information needed to predict the consequences of actions. General-purpose pretrained visual representations offer reusable features learned beyond an agent's online experience, motivating their use as a prediction space for world modeling, while turning these features into imagined trajectories requires learning action-conditioned transitions and accounting for uncertain future states. We introduce PreShort, an online world model that employs frozen visual features as both observation representations and continuous imagination targets. Its core component, Differential Shortcut Dynamics, encodes latent-action history with differential attention and conditions a stochastic shortcut predictor on the resulting context. The context is reused across a few refinement steps to generate each successor, and imagined trajectories support actor-critic learning. Freezing the visual interface also enables feature caching and confines online world model updates to downstream modules. Experiments show that PreShort achieves a mean human-normalized score of 146.7% on Atari 100K, a mean return of 821.5 on the DeepMind Control Suite, and a return of 10.7 on Crafter, exceeding strong worldmodel baselines in mean aggregate performance across all three benchmarks. Further analyses assess rollout accuracy, compatibility with different pretrained visual backbones, and computational efficiency in terms of trainable parameters and training cost.

1 INTRODUCTION

Deep reinforcement learning has achieved substantial progress in decision-making, yet learning effective policies from limited interaction remains challenging, particularly in visual environments (Schulman et al., 2017; Haarnoja et al., 2018; Hafner et al., 2025a). World model-based reinforcement learning addresses this challenge by learning latent dynamics from experience and optimizing policies on imagined trajectories, thereby reducing the need for additional environment interaction (Hafner et al., 2020; 2025a). Its effectiveness depends on informative history representations and successor prediction that support reliable policy learning over repeated imagined rollouts.

World models learn action-conditioned dynamics over visual representations, often training both jointly through reconstruction and transition modeling (Hafner et al., 2025a; Zhang et al., 2023). This requires learning which visual details to preserve alongside their temporal evolution, while approaches based on variational autoencoders (VAE) incur additional reconstruction costs (Morihira et al., 2026). For example, in Atari's game *Pong*, the ball occupies few pixels, yet tracking its position is essential for predicting its trajectory and paddle interactions (Zhang et al., 2025). If the encoder fails to preserve these details, the temporal model receives incomplete information, making accurate dynamics learning more difficult, particularly during early training (Zhang et al., 2026).

Pretrained visual encoders offer representations learned beyond the buffer's limited online experience. When these representations preserve the visual details needed for control, they provide an informative basis for learning action-conditioned transitions and reduce the burden of learning perception and dynamics jointly. Freezing the encoder further maintains a stable prediction space as the world model learns from newly collected experience. DINO-WM (Zhou et al., 2025) and V-JEPA 2-AC (Assran et al., 2025) learn dynamics over frozen features for goal-conditioned planning, while OC-STORM (Zhang et al., 2026) augments an online world model with object representations from

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.

an additional pretrained segmentation network, supplying richer decision-relevant information. RAE (Zheng et al., 2026) further demonstrates that visual representations can support generative modeling. Building on these findings, we incorporate pretrained visual encoders into world models that generate imagined trajectories for reinforcement learning agents. Building on this idea, we learn action-conditioned temporal representations and stochastic dynamics online to support imagination-based policy learning.

Modeling uncertain futures in a continuous pretrained representation space requires representing multiple possible successors. Deterministic predictions trained with squared error tend toward the conditional mean, potentially averaging over distinct outcomes. In contrast, categorical VAE models address such uncertainty through distributions over discrete latent states (Zhang et al., 2023; Hafner et al., 2025a), while continuous generative dynamics offer a way to model it directly from the noise space to the pretrained feature space. For iterative generators, however, sampling requires repeated model evaluations for each imagined transition (Alonso et al., 2024). Reusing a temporal summary as the generator's history input reduces repeated history access (Meo et al., 2025), and therefore makes the information retained during compression critical. Under partial observability, this compact state must preserve action-relevant dependencies for both dynamics prediction and policy learning.

In this paper, we introduce *PreShort*, an online world model architecture that performs self-supervised training and accurate imagination in the continuous representation space of a frozen pretrained visual encoder. PreShort uses *differential shortcut dynamics (DSD)*, a context-conditioned stochastic architecture, to model transitions efficiently in this space. DSD first aggregates latent states and actions into a compact context representation using differential attention. Successor generation starts from Gaussian noise, introducing stochasticity and allowing different possible futures to be sampled under the same context. DSD further incorporates token attention into a shortcut predictor to combine the context with the evolving candidate latent and refinement conditions. Reusing the same context within each transition avoids repeated history processing, while shortcut learning supports conditional successor generation in a few refinement steps. Our main contributions are:

- **Pretrained representations for online imagination.** We show that frozen pretrained visual representations can support stochastic imagination and sample-efficient policy learning in online MBRL.
- **Temporal adaptation for stochastic dynamics.** We develop Differential Shortcut Dynamics (DSD), which learns action-conditioned history summaries under dynamics and task supervision and reuses them throughout shortcut successor generation.
- **Efficient and backbone-agnostic performance.** PreShort achieves 146.7% mean HNS on Atari 100K, a mean return of 821.5 on DeepMind Control, and 10.7 on Crafter at 1M steps, with faster training and fewer online-trainable parameters than the compared world models; the same formulation transfers across frozen visual backbones with only input-interface adjustments.

2 Related Work

A prominent line of online model-based reinforcement learning represents uncertainty through discrete stochastic latent states: DreamerV3 (Hafner et al., 2025a) and STORM (Zhang et al., 2023) learn observation-conditioned posteriors and history-conditioned predictive priors in a categorical latent space. In contrast, PreShort models stochastic transitions directly over frozen continuous visual representations, and its DSD uses differential attention (Ye et al., 2025) to selectively aggregate histories into a compact context.

Dreamer 4 combines a pretrained continuous tokenizer with shortcut forcing but learns policies in imagination from offline data (Hafner et al., 2025b), while JEDI jointly learns its visual encoder, latent diffusion dynamics, and task predictors online (Lim et al., 2026). PreShort instead integrates an externally pretrained, frozen visual encoder into the online loop: it adapts temporal context and stochastic dynamics from newly collected experience and trains a policy on imagined rollouts in the same continuous feature space, reusing pretrained perception while preserving the cycle of environment interaction, world-model adaptation, and imagination-based policy improvement.

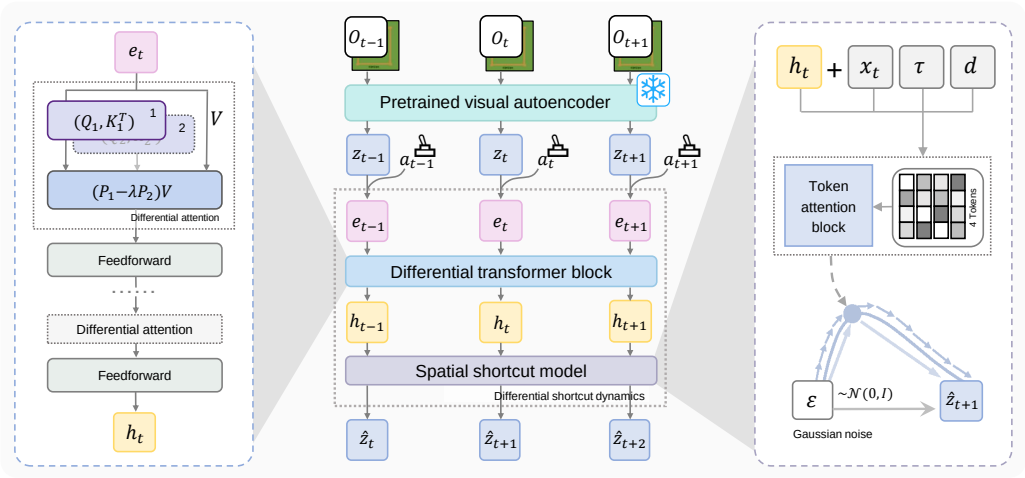

Figure 1: Overview of PreShort. A frozen visual encoder provides continuous latents, while the differential shortcut dynamics combines temporal context encoding (left) with token attention shortcut generation (right) for prediction.

DINO-WM (Zhou et al., 2025) and V-JEPA 2-AC (Assran et al., 2025) learn action-conditioned transitions over frozen visual representations, but their rollouts only evaluate candidate action sequences for goal-conditioned planning rather than training an imagination-based actor critic. OC-STORM (Zhang et al., 2026) demonstrates the value of frozen object representations for online world modeling. The pretraining objective further shapes the space available for prediction: visual autoencoders such as Wan-VAE (Wan, 2025) learn compact representations for reconstruction, while representation autoencoders pair frozen pretrained encoders with learned decoders for generative modeling (Zheng et al., 2026). PreShort studies frozen visual representations as continuous successor-prediction targets for stochastic imagination and online policy optimization, accommodating these diverse backbones by modeling dynamics over the flattened continuous space, without relying on a particular feature layout. A detailed discussion of related work is provided in Appendix A.

3 Method

We formulate online visual control as a partially observable Markov decision process (POMDP) defined by the tuple $(\mathcal{O}, \mathcal{A}, \mathcal{T}, R, \gamma)$. Here, $\mathcal{O}$ and $\mathcal{A}$ denote the observation and action spaces, $\mathcal{T}$ the environment transition function, R the reward function, and $\gamma \in [0, 1)$ the discount factor. At time t, the agent receives an image observation $o_t \in \mathcal{O}$, selects an action $a_t \in \mathcal{A}$, and receives a reward r_t and a continuation indicator $c_t \in \{0, 1\}$. Because a single observation may not fully reveal the underlying environment state, the policy conditions on the interaction history and specifies an action distribution $\pi_\theta(a_t \mid o_{\leq t}, a_{<t})$ and is optimized to maximize the expected discounted return $J(\theta) = \mathbb{E}_{\pi_\theta, \mathcal{T}}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$, where γ determines the relative weight of future rewards.

3.1 Continuous World Modeling

Figure 1 illustrates the architecture of PreShort. Given observation and action sequences, a frozen pretrained visual model provides stable visual representations, while a temporal model compresses the visual and action history into summarized hidden states. Shortcut dynamics then predict the next latent, and the hidden state is also used to estimate the reward and continuation signals.

3.1.1 World Model Structure

We aim to design a world model that accommodates continuous representations from different pretrained visual encoders without relying on a particular spatial feature layout (Siméoni et al., 2026; Wan, 2025). We therefore flatten the encoder output into a continuous vector and use this vector as

the common interface for action conditioning and successor prediction. This design allows the dynamics formulation to accommodate different visual backbones through their vector representations, without requiring a backbone-specific architecture. Writing the complete frozen visual interface as E_{pre}, we obtain

$$\text{Visual encoder:} \qquad z_t = \text{sg}(E_{\text{pre}}(o_t)) \,. \tag{1}$$

Here, $\text{sg}(\cdot)$ denotes the stop-gradient operator, and the pretrained visual encoder remains frozen throughout online learning. To account for differences in numerical scale across pretrained visual representations, we apply a fixed normalization and bounding transformation to stabilize continuous latent dynamics learning (see Appendix C.1 for details).

The visual latent z_t and action encoding are concatenated and passed through a shared MLP to form a temporal token e_t. A differential transformer encodes these tokens into context representations $h_{1:T}$, which condition successor prediction and the reward and continuation heads:

$$\begin{aligned}
\text{Action mixer:} \qquad & e_t = m_\phi(z_t, a_t), \\
\text{Sequence model:} \qquad & h_{1:T} = f_\phi(e_{1:T}), \\
\text{Dynamics predictor:} \qquad & \hat{z}_{t+1} = g_\phi^{\text{sc}}(x_{t+1,\tau}, \tau, d \mid h_t), \\
\text{Reward predictor:} \qquad & \hat{r}_t = g_\phi^R(h_t), \\
\text{Continuation predictor:} \qquad & \hat{c}_t = g_\phi^C(h_t).
\end{aligned} \tag{2}$$

The path construction, step-size conditioning τ and d, intermediate latent state x_τ, and shortcut targets of g_ϕ^{sc} are detailed in the following subsection.

3.1.2 Differential Shortcut Dynamics

Learning dynamics over frozen continuous representations requires preserving meaningful latent states during autoregressive imagination. In categorical world models such as DreamerV3 (Hafner et al., 2025a) and subsequent approaches built on the paradigm, the stochastic state comprises 32 categorical variables, each with 32 classes. Each variable is sampled as a one-hot vector, restricting predictions to a predefined set of discrete codes. A prediction error therefore changes the selected code while preserving its discrete structure. Continuous predictions lack this constraint: even a small numerical deviation can move a predicted latent outside the regions occupied by valid observation encodings. Such deviations may yield semantically invalid representations, rather than merely inaccurate predictions of valid states. During autoregressive imagination, these representations feed subsequent transitions, allowing errors to compound and undermine the reward and value estimates used for policy learning. This challenge is particularly relevant during online learning, when dynamics are learned from limited and continually collected experience. We therefore introduce Differential Shortcut Dynamics to efficiently generate plausible continuous successor states while preserving stochasticity during multi-step rollouts.

Differential attention for temporal modeling The sequence model must summarize the interaction context into a compact state that supports both dynamics prediction and policy learning, retaining informative temporal dependencies under partial observability while limiting interference from redundant observations. Since increasing the depth or width of a transformer does not improve final performance in the evaluated settings (Zhang et al., 2023; 2026), we instead improve how a fixed-size temporal encoder uses its representational capacity under a limited interaction budget.

We adopt differential attention, which subtracts two softmax attention maps with a learnable coefficient to suppress shared attention noise and emphasize relevant contextual information (Ye et al., 2025). Applied causally to the representation and action tokens, it produces a history representation h_t that conditions both successor-state generation and task prediction; during imagination, the same representation serves transition generation and action selection. Differential attention introduces no additional parameters over standard self-attention: we halve the dimensions of the query, key, and value projections to perform the differential operation, ensuring a fair comparison in parameter count. We compare differential attention with standard self-attention in Section 4.3, with implementation details provided in Appendix G.2.

Token attention for shortcut prediction. The context representation supplies information about the past, but the future state may remain uncertain. We therefore use a shortcut model to generate

successor latents from Gaussian noise conditioned on the context representation h_t. Noise initialization introduces stochasticity into dynamics modeling of continuous visual representations, allowing potentially different future states to be generated under the same context. Step-size conditioning and shortcut consistency support generation in a few refinement steps, limiting the cost of repeatedly applying this model during imagination (Frans et al., 2025).

At each refinement step, successor prediction depends on the temporal context, the intermediate latent state x_τ, the refinement time τ, and the step size d. We further introduce token attention into the shortcut model to integrate these inputs. Unlike Dreamer 4, which processes sequences of actions and corrupted representations using an autoregressive transformer, DSD first summarizes the temporal context into h_t and then performs shortcut generation through token attention within each transition. This separation allows the temporal context to be reused across refinement steps. Separate embeddings map h_t, $x_{t+1,\tau}$, τ, and d into four tokens. After the projection, standard token attention is applied across these tokens within each transition. The updated latent token is read by an endpoint head to predict the clean successor. We adopt x-space prediction to directly estimate the clean successor latent rather than the velocity. This choice is motivated by the observation that high-frequency outputs associated with velocity prediction can introduce errors that accumulate during autoregressive generation (Hafner et al., 2025b). Predicting clean latent endpoints aims to support stable imagination, with the refinement direction computed from the predicted endpoint.

The finest finite step is $d_{\min} = 1/M$, with $M = 64$. For direct supervision, we use $d = 0$ and sample $\tau \sim \mathcal{U}[0, 1)$. For shortcut consistency, we sample $d \sim \mathcal{U}\{1, 1/2, \ldots, 2d_{\min}\}$ and a compatible path position $\tau \sim \mathcal{U}\{0, d, \ldots, 1 - d\}$. The network maps step inputs at or below $d_{\min}$ to the zero-step condition. Given a noise point, the intermediate latent, endpoint prediction, and corresponding update direction are

$$x_{t+1,\tau} = (1 - \tau)x_{t+1,0} + \tau z_{t+1}, \tag{3a}$$

$$\widehat{z}_{t+1|\tau,d} = g_\phi^{\mathrm{sc}}(h_t, x_{t+1,\tau}, \tau, d), \tag{3b}$$

$$\widehat{v}_{t+1|\tau,d} = \frac{\hat{z}_{t+1|\tau,d} - x_{t+1,\tau}}{1 - \tau}. \tag{3c}$$

Here, $x_{t+1,0}$ is obtained from Gaussian noise using the fixed bounding transformation described in Appendix C. The network predicts the clean endpoint in latent space, and the update direction is derived from this prediction. The training objectives are given in Section 3.1.3, and the inference procedure for imagination is described in Section 3.2.

3.1.3 World Model Training Objective

The world model is optimized on replay sequences while keeping the pretrained visual autoencoder frozen. The trainable components include the action mixer, sequence model, shortcut dynamics, and predictors in Equation 2. The overall objective combines dynamics learning, shortcut consistency, reward prediction, and continuation prediction:

$$\mathcal{L}_{\mathrm{world}} = \beta_{\mathrm{sc}}\mathcal{L}_{\mathrm{sc}} + \mathcal{L}_{\mathrm{rew}} + \mathcal{L}_{\mathrm{con}} + \mathcal{L}_{\mathrm{dyn}}, \tag{4}$$

where $\beta_{\mathrm{sc}} = 0.25$ controls the weight of shortcut consistency. $\mathcal{L}_{\mathrm{rew}}$ and $\mathcal{L}_{\mathrm{con}}$ supervise the reward and continuation prediction using symlog two-hot cross entropy and binary cross entropy:

$$\begin{aligned} \mathcal{L}_{\mathrm{rew}} &= \frac{1}{BT}\sum_{b=1}^{B}\sum_{t=1}^{T} \mathrm{SymlogCrossEnt}(r_t, \hat{r}_t)\,, \\ \mathcal{L}_{\mathrm{con}} &= \frac{1}{BT}\sum_{b=1}^{B}\sum_{t=1}^{T} \mathrm{BinaryCrossEnt}(c_t, \hat{c}_t)\,. \end{aligned} \tag{5}$$

Here, B is the batch size and T is the number of transitions per sequence. Both objectives provide task-relevant supervision for the temporal context representation. Using the sampling scheme and the endpoint prediction in Equation 3, we supervise the zero-step branch with the observed successor latent as follows

$$\mathcal{L}_{\mathrm{dyn}} = \frac{1}{BT}\sum_{b=1}^{B}\sum_{t=1}^{T} \left\| \widehat{z}_{b,t+1|\tau,0} - \mathrm{sg}(z_{b,t+1}) \right\|_2^2. \tag{6}$$

For shortcut consistency, two consecutive half-step predictions construct a teacher endpoint. With the batch index suppressed and $x = x_{t+1,\tau}$, the half-step directions and teacher target are

$$\begin{aligned} v_1 &= \frac{g_{\mathrm{sc}}(h_t, x, \tau, d/2) - x}{1-\tau}, & x' &= x + \frac{d}{2} v_1, \\ v_2 &= \frac{g_{\mathrm{sc}}(h_t, x', \tau + d/2, d/2) - x'}{1-\tau-d/2}, & z^{\mathrm{teach}}_{t+1|\tau,d} &= \mathrm{sg}\left[x + (1-\tau)\frac{v_1 + v_2}{2} \right]. \end{aligned} \tag{7}$$

Both half-step evaluations use the same detached context and run without gradients. The average update direction is converted into an endpoint in x-space, which supervises the coarse prediction

$$\mathcal{L}_{\mathrm{sc}} = \frac{1}{BT} \sum_{b=1}^{B} \sum_{t=1}^{T} \left\| \widehat{z}_{b,t+1|\tau,d} - z^{\mathrm{teach}}_{b,t+1|\tau,d} \right\|_2^2. \tag{8}$$

Base noise and refinement variables are sampled for each transition, with their batch and time indices omitted for readability. Each transition contributes to both losses, which are averaged separately.

3.2 Policy Learning

The learned world model supports policy optimization through on-policy imagination in continuous latent space. DSD generates trajectories under the current policy, providing predicted rewards and continuation signals for actor and critic training.

Imagination Rollout Each imagined trajectory starts from a short context sampled from replay. The current policy selects actions, and DSD generates trajectories autoregressively. Following the established few-step setting of shortcut models (Frans et al., 2025), we use $K = 4$ equal steps, with $d = 1/K$ and $\tau = k/K$ for $k = 0, \ldots, K-1$, and take $x_{t+1,1}$ as the generated successor. The resulting trajectories are generated without gradients and used to train the actor and critic.

Actor-Critic Learning To isolate the contribution of world modeling, we use the same actor–critic learning procedure as in prior MBRL methods such as DreamerV3 and STORM. The policy is trained exclusively on imagined trajectories generated by PreShort, where the actor and critic share the decision state $s_t = [z_t, h_{t-1}]$. The only difference is that z_t lies in the frozen pretrained visual representation space.

4 Experiments

In this section, we evaluate whether frozen pretrained visual representations and differential shortcut dynamics can support effective and computationally efficient online policy learning. Specifically, we address the following questions:

1. How does PreShort perform compared with established online world model-based agents across different benchmarks under limited interaction budgets? (Section 4.2)
2. How do differential attention and token attention-based shortcut dynamics contribute to control performance when policies are trained on imagined trajectories? (Section 4.3)
3. How accurately does PreShort predict future states during autoregressive imagination as the rollout horizon increases over multiple transitions? (Section 4.4.1)
4. What computational savings does PreShort achieve by freezing the visual backbone and focusing online optimization on temporal encoding and latent dynamics? (Section 4.4.2)
5. How well does PreShort generalize across frozen pretrained visual backbones with different pretraining objectives and representation properties? (Appendix D.1)
6. Can simply incorporating pretrained visual features into a categorical world model improve control performance while retaining its categorical dynamics formulation? (Appendix D.2)

We report mean and median human-normalized scores (HNS) across Atari 100K games, mean and median returns across DeepMind Control Suite tasks, and final returns on Crafter. Detailed evaluation protocols and hyperparameter settings are provided in Appendices F and F.2. Unless otherwise stated, all results are reported as the mean over 5 random seeds.

4.1 BENCHMARKS

We evaluate PreShort on benchmarks covering discrete and continuous control, as well as exploration and long-horizon decision making.

- **Atari 100K.** 26 games with 100K action steps per game, testing sample-efficient discrete control. We report mean and median HNS $= (S_{\text{agent}} - S_{\text{random}})/(S_{\text{human}} - S_{\text{random}})$.
- **DeepMind Control Suite.** 20 visual continuous-control tasks, each learned from images within 1M environment steps. We report mean and median returns across tasks.
- **Crafter.** A procedurally generated survival environment with visual observations and discrete actions. Agents train for 1M environment steps; we report the final return.

4.2 MAIN EXPERIMENTS

We compare PreShort with a set of model-based methods that employ architectures including a recurrent state-space model (RSSM), a transformer, and a score-based diffusion model. Model-free, lookahead-search, offline imitation or planning-based approaches (such as DINO-WM and Dreamer 4) are excluded, since our focus is on improving online world models and differences in data access and control objectives make direct comparisons unfair. The main experiments adopt the frozen Wan-VAE encoder for its continuous representation, since its native decoder supports the visualizations presented in (Wan, 2025). Compatibility with additional pretrained visual encoders without decoders is examined in Appendix D.1. To ensure a fair comparison, we use 64×64 visual observations as inputs to the pretrained visual encoder, matching the input resolution of the baselines and avoiding an advantage from higher-resolution observations.

The results are shown in Figure 3 with blue lines and Table 1. Compared with baselines, PreShort achieves superior performance, particularly on tasks where reward-relevant targets are difficult to capture, such as *Pong* and *Gopher*. This advantage extends to *Freeway*, a challenging sparse-reward task on which most baselines obtain zero return (without demonstration). PreShort rapidly achieves a non-zero return early in training (as the detailed curve shows in Appendix E.1). We attribute this to the pretrained visual representations, which accurately encode these visual features and provide sufficient information for decision-making. PreShort also excels on tasks with complex dynamics, such as *Boxing* and *Crafter*, where DSD accurately models the environment transitions (as we demonstrate in Subsection 4.4.1). As shown in Figure 2, after only 10K environment interactions, PreShort already generates substantially more accurate imagined trajectories than the online baselines. Notably, these improvements are attained with substantially improved computational efficiency, requiring only 1.1 hours per Atari 100K task on a single NVIDIA RTX 4090 graphics card, compared with 3.2 hours for STORM (reproduced), 7.9 hours for DreamerV3-torch (reproduced), and 3.09 days for DIAMOND (reproduced).

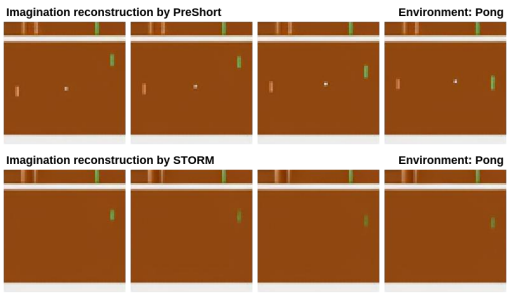

Figure 2: Imagined trajectories in *Pong* after 10K interactions. Successive decoded frames imagined by PreShort (top) and STORM (bottom) under the same context and action sequence. PreShort preserves small reward-relevant objects throughout the rollout.

4.3 ABLATION STUDIES

We next examine how the proposed visual interface and dynamics formulation support online control. We evaluate the roles of shortcut generation and differential attention in efficient stochastic imagination and action-conditioned temporal modeling. The architectures of all ablated models are detailed in Appendix G. Due to space constraints, ablations on pretrained visual encoders and categorical world models are deferred to Appendix D.1 and D.2.

Table 1: Aggregate scores of PreShort and the selected baselines.

Atari 100K					
	DreamerV3	STORM	DIAMOND	OC-STORM	PreShort
HNS Mean	125%	122.3%	146.1%	134.8%	**146.7%**
HNS Median	49%	42.8%	37.3%	43.8%	**76%**
DeepMind Control Suite					
	DreamerV3	STORM	TD-MPC2	TWISTER	PreShort
Task Mean	786	737.2	720.9	801.8	**821.5**
Task Median	861	801.1	795.9	**907.6**	877.7
Crafter @ 1M					
	IRIS	Δ-IRIS	DreamerV3	STORM	PreShort
Return@1M	5.5	7.7	9.4	8.7	**10.7**

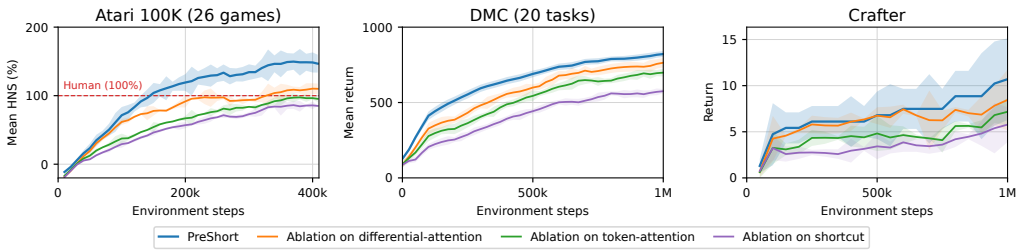

Figure 3: Mean training curves on Atari 100K (26 games, mean HNS), DMC (20 tasks, mean return), and Crafter (return). PreShort (blue) is compared against several ablation curves: standard self-attention replacing differential attention (orange); concatenated conditions replacing the four-token attention interface (green); and a parameter-matched deterministic MLP replacing stochastic shortcut generation (purple). Shaded regions denote the standard deviation across 5 seeds.

We first replace differential attention in the sequence model with standard self-attention to assess its contribution to control performance. Learning policies from imagined trajectories requires modeling uncertainty in continuous latent transitions while maintaining inference efficiency. We therefore also replace the shortcut dynamics in PreShort with a parameter-matched MLP to investigate whether the stochasticity introduced by shortcut dynamics improves the quality of imagined trajectories. To isolate the contribution of the four-token attention interface itself, we further concatenate h_t, x, τ, d into a single vector and feed it to a fixed-size, parameter-matched network, which receives the same conditions but removes the candidate latent's explicit query over the individual inputs.

The results are shown in Figure 3, and each simplification reduces control performance: standard self-attention retains less useful temporal context, the deterministic MLP accumulates larger rollout errors, and concatenating the conditions into one vector removes the benefit of the token attention interface. The full model performs best, supporting the contribution of both differential temporal encoding and token-conditioned stochastic shortcut generation.

4.4 Further Analysis

Beyond the policy's performance, we examine the quality of imagined trajectories and the resources required for online adaptation. In Section 4.4.1, we measure the pixel-level fidelity of 16-step autoregressive rollouts by decoding the predicted features and computing PSNR against ground-truth frames across the rollout horizon. Section 4.4.2 then quantifies the trainable parameter savings from freezing the visual backbone and examines the resulting allocation of online updates to task-specific representations and dynamics.

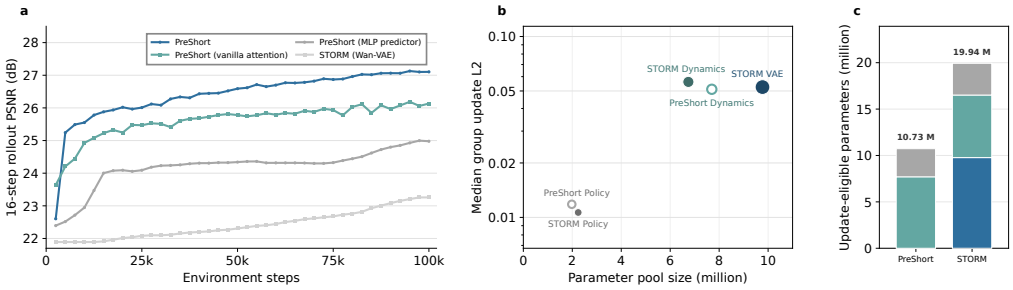

Figure 4: Rollout accuracy and parameter efficiency. (a) PSNR of decoded 16-step rollouts on Boxing throughout online training. (b) Median group update L2 magnitude versus parameter count. (c) Update-eligible parameter counts for PreShort and STORM. The rollout comparison in (a) uses STORM with frozen Wan-VAE features; (b,c) use STORM with its trainable visual autoencoder.

4.4.1 Rollout Accuracy

We evaluate rollout accuracy on *Boxing* using four variants from the preceding ablation studies. All variants are conditioned on the same observed history and rolled out autoregressively under the same recorded action sequence. We decode the predicted visual features using the frozen Wan-VAE decoder and compute the peak signal-to-noise ratio (PSNR) against the corresponding ground-truth frames at each prediction step, assessing how prediction accuracy changes with the rollout horizon (Hansen & Wang, 2026).

The results are shown in Figure 4(a). Full PreShort achieves the highest PSNR among the four variants, followed by the standard self-attention variant, the parameter-matched MLP variant, and the adapted STORM baseline. The decrease in prediction accuracy when differential attention was replaced with standard self-attention supports its contribution to accurate rollout prediction. Replacing shortcut dynamics with a parameter-matched MLP led to a larger reduction, suggesting that parameter count alone does not explain the benefit of the shortcut dynamics architecture. STORM achieved the lowest PSNR despite using the same frozen Wan-VAE features, indicating that the shared visual representation alone was insufficient to match PreShort's prediction accuracy. Together, these comparisons support the contributions of both differential attention and shortcut dynamics to the fidelity of autoregressive rollouts.

4.4.2 Parameter Efficiency

By reusing a frozen visual backbone, PreShort avoids updating the visual encoder and decoder during online learning. We compare the update-eligible parameter counts and median group update L2 magnitudes with those of STORM using its trainable visual autoencoder. Figure 4(c) shows that PreShort updates 10.73M parameters online, 46.2% fewer than STORM's 19.94M. Its dynamics and policy groups exhibit update magnitudes similar to those of their STORM counterparts, while avoiding updates to the visual autoencoder (Figure 4(b)).

5 Conclusion

In this work, we present PreShort, an online world model that learns and imagines in the continuous representation space of a frozen pretrained visual encoder. Its core component, Differential Shortcut Dynamics, uses differential attention to compress latent–action histories into a context that is computed once per transition, and a token attention shortcut predictor that turns Gaussian noise into stochastic successors in as few as four refinement steps. On Atari 100K, the DeepMind Control Suite, and Crafter, PreShort attains a 146.7% mean HNS, a mean return of 821.5, and a return of 10.7, respectively, while training an Atari agent in 1.1 hours on a single RTX 4090 and updating 46.2% fewer online-trainable parameters than baseline. These results support frozen pretrained visual representations as a practical interface for online imagination-based reinforcement learning, separating perception from online dynamics and policy adaptation.

AI USE STATEMENT

In this work, we used generative AI tools for literature search, manuscript editing, and source-code formatting. We did not use generative AI tools for research conceptualization, method design and experimental design, and other required-disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for code explanation, layout suggestions, and simulated peer review. We reviewed all AI-assisted work, which was verified and tested for correctness by all authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, we provide additional implementation details in Appendix C, experimental settings, model architecture, and hyperparameters in Appendix F, and detailed ablation structures in Appendix G.

REFERENCES

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

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15619–15629, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew J. Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *CoRR*, abs/2506.09985, June 2025.

M. G. Bellemare, Y. Naddaf, J. Veness, and M. 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. Learning transformer-based world models with contrastive predictive coding. In *International Conference on Learning Representations*, volume 2025, pp. 6949–6970, 2025.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision*, pp. 9630–9640, 2021.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations*, volume 2025, pp. 34668–34684, 2025.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, volume 80, pp. 1861–1870, 2018.

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

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

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025a.

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

Nick Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, volume 2024, pp. 47376–47405, 2024.

Nicklas Hansen and Xiaolong Wang. Hallucination in world models is predictable and preventable. *arXiv preprint arXiv:2606.27326*, 2026.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15979–15988, 2022.

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

Jing Yu Lim, Rushi Shah, Zarif Ikram, Samson Yu, Haozhe Ma, Tze-Yun Leong, and Dianbo Liu. JEDI: Joint embedding diffusion world model for online model-based reinforcement learning. *arXiv preprint arXiv:2605.13013*, 2026.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: Towards universal visual reward and representation via value-implicit pre-training. In *International Conference on Learning Representations*, 2023.

Cristian Meo, Mircea Tudor Lică, Zarif Ikram, Akihiro Nakano, Vedant Shah, Aniket Rajiv Didolkar, Dianbo Liu, Anirudh Goyal, and Justin Dauwels. Masked generative priors improve world models sequence modelling capabilities. In *ICLR Workshop on World Models: Understanding, Modelling and Scaling*, 2025.

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

Vincent Micheli, Eloi Alonso, and François Fleuret. Efficient world models with context-aware tokenization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, 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*, 2026.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *Conference on Robot Learning*, 2022.

NVIDIA. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seung Eun Yi, Michael Ramamonjisoa, Francisco Massa, Daniel HAZIZA, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Herve Jegou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *Transactions on Machine Learning Research*, 2026.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy P. Lillicrap, and Martin A. Riedmiller. Deepmind control suite. *CoRR*, abs/1801.00690, 2018.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Team Wan. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. Differential transformer. In *International Conference on Learning Representations*, volume 2025, pp. 144–164, 2025.

Lijun Yu, José Lezama, Nitesh Bharadwaj Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David Ross, and Lu Jiang. Language model beats diffusion - tokenizer is key to visual generation. In *International Conference on Learning Representations*, volume 2024, pp. 765–783, 2024.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision*, pp. 11941–11952, 2023.

Boxuan Zhang, Runqing Wang, Wei Xiao, Weipu Zhang, Jian Sun, Gao Huang, Jie Chen, and Gang Wang. DyMoDreamer: World modeling with dynamic modulation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 135488–135523, 2025.

Weipu Zhang, Gang Wang, Jian Sun, Yetian Yuan, and Gao Huang. STORM: Efficient stochastic transformer based world models for reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 27147–27166, 2023.

Weipu Zhang, Adam Jelley, Trevor McInroe, Amos Storkey, and Gang Wang. Object-centric world models from few-shot annotations for sample-efficient reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 91366–91394, 2026.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *International Conference on Learning Representations*, volume 2026, pp. 35791–35820, 2026.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *International Conference on Machine Learning*, 2025.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. Image BERT pre-training with online tokenizer. In *International Conference on Learning Representations*, 2022.

A Further Related Works

We clarify the distinctions between PreShort and recent approaches in the world model as follows:

Table 2: Comparison between PreShort and closely related world-model approaches.

Attribute	STORM	Dreamer 4	DINO-WM	JEDI	PreShort (ours)
Online/Offline	Online	Offline	Offline	Online	Online
Agent training	Actor–critic	Imagination RL	Zero-shot	Actor–critic	Actor–critic
Sequence model	Transformer	Transformer	ViT	Diffusion	Diff. transformer
Transition	MLP	Shortcut			Token-attn. shortcut
Visual interface	Categorical VAE	Causal tokenizer	DINOv2	Conv encoder	Wan-VAE
Pixel recon.	Yes	Yes	No	No	No
Latent structure	Discrete	Continuous	Continuous	Continuous	Continuous
Encoder update	Jointly optimized	Frozen	Frozen	End-to-end	Frozen

- **STORM** combines a categorical VAE with a GPT-like Transformer to support online actor-critic learning from imagined rollouts (Zhang et al., 2023). Its encoder and dynamics are jointly optimized in a discrete latent space. PreShort separates the visual interface from online dynamics adaptation by freezing a pretrained visual encoder and learns stochastic action-conditioned transitions in its continuous representation space through DSD. As a result, PreShort avoids the burden of jointly optimizing perception and dynamics and confines online updates to a lightweight dynamics module, yielding a stable prediction space with fewer trainable parameters.
- **Dreamer 4** is an offline video world model built around a causal tokenizer and an interactive Transformer dynamics model. Its shortcut denoising objective generates continuous representations in a few refinement steps, targeting efficient interactive rollout generation from large-scale video data (Hafner et al., 2025b). PreShort shares this emphasis on efficient continuous successor generation, but applies it in an online setting with frozen pretrained features. In DSD, differential attention first compresses latent-action histories into a compact context; reusing this context during refinement avoids repeated history processing, while Gaussian initialization permits multiple stochastic successors under the same context. This design brings few-step continuous imagination into the online regime, supporting continual world-model and policy updates without large-scale offline video pretraining or repeated history access during rollout generation.
- **DINO-WM** develops an offline world model for zero-shot planning across tasks. It fits a ViT transition model to frozen DINOv2 patch features from collected trajectories and optimizes action sequences in feature space toward a specified goal, without online model adaptation or actor-critic policy training (Zhou et al., 2025). PreShort addresses a different setting: it performs online actor-critic MBRL with a frozen Wan-VAE interface, learning stochastic action-conditioned dynamics from newly collected experience and using DSD to generate imagined rollouts for policy improvement. In this online formulation, the world model and policy co-adapt as new experience arrives, and the stochastic successors capture uncertainty over future outcomes that deterministic feature-space prediction would average out.
- **JEDI** is an online world model based on latent diffusion that jointly trains a convolutional encoder and conditional denoising dynamics. Its predictive diffusion objective shapes the continuous latent space through prediction of future states (Lim et al., 2026). PreShort takes the complementary route of fixing a pretrained Wan-VAE and adapting only the downstream dynamics. DSD then combines differential history aggregation with stochastic shortcut generation conditioned on context, enabling successor prediction in a few steps without repeatedly encoding the history. By relieving the dynamics model of representation learning and amortizing history aggregation across refinement steps, PreShort makes stochastic few-step rollouts inexpensive enough for online actor-critic training.

B Review of Pretrained Representation

We review representative pretrained visual models spanning reconstruction-oriented and representation-oriented approaches. Table 3 summarizes their learning objectives, latent forms, and visual decoding capabilities.

Table 3: A brief review of pretrained visual representation across reconstruction-oriented and representation-oriented approaches.

Reconstruction-oriented	
Model	Description
VAE	Probabilistic autoencoder for learning continuous stochastic latent representations. Encodes inputs into a latent distribution and reconstructs them with a decoder under KL regularization(Kingma & Welling, 2014).
VQ-VAE	Discrete autoencoder that quantizes encoder outputs through a learned codebook, producing discrete latent representations that are decoded for reconstruction(van den Oord et al., 2017).
Wan-VAE	Causal spatio-temporal variational autoencoder for video representation and reconstruction. It provides continuous probabilistic latents through KL-regularized encoding, together with a native causal decoder for video reconstruction(Wan, 2025).
Cosmos Tokenizer	Causal image/video tokenizer offering both continuous and discrete latent interfaces. Its continuous variant is a deterministic vanilla autoencoder without KL prior regularization, while the discrete variant uses finite scalar quantization (FSQ)(NVIDIA, 2025).
MAGVIT-v2	Visual tokenizer for image and video generation that maps inputs to highly compressed discrete latent tokens through lookup-free quantization (LFQ). The native decoder reconstructs visual content from these discrete tokens(Yu et al., 2024).
Representation-oriented	
Model	Description
DINOv3	Large-scale self-supervised vision model that learns transferable global and dense visual features. As a member of the DINO family, it builds on the self-supervised vision-transformer line established by DINO and DINOv2 (Caron et al., 2021; Oquab et al., 2024). It exposes continuous encoder representations without a native pixel-reconstruction decoder (Siméoni et al., 2026).
iBOT	Self-supervised vision transformer combining masked image modeling with online-tokenizer self-distillation. It outputs continuous patch and class features, without a native pixel-reconstruction decoder(Zhou et al., 2022).
MAE	Masked autoencoder that learns visual representations by reconstructing randomly masked image patches from the visible ones. Its asymmetric decoder serves the pretraining reconstruction objective, while downstream use typically relies on the continuous encoder features(He et al., 2022).
I-JEPA	Predictive self-supervised model that learns visual representations by predicting the embeddings of target image regions from contextual regions. It operates in representation space and does not provide a pixel-reconstruction decoder(Assran et al., 2023).
SigLIP2	Vision–language encoder that extends SigLIP(Zhai et al., 2023) with captioning, self-distillation and masked prediction objectives. It provides continuous multilingual, localized and dense visual features without a native pixel-reconstruction decoder(Tschannen et al., 2025).
R3M	Visual representation model trained on human videos with temporal contrastive learning and video–language alignment for robot manipulation. It provides transferable continuous embeddings without a native pixel-reconstruction decoder(Nair et al., 2022).
VIP	Value-implicit pretraining method that learns visual representations from offline trajectories by exploiting value-related supervision. It provides control-oriented continuous embeddings without a general-purpose pixel-reconstruction decoder(Ma et al., 2023).

C Additional Method Details

This appendix gives implementation-level details omitted from the main Method section.

C.1 Fixed Latent Coordinates

Equation 1 uses E_{pre} as shorthand for the complete frozen visual interface. In the implementation, the raw encoder features y_t are flattened and mapped to the model coordinates z_t by a fixed transformation:

$$\begin{aligned} y_t &= \mathrm{flatten}(E_{\mathrm{raw}}(o_t)), \\ z_t &= \tanh\left(\frac{y_t - \mu}{\alpha\bar{\sigma}}\right), \qquad \bar{\sigma} = \max(\sigma, \sigma_{\min}), \\ y_t &= \mu + \alpha\bar{\sigma}\,\mathrm{atanh}(z_t). \end{aligned} \tag{9}$$

The main configuration uses Wan-VAE features with $d_z = 768$. The visual features are calibrated once after replay warmup using all populated current and successor feature entries. We estimate a separate mean and population standard deviation for each coordinate, use $\alpha = 3$ and $\sigma_{\min} = 10^{-3}$, and store these statistics as fixed buffers. The transformation introduces no trainable latent adapter or stochastic posterior. The tanh transformation is evaluated in FP32 and clipped to $\pm(1 - 10^{-6})$. The same bounded transformation is applied to initialization noise and predicted endpoints. The inverse mapping is applied only before frozen decoding for visualization, with its input clipped to the same bounds. Clipping makes the mapping only approximately invertible near the bounds. An alternative linear configuration uses $z_t = (y_t - \mu)/(\alpha\bar{\sigma})$ and $y_t = \mu + \alpha\bar{\sigma} z_t$, omits tanh from initialization and the endpoint head, and uses standard Gaussian initialization.

C.2 Differential Temporal Block

Differential Temporal Context Encoding. Under partial observability, predicting the consequences of an action can require information from earlier observations. The temporal encoder must therefore aggregate past latents and actions into a context representation that retains useful temporal dependencies. We introduce differential attention into this encoder to make context aggregation more selective. Following Differential Transformer (Ye et al., 2025), the mechanism forms two softmax attention maps and subtracts them with a learnable coefficient:

$$\begin{aligned} A_j &= \mathrm{softmax}\left(\frac{Q_j K_j^\top}{\sqrt{d_k}} + \mathcal{M}\right), \quad j \in \{1, 2\}, \\ Y &= (A_1 - \lambda A_2)V. \end{aligned} \tag{10}$$

Here, Q_j and K_j are separate query and key projections, V supplies the value features, and $\mathcal{M}$ is a causal mask. The subtraction allows shared contributions in the two maps to be attenuated. Applied to tokens containing latents and actions, it provides a mechanism for selectively aggregating information relevant to successor prediction. The resulting context representation h_t summarizes the sequence up to the current action and is jointly supervised by dynamics, reward, and continuation objectives. This connects temporal context encoding directly to the predictions used during imagination. In Equation 10, each baseline head splits its Q/K projections into two branches, while the value projection supplies the concatenated value channels. The difference coefficient is parameterized as

$$\lambda_i = \exp(\langle \lambda_{q1,i}, \lambda_{k1,i} \rangle) - \exp(\langle \lambda_{q2,i}, \lambda_{k2,i} \rangle) + \lambda_{\mathrm{init}}(i). \tag{11}$$

The output is $\mathrm{RMSNorm}(A_i V_i)$ followed by the STORM-style post-LayerNorm residual block and ReLU feed-forward network. The current temporal model uses width 512, two layers, eight baseline heads, four differential heads, and interleaved RoPE. Attention scores, softmax, subtraction, value multiplication, λ_i, and RMSNorm are evaluated in FP32. KV caching stores only the newly appended latent–action pair. Token attention applies standard multi-head attention over the four conditioning tokens within each transition.

C.3 Actor-Critic

Policy learning follows the actor–critic formulation of DreamerV3 (Hafner et al., 2025a). The agent is trained on imagined trajectories generated by the world model. Each imagined rollout is initialized

from a contextual trajectory sampled from the replay buffer. The world model encodes this real context and subsequently generates latent transitions without access to future observations.

At each imagination step, the policy state concatenates the generated latent state z_t with its history representation h_{t-1}. The actor and critic are implemented as separate multilayer perceptrons:

$$\begin{aligned} &\text{State:} && s_t = [z_t, h_{t-1}], \\ &\text{Critic:} && V_\psi(s_t) \approx \mathbb{E}_{\pi_\theta,\, p_\phi}\left[\sum_{k=0}^{T-t} \gamma^k \hat{r}_{t+k}\right], \\ &\text{Actor:} && a_t \sim \pi_\theta(a_t \mid s_t). \end{aligned} \tag{12}$$

Here, p_ϕ denotes the learned world model, π_θ denotes the policy, and T is the imagination horizon. The symbols $\hat{r}_t$ and $\hat{c}_t$ denote the reward and continuation predicted by the world model, respectively. In particular, $\hat{c}_t$ estimates whether the imagined trajectory continues after step t; it is distinct from the reward.

We estimate the return using the recursively defined λ-return:

$$G_t^\lambda = \hat{r}_t + \gamma \hat{c}_t \left[(1-\lambda) V_\psi(s_{t+1}) + \lambda G_{t+1}^\lambda\right], \tag{13}$$

$$G_T^\lambda = V_\psi(s_T). \tag{14}$$

The continuation factor $\hat{c}_t$ suppresses value propagation beyond a predicted termination. During world model training, its target is $c_t = 1 - \delta_t$, where $\delta_t \in \{0, 1\}$ denotes the observed termination indicator.

To stabilize value learning, we maintain a slow critic $V_{\bar{\psi}}$ through an exponential moving average of the online critic parameters:

$$\bar{\psi} \leftarrow \sigma \bar{\psi} + (1-\sigma)\psi, \tag{15}$$

where σ is the decay coefficient. A corresponding slow λ-return, $\bar{G}_t^\lambda$, is computed by replacing V_ψ with $V_{\bar{\psi}}$ in Equations 13 and 14.

Following DreamerV3, the policy advantage is normalized using the range between the 95th and 5th percentiles of the λ-returns:

$$S = \mathrm{EMA}\left[\mathrm{percentile}(G^\lambda, 95)\right] - \mathrm{EMA}\left[\mathrm{percentile}(G^\lambda, 5)\right]. \tag{16}$$

The normalized advantage is

$$A_t = \frac{G_t^\lambda - V_\psi(s_t)}{\max(1, S)}. \tag{17}$$

The actor minimizes the negative policy-gradient objective with entropy regularization:

$$\mathcal{L}_{\mathrm{actor}}(\theta) = \mathbb{E}_{\pi_\theta,\, p_\phi}\left[-\,\mathrm{sg}(A_t) \log \pi_\theta(a_t \mid s_t) - \eta\, \mathcal{H}(\pi_\theta(\cdot \mid s_t))\right], \tag{18}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient, $\mathcal{H}(\cdot)$ is the policy entropy, and η controls the entropy regularization strength.

The critic is optimized using symlog two-hot regression against both the online and slow λ-return targets:

$$\mathcal{L}_{\mathrm{actor}}(\psi) = \frac{1}{BL} \sum_{n=1}^{B} \sum_{t=1}^{L} \left[\mathcal{L}_{\mathrm{symlog}}\left(V_\psi(s_t) - \mathrm{sg}(G_t^\lambda)\right) + \mathcal{L}_{\mathrm{symlog}}\left(V_\psi(s_t) - \mathrm{sg}\big(V_{\psi^{\mathrm{EMA}}}(s_t)\big)\right)\right]. \tag{19}$$

The complete policy-learning objective is

$$\mathcal{L}_{\mathrm{AC}} = \mathcal{L}_{\mathrm{actor}} + \mathcal{L}_{\mathrm{critic}}. \tag{20}$$

D Further Ablation Studies

D.1 Generalization across pretrained visual autoencoders

To assess the generality of our framework across pretrained visual interfaces, we replace the default Wan-VAE with DINOv3 and the continuous Cosmos Tokenizer. DINOv3 is a self-supervised visual encoder that provides transferable features without a native pixel decoder, testing whether DSD extends beyond autoencoder (AE) representations (Siméoni et al., 2026). The continuous Cosmos Tokenizer instead uses a vanilla AE interface, contrasting with the KL-regularized variational latent of Wan-VAE (NVIDIA, 2025; Wan, 2025). A review of the other pretrained visual models considered in this work is provided in Appendix B. We adjust only the input preprocessing and projection layers to accommodate differences in feature geometry and dimensionality; all subsequent components remain unchanged. Further implementation details and interface-specific adaptations are provided in the following Appendix G.1.

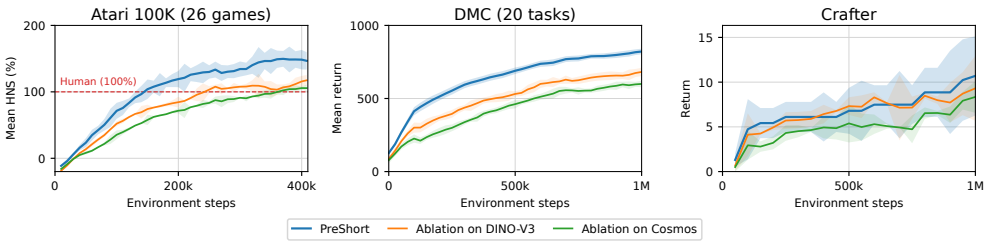

Figure 5: Control performance across frozen pretrained visual interfaces.

Figure 5 reports the control performance of the three configurations, and all three frozen interfaces support effective policy learning, confirming that the dynamics formulation does not depend on a particular pretraining objective or feature layout. Their ordering is consistent, with Wan-VAE performing best, followed by DINOv3 and the Cosmos Tokenizer. Wan-VAE's reconstruction-oriented objective preserves fine-grained visual detail, and its KL-regularized latent provides a smooth, bounded prediction space that stabilizes autoregressive imagination. DINOv3's self-supervised features capture scene semantics but discard precise localization through patch-level aggregation, while the Cosmos latent lacks variational regularization and is pretrained from the evaluated environments. The formulation is not tied to any particular visual backbone, but reconstruction-oriented and well-regularized representation spaces work best for online control.

D.2 Pretrained representations with categorical dynamics

We investigate whether the benefits of pretrained visual representations can be obtained by directly incorporating them into categorical world models. These baselines use the same frozen visual backbone as PreShort (Wan-VAE) while retaining categorical latent modeling and the associated training objectives. We adapt and evaluate both recurrent and Transformer-based categorical world models (DreamerV3 and STORM) (Hafner et al., 2025a; Zhang et al., 2023). The main modification is to replace the original VAE operating on images with one that encodes and reconstructs pretrained visual features, implemented by replacing its convolutional-based encoder and decoder with MLPs (see the following Appendix G.4 for modification details).

Despite using the same pretrained visual backbone, the categorical variants remain less competitive than PreShort. This gap suggests that strong visual representations alone are insufficient: the downstream latent model must also preserve the information they capture. Encoding these features into categorical latent variables may discard fine-grained information about small objects and local state changes (Zhang et al., 2025), even when such details are present in the pretrained representations. These distinctions can be important for predicting action outcomes and learning effective policies. By modeling transitions directly in the continuous feature space, PreShort avoids this additional categorical bottleneck, potentially retaining more of the visual information needed for control.

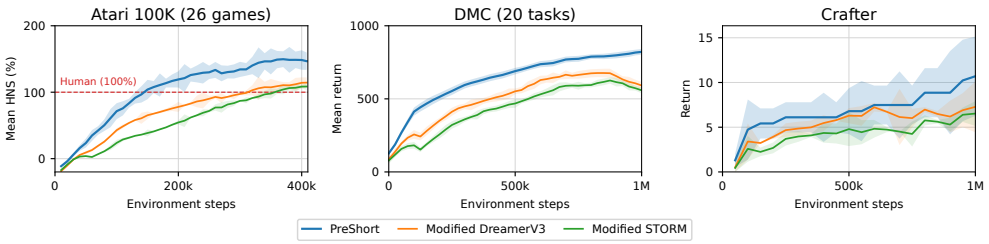

Figure 6: Control performance of categorical dynamics with pretrained features.

E Complete Benchmark Results

We evaluate PreShort on Atari 100K, the DeepMind Control Suite, and Crafter (Bellemare et al., 2013; Tassa et al., 2018; Hafner, 2022). Baseline performance is taken from the references whenever available (Hafner et al., 2025a; Zhang et al., 2023; Alonso et al., 2024; Zhang et al., 2026; Hansen et al., 2024; Burchi & Timofte, 2025; Micheli et al., 2023; 2024); otherwise, we run the official implementations under the same experimental settings as PreShort, as in the case of STORM on the DeepMind Control Suite. For the computational-cost comparison in Section 4.2, we run the official implementations on our hardware.

E.1 Atari 100K Evaluation

Table 4: Game scores and overall human-normalized performance on the 26 games in the Atari 100K benchmark.

Game	Random	Human	DreamerV3	STORM	DIAMOND	OC-STORM	PreShort
Alien	228	7128	**1118**	984	744	1101	**1171**
Amidar	6	1720	97	205	**226**	163	175
Assault	222	742	683	801	**1526**	1270	807
Asterix	210	8503	1062	1028	**3698**	1754	1262
Bank Heist	14	753	398	641	20	**1075**	**1023**
Battle Zone	2360	37188	**20300**	13540	4702	4590	12220
Boxing	0	12	82	80	87	**92**	**92**
Breakout	2	30	10	16	**132**	53	30
Chopper Command	811	7388	**2222**	1888	1370	2090	956
Crazy Climber	10780	35829	86225	66776	**99168**	84111	91036
Demon Attack	152	1971	**577**	165	288	411	288
Freeway	0	30	0	0	**33**	0	21
Frostbite	65	4335	**3377**	1316	274	260	1208
Gopher	258	2413	2160	8240	5898	4457	**10270**
Hero	1027	30826	**13354**	11044	5622	6441	9119
James Bond	29	303	540	509	427	347	**668**
Kangaroo	52	3035	2643	4208	5382	4218	**6656**
Krull	1598	2666	8171	8413	8610	**9715**	8473
Kung Fu Master	256	22736	**25900**	**26182**	18714	**24988**	22716
Ms Pacman	307	6952	1521	**2673**	1958	2401	**2781**
Pong	-21	15	-4	11	**20**	**20.6**	**21**
Private Eye	25	69571	3238	**7781**	114	85	3112
Qbert	164	13455	2921	**4522**	**4499**	**4546**	3778
Road Runner	12	7845	19230	17564	**20673**	**20482**	19480
Seaquest	68	42055	**962**	525	551	712	470
Up N Down	533	11693	**46910**	7985	3856	6623	9771
Human Mean	0%	100%	125%	122.3%	146%	134.8%	**146.7%**
Human Median	0%	100%	49%	42.8%	37%	43.8%	**76%**

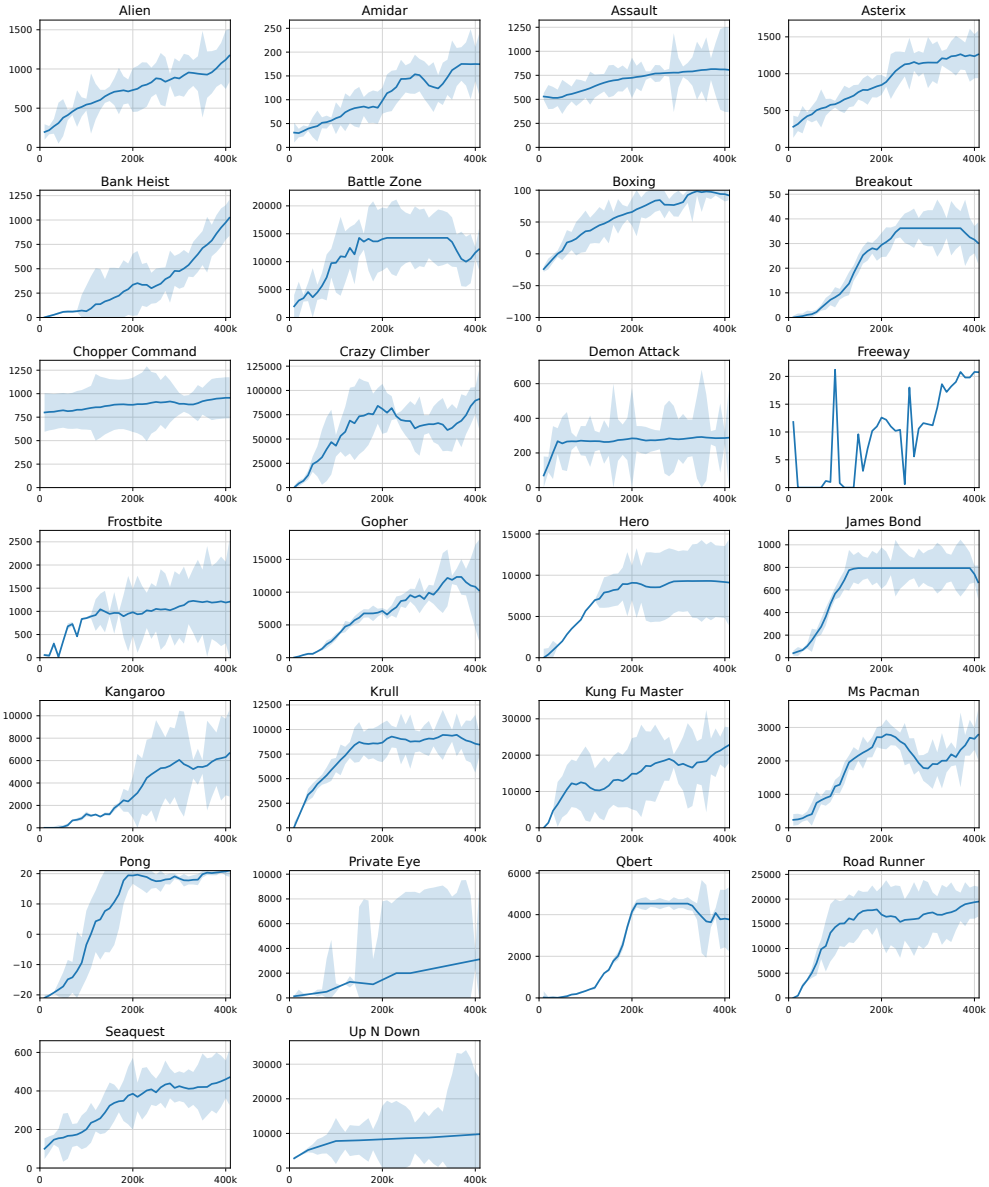

Figure 7: Game-score training curves for the 26 games in the Atari 100K benchmark.

E.2 DeepMind Control Suite Evaluation

Table 5: Task returns on the DeepMind Control Suite.

Task	DreamerV3	STORM	TD-MPC2	TWISTER	PreShort
Environment steps	1M	1M	1M	1M	1M
Acrobot Swingup	229	120	216	**239**	**244**
Ball In Cup Catch	**972**	**949**	717	**967**	**990**
Cartpole Balance	**993**	**999**	931	**998**	**1000**
Cartpole Balance Sparse	**964**	932	**1000**	**1000**	**1000**
Cartpole Swingup	**861**	661	808	819	**875**
Cartpole Swingup Sparse	759	767	739	735	**866**
Cheetah Run	**836**	674	550	694	**860**
Finger Spin	589	**960**	**986**	**976**	637
Finger Turn Easy	878	800	789	**924**	**929**
Finger Turn Hard	**904**	589	872	**910**	**943**
Hopper Hop	227	**326**	211	**314**	231
Hopper Stand	903	829	803	**932**	**971**
Pendulum Swingup	744	588	743	**832**	747
Quadruped Run	617	**692**	362	652	**682**
Quadruped Walk	811	818	253	**905**	**877**
Reacher Easy	**951**	909	**971**	**933**	**971**
Reacher Hard	**862**	498	**877**	566	**879**
Walker Run	684	**802**	728	711	750
Walker Stand	**976**	**961**	916	**977**	**988**
Walker Walk	**961**	872	**945**	**951**	**989**
Task mean	786	737	721	802	**821**
Task median	861	801	796	**908**	878

E.3 Crafter @ 1M Evaluation

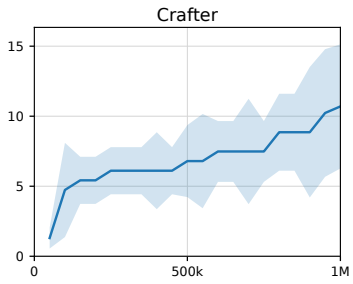

Figure 9: Crafter training curve over 1M agent–environment interaction steps.

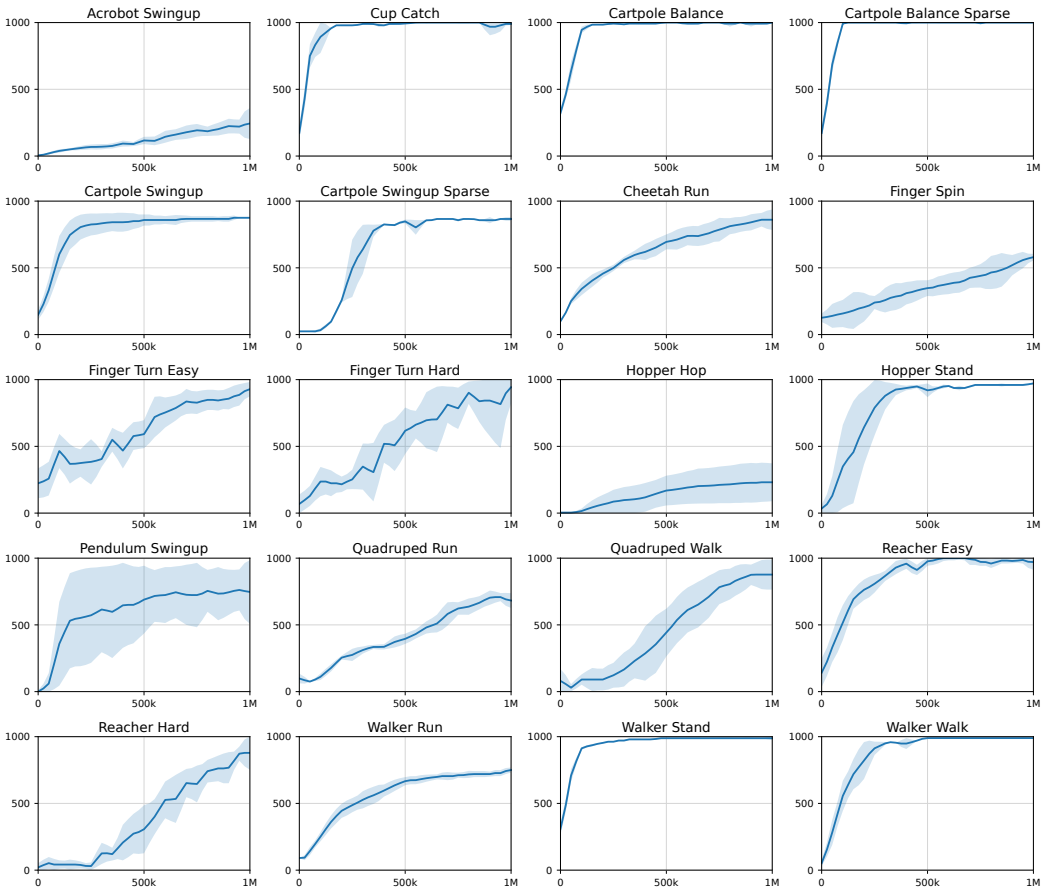

Figure 8: Task-return training curves on the DeepMind Control Suite over 1M environment steps.

F EXPERIMENTAL DETAILS

F.1 MODEL ARCHITECTURE

The following tables summarize the architecture used in the main experiments. The frozen Wan2.2 VAE encodes each observation independently, while all downstream components are trained from online interaction.

Table 6: Frozen visual interface and action mixer.

Submodule	Output shape
Observation o_t	$3 \times 64 \times 64$
Frozen Wan2.2 VAE	$48 \times 4 \times 4$
Flatten	768
Visual projection	512
One-hot action a_t	A
Action projection	512
Concatenation	1024
Fusion stem	512
Temporal token	512

Table 7: Differential temporal Transformer. The model consists of two causal blocks. Each block uses four differential heads with two 64-dimensional query/key branches and one shared 128-dimensional value branch. Interleaved RoPE is applied to queries and keys, and projected keys and values are cached during rollout.

Submodule	Module alias	Output shape
Input temporal tokens $(e_{1:T})$		$T \times 512$
LayerNorm		$T \times 512$
Differential self-attention		$T \times 512$
Linear + Dropout	Differential Attention	$T \times 512$
Residual + LayerNorm		$T \times 512$
Linear + ReLU		$T \times 1024$
Linear + Dropout	Feed Forward	$T \times 512$
Residual + LayerNorm		$T \times 512$
Output history $(h_{1:T})$		$T \times 512$
General		**Value**
Model width		512
Number of blocks		2
Maximum sequence length		64

Table 8: Token attention-based shortcut predictor. The four inputs are embedded independently with SiLU MLPs and mixed by one standard eight-head Transformer block. The feed-forward network uses ReLU, and the endpoint head uses SiLU.

Submodule	Module alias	Output shape
History state h_t	Input embeddings	512
Noisy latent $x_{t+1,\tau}$		512
Path position τ		512
Step size d		512
Token sequence		4×512
LayerNorm		4×512
Multi-head self-attention		4×512
Linear + Dropout	Token Attention	4×512
Residual + LayerNorm		4×512
Linear + ReLU		4×1024
Linear + Dropout	Feed Forward	4×512
Residual + LayerNorm		4×512
Endpoint head		768
General		**Value**
Embedding width		512
Number of blocks		1
Number of attention heads		8
Token count		4

Table 9: Prediction and policy networks. The reward and continuation predictors use SiLU after their hidden layer. Hidden layers in the actor and critics use LayerNorm and ReLU. The slow critic shares the critic architecture and is updated by exponential moving average.

Network	**Hidden layers**	**Input**	**Hidden dimension**	**Output dimension**	**Output distribution**
Reward predictor	1	h_t	512	255	Symlog two-hot
Continuation predictor	1	h_t	512	1	Bernoulli
Actor	2	$s_t = [z_t, h_{t-1}]$	512	A	Categorical
Critic	2	s_t	512	255	Symlog two-hot
Slow critic	2	s_t	512	255	Symlog two-hot

F.2 HYPERPARAMETERS

Table 10 lists the shared model and optimization settings used in the main experiments.

Table 10: Shared hyperparameters used in the main experiments.

Hyperparameter	Symbol	value
Temporal Transformer layers	–	2
Temporal feature dimension	d_h	512
Differential attention heads	–	4
Dropout probability	p	0.1
Token attention blocks	–	1
Token embedding dimension	–	512
Token attention heads	–	8
Shortcut grid size	–	64
Generation steps	K	4
World-model batch size	B	16
World-model sequence length	T	64
Imagination batch size	–	1024
Imagination context length	C	8
Imagination horizon	H	16
World-model updates per step	–	1
Actor–critic updates per step	–	1
Discount factor	γ	0.985
Return parameter	λ	0.95
Entropy coefficient	η	3×10^{-4}
Return-normalization EMA decay	–	0.99
Slow-critic EMA decay	–	0.98
Optimizer	–	Adam
World-model learning rate	–	1×10^{-4}
World-model gradient clipping	–	100
Actor–critic learning rate	–	3×10^{-5}
Actor–critic Adam epsilon	ϵ	1×10^{-5}
Actor–critic gradient clipping	–	100

G Details of Ablation Structures

G.1 Pretrained Visual Interfaces

We compare three frozen visual interfaces to assess whether PreShort accommodates representations with different pretraining objectives and output geometries. `PreShort-Wan` uses the default KL-regularized variational interface of Wan-VAE, `PreShort-DINO` uses DINOv3 ViT-S/16 features, and `PreShort-Cosmos` uses the continuous CI8×8 Cosmos Tokenizer (Wan, 2025; Siméoni et al., 2026; NVIDIA, 2025). Each configuration preserves the native continuous feature space of its visual encoder. Interface-specific input mappings connect these features to the shared temporal and shortcut architecture.

Shared model interface. Let $v \in \{\mathrm{Wan}, \mathrm{DINO}, \mathrm{Cosmos}\}$ index the visual interface, and let D_v denote its flattened feature dimension. The frozen encoder produces

$$z_t^{(v)} = \mathrm{sg}\Big(\mathrm{flatten}\Big[A_v\Big(E_{\mathrm{pre}}^{(v)}(\rho_v(o_t))\Big)\Big]\Big) \in \mathbb{R}^{D_v}, \tag{21}$$

where o_t is the observation, ρ_v is the encoder-specific preprocessing, $E_{\mathrm{pre}}^{(v)}$ is the frozen visual encoder, A_v is the interface-specific spatial aggregation, and $\mathrm{sg}(\cdot)$ denotes stop gradient. The visual latent and action are embedded separately before fusion:

$$\begin{aligned} q_t^{z,(v)} &= W_z^{(v)} z_t^{(v)} \in \mathbb{R}^{512}, & q_t^a &= W_a a_t \in \mathbb{R}^{512}, \\ e_t^{(v)} &= s_\phi\Big([q_t^{z,(v)}; q_t^a]\Big) \in \mathbb{R}^{512}, & h_{1:T}^{(v)} &= f_\phi\Big(e_{1:T}^{(v)}\Big). \end{aligned} \tag{22}$$

Here, $W_z^{(v)}$ maps the native visual dimension D_v to the temporal width, $a_t \in \mathbb{R}^{d_a}$ is the action vector, and W_a maps it to the same width. The fusion network s_ϕ maps their 1024 dimensional concatenation to the 512 dimensional token $e_t^{(v)}$, from which the differential temporal encoder produces $h_t^{(v)}$. Within the shortcut predictor, the noisy latent is embedded from D_v to the shared 512 dimensional internal width, while the endpoint head returns the predicted successor to $\mathbb{R}^{D_v}$. This preserves each encoder's feature coordinates throughout latent prediction.

DINOv3 interface. `PreShort-DINO` uses a frozen DINOv3 ViT-S/16 encoder, whose patch size is 16and whose patch embedding width is 384 (Siméoni et al., 2026). The 64×64 environment observation is transformed by ρ_{DINO} to the encoder input format. We retain the spatial patch features and exclude the class and register tokens. The resulting feature map is spatially aggregated and flattened to obtain $z_t^{(\mathrm{DINO})} \in \mathbb{R}^{D_{\mathrm{DINO}}}$. Its visual input mapping $W_z^{(\mathrm{DINO})}$ and the outer layers of the shortcut predictor are dimensioned according to D_{DINO}. DINOv3 provides no native pixel reconstruction decoder, and this configuration introduces no decoder or reconstruction objective.

Cosmos interface. `PreShort-Cosmos` uses the frozen continuous CI8×8 Cosmos Tokenizer. Its vanilla autoencoder provides a 16 channel continuous latent with an 8×8 spatial compression ratio (NVIDIA, 2025). A 64×64 observation produces a feature map of shape $16 \times 8 \times 8$, giving $D_{\mathrm{Cosmos}} = 1024$ after flattening. The visual input mapping uses $W_z^{(\mathrm{Cosmos})} \in \mathbb{R}^{512 \times 1024}$, and the shortcut predictor operates on 1024 dimensional inputs and targets through its 512 dimensional internal representation. The pretrained Cosmos decoder is retained only for analyses that require pixel visualization and receives no online updates.

Interface details. Table 11 summarizes the visual interfaces and the dimension-dependent mappings used by PreShort.

G.2 Differential Attention

We construct `No DiffAttn` to isolate the contribution of differential temporal aggregation to control performance. This variant replaces differential attention in the temporal sequence model with standard causal self-attention under a matched parameter count. The frozen visual interface, latent and action token construction, shortcut dynamics, reward and continuation predictors, world model objectives, and policy learning procedure remain unchanged. The comparison therefore concerns how the observed history is summarized, rather than how successor latents are generated.

Table 11: Visual interfaces used in the pretrained-model ablation. All visual encoders are frozen. Each native feature is flattened to D_v, projected to the shared temporal width of 512, and used as the input and prediction target of the shortcut dynamics. The downstream world-model and policy learning components are identical across configurations.

Configuration	PreShort-Wan	PreShort-DINO	PreShort-Cosmos
Visual interface			
Visual model	Wan-VAE	DINOv3 ViT-S/16	Cosmos Tokenizer CI8×8
Interface type	VAE	Representation encoder	Vanilla AE
Native pixel decoder	Yes	No	Yes
Native representation			
Feature organization	Spatial map	Patch map	Spatial map
Feature channels	48	384	16
Spatial scale	4×4	4×4	8×8
Native feature shape	$48 \times 4 \times 4$	$384 \times 4 \times 4$	$16 \times 8 \times 8$
Flattened dimension D_v	768	6144	1024

Standard self-attention temporal block. Both variants retain the temporal interface used by PreShort:

$$\begin{aligned} e_t &= m_\phi(z_t, a_t), \\ h_{1:T} &= f_\phi(e_{1:T}), \end{aligned} \tag{23}$$

where z_t is the continuous visual latent, a_t is the corresponding action, e_t is their fused temporal token, and h_t summarizes the sequence through the current action. In `No DiffAttn`, each temporal attention layer computes

$$\begin{aligned} A &= \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + M\right), & Y_{\mathrm{NoDiff}} &= AV, \\ A_j &= \mathrm{softmax}\left(\frac{Q_j K_j^\top}{\sqrt{d_k}} + M\right), \quad j \in \{1,2\}, & Y_{\mathrm{PreShort}} &= (A_1 - \lambda A_2)V. \end{aligned} \tag{24}$$

Here, Q, K, and V are the query, key, and value projections, d_k is the key dimension, and M is the causal mask that excludes future tokens. PreShort forms two attention maps from separate query and key branches and combines them using the learned differential coefficient λ (Ye et al., 2025). By contrast, `No DiffAttn` uses one attention map and contains neither the second query–key branch nor λ.

Controlled comparison. The temporal models retain the same depth, width, input and output interfaces, positional encoding, and causal sequence structure. The two variants also use the same shortcut predictor, including its token attention within each transition, and are optimized with the same dynamics, shortcut consistency, reward, and continuation objectives defined in the main text. We further hold fixed the interaction budget, replay data, policy optimization, and evaluation protocol. Projection sizes are adjusted to match the trainable parameter count, so this ablation does not attribute differences to additional parameters; it does not assume identical computational cost.

Architecture details. Table 12 reports the temporal-block configuration used in this comparison.

G.3 Shortcut Dynamics

We construct `No Shortcut` to isolate the contribution of the shortcut transition formulation. This variant replaces the stochastic shortcut predictor with a deterministic multilayer perceptron (MLP) under a matched transition parameter count. The frozen visual interface, latent and action token construction, differential temporal encoder, reward and continuation predictors, and policy learning procedure remain unchanged. The comparison therefore evaluates the joint effect of stochastic initialization, conditional refinement, and shortcut consistency, rather than attributing any difference to stochasticity alone.

Table 12: Temporal sequence-model configurations for the differential-attention ablation. PreShort uses differential attention with two query–key branches and a learnable coefficient λ, whereas No DiffAttn uses standard causal self-attention. Both variants use interleaved RoPE, two temporal layers, a model width of 512, a feed-forward width of 1024, post-LayerNorm residual blocks, dropout probability 0.1, and KV caching.

Configuration	PreShort	No DiffAttn
Attention		
Attention maps per head	2	1
Attention heads	4	8
Query/key width per head	64	64
Value width per head	128	64
Temporal block		
Number of layers	2	2
Model width	512	512
Feed-forward width	1024	1024

Deterministic MLP dynamics. Both variants retain the temporal interface of PreShort:

$$\begin{aligned} e_t &= m_\phi(z_t, a_t), \\ h_{1:T} &= f_\phi(e_{1:T}), \end{aligned} \tag{25}$$

where $z_t \in \mathbb{R}^{768}$ is the continuous visual latent, a_t is the corresponding action, e_t is their fused temporal token, and $h_t \in \mathbb{R}^{512}$ summarizes the latent and action history through the current action. PreShort generates a successor from Gaussian noise through conditional shortcut refinement (Frans et al., 2025). In `No Shortcut`, the same context is instead mapped directly to the next latent:

$$\hat{z}_{t+1}^{\text{mlp}} = g_\phi^{\text{mlp}}(h_t). \tag{26}$$

The MLP produces a 768 dimensional successor in one forward pass. It receives neither Gaussian noise nor the shortcut variables $x_{t+1,\tau}$, τ, and d, and therefore requires no token attention or iterative refinement. The differential temporal encoder remains unchanged.

Training objective. Because `No Shortcut` does not construct coarse and half step predictions, the shortcut consistency objective $\mathcal{L}_{\text{sc}}$ is removed. The deterministic transition is trained against the observed successor latent using

$$\mathcal{L}_{\text{dyn}}^{\text{mlp}} = \frac{1}{BT} \sum_{b=1}^{B} \sum_{t=1}^{T} \left\| \hat{z}_{b,t+1}^{\text{mlp}} - \text{sg}(z_{b,t+1}) \right\|_2^2, \tag{27}$$

where B is the batch size, T is the number of transitions per sequence, and $\text{sg}(\cdot)$ denotes stop gradient. The reward and continuation objectives are unchanged, giving

$$\mathcal{L}_{\text{world}}^{\text{NoSC}} = \mathcal{L}_{\text{dyn}}^{\text{mlp}} + \mathcal{L}_{\text{rew}} + \mathcal{L}_{\text{con}}. \tag{28}$$

Architecture details. Table 13 summarizes the transition structures used in this comparison.

G.4 Modifications to Categorical World Models

We adapt DreamerV3 and STORM to use the same frozen Wan-VAE features as PreShort while retaining their categorical latent representations and original world-model objectives (Hafner et al., 2025a; Zhang et al., 2023; Wan, 2025). In both variants, the convolutional image encoder and pixel decoder are replaced with MLPs that encode and reconstruct pretrained visual features. Their main architectural difference remains unchanged: DreamerV3 models categorical dynamics recurrently, whereas STORM uses a causal Transformer.

Table 13: Transition-predictor configurations for the shortcut-dynamics ablation. Both variants use the same temporal model, a 512-dimensional history state, a 768-dimensional latent output, and matched transition-model parameter counts. PreShort uses a four-token shortcut predictor with one eight-head attention block, whereas No Shortcut uses a deterministic MLP.

Configuration	**PreShort**	**No Shortcut**
Predictor type	Shortcut predictor	Deterministic MLP
Stochastic initialization	Gaussian	None
Token attention	4 tokens	None
Refinement steps	4	1
Shortcut consistency	Yes	No
Output dimension	768	768

Model structure. Given an image observation o_t, the frozen Wan-VAE encoder produces a 768-dimensional continuous visual feature

$$z_t = \mathrm{sg}\left(\mathrm{flatten}\left(E_{\mathrm{Wan}}(o_t)\right)\right) \in \mathbb{R}^{768}, \tag{29}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient. A trainable MLP then maps z_t to the feature u_t used by the categorical posterior:

$$u_t = f_\phi^{\mathrm{enc}}(z_t). \tag{30}$$

The mapping itself remains continuous. The posterior encoder converts u_t, together with the temporal context required by the corresponding model, into a categorical distribution:

$$X_t \sim q_\phi\left(X_t \mid u_t, \chi_t\right), \tag{31}$$

where z_t denotes the frozen continuous feature, u_t its trainable MLP projection, and X_t the final discrete latent used by the world model. A trainable MLP decoder reconstructs the frozen feature z_t from the categorical model state. Thus, the modified models reconstruct pretrained features rather than pixels.

To distinguish quantities belonging to the two adaptations, superscript D denotes the DreamerV3 variant and superscript S denotes the STORM variant. For example, X_t^{D} is the categorical latent of DreamerV3, whereas X_t^{S} is the categorical latent of STORM. These superscripts are model labels rather than exponents.

DreamerV3 dynamics. The DreamerV3 variant retains its recurrent state-space dynamics. Its recurrent state h_t^{D} summarizes the preceding categorical latent and action. The posterior additionally receives the current mapped feature u_t^{D}, whereas the prior predicts the categorical latent from the recurrent state alone:

$$\begin{aligned}
\text{Sequence model:} \quad & h_t^{\mathrm{D}} = f_\phi^{\mathrm{D}}\left(h_{t-1}^{\mathrm{D}}, X_{t-1}^{\mathrm{D}}, a_{t-1}\right), \\
\text{Categorical posterior:} \quad & X_t^{\mathrm{D}} \sim q_\phi^{\mathrm{D}}\left(X_t^{\mathrm{D}} \mid h_t^{\mathrm{D}}, u_t^{\mathrm{D}}\right), \\
\text{Categorical prior:} \quad & \widehat{X}_t^{\mathrm{D}} \sim p_\phi^{\mathrm{D}}\left(\widehat{X}_t^{\mathrm{D}} \mid h_t^{\mathrm{D}}\right), \\
\text{Feature decoder:} \quad & \hat{z}_t^{\mathrm{D}} = d_\phi^{\mathrm{D}}\left(h_t^{\mathrm{D}}, X_t^{\mathrm{D}}\right), \\
\text{Reward predictor:} \quad & \hat{r}_t^{\mathrm{D}} = g_\phi^{R,\mathrm{D}}\left(h_t^{\mathrm{D}}, X_t^{\mathrm{D}}\right), \\
\text{Continuation predictor:} \quad & \hat{c}_t^{\mathrm{D}} = g_\phi^{C,\mathrm{D}}\left(h_t^{\mathrm{D}}, X_t^{\mathrm{D}}\right).
\end{aligned} \tag{32}$$

The recurrent dynamics and categorical prior and posterior are retained from DreamerV3; only the original image encoder and pixel decoder are replaced by the feature-space MLPs.

STORM dynamics. The STORM variant retains its causal Transformer dynamics. The posterior maps the current feature u_t^{S} to the categorical latent X_t^{S}. This latent is combined with the corresponding action to form the sequence-model input, from which the next categorical distribution,

reward, and continuation are predicted:

$$\begin{aligned}
\text{Categorical posterior:} \quad & X_t^{\mathrm{S}} \sim q_\phi^{\mathrm{S}}\left(X_t^{\mathrm{S}} \mid u_t^{\mathrm{S}}\right), \\
\text{Action mixer:} \quad & e_t^{\mathrm{S}} = m_\phi^{\mathrm{S}}\left(X_t^{\mathrm{S}}, a_t\right), \\
\text{Sequence model:} \quad & h_{1:T}^{\mathrm{S}} = f_\phi^{\mathrm{S}}\left(e_{1:T}^{\mathrm{S}}\right), \\
\text{Categorical prior:} \quad & \widehat{X}_{t+1}^{\mathrm{S}} \sim p_\phi^{\mathrm{S}}\left(\widehat{X}_{t+1}^{\mathrm{S}} \mid h_t^{\mathrm{S}}\right), \\
\text{Feature decoder:} \quad & \hat{z}_t^{\mathrm{S}} = d_\phi^{\mathrm{S}}\left(X_t^{\mathrm{S}}\right), \\
\text{Reward predictor:} \quad & \hat{r}_t^{\mathrm{S}} = g_\phi^{R,\mathrm{S}}\left(h_t^{\mathrm{S}}\right), \\
\text{Continuation predictor:} \quad & \hat{c}_t^{\mathrm{S}} = g_\phi^{C,\mathrm{S}}\left(h_t^{\mathrm{S}}\right).
\end{aligned} \tag{33}$$

The causal Transformer and categorical transition remain unchanged from STORM. The modification is confined to replacing its image-based observation interface and pixel reconstruction target with the frozen-feature interface.

Loss functions. Both categorical variants retain their original reward, continuation, dynamics, and representation objectives. The original pixel reconstruction term is replaced by mean-squared reconstruction of the frozen Wan-VAE feature. For $A \in \{\mathrm{D}, \mathrm{S}\}$, the total world-model loss is

$$\mathcal{L}(\phi_A) = \frac{1}{BT} \sum_{n=1}^{B} \sum_{t=1}^{T} \left[\mathcal{L}_{n,t}^{\mathrm{feat},A} + \mathcal{L}_{n,t}^{\mathrm{rew},A} + \mathcal{L}_{n,t}^{\mathrm{con},A} + \beta_{\mathrm{dyn}}^{A} \mathcal{L}_{n,t}^{\mathrm{dyn},A} + \beta_{\mathrm{rep}}^{A} \mathcal{L}_{n,t}^{\mathrm{rep},A} \right], \tag{34}$$

where B and T denote the batch size and sequence length, respectively. The prediction losses are

$$\mathcal{L}_t^{\mathrm{feat},A} = \left\| \hat{z}_t^A - z_t \right\|_2^2, \tag{35}$$

$$\mathcal{L}_t^{\mathrm{rew},A} = \mathcal{L}^{\mathrm{sym}}\left(\hat{r}_t^A, r_t\right), \tag{36}$$

$$\mathcal{L}_t^{\mathrm{con},A} = -c_t \log \hat{c}_t^A - (1 - c_t) \log\left(1 - \hat{c}_t^A\right), \tag{37}$$

where $\mathcal{L}^{\mathrm{sym}}$ denotes the symlog two-hot reward loss and c_t is the continuation target.

Let q_t^A and p_t^A denote the matched categorical posterior and transition prior defined by the corresponding dynamics above. The dynamics and representation losses are

$$\mathcal{L}_t^{\mathrm{dyn},A} = \max\left(b_{\mathrm{free}}^A, D_{\mathrm{KL}}\left[\mathrm{sg}\left(q_t^A\right) \,\middle\|\, p_t^A\right]\right), \tag{38}$$

$$\mathcal{L}_t^{\mathrm{rep},A} = \max\left(b_{\mathrm{free}}^A, D_{\mathrm{KL}}\left[q_t^A \,\middle\|\, \mathrm{sg}\left(p_t^A\right)\right]\right), \tag{39}$$

where $b_{\mathrm{free}}^A = 1$ denotes the free-bits threshold of the corresponding baseline. The dynamics loss trains the transition prior toward the observation-conditioned posterior, whereas the representation loss updates the posterior toward the detached prior.

Apart from replacing pixel reconstruction with frozen-feature reconstruction, both adaptations preserve their original categorical dynamics and policy-learning procedures. They use the same frozen Wan-VAE encoder and evaluation setting as PreShort, allowing the comparison to isolate the effect of modeling pretrained visual features through categorical rather than continuous latent dynamics.

H LIMITATIONS AND FUTURE WORK

We identify two limitations of this work, each suggesting a direction for future improvement.

Task adaptation of frozen representations. Although RAE (Zheng et al., 2026) and our experiments indicate that pretrained representations can preserve fine-grained visual details and support control, freezing the visual encoder prevents the representation space from adapting to task-specific control requirements, which may limit further improvements in control performance. Further task-aware adaptation may therefore improve control performance.

Limited real-world validation. Our evaluation is confined to simulated visual-control benchmarks and does not yet include an explicit real-world application. Future work should evaluate PreShort in physical environments to assess its transfer beyond simulation.