STARWM: Self-Supervised Trained Attention Routing for Robust World Models

Zeqiang Zhang[1], **Fabian Wurzberger**[1], **Maximilian Otte**[1], **Daniel Schmid**[1], **Sebastian Gottwald**[1], **Arne Peter Raulf**[2], **Daniel Alexander Braun**[1]

[1]Institute of Neural Information Processing, Ulm University
[2]German Aerospace Center (DLR)
{zeqiang.zhang, daniel.braun}@uni-ulm.de

Abstract

A robust world model must strike the balance between faithfully capturing environmental dynamics and abstracting away from irrelevant content. While reconstruction-based world models ensure faithful supervision, they misallocate representational capacity by pixel area rather than dynamics relevance for visual tasks, which can cause task-irrelevant content to dominate the learned representation. Alternatively, reconstruction-free methods avoid this bias but risk discarding possibly relevant information. We propose STARWM, which uses a cross-attention module trained on self-supervised dynamics to decide where reconstruction applies. A dual-stream decoder then restricts reconstruction to the attended regions, with stop-gradient barriers preventing interference between the two objectives. These components allows reconstruction to supervise the visual content of attended regions without contaminating the latent with non-predictive information. On DeepMind Control with dynamic video backgrounds, default (reward-free) STARWM achieves the strongest performance under random-frame distractors and substantially outperforms reconstruction-based baselines under sequential video. In addition, its reward-augmented variant matches or exceeds reconstruction-free methods on sequential video, achieving the highest overall return across all distractor regimes. Mechanistic probing confirms STARWM preserves state attributes with near-perfect fidelity through long-horizon imagination while systematically discarding distractors. Code is available on GitHub[1].

1 Introduction

World models in the context of model-based reinforcement learning ideally allow simulating environmental dynamics in a compact latent space, enabling sample-efficient policy optimization [8]. Intuitively, a world model should capture environment dynamics with temporal coherence over long horizons, remain unaffected by task-irrelevant distractors, and align its representation with the overall objective, if available.

Realizing these properties depends on how the representation is learned. Reconstruction-based world models such as DreamerV3 [11] ground their representations through dense pixel reconstruction, but the gradient scales with pixel area rather than dynamical relevance, biasing the encoder toward whatever covers most of the frame (reconstruction bias) [4, 7]. Reconstruction-free methods (DreamerPro [4], R2-Dreamer [21]) replace reconstruction with auxiliary objectives such as latent consistency, contrastive prediction, or redundancy reduction. Self-supervised signals, such as inverse dynamics [25] and contrastive learning [17, 29], provide meaningful gradients on action-conditional

[1]https://github.com/zeqiang-zhang/StarWM

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

or temporally predictable regions, but capture only specific aspects of the underlying reality without pixel-level grounding [27].

In the literature there have been multiple attempts to combine the pixel-level grounding of reconstruction with the flexibility of reconstruction-free methods that avoid the reconstruction bias in favor of a priori chosen auxiliary objectives. For example, latent-level decomposition methods (TIA [7], DenoisedMDP [33], Iso-Dream [23]) allocate both reconstruction-based and -free signals to a shared latent representation. But this puts the two signals in constant gradient tension: reconstruction pushes the encoder to preserve all content, while the auxiliary objective pushes specific content into specific subspaces. The resulting allocation depends on carefully balancing these conflicting gradients.

We propose STARWM (**S**elf-supervised **T**rained **A**ttention **R**outing **W**orld **M**odel), which resolves the conflict between the two signals by architectural separation. STARWM performs the decomposition on the encoder's spatial feature map and isolates the supervision sources at the parameter level through attention routing. Our contributions are:

- **A dual-stream architecture with self-supervised attention routing** that separates the where-to-attend decision from the what-to-extract decision. A cross-attention module produces a spatial mask, supervised exclusively by auxiliary objectives, with stop-gradient barriers ensuring that the two supervision sources update disjoint parameter subsets. The dual-stream decoder uses the spatial mask to route relevant content to the latent dynamics model, while the rest is passed to a residual stream that absorbs irrelevant content.
- **NoisyShape, a diagnostic environment** for mechanistic analysis of learned representations under visual distractions, providing controllable, easily interpreted dynamics and randomized video distractors. Ground-truth labels for color, shape, size, and position enable probe analysis of both immediate encoding and long-horizon imagination.
- **Comprehensive qualitative and quantitative evaluations** of STARWM's behavior under visual distractions. With the same hyperparameters, STARWM on average outperforms all baselines across distractor setups. On NoisyShape (reward-free), attention-based routing remains effective while alternatives degrade. Probe analysis confirms that STARWM retains task-relevant attributes while discarding irrelevant ones, achieving near-perfect temporal coherence over long horizons. Cross-sample image composition directly visualizes the three components' specialization without any explicit disentanglement objective.

2 Preliminaries

Problem Setting. We consider a visual control task where the observation $o_t \in \mathbb{R}^{H\times W\times 3}$ is generated from two unobserved state components: a *relevant* state that the world model should faithfully capture, and an *irrelevant* state that should be filtered out. What counts as relevant depends on the chosen objective. Given any such objective, our goal is a general world model whose latent dynamics capture the *relevant* state while remaining uncontaminated by the *irrelevant* state.

Reconstruction-Based World Models. Reconstruction-based world models learn a latent dynamics model by training an encoder–decoder architecture to reconstruct observations while predicting forward in latent space. A representative example is DreamerV3 [11], a world model centered on the Recurrent State Space Model (RSSM):

$$\text{Sequence model:} \quad h_t = f_\theta(h_{t-1}, z_{t-1}, a_{t-1}), \tag{1}$$

$$\text{Posterior:} \quad z_t \sim q_\theta(z_t \mid h_t, o_t), \tag{2}$$

$$\text{Prior:} \quad \hat{z}_t \sim p_\theta(\hat{z}_t \mid h_t), \tag{3}$$

where h_t is a deterministic recurrent state and z_t a stochastic posterior inferred from the observation. The model is trained by maximizing an Evidence Lower Bound:

$$\mathcal{L}_{\text{WM}} = \mathbb{E}\Big[\underbrace{-\ln p_\theta(o_t \mid h_t, z_t)}_{\text{Image loss } (\mathcal{L}_{\text{img}})} \underbrace{-\ln p_\theta(r_t \mid h_t, z_t)}_{\text{Reward loss } (\mathcal{L}_{\text{reward}})} \underbrace{-\ln p_\theta(c_t \mid h_t, z_t)}_{\text{Cont. loss } (\mathcal{L}_{\text{cont}})} + \underbrace{\beta\,\mathrm{KL}\big[\, q_\theta(\cdot \mid h_t, o_t) \,\|\, p_\theta(\cdot \mid h_t) \,\big]}_{\mathcal{L}_{\text{KL}}^{\text{RSSM}}}\Big]. \tag{4}$$

To learn a policy, DreamerV3 trains an actor-critic agent on latent trajectories imagined by the RSSM. The actor maximizes the critic's estimated return over rollouts from encoded states.

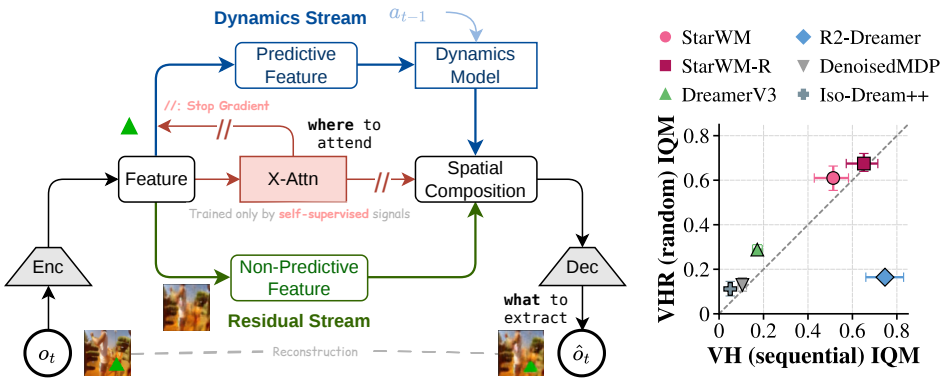


Figure 1: **StarWM disentangles *where-to-attend* (X-Attn) from *what-to-extract* (reconstruction $\hat{o}_t$). *Left:*** Self-supervised objective signals train a cross-attention module to localize predictable, action-conditional content, while the content of the attended regions is shaped by a reconstruction objective through a dual-stream decoder. Stop-gradient barriers (//) keep the two objectives from interfering. For detailed architecture diagram see Appendix B.1. ***Right:*** Interquartile mean (IQM) aggregated across 6 DMC tasks under sequential (VH) and random (VHR) video distractors, normalized by DreamerV3 clean performance (recovery ratio). StarWM and its variant are the only methods robust to both distractor types.

3 Method: StarWM

StarWM addresses the reconstruction bias with a minimal modification to existing reconstruction-based world models: only the encoder and decoder are changed, while the latent dynamics model and policy learning procedure are inherited unchanged. The modification realizes two design principles, illustrated in Figure 1: **(1) Dual-stream routing.** A cross-attention module splits the encoder's spatial features into a dynamics stream (entering the latent dynamics model) and a residual stream (auxiliary reconstruction only, discarded during imagination), preventing the dynamics latent from encoding objective-irrelevant content. **(2) Self-supervised location/content separation.** Self-supervised signals train the attention to decide *where* to route, while reconstruction trains the decoder to decide *what* to extract. Stop-gradient barriers keep the two objectives from interfering.

Though we instantiate them on DreamerV3, these principles are independent of the specific dynamics backbone. Specifically, instead of applying the RSSM to the raw observations o_t, our approach applies it to the attention gated features e_t. The following subsections detail each component.

3.1 Cross-Attention Spatial Routing

StarWM intercepts the encoder at an intermediate feature map of shape $C \times H \times W$ (see Appendix B.1), flattened into $L = HW$ spatial tokens $F_t \in \mathbb{R}^{L \times C}$. As attention is permutation-invariant, we form keys by adding 2D sinusoidal positional encodings $E_{\text{pos}} \in \mathbb{R}^{L \times C}$ [3] to enable location-aware attention, and values by appending normalized spatial coordinates $\mathrm{E}_{\text{coord}} \in \mathbb{R}^{L \times 2}$ [19, 18] to preserve explicit positional information for the dynamics model. Specifically, we set $K_t = F_t + E_{\text{pos}}$ and $V_t = [F_t;\ E_{\text{coord}}]$.

To track distinct entities or articulated body parts independently, we employ N learnable query vectors $Q \in \mathbb{R}^{N \times C}$. The queries attend over these features via scaled dot-product attention:

$$A_t = \text{Softmax}\left(\frac{(QW_Q)(K_t W_K)^\top}{\sqrt{C}}\right) \in \mathbb{R}^{N \times L}, \tag{5}$$

where $W_Q, W_K \in \mathbb{R}^{C \times C}$ are learnable projections. Each row of A_t is a soft spatial probability mask localizing one entity. We use a multi-head implementation in practice (Appendix B.1).

Crucially, A_t routes the visual features into two complementary streams: a *dynamics stream* of attended entities passed to the RSSM, and a *residual stream* (handled by an auxiliary VAE) that aids

reconstruction but is discarded during imagination. The following sections detail how this attention is trained (§3.2) and decoded (§3.3).

3.2 Self-Supervised Location/Content Separation

We train the cross-attention module using self-supervised signals that each instantiate a notion of relevance grounded in environment dynamics. To compute these losses, we extract coordinate-free attention-pooled features $e_t^{\text{attn}} = \text{flatten}(A_t F_t)$. Omitting spatial coordinates forces it to localize entities based purely on visual dynamics rather than absolute positions.

Inverse dynamics. A learnable inverse dynamics mapping [25] ϕ_{inv} predicts the action a_t from consecutive attention-pooled features, trained by optimizing

$$\mathcal{L}_{\text{inv}} = \ell\big(\phi_{\text{inv}}(e_t^{\text{attn}},\, e_{t+1}^{\text{attn}}),\, a_t\big), \tag{6}$$

where ℓ denotes cross-entropy or squared error depending on the action space. This objective guides attention to regions whose visual changes are caused by the agent's actions, while action-independent content produces no useful gradient.

Temporal contrastive learning. A learnable projection mapping ϕ_{ctr} is trained by temporal contrastive learning [22, 17], pulling consecutive features together and pushing apart random negatives, resulting in the loss

$$\mathcal{L}_{\text{ctr}} = -\log\sigma\Big(\tfrac{\phi_{\text{ctr}}(e_t^{\text{attn}})^\top \phi_{\text{ctr}}(e_{t+1}^{\text{attn}})}{\tau}\Big) - \log\sigma\Big(-\tfrac{\phi_{\text{ctr}}(e_t^{\text{attn}})^\top \phi_{\text{ctr}}(\tilde{e}^{\text{attn}})}{\tau}\Big), \tag{7}$$

where τ a temperature, and $\tilde{e}^{\text{attn}}$ a random negative. This guides attention toward temporally predictable content.

Together, these signals route attention toward controllable and predictable entities, leaving the rest as irrelevant. While they serve as our default, the cross-attention pathway is agnostic to the choice of signal; for instance, adding reward prediction yields STARWM-R, which routes by task-relevance and better tracks passively moving task-relevant entities (see Appendix E.1).

Gradient isolation. All downstream components access a stop-gradiented copy of the attention map $\bar{A}_t = \text{sg}(A_t)$, numerically identical to A_t but carrying no gradient. The entity tokens $e_t^{\text{dyn}} \in \mathbb{R}^{N\times(C+2)}$ entering the RSSM are computed as $e_t^{\text{dyn}} = \bar{A}_t V_t$, as the model's estimate of *relevant* state. Additionally, the dual-stream decoder (§3.3) uses $\bar{A}_t$ as the spatial routing mask. The attention parameters $\{Q, W_Q, W_K\}$ are trained purely by the self-supervised signals above, while the encoder continues to receive gradients from all three sources: self-supervised losses through A_t, RSSM losses through V_t, and reconstruction losses through the decoder.

3.3 Dual-Stream Decoder

With the attention map fixed by the stop-gradient barrier, the decoder composes the two streams into a reconstruction target that allows gradients to shape only entity content without altering the routing.

Dynamics stream via spatial broadcast. Following the spatial broadcast decoder design [36], we first decode the dynamics features into predicted representations $\hat{e}_t = \phi_{\text{fg}}([h_t; z_t])$. We then render each representation back to its attended spatial positions by inverse broadcasting, computing $\hat{F}_t^{\text{fg}} = \bar{A}_t^\top \hat{e}_t$ with $\bar{A}_t$ as the composition mask. This "paints" each attended region back onto its spatial positions. The decoder reads the content entirely from dynamic features, while the attention map specifies only where to draw. This forces the RSSM to retain rich attribute information.

Residual stream. Parallel to the dynamics stream, the full encoder output F_t is passed through a residual mapping ϕ_{bg} producing the background features $\hat{F}_t^{\text{bg}}$. In our implementation ϕ_{bg} uses a Gaussian bottleneck regularized by unit Gaussian prior minimizing $\beta_{\text{bg}}\mathcal{L}_{\text{KL}}^{\text{bg}}$ similar to a VAE. The stochastic bottleneck prevents the residual stream from encoding the entire scene. Instead, it absorbs objective-irrelevant background details to satisfy the dense reconstruction objective. The background latent is not fed into the RSSM and is discarded during imagination rollouts. The dynamics latent handles only attended content, while the reconstruction of objective-irrelevant content is absorbed by the residual stream.

Spatial composition. The aggregate coverage mask $M_t = \text{clamp}(\sum_n \bar{A}_{t,n},\, 0,\, 1)$ measures total objective-relevant content presence at each spatial location by summing over the N queries. Because the attention in Eq. 5 is normalized over spatial tokens rather than over queries, individual query maps are not mutually exclusive and can place attention on shared locations; clamping ensures M_t remains a valid blending weight in $[0, 1]$. The two streams are merged at the feature level:

$$\hat{F}_t = \hat{F}_t^{\text{fg}} \odot M_t \;+\; \hat{F}_t^{\text{bg}} \odot (1 - M_t). \tag{8}$$

A shared transposed convolution upsamples $\hat{F}_t$ back into observation space, resulting in the reconstruction loss $\mathcal{L}_{\text{img}}$. Composing at the feature level rather than at pixel level prevents the soft mask from implicitly encoding shape through its spatial extent [19].

Combining all components, the overall training objective for STARWM is:

$$\mathcal{L} = \underbrace{\mathcal{L}_{\text{img}} + \mathcal{L}_{\text{reward}} + \mathcal{L}_{\text{cont}} + \mathcal{L}_{\text{KL}}^{\text{RSSM}}}_{\text{Dreamer backbone}} + \underbrace{\lambda_{\text{inv}}\,\mathcal{L}_{\text{inv}} + \lambda_{\text{ctr}}\,\mathcal{L}_{\text{ctr}}}_{\text{X-Attention guidance}} + \underbrace{\beta_{\text{bg}}\,\mathcal{L}_{\text{KL}}^{\text{bg}}}_{\text{Residual regularization}}. \tag{9}$$

4 Experiments

Our evaluation addresses four questions about STARWM's design: whether the latent representation cleanly separates objective-relevant from objective-irrelevant content (§4.1); whether this separation translates into policy robustness across distractor types (§4.2); what should train the attention, and whether isolating it from reconstruction is necessary (§4.3); and under which conditions the default self-supervised routing signal suffices, and when task reward is needed to complete it (§4.4).

Baselines and implementation setup. We compare STARWM against four baselines representing distinct distraction-handling paradigms: **DreamerV3** [11], our reconstruction-based backbone; **DenoisedMDP** [33], latent decomposition based on controllability and reward-relevance; **Iso-Dream++** [24], which splits the RSSM into controllable and non-controllable branches via inverse dynamics; and **R2-Dreamer** [21], a concurrent reconstruction-free approach replacing the pixel decoder with a redundancy-reduction objective. Our evaluation uses two complementary environments: a controlled NoisyShape diagnostic environment with ground-truth attribute labels for mechanistic analysis, and the DeepMind Control Suite with video distractors for policy benchmarking. All methods use identical observation resolutions and training budgets (see Appendix B.3); all results report mean and standard deviation across 3 random seeds.

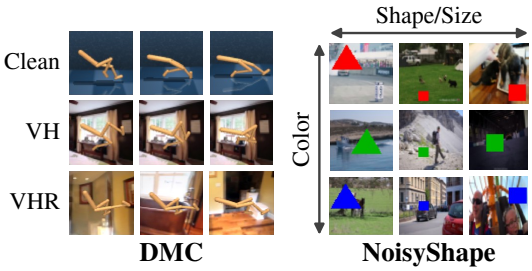


Figure 2: **Evaluation environments.** ***Left:*** DeepMind Control Suite with three background conditions: clean, Video Hard Sequential, and Video Hard Random. ***Right:*** NoisyShape: a geometric shape with controllable attributes (shape, color, size) on DAVIS video backgrounds, used for mechanistic analysis with ground-truth attribute probing; position is randomly re-sampled each step. Arrows denote the four possible actions.

4.1 Latent Separation: Probe Analysis on NoisyShape

We propose NoisyShape as a diagnostic environment for mechanistic analysis of representation learning under controlled distractors (Figure 2; Appendix C.1). For each state, a geometric shape appears across randomly sampled DAVIS [26] video backgrounds. The environment is constructed around three properties targeting distinct challenges: (i) *objective-relevant attributes*, where shape and size are coupled while color cycles independently, all driven by action; (ii) *objective-irrelevant noise*, including randomly re-sampled spatial positions and random video backgrounds that carry no predictive information; and (iii) a *reward-free setup* that isolates representation learning from policy optimization. Combined with ground-truth labels for probing, these properties make NoisyShape suitable for two complementary analyses: evaluating baseline representation paradigms, and conducting incremental architectural ablations on the DreamerV3 backbone.

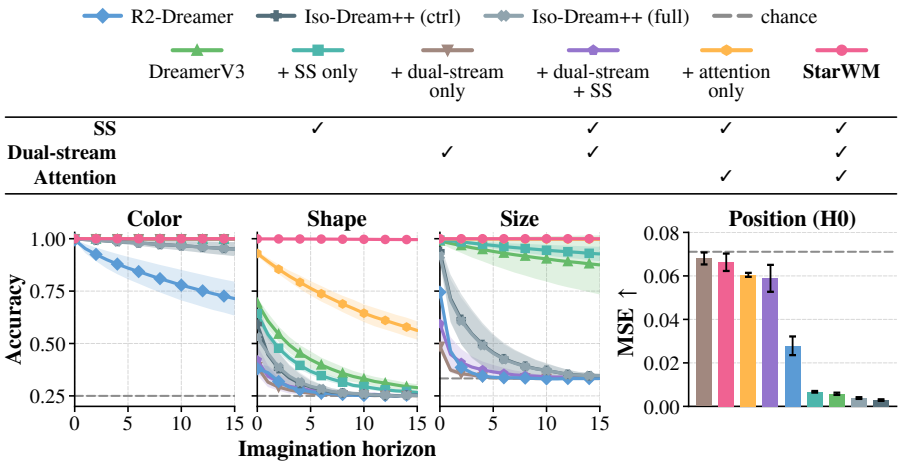


Figure 3: **Latent probe results on NoisyShape.** Probe accuracy for objective-relevant attributes (color, shape, size) from H0 (encoding) through H15 (15 steps of imagination), and objective-irrelevant attribute (position MSE) at H0 (H15 omitted as ground-truth position at $t+15$ is independent of the state at t; higher MSE indicates correct irrelevant treatment); dashed line = chance level.

We train probes on the RSSM latent to predict color, shape, and size at two stages: the encoding step (H0) and after 15 steps of open-loop imagination (H15). H0 measures whether the encoder captures objective-relevant attributes from observations; H15 tests whether the dynamics model preserves these attributes in latent space without further visual input. A representation contaminated by irrelevant content typically encodes well at H0 but loses the relevant signal during long-horizon rollouts, since the dynamics model cannot predict irrelevant noise. As NoisyShape is reward-free, our cross-method comparison here is at the level of latent-probe accuracy rather than policy return; policy comparisons are conducted on DMC (§4.2). Full evaluation protocol is in Appendix C.2 and evaluation on the variants of the NoisyShape is in Appendix E.5.

Objective-relevant attribute retention. Figure 3 shows that STARWM achieves near-perfect retention across all relevant attributes through imagination, while every baseline degrades. DreamerV3's shape retention collapses through imagination, confirming the reconstruction bias; R2-Dreamer degrades already at encoding, since its redundancy-reduction objective lacks signal under random backgrounds; Iso-Dream++'s action-driven separation operates at the latent level and likewise fails to filter irrelevant content; and DenoisedMDP cannot be evaluated here as its decomposition strictly requires reward signals. To isolate which architectural component closes this gap, we evaluate ablations of STARWM that progressively add self-supervised attention guidance, the dual-stream decoder, and cross-attention routing to the DreamerV3 backbone. Every ablation configuration degrades on at least one attribute, indicating that all three components are necessary for distraction-robust representation learning in this reward-free setting.

Position as a diagnostic feature. The position attribute provides a test of selective encoding: an ideal dynamics latent encodes the attribute only when it is dynamically predictable. Though position is sampled uniformly at each step, DreamerV3 faithfully encodes position at H0 (near-zero MSE), driven by its dense reconstruction objective. STARWM discards most of this information (high MSE), correctly treating the random noise as irrelevant (Fig. 3, right). This filtering is driven by the dual-stream decoder: ablations without it retain position precisely. To rule out that STARWM simply drops spatial information regardless of its predictability, we evaluate on a variant where the object follows a predictable trajectory (Appendix E.4). In this variant, STARWM successfully retains object's position through 15 steps of imagination, confirming that the RSSM preserves spatial features when they are dynamically meaningful.

Cross-sample composition. To directly show how the three architectural components specialize, we decode composite images that mix each component across two source frames (Figure 4). Each substitution cleanly swaps the corresponding visual factor while leaving the others intact, revealing

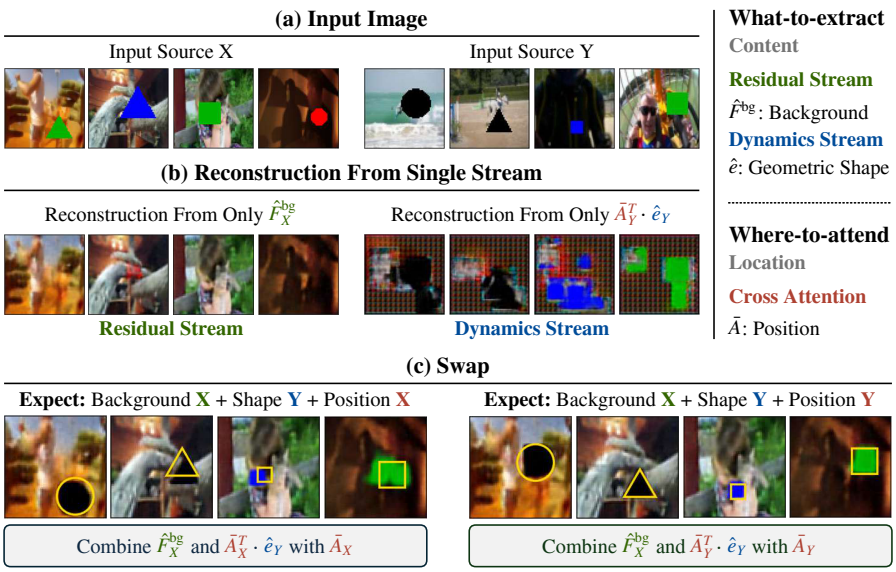


Figure 4: **Cross-sample composition reveals emergent factorization.** **(a.)** Input frames from sources X and Y, with **(b.)** each stream's isolated reconstruction beneath. The residual stream renders only background context, while the dynamics stream renders entity content. **(c.)** Swapping components across X and Y produces controlled outputs: Y's entity on X's background, with position determined by whichever attention mask ($\bar{A}_X$ or $\bar{A}_Y$) is used. The three components specialize into distinct emergent roles, mapping onto the content and location separation of §3 without any explicit disentanglement objective.

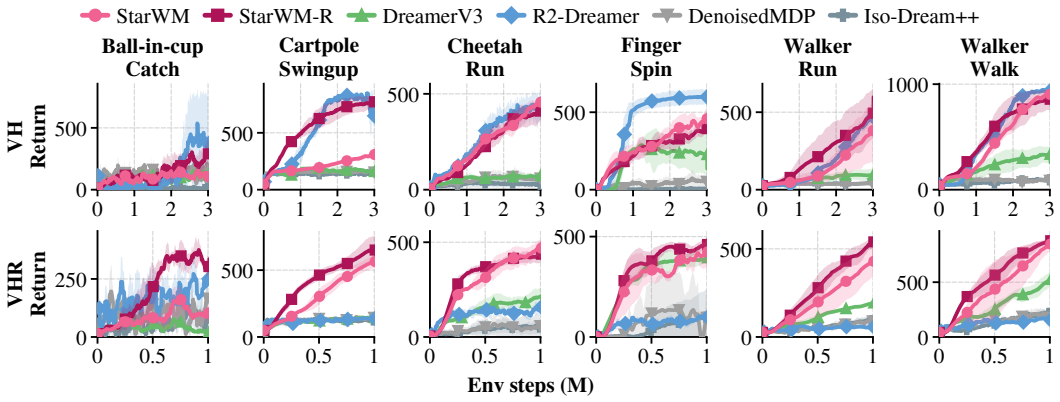


Figure 5: **DMC learning curves.** STARWM and STARWM-R are the only methods that perform strongly under both sequential (VH) and random (VHR) video backgrounds.

an emergent three-way factorization: the residual stream carries background context, the dynamics stream carries entity content, and the attention map carries spatial location. This factorization arises from the architectural design alone, without any explicit disentanglement objective.

4.2 Policy Performance: DMC under Video Distractors

Figure 5 reports per-task evaluation returns on DMC under both video distractor settings, and Figure 1 (right) summarizes cross-condition robustness via IQM aggregated across the six tasks, normalized to DreamerV3 clean performance, yielding a recovery ratio measuring how much of the clean-task return each method retains under distractors. STARWM's additional architectural components do not come at a cost in clean environments, where it matches DreamerV3. Full numerical results, together with aggregate statistics under 95% stratified bootstrap CIs, performance profiles, and

pairwise probability-of-improvement heatmaps following Agarwal et al. [1], are in Appendix D. See Appendix E.3 for hyperparameter sensitivity.

Figure 6 visualizes STARWM's attention maps under both clean and distractor (VH) settings: queries lock to the agent's body parts and do not drift to the visually dominant background. This is direct mechanistic evidence that the routing decision is governed by self-supervised signals rather than reconstruction's pixel. Appendix F.3 confirms that routing stays stable across matched textures, low-contrast agents, and photorealistic scenes.

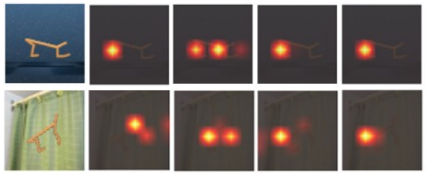


Figure 6: **Learned attention maps on DMC.** STARWM's cross-attention consistently localizes the agent through 4 queries under both clean and distracting settings.

Under random distractors (VHR), where temporal coherence is destroyed, default STARWM achieves the strongest performance on 5 of 6 tasks with substantial margins. This demonstrates the robustness of purely self-supervised cross-attention routing under unstructured noise, where baseline representations struggle to isolate the agent. Under sequential distractors (VH), R2-Dreamer leads on most tasks, since temporally coherent backgrounds align well with redundancy reduction. This same coherence partially co-opts STARWM's contrastive guidance, which rewards any temporally consistent feature and cannot by itself disambiguate agent from background (Appendix E.3.1); despite this, default STARWM remains competitive on high-DOF locomotion (Walker Walk, Cheetah Run), where inverse dynamics provides a strong agent-localizing signal.

In this regime where self-supervised signals underdetermine the routing target, the modular attention pathway becomes useful. As our core routing architecture is inherently agnostic to the supervision source, it naturally accommodates the task reward already available in any RL setting. STARWM-R adds a reward prediction head that directs the reward gradient to guide the spatial mask, matching or exceeding R2-Dreamer on the tasks where default STARWM most strongly underperformed.

Across the six tasks, STARWM and STARWM-R are the only methods that achieve high returns under both VH and VHR (Figure 1, right), and on the combined VH+VHR pool STARWM-R attains the highest IQM and lowest optimality gap among all evaluated methods (Appendix D).

4.3 What Should Train the Attention?

The cross-attention parameters can be trained by three signals: self-supervised losses, reconstruction gradients through the decoder, and RSSM gradients through the entity tokens. STARWM (★) uses stop-gradient barriers to leave the self-supervised losses as the sole signal training the attention parameters. We assess whether this combination is necessary by comparing against two ablations: one (☆1) that removes the stop-gradients so all three sources reach the attention, and one (☆3) that removes the self-supervised loss and makes RSSM gradients the only signal. On NoisyShape, all three configurations yield near-optimal shape retention: the object easiest to reconstruct (the geometric shape) coincides with the action-conditional one, so reconstruction-based gradients guide attention to the correct target without further supervision. On DMC with video distractors, this alignment breaks down and self-supervised guidance becomes critical. Figure 7 shows that both ablations collapse under VHR while STARWM maintains performance; the two design choices are jointly necessary because they together restrict the attention signal to a dynamics-grounded source. Full ablations are in Appendix E.2.

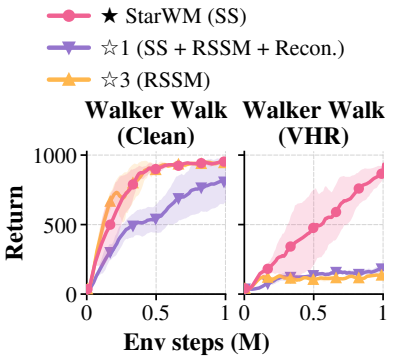


Figure 7: **Ablation study of sources of the attention signal.** Self-supervised guidance and gradient isolation are jointly necessary.

4.4 When Does Self-Supervised Routing Suffice?

What each signal captures. The two default signals select different content. Inverse dynamics points attention to regions whose changes are explained by the agent's action, an unambiguous target: on NoisyShape, it alone suffices for near-optimal routing. Temporal contrastive learning only asks consecutive frames to be more similar than frames sampled from the batch, which mostly come from other episodes. It therefore favors any content that stays consistent within an episode and differs across episodes, and on its own it fails to single out the agent on NoisyShape, consistent with Schneider et al. [28]. On DMC, contrastive learning helps under VHR, where the agent is the only such content, but is partially co-opted under VH, where the sequential video qualifies as well (Appendix E.3.1).

Passive task-relevant content. Some task-relevant content is not driven by the agent's actions: a static goal, or an object that moves only passively. Inverse dynamics gives no signal for it, so whether it is retained depends on the contrastive signal alone. We test this with a NoisyShape variant containing a passive object whose color and position stay fixed within an episode, DMC Reacher (static target), and Ball-in-cup (passively moving ball) (Appendix F.4).

Table 1: **When default routing retains passive task-relevant content.** Retained if the content is static within an episode and differs across episodes, unless a coherent video background already plays that role. Reward routing (STARWM-R) fixes the failure cases. n/a: reward-free environment.

Passive content	Background	Static within episode?	Coherent video?	STARWM	STARWM-R
Object position (R^2)	random	✓	✗	**0.80**	–
	sequential	✓	✓	0.13	–
Reacher target (return)	VHR	✓	✗	**690**	657
	VH	✓	✓	391	**789**
Ball-in-cup ball (return)	VHR	✗	✗	98	**303**
	VH	✗	✓	114	**304**

Table 1 follows directly from what the contrastive signal favors. A static passive object stays fixed within an episode and differs across episodes, so it is retained, but only when nothing else fits better. A sequentially played video fits equally well, takes over the contrastive signal, and the passive object is dropped. A passively moving object, like the ball, is not consistent within an episode, so it is lost under both backgrounds.

This is a limitation of self-supervision in general: from observations alone, a model cannot tell whether predictable but uncontrollable content matters for the task. Reward provides exactly this missing information. Because the routing module accepts any training signal, adding reward prediction (STARWM-R) requires no architectural change and recovers all failure cases in Table 1.

5 Related Work

World models and distraction-robust representation learning. Visual world models divide broadly into reconstruction-based approaches (e.g., the Dreamer family [9, 10, 11]) and reconstruction-free methods (e.g., TD-MPC [14, 13]). While reconstruction provides dense visual grounding, it suffers from reconstruction bias toward dominant pixel areas; conversely, reconstruction-free methods eliminate this bias but discard pixel-level grounding. STARWM bridges this divide by retaining dense reconstruction for content supervision while redirecting capacity away from distractors via self-supervised attention routing.

To tackle visual distractions, prior reconstruction-based methods either substitute reconstruction with auxiliary objectives (DreamerPro [4], R2-Dreamer [21]) or decompose a shared latent space (TIA [7], DenoisedMDP [33], Iso-Dream [23]), but the latter introduces gradient tension across objectives. Spatial approaches localize relevant regions via unsupervised keypoints [34], simulator masks [16], or SAM-based foundation models [35], yet rely on external supervision. SMG [40] uses a dual-branch structure to isolate task-relevant reconstruction, but it is model-free (built on DrQ-v2 [38]) and requires per-task image augmentations. In contrast, STARWM derives its spatial

routing endogenously from environment dynamics, requiring neither external priors nor task-specific augmentations while preserving dense reconstruction for attended content.

Object-centric learning and cross-attention. Slot Attention [19] finds objects using repeated attention steps guided by image reconstruction, and it rebuilds scenes using spatial broadcast decoding. DETR [3] finds objects using learned query vectors in a single attention step. STARWM combines this single-step attention with spatial broadcast decoding. In Slot Attention, pixel reconstruction guides the attention weights, which easily leads the model to focus on large or complex backgrounds. In contrast, STARWM trains its attention using self-supervised dynamics and blocks all gradients from reconstruction. In addition, using a single attention step makes it easy to add stop-gradient barriers between attention routing and feature extraction, which repeated attention steps cannot do.

Self-supervised methods for RL. Prior self-supervised RL methods often use inverse dynamics [25], contrastive learning [17, 22], or predictive objectives [29] to learn whole-image representations, which drops spatial layout. In contrast, STARWM uses these dynamics signals to train a spatial attention module that routes information, while keeping image reconstruction to supervise the visual details of attended parts. The attention gating mechanism is also supervision-agnostic. We default to inverse dynamics and contrastive learning for reward-free applicability, but task-specific signals such as reward prediction (§4.2) can be added.

6 Conclusion

A central challenge in robust world model learning is the tension between reconstruction-based objectives, which provide pixel-level grounding to the observation space but overemphasize irrelevant detail, and reconstruction-free objectives, which ignore noise but can lose key environment information. We introduced STARWM, which unites the strengths of both approaches. By using learned attention to guide reconstruction only to relevant regions, STARWM stops background distractors from entering the latent dynamics model. This prevents the model from wasting capacity on large irrelevant areas, enabling accurate predictions over long horizons. Our default setup uses inverse dynamics and contrastive learning to find controllable and predictable objects without any task rewards, clearly outperforming reconstruction-based baselines under visual distractions. Moreover, this attention design is modular and can easily include other signals, such as task reward in STARWM-R. More broadly, the dual stream decomposition in STARWM echoes the classical what-versus-where pathways in visual neuroscience [20], hinting at the possibility of a more general design principle.

Limitations. Our experiments focus on single-agent continuous control from images. Settings with multiple agents, non-spatial inputs, or real-world domain shifts remain untested, although preliminary tests (Appendix F.2) show that attention queries can separate across multiple objects. Additionally, the 8×8 attention grid can be too coarse for tiny objects (Appendix F.1). Our default attention signals also have trade-offs: inverse dynamics gives weak guidance for passively moving objects (such as the ball in Ball-in-Cup), static targets under coherent backgrounds (Appendix F.4), or delayed action effects (Appendix F.5), while contrastive learning can be fooled by moving video backgrounds. While adding reward prediction fixes these issues in STARWM-R, finding a single self-supervised signal that works in every setting remains an open question (Appendix E.1.1). Finally, keeping an image decoder requires more compute than reconstruction-free methods like R2-Dreamer, though the training cost matches the original DreamerV3 backbone (Appendix B.3).

Acknowledgments and Disclosure of Funding

The authors acknowledge funding and support from the German Aerospace Center (DLR), the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant GRK 3012 – 520750254 (kemai.uni-ulm.de), and the state of Baden-Württemberg through bwHPC and DFG grant INST 35/1597-1 FUGG.

References

[1] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in Neural*

Information Processing Systems, 2021.

[2] 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.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 213–229, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58452-8.

[4] Fei Deng, Ingook Jang, and Sungjin Ahn. DreamerPro: Reconstruction-free model-based reinforcement learning with prototypical representations. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 4956–4975. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/deng22a.html.

[5] Yonathan Efroni, Dylan J Foster, Dipendra Misra, Akshay Krishnamurthy, and John Langford. Sample-efficient reinforcement learning in the presence of exogenous information. In Po-Ling Loh and Maxim Raginsky, editors, *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 5062–5127. PMLR, 02–05 Jul 2022. URL https://proceedings.mlr.press/v178/efroni22a.html.

[6] Stefano Ferraro, Pietro Mazzaglia, Tim Verbelen, and Bart Dhoedt. Focus: object-centric world models for robotic manipulation. *Frontiers in Neurorobotics*, Volume 19 - 2025, 2025. ISSN 1662-5218. doi: 10.3389/fnbot.2025.1585386. URL https://www.frontiersin.org/journals/neurorobotics/articles/10.3389/fnbot.2025.1585386.

[7] Xiang Fu, Ge Yang, Pulkit Agrawal, and Tommi Jaakkola. Learning task informed abstractions. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3480–3491. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/fu21b.html.

[8] David Ha and Jürgen Schmidhuber. World models. *CoRR*, abs/1803.10122, 2018. URL http://arxiv.org/abs/1803.10122.

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

[10] Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=0oabwyZbOu.

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

[12] Nicklas Hansen and Xiaolong Wang. Generalization in reinforcement learning by soft data augmentation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, page 13611–13617. IEEE Press, 2021. doi: 10.1109/ICRA48506.2021.9561103. URL https://doi.org/10.1109/ICRA48506.2021.9561103.

[13] Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Oxh5CstDJU.

[14] Nicklas A Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 8387–8406. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/hansen22a.html.

[15] Kaichen Huang, Shenghua Wan, Minghao Shao, Hai-Hang Sun, Le Gan, Shuai Feng, and De-Chuan Zhan. Leveraging separated world model for exploration in visually distracted environments. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 82350–82374. Curran Associates, Inc., 2024. doi: 10.52202/079017-2618. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/96189e90e599ccc43f00434ff3ed0312-Paper-Conference.pdf.

[16] Kyungmin Kim, JB Lanier, and Roy Fox. Make the pertinent salient: Task-relevant reconstruction for visual control with distractions. *Reinforcement Learning Journal*, 6:1334–1364, 2025.

[17] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. CURL: Contrastive unsupervised representations for reinforcement learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5639–5650. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/laskin20a.html.

[18] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 9628–9639, Red Hook, NY, USA, 2018. Curran Associates Inc.

[19] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 11525–11538. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/8511df98c02ab60aea1b2356c013bc0f-Paper.pdf.

[20] Mortimer Mishkin, Leslie G. Ungerleider, and Kathleen A. Macko. Object vision and spatial vision: two cortical pathways. *Trends in Neurosciences*, 6:414–417, 1983. ISSN 0166-2236. doi: https://doi.org/10.1016/0166-2236(83)90190-X. URL https://www.sciencedirect.com/science/article/pii/016622368390190X.

[21] Naoki Morihira, Amal Nahar, Kartik Bharadwaj, Yasuhiro Kato, Akinobu Hayashi, and Tatsuya Harada. R2-dreamer: Redundancy-reduced world models without decoders or augmentation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Je2QqXrcQq.

[22] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[23] Minting Pan, Xiangming Zhu, Yunbo Wang, and Xiaokang Yang. Iso-dream: Isolating and leveraging noncontrollable visual dynamics in world models. *Advances in neural information processing systems*, 35:23178–23191, 2022.

[24] Minting Pan, Xiangming Zhu, Yitao Zheng, Yunbo Wang, and Xiaokang Yang. Model-based reinforcement learning with isolated imaginations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):2788–2803, 2024. doi: 10.1109/TPAMI.2023.3335263.

[25] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2778–2787. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/pathak17a.html.

[26] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 724–732, 2016. doi: 10.1109/CVPR.2016.85.

[27] Kate Rakelly, Abhishek Gupta, Carlos Florensa, and Sergey Levine. Which mutual-information representation learning objectives are sufficient for control? In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

[28] Felix Schneider, Xia Xu, Markus R. Ernst, Zhengyang Yu, and Jochen Triesch. Contrastive learning through time. In *SVRHM 2021 Workshop @ NeurIPS*, 2021. URL https://openreview.net/forum?id=HTCRs8taN8.

[29] Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uCQfPZwRaUu.

[30] Gautam Singh, Fei Deng, and Sungjin Ahn. Illiterate DALL-e learns to compose. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=h0OYV0We3oh.

[31] Ruixiang Sun, Hongyu Zang, Xin Li, and Riashat Islam. Learning latent dynamic robust representations for world models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[32] Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020. ISSN 2665-9638. doi: https://doi.org/10.1016/j.simpa.2020.100022. URL https://www.sciencedirect.com/science/article/pii/S2665963820300099.

[33] Tongzhou Wang, Simon Du, Antonio Torralba, Phillip Isola, Amy Zhang, and Yuandong Tian. Denoised MDPs: Learning world models better than the world itself. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 22591–22612. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/wang22c.html.

[34] Xudong Wang, Long Lian, and Stella X. Yu. Unsupervised visual attention and invariance for reinforcement learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6673–6683, 2021. doi: 10.1109/CVPR46437.2021.00661.

[35] Ziyu Wang, Yanjie Ze, Yifei Sun, Zhecheng Yuan, and Huazhe Xu. Generalizable visual reinforcement learning with segment anything model. *arXiv preprint arXiv:2312.17116*, 2023.

[36] Nick Watters, Loic Matthey, Chris P. Burgess, and Alexander Lerchner. Spatial broadcast decoder: A simple architecture for disentangled representations in VAEs, 2019. URL https://openreview.net/forum?id=S1x7WjnzdV.

[37] Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. Slotformer: Unsupervised visual dynamics simulation with object-centric models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=TFbwV6I0VLg.

[38] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=_SJ-_yyes8.

[39] Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=-2FCwDKRREu.

[40] Di Zhang, Bowen Lv, Hai Zhang, Feifan Yang, Junqiao Zhao, Hang Yu, Chang Huang, Hongtu Zhou, Chen Ye, and changjun jiang. Focus on what matters: Separated models for visual-based RL generalization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=wz2KvvEk44.

Appendix: Table of Contents

A Extended Related Work **15**

A.1 Taxonomy of Observation Components 15

A.2 Self-Supervised Signals and Their Limitations 16

A.3 DreamerV3's Built-in Noise Handling 16

A.4 Comparison with Object-Centric and Slot-Based Methods 17

B Method & Implementation Details **17**

B.1 Algorithm 17

B.2 Hyperparameters 19

B.3 Implementation of Baseline Algorithms and Compute Resources 19

C Environments & Evaluation Protocols **21**

C.1 Shape Environments 21

C.2 Probe Analysis Methodology for NoisyShape 23

C.3 DeepMind Control Suite Environments 23

D Full results & Aggregate Metrics **24**

D.1 Full results in NoisyShape Environment 24

D.2 Full results in DMC Environments 25

E Ablation Studies & Core Mechanistic Analysis **26**

E.1 Reward as Additional Attention Routing Signal 26

E.2 Gradient Isolation Analysis 28

E.3 Hyperparameter Sensitivity 31

E.4 Selective Spatial Encoding Analysis 34

E.5 Progressive Distraction Analysis across NoisyShape Variants 35

F Stress Testing, Generalization & Boundary Cases **36**

F.1 Spatial Granularity & Small Objects 36

F.2 Multi-Object Extension: NoisyTwoShape 38

F.3 Texture, Low Contrast, and Photorealistic Scenes 39

F.4 Passive and Non-Action-Driven Task-Relevant Objects 41

F.5 Delayed Action Effects 44

G Broader Impacts **45**

NeurIPS Paper Checklist **46**

A Extended Related Work

This section provides an in-depth discussion of how STARWM relates to and differs from existing methods across several research axes.

A.1 Taxonomy of Observation Components

A visual observation in a typical RL environment contains multiple types of information, which can be classified along three orthogonal axes:

- **Predictable vs. unpredictable.** Can the component's future values be predicted from past states and actions? The agent's body dynamics are predictable; randomly sampled background textures are not.
- **Controllable vs. uncontrollable.** Is the component's transition function conditionally dependent on the agent's action? The agent's joint angles are controllable; a moving cloud in the background is not.
- **Task-relevant vs. task-irrelevant.** Does the component affect the reward function? The ball's position in a catching task is task-relevant; wall texture is not.

Different prior methods implicitly target different subsets of this taxonomy:

- **DBC** [39]: separates task-relevant from task-irrelevant via bisimulation metrics. Requires a meaningful reward signal.
- **TIA** [7]: separates task-relevant from task-irrelevant via reward-prediction adversarial learning. Brittle under sparse rewards and non-transferable across tasks.
- **DenoisedMDP** [33]: provides a four-way decomposition based on controllability $\times$ reward-relevance via information-theoretic objectives. Operates at the final encoder output without explicit spatial reasoning.
- **Iso-Dream / Iso-Dream++** [23, 24]: separate controllable from noncontrollable latent transitions using inverse dynamics, then roll out the two branches separately for behavior learning. Like STARWM, uses inverse dynamics as the separation signal, but applies it at the latent level rather than the spatial level.
- **HRSSM** [31]: combines spatio-temporal cube masking with bisimulation-based similarity to learn compact reward-aware representations, using a hybrid RSSM where a mask branch and an EMA raw branch share history. Operates at the global feature level rather than spatially, and the bisimulation component requires reward signals.
- **SeeX** [15]: targets unsupervised exploration with a dual-branch world model that mirrors STARWM's relevant/irrelevant split at the architectural level. Separation is driven by an exogenous reconstruction loss that requires s^- to reconstruct the full observation, exploiting the prior that task-relevant content occupies a small portion of pixels. Unlike STARWM, the two branches operate on entire images without spatial routing, and the policy is learned via intrinsic reward maximization rather than task reward.
- **ExoRL** [5]: formally defines the ExoMDP framework, separating endogenous (action-affected) from exogenous (action-independent) state factors. It provides theoretical regret guarantees but does not address the pixel-level representation challenges in visual RL.

STARWM differs from these methods on two axes. First, the decomposition operates at spatial resolution through explicit cross-attention, rather than at the latent level as in DenoisedMDP and Iso-Dream/Iso-Dream++. Second, STARWM is a framework rather than a fixed objective: any differentiable signal applied to the live attention path trains the routing without interfering with reconstruction, by virtue of the stop-gradient isolation. Our default instantiation combines inverse dynamics (controllability axis) with temporal contrastive learning (predictability axis), neither of which requires reward; when reward is available, it can be added as an additional attention-guidance signal along the task-relevance axis (§E.1). The choice of attention loss thus becomes a modular design decision based on task characteristics, rather than a fixed architectural commitment.

A.2 Self-Supervised Signals and Their Limitations

Self-supervised signals provide an attractive alternative to reward-based supervision because they are derived from the agent's interaction with the environment without requiring task labels. The most popular such signals fall into three families: state-only contrastive learning (e.g., CURL [17]), forward-dynamics prediction (e.g., SPR [29]), and inverse-dynamics prediction [25].

Rakelly et al. [27] analyze these three families theoretically and show that none of them is a reliable representation learner for control alone. State-only contrastive objectives can yield representations that alias states with different optimal actions. Inverse-dynamics objectives can discard task-relevant state information that is not necessary for predicting actions. Forward-dynamics objectives are the most robust of the three, but retain any state information that is temporally correlated.

These results motivate STARWM's design. Rather than rely on any single self-supervised objective as the primary representation learner, we use them as attention guidance signals: inverse dynamics steers attention toward action-conditional regions, and contrastive learning provides complementary temporal supervision in transitions where inverse dynamics is silent. The visual content of attended regions is supervised separately by reconstruction through the dual-stream decoder, and stop-gradient isolation prevents reconstruction gradients from corrupting the attention parameters or vice versa. In this way, each signal operates within the regime where its inductive bias is robust, and no single signal carries the full burden of representation learning.

A.3 DreamerV3's Built-in Noise Handling

DreamerV3 already incorporates several features designed to handle stochastic environments:

- **Dual latent variables.** The RSSM maintains both a deterministic recurrent state h_t (encoding history) and a stochastic categorical state z_t (capturing single-step uncertainty). In principle, the stochastic component can absorb unpredictable variation, while the deterministic component tracks persistent dynamics.
- **KL balancing and free nats.** KL balancing prevents posterior collapse by asymmetrically scaling the KL divergence between the posterior and prior. The free nats mechanism sets a minimum KL threshold, allowing the model to maintain a minimum level of stochasticity in z_t without being penalized.
- **Symlog scaling.** Symlog transformations handle varying signal magnitudes, improving stability across diverse domains.

These mechanisms are effective for latent-level noise handling: the stochastic z_t can represent uncertainty over globally unpredictable factors, and the prior $p(z_t|h_t)$ can learn to "ignore" unpredictable components whose KL cost exceeds their reconstruction benefit. However, they operate after the encoder has already compressed the full 64×64 observation into a single vector. Three core vulnerabilities remain upstream:

- **The encoder bottleneck.** The dense image decoder requires the encoder to faithfully encode all visual content. The RSSM's noise-handling mechanisms cannot undo this upstream encoding pressure; they can only attempt to filter the already-contaminated latent.
- **Spatially entangled representations.** Without spatial attention, attended and background features are entangled in the same latent vector. The stochastic z_t lacks the expressivity to cleanly separate spatially localized entities from distributed background patterns. Even with free nats, the KL budget is shared between genuine action-induced uncertainty and background stochasticity.
- **Imagination rollouts.** During imagination, the world model generates z_t from the prior $p(z_t|h_t)$. If the posterior z_t was contaminated by background information during encoding, the prior must also model this contamination to match. This leads to hallucinated background dynamics in imagined trajectories, directly degrading policy quality.

STARWM addresses this gap by intervening before the RSSM: the cross-attention module spatially separates attended entities from backgrounds at the feature map level, feeding only objective-relevant entity features into the RSSM. The dual-stream decoder removes the reconstruction pressure on the

RSSM by handling background reconstruction through a separate VAE pathway that is discarded during imagination. The RSSM's built-in noise handling mechanisms can then focus entirely on the dynamics of endogenous entities, rather than wasting capacity on filtering background contamination.

A.4 Comparison with Object-Centric and Slot-Based Methods

Object-centric methods. Among object-centric methods, Slot Attention [19] is the most closely related line to STARWM in mechanism, sharing the use of attention to extract entity-level representations from spatial feature maps. However, the two approaches differ fundamentally in what trains the attention:

	Slot Attention	**STARWM**
Attention supervision	Reconstruction	Self-supervised Signals
Attention refinement	Iterative	Single-pass
Slot competition	Softmax over slots	Independent softmax per query
Gradient from recon.	Flows through attention	Blocked by stop-gradient

For reconstruction-supervised attention, the mechanism is trained to minimize reconstruction error, which is proportional to the number of pixels covered. This incentivizes slots to attend to visually complex, spatially dominant regions, such as the distractor backgrounds that should be ignored. Additionally, Slot Attention's iterative refinement couples the attention weights to the decoder, making it infeasible to apply a simple stop-gradient fix without breaking the iterative convergence.

STARWM's self-supervised attention training is immune to this bias: inverse dynamics provides gradients only for action-conditional regions, while the stop-gradient ensures that reconstruction gradients never reach the attention parameters. This makes the attention mechanism robust to arbitrary background complexity without requiring any modification to the loss functions.

Object-centric world models. The broader landscape of object-centric world models includes both slot-attention-based methods (e.g., SlotFormer [37], SLATE [30]) and methods relying on external segmentation supervision (e.g., FOCUS [6], which uses ground-truth masks from simulators or SAM). These methods aim at scene decomposition into discrete objects for prediction or control, typically in environments with well-defined object boundaries. STARWM addresses a different regime: distractor robustness in visual control, where the failure mode is the reconstruction trap rather than object decomposition. It is not designed for full multi-object scene parsing; nevertheless, our two-object experiment (Appendix F.2) shows that attention queries spontaneously partition across relevant entities even in this distractor regime, suggesting compatibility with multi-object settings as a future direction.

B Method & Implementation Details

B.1 Algorithm

Algorithm 1 presents the complete training procedure for one batch. Below we detail implementation choices omitted from the main text for brevity.

Encoder interception. Standard DreamerV3 compresses a 64×64 image through four convolutional stages to a 4×4 feature map (16 spatial tokens) before the RSSM. STARWM intercepts the encoder after its third convolutional stage, retaining an intermediate 8×8 feature map F_8 with 128 channels ($L{=}64$ spatial tokens). A 1×1 convolution projects F_8 to $C = 256$ channels. The fourth stage continues normally to produce F_4 (256 channels, 4×4), which is flattened to a 4096-dimensional vector for the background VAE stream (§3.3).

Multi-head attention aggregation. The cross-attention module (§3.1) uses $H{=}4$ heads with head dimension $C/H{=}64$. Query and key projections W_Q, W_K operate per-head; attention logits are computed independently per head and are softmaxed. The final attention map $A_t \in \mathbb{R}^{N\times L}$ is the arithmetic mean across heads. Entity tokens are then extracted by applying this averaged map to the unprojected values, i.e. $e_t^{dyn} = \bar{A}_t \cdot V_t$.

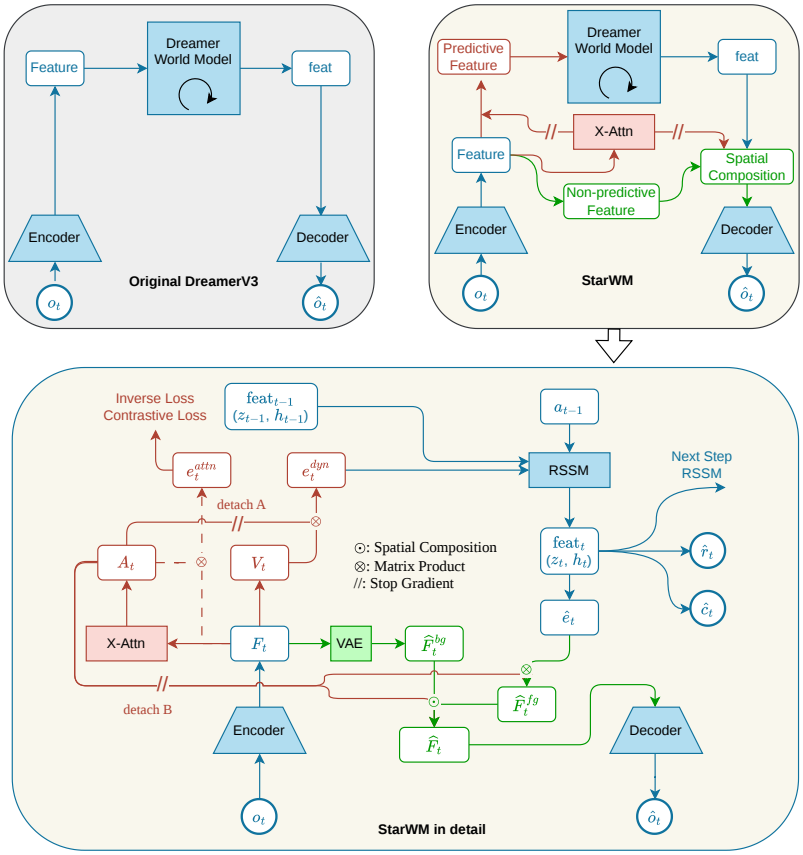


Figure 8: **StarWM architecture.** Top: comparison with DreamerV3. StarWM splits the encoder output via cross-attention into an attended stream (entering the RSSM) and a residual stream (auxiliary VAE, discarded during imagination). Bottom: detailed architecture. The cross-attention module (red) extracts entity tokens guided by self-supervised losses; the dual-stream decoder (green) composes attended foreground content with residual background content for reconstruction. Blue components are inherited from DreamerV3. Stop-gradient barriers (//) isolate attention parameters from reconstruction gradients.

Image loss. StarWM uses per-pixel MSE loss summed over spatial dimensions, replacing DreamerV3's symlog-discretized distribution head for the image decoder. This simplification is sufficient because the dual-stream architecture handles signal magnitude differences between foreground and background through explicit spatial routing rather than output normalization.

Inverse dynamics for continuous vs. discrete actions. The inverse dynamics loss $\mathcal{L}_{\text{inv}}$ (Eq. 6) uses cross-entropy for discrete action spaces (NoisyShape, 4 actions) and mean squared error for continuous action spaces (DMC).

Contrastive negative sampling. The temporal contrastive loss (Eq. 7) uses binary InfoNCE with one positive (the temporally adjacent frame) and one negative (a randomly sampled frame from the same batch). This is computationally lightweight and empirically sufficient given that the primary attention guidance comes from inverse dynamics.

Background VAE KL annealing. The background KL scale β_{bg} is annealed linearly from $\beta_{\text{bg}}^{\text{init}} = 5.0$ to $\beta_{\text{bg}}^{\text{final}} = 0.5$ over the first 15,000 training steps. The high initial value encourages the background VAE to produce near-prior samples early in training, preventing it from encoding

Algorithm 1 STARWM World Model Training (one batch)

Require: Batch of sequences $\{(o_t, a_t, r_t, c_t)\}_{t=1}^{T}$

// Encoder interception

$F_t^8 \leftarrow \text{Encoder}_{\text{layers 1-3}}(o_t)$ ▷ Intermediate 8×8 features

$F_t \leftarrow \text{flatten}(\text{Conv}_{1\times1}(F_t^8))$ ▷ Project to $L \times C$

$F_t^4 \leftarrow \text{Encoder}_{\text{layer 4}}(F_t^8)$ ▷ Full encoder output at 4×4

// Cross-attention spatial routing

$K_t \leftarrow F_t + E_{\text{pos}}$ ▷ Keys with positional encoding

$V_t \leftarrow [F_t;\ \text{E}_{\text{coord}}]$ ▷ Values with coordinate grid

$A_t \leftarrow \text{MultiHeadAttn}(Q, K_t)$ ▷ Attention weights $\in \mathbb{R}^{N\times L}$

$e_t^{\text{attn}} \leftarrow \text{flatten}(A_t F_t)$ ▷ Entity features (live A)

// Self-supervised attention training (gradient flows through A)

$\mathcal{L}_{\text{inv}} = \ell\big(\phi_{\text{inv}}([e_t^{\text{attn}};\ e_{t+1}^{\text{attn}}]),\ a_t\big)$

$\mathcal{L}_{\text{ctr}} = -\log\sigma\Big(\frac{\phi_{\text{ctr}}(e_t^{\text{attn}})^\top \phi_{\text{ctr}}(e_{t+1}^{\text{attn}})}{\tau}\Big) - \log\sigma\Big(-\frac{\phi_{\text{ctr}}(e_t^{\text{attn}})^\top \phi_{\text{ctr}}(\tilde{e}^{\text{attn}})}{\tau}\Big)$

// Stop-gradient decoupled pathway (gradient blocked at $\bar{A}$)

$\bar{A}_t \leftarrow \text{sg}(A_t)$ ▷ Block gradients to attention params

$e_t^{\text{dyn}} \leftarrow \phi_{\text{dyn}}(\bar{A}_t V_t)$ ▷ RSSM input

$\text{feat}_t\ ([h_t, z_t]), \leftarrow \text{RSSM}$ ▷ Latent dynamics

$\mathcal{L}_{\text{KL}}^{\text{RSSM}} \leftarrow \text{RSSM}(e_t^{\text{dyn}}, a_{t-1})$ ▷

$\mathcal{L}_{\text{reward}} \leftarrow -\ln p_\theta(r_t \mid \text{feat}_t)$

$\mathcal{L}_{\text{cont}} \leftarrow -\ln p_\theta(c_t \mid \text{feat}_t)$

// Dual-stream decoder

$\hat{F}_t^{\text{fg}} \leftarrow \bar{A}_t^\top \cdot \phi_{\text{fg}}(\text{feat}_t)$ ▷ attended: inverse broadcast

$z_t^{\text{bg}} \sim q_\psi(z_t^{\text{bg}} \mid \text{flatten}(F_t^4))$ ▷ Background: VAE encoding

$\hat{F}_t^{\text{bg}} \leftarrow \phi_{\text{bg}}(z_t^{\text{bg}})$

$M \leftarrow \text{clamp}(\sum_n \bar{A}_{t,n},\ 0,\ 1)$ ▷ Coverage mask

$\hat{o}_t \leftarrow \text{Decode}(\hat{F}_t^{\text{fg}} \odot M + \hat{F}_t^{\text{bg}} \odot (1-M))$

$\mathcal{L}_{\text{img}} \leftarrow \|\hat{o}_t - o_t\|^2$

// Joint optimization (stop-gradient prevents conflicting gradients)

$$\mathcal{L} = \underbrace{\mathcal{L}_{\text{img}} + \mathcal{L}_{\text{reward}} + \mathcal{L}_{\text{cont}} + \mathcal{L}_{\text{KL}}^{\text{RSSM}}}_{\text{Dreamer backbone}} + \underbrace{\lambda_{\text{inv}}\,\mathcal{L}_{\text{inv}} + \lambda_{\text{ctr}}\,\mathcal{L}_{\text{ctr}}}_{\text{X-Attention guidance}} + \underbrace{\beta_{\text{bg}}\,\mathcal{L}_{\text{KL}}^{\text{bg}}}_{\text{Residual regularization}}\ .$$

Update all parameters with $\nabla\mathcal{L}$

foreground content before the attention module has learned to localize entities. After annealing, the lower β_{bg} permits the background stream to reconstruct richer background detail without competing with the dynamics pathway.

B.2 Hyperparameters

Table 2 lists the key hyperparameters of STARWM used throughout all experiments.

B.3 Implementation of Baseline Algorithms and Compute Resources

We compare our method against three representative model-based reinforcement learning baselines: DreamerV3 [11], R2-Dreamer [21], DenoisedMDP [33], and Iso-Dream++ [24]. Original implementation of DreamerV3 is in JAX, and We use the PyTorch reimplementation on GitHub[NM512/dreamerv3-torch]. For other baselines, we use the official open-source implementation and adopt the recommended default hyperparameters. To ensure that all methods are evaluated under identical observation conditions, we make minimal modifications to integrate a

Table 2: Hyperparameters of STARWM.

Component	Hyperparameter	Value
Cross-attention	Number of queries N	4
	Number of heads H	4
	Head dimension d/H	64
	Projected channels d	256
	Intercept resolution	8×8
	Positional encoding	2D sinusoidal
Background VAE	Latent dimension D_{bg}	128
	KL free bits κ	1.0
	KL scale β_{bg} (final)	0.5
	KL scale β_{bg} (initial)	5.0
	Anneal steps T_{anneal}	15,000
Self-supervised	Inverse dynamics scale λ_{inv}	1.0
	Contrastive scale λ_{ctr}	1.0
	Contrastive temperature τ	0.1
	Projection dimension D_p	128
Decoder	Upsampling stages	3 ($8 \to 64$)
	Channel progression	$256 \to 128 \to 64 \to 3$
	Kernel size	4
DreamerV3 backbone	RSSM deterministic units	512
	Discrete latents z	32 classes $\times$ 32 categories
	Encoder CNN depth	32 (channels: $32 \to 64 \to 128 \to 256$)
	RSSM KL scale β	0.5
	KL free bits	1.0

unified Video Hard background distractor wrapper, while leaving the underlying agent architecture, optimization procedure, and learning hyperparameters untouched. For dmc tasks, We use the dmc_vision configuration as provided in the official codebase for each baseline.

All baselines and our method are run within the same conda environment, on a single NVIDIA A100 / H100 GPU per training run. Each training run uses three random seeds; for each task–method combination, the three seeds are packed onto a single GPU and trained concurrently. Training budgets are set to 3M environment steps for the VH variant and 1M environment steps for the VHR variant. Table 3 reports the resulting average GPU-hours per task. The reported wall-clock numbers should be interpreted as approximat as the same GPU model exhibits speed variation depending on cluster load and concurrent jobs, and A100/H100 differ in raw throughput.

Table 3: Average GPU-hours per task on a single NVIDIA H100/A100 (one training run packs 3 random seeds onto the same GPU). Numbers are total wall-clock time to reach the target step budget (1M for Clean and VHR; 3M for VH).

Algorithm	Clean (1M)	VH (3M)	VHR (1M)
StarWM (ours)	35	94	34
Orig DreamerV3	30	104	37
Iso-Dream++	—	103	37
DenoisedMDP	—	37	15
R2-Dreamer	—	26	10

STARWM runs at comparable speed to other reconstruction-based world models (DreamerV3, Iso-Dream++) and is slower than R2-Dreamer, which forgoes the image decoder. This is consistent with our limitations discussion: retaining reconstruction provides visual grounding at the cost of higher per-step compute. The cross-attention and self-supervised auxiliary heads add minor overhead relative to the DreamerV3 backbone STARWM builds on.



Figure 9: **C.1: Illustration of NoisyShape Environments.**

C Environments & Evaluation Protocols

C.1 Shape Environments

The Shape family of environments provides a controlled and interpretable testbed for analyzing how world models encode and retain objective-relevant attributes under visual distractions.

- **Shape and Scale:** There are four geometric shapes (circle, triangle, square, and pentagon) arranged in a fixed cycle. Each shape appears in one of three sizes: small, medium, or large. When the agent applies the *forward shape* action, the object increases in size; once it reaches the largest size, it transitions to the next shape in its smallest form. The *backward shape* action reverses this progression.
- **Color:** The object color cycles through red $\rightarrow$ green $\rightarrow$ blue $\rightarrow$ black and vice versa, controlled by the other two discrete actions (*forward color* and *backward color*).
- **Position:** The spatial location of the object is randomized independently at each time step and does not respond to agent actions.
- **Noise:** Noisy backgounds.

The agent's action space is thus composed of four discrete actions, corresponding to the forward/backward transitions of shape and color.

Shape environment details. The observations in the Shape environment are rendered as images with an initial resolution of 64×64 pixels and 3 channels (RGB).

The position of the geometric shape in each image is randomized using a uniform distribution, constrained to ensure that the shape is fully contained within the image boundaries. Every shape, whether a circle, triangle, square, or pentagon, uses the same radius value for each size, meaning small, medium, and large mean the same thing across all of them. However, due to their differing geometries, the resulting areas vary across shape types.

The mapping between shape type, size, and the corresponding radius and area is summarized in Table 4.

Features of the environment. The Shape environment contains diverse observational features that differ in predictability and semantic relevance. Noise is entirely stochastic and neither predictable nor reconstructable, acting as irrelevant variation that should ideally be ignored by the representation. Position, though fully reconstructable from the observation, is unpredictable due to random sampling at each step. It is often retained by reconstruction-based methods despite being semantically uninformative. Shape and size form a coupled, action-dependent dynamic; size changes more rapidly and triggers shape transitions, making both features highly predictable and essential for modeling controllable structure. However, note that only size information is essential to predict actions between neighbor observations. In contrast, color evolves independently of shape and size. Its small discrete state space and weak temporal continuity make it harder to reliably model using temporal objectives

Table 4: Parameters for shapes.

	Small, $r = 6.4$	Medium, $r = 11.52$	Large, $r = 16.64$
Circle	$S = 129$ (3.14%)	$S = 417$ (10.18%)	$S = 870$ (21.24%)
Triangle	$S = 53$ (1.30%)	$S = 172$ (4.21%)	$S = 360$ (8.78%)
Square	$S = 82$ (2.00%)	$S = 265$ (6.48%)	$S = 554$ (13.52%)
Pentagon	$S = 97$ (2.38%)	$S = 316$ (7.70%)	$S = 658$ (16.07%)

such as contrastive learning. This decomposition illustrates the challenges in disentangling useful dynamics from distractors and guides the design of robust representation objectives.

Chance level calculation. For categorical attributes, chance accuracy follows directly from the number of classes: $1/4$ for color (4 colors), $1/4$ for shape (4 shapes), and $1/3$ for size (3 sizes).

For position, the target object's normalized coordinates (x, y) are sampled independently and uniformly within $[m, 1-m]^2$, where m is the object-size-dependent margin that prevents the object from extending outside the frame. Since both coordinates are symmetrically distributed around 0.5, the optimal constant predictor (in MSE) is $\hat{x} = \hat{y} = 0.5$, and its expected squared error equals the sum of the variances of x and y:

$$E\left[(x-0.5)^2 + (y-0.5)^2\right] = 2 \cdot \mathrm{Var}(x) = \frac{(1-2m)^2}{6}. \tag{10}$$

The margin m is determined by the object's radius (Table 4): $m = r/64 \in \{0.10, 0.18, 0.26\}$ for small, medium, and large sizes respectively. Since the environment cycles through sizes deterministically with equal long-run frequency $1/3$, the chance-level position MSE is

$$\frac{1}{3} \sum_{m \in \{0.10, 0.18, 0.26\}} \frac{(1-2m)^2}{6} \approx 0.0711. \tag{11}$$

We use this value as the dashed chance line in Figure 3.

Design rationale. The object occupies $2\% \sim 20\%$ of pixels while the background covers $80\% \sim 98\%$, creating a direct test of pixel-area bias. Rewards are uniform random noise, ensuring that no reward signal can guide representation learning. This forces the model to rely entirely on its inductive biases and training objectives for entity identification.

Variants. We have four background conditions:

- **Shape Env** (clean): Solid white background. Position is randomized each step. This isolates the dynamics model's ability to encode shape under minimal distraction.
- **Shape Env-Fixed** (clean, fixed position): shape at a fixed screen position on a plain background. This baseline removes *all* sources of variation except the objective-relevant attributes (color, shape, size), establishing that DreamerV3 can perfectly encode these attributes when no confounding information competes for representational capacity.
- **Noisy Shape Env** (random noise): Each step independently samples a DAVIS [26] video frame as the background (64×64, drawn from all 6208 frames of the 90 DAVIS 2017 train+val sequences). Both position and background are resampled every step, creating maximal stochasticity.
- **Noisy Shape Sequential Env** (sequential noise): Similar to Noisy Shape Env, but the background video frames advance sequentially within each episode (one frame per step from a DAVIS video) rather than being independently sampled. This creates a more realistic but also more challenging distraction pattern: the background has temporal autocorrelation, which may cause the contrastive loss to reward encoding background dynamics.

- **Noisy Shape Env with Uniform Motion** (random noise, positions are predictable): The shape moves in uniform linear motion: at each step, position advances by a fixed velocity vector, reflecting off the image boundaries on contact(a margin of 0.05, i.e. positions stay in $[0.05, 0.95]^2$). Initial position and velocity are sampled once per episode and held constant thereafter. All other attributes (color, shape, size, action space, reward noise) follow the NoisyShape setup. Background frames are sampled from a fixed pool of 20 DAVIS frames (the first 20 frames in sorted order, all from a single sequence) and switched independently at each step, providing a clean test of objective-irrelevant signal suppression while position becomes a dynamically predictable attribute. See Appendix E.4 for results and discussion.

C.2 Probe Analysis Methodology for NoisyShape

The diagnostic probe analysis in §4.1 evaluates how well the RSSM latent state encodes ground-truth environment attributes. We describe the data collection, probe architecture, and training protocol below.

Data collection. We collect trajectories by running a random policy in the target environment. For each experiment, we gather 500 episodes (each 200 steps) for training the probe classifiers, and a separate set of 200 episodes for evaluating imagined rollouts. Both sets are collected once and cached across all models to ensure fair comparison. Actions are sampled uniformly from the discrete action space (4 actions).

Feature extraction. For each model, we encode the collected trajectories step-by-step through the frozen world model's encoder and RSSM. At each timestep, we extract the RSSM feature vector $\text{feat}_t = [\, h_t;\ z_t \,] \in \mathbb{R}^{1536}$ (512 deterministic + 32×32 stochastic). For STARWM, encoding follows the cross-attention pathway: the encoder produces F_8, which is projected by the $1{\times}1$ convolution, pooled via cross-attention, and passed through proj_{dyn} before entering the RSSM. For imagined evaluation, we additionally perform open-loop imagination from each encoded state: starting from the posterior latent, we apply $H{=}15$ consecutive `img_step` calls with random actions, extracting the imagined feature at each horizon.

Probe architecture. Each probe is a 3-layer MLP: $\text{Linear}(1536, 256) \to \text{ReLU} \to \text{Dropout}(0.2) \to \text{Linear}(256, 256) \to \text{ReLU} \to \text{Dropout}(0.2) \to \text{Linear}(256, C)$, where C is the number of classes for classification (4 for color, 4 for shape, 3 for size) or 1 for regression (position).

Training protocol. The dataset is split 70/30 into training and test sets (stratified by label for classification). We train with Adam (learning rate 10^{-3}, weight decay 10^{-4}), batch size 512, cross-entropy loss for classification and MSE loss for regression, for up to 100 epochs with early stopping (patience $=$ 10 epochs on validation loss). The best checkpoint by validation loss is restored for final evaluation. Test accuracy (classification) or test MSE (regression) is reported.

Imagined label computation. Ground-truth labels at imagination horizon h are computed deterministically from the starting labels and the sequence of imagined actions, following the environment's action logic. For NoisyShape environments, actions 0/1 cycle the color backward/forward among 4 options, and actions 2/3 cycle the combined shape-size index backward/forward among 12 options (4 shapes $\times$ 3 sizes). For NoisyShape with uniform motion, position additionally follows linear motion with elastic boundary reflection, independent of actions; position labels at each horizon are computed by simulating this dynamics forward.

C.3 DeepMind Control Suite Environments

We evaluate on six continuous control tasks from the DeepMind Control Suite [32], standard in prior distraction-robust RL work [7, 33, 4]: `Walker Walk`, `Walker Run`, `Cheetah Run`, `Cartpole Swingup`, `Ball-in-cup Catch`, and `Finger Spin`. All tasks use 64×64 RGB observations and action repeat 2. We train for 1M environment steps under Clean and VHR settings, and 3M steps under the VH setting to allow convergence under temporally coherent distractors.

Clean. Standard DMC rendering with the default background. This serves as a regression test to verify that STARWM's additional modules do not degrade performance when no distractions are present.

Video Hard Sequential (VH). The background is replaced by a natural video clip from the DMC-GB *video_hard* set that plays sequentially within each episode: each simulator step advances the video by one frame, creating a temporally coherent but uncontrollable background distraction. This setting tests robustness to structured, predictable but action-independent visual dynamics. The temporally coherent background is particularly challenging because methods that exploit temporal consistency (e.g., contrastive learning) may inadvertently attend to the background. Rendering details are described below.

Video Hard Random (VHR). Same video pool as VH, but each step independently samples a random frame of the current episode's video, destroying all temporal structure. This creates i.i.d. high-entropy distractions that are unpredictable by the dynamics model. VHR isolates the effect of spatial visual clutter from temporal distractor dynamics: any method that succeeds under VHR must be filtering distractors based on spatial or action-conditional cues rather than temporal prediction.

Rendering details. Our video background settings are based on the DMControl Generalization Benchmark [12, 2]. The original benchmark defines two difficulty levels: *video_easy*, which retains the ground plane and composites the video behind scene geometry, and *video_hard*, which removes the ground plane so that the video covers the entire background. We adopt the video_hard setting for both VH and VHR, as the absence of the ground plane prevents the agent from exploiting a fixed texture as a stable spatial reference. The backgrounds are the 100 *video_hard* clips distributed with DMC-GB (a subset of RealEstate10K). The first episode of each environment instance starts from a clip chosen by the environment's random seed, and every subsequent episode advances to the next clip of the pool. For VH, the clip is played from its first frame, one frame per simulator step, and loops if the episode exceeds the clip length. For VHR, at each step a frame of the current clip is sampled uniformly. All methods use the same wrapper and video pool. For `Reacher`, whose top-down camera does not see the backdrop, DAVIS 2017 [26] frames are instead composited behind the arm and target at pixel level (VH: a random clip played sequentially from a random start frame; VHR: a random frame of a random clip at every step).

D Full results & Aggregate Metrics

This section provides the full numerical results underlying the main text's figures and tables, including standard deviations across seeds.

D.1 Full results in NoisyShape Environment

Table 5 reports Probe accuracy on the NoisyShape diagnostic environment (§4.1) with standard deviations across 3 seeds.

Table 5: **D.1: Latent probe accuracy on Noisy Shape.** H0 = encoding step; H15 = 15 steps of open-loop imagination. Each ablation isolates one component of STARWM.

	H0 (Encoding)				H15 (Imagination)		
Model	Color	Shape	Size	Position MSE ↑	Color	Shape	Size
DreamerV3	**99.92** ±0.01	69.37 ±2.41	98.92 ±0.93	0.006 ±0.001	**99.73** ±0.11	28.95 ±1.65	87.49 ±14.10
+ dual-stream only	**100.0** ±0.00	40.01 ±1.13	51.55 ±2.49	**0.068** ±0.003	**99.98** ±0.00	25.04 ±0.10	33.31 ±0.29
+ SS only	**99.89** ±0.02	66.83 ±1.67	99.39 ±0.09	0.007 ±0.000	**99.70** ±0.14	26.65 ±0.53	92.73 ±3.44
+ dual-stream + SS	**100.0** ±0.01	43.65 ±7.38	62.20 ±9.26	**0.059** ±0.006	**99.94** ±0.03	25.27 ±0.27	33.80 ±0.54
+ attention only	**99.97** ±0.01	92.95 ±1.21	**99.92** ±0.02	**0.060** ±0.001	**99.94** ±0.01	56.22 ±4.35	**99.71** ±0.13
STARWM	**99.99** ±0.00	**99.92** ±0.02	**100.0**±0.00	**0.066** ±0.004	**100.0** ±0.00	**99.57** ±0.18	**99.99** ±0.01
R2-Dreamer	99.42 ±0.35	37.92 ±2.52	74.51 ±2.92	0.028 ±0.004	71.41 ±8.11	25.38 ±0.23	33.30 ±0.14

D.2 Full results in DMC Environments

D.2.1 Per-task results

Table 6 reports full evaluation returns with standard deviations across 3 seeds for all DMC tasks and settings. For the Clean setting, we report two DreamerV3 reference values. DreamerV3 (Original Paper) lists the scores reported in Hafner et al. [11]. DreamerV3 (PyTorch) lists the scores obtained by running the widely-used open-source PyTorch reimplementation under identical training budgets. STARWM is also built on top of this PyTorch codebase.

Table 6: **D.2.1: DMC evaluation returns. (mean $\pm$ standard deviation over 3 seeds, at the end of training)** VH: sequential video (3M steps). VHR: random video (1M steps). Best per (task, setting) in **bold**.

		Walker Walk	Walker Run	Cheetah Run	Cartpole	Finger Spin	Ball-in-cup
Clean	DreamerV3 (Original Paper)	966	797	846	843	886	975
	DreamerV3 (Pytorch)	954.40 $\pm$3.15	749.93 $\pm$10.64	894.54 $\pm$3.08	840.10 $\pm$31.89	554.74 $\pm$97.56	964.65 $\pm$1.53
	STARWM	944.06 $\pm$2.57	686.81 $\pm$39.45	762.77 $\pm$54.33	863.66 $\pm$5.62	559.35 $\pm$92.24	948.07 $\pm$9.50
VH	DreamerV3	327.39 $\pm$68.25	95.36 $\pm$40.38	73.30 $\pm$8.46	160.13 $\pm$12.17	217.50 $\pm$119.63	118.26 $\pm$23.78
	DenoisedMDP	93.14 $\pm$24.69	39.70 $\pm$2.91	60.93 $\pm$25.55	161.14 $\pm$7.17	53.96 $\pm$36.57	168.34 $\pm$46.43
	Iso-Dream++	98.16 $\pm$12.49	43.79 $\pm$6.89	28.01 $\pm$14.61	139.92 $\pm$5.77	6.35 $\pm$0.71	15.76 $\pm$4.05
	R2-Dreamer	**936.49** $\pm$8.87	471.63 $\pm$39.33	426.69 $\pm$70.67	652.00 $\pm$82.78	**600.12** $\pm$50.34	**358.53** $\pm$405.18
	STARWM	904.66 $\pm$33.78	378.60 $\pm$190.65	**452.77** $\pm$22.76	308.50 $\pm$10.44	464.76 $\pm$28.24	114.21 $\pm$27.46
	STARWM-R	856.98 $\pm$98.32	**502.63** $\pm$146.12	405.28 $\pm$53.30	**770.81** $\pm$52.29	390.72 $\pm$25.92	303.74 $\pm$99.62
VHR	DreamerV3	517.48 $\pm$68.05	191.37 $\pm$2.64	212.62 $\pm$45.31	143.89 $\pm$3.23	390.74 $\pm$21.53	23.12 $\pm$11.97
	DenoisedMDP	183.85 $\pm$79.73	90.49 $\pm$20.30	54.64 $\pm$35.79	141.36 $\pm$18.50	44.08 $\pm$60.37	137.53 $\pm$52.80
	Iso-Dream++	209.89 $\pm$41.89	92.45 $\pm$24.10	56.02 $\pm$53.80	122.07 $\pm$7.67	95.86 $\pm$134.01	52.18 $\pm$38.04
	R2-Dreamer	158.39 $\pm$16.06	56.80 $\pm$4.20	150.12 $\pm$30.88	139.85 $\pm$6.57	97.99 $\pm$17.11	239.40 $\pm$34.30
	STARWM	851.24 $\pm$51.07	435.41 $\pm$73.46	**468.81** $\pm$44.04	562.96 $\pm$50.04	414.15 $\pm$70.40	97.64 $\pm$35.66
	STARWM-R	**872.66** $\pm$23.07	**536.26** $\pm$30.91	432.32 $\pm$41.24	**650.64** $\pm$102.31	**459.80** $\pm$25.19	**302.88** $\pm$33.24

D.2.2 Aggregate Statistical Analysis

To complement the per-task learning curves and tables in Appendices D.2.1, we report aggregate performance using the reliable RL evaluation framework of Agarwal et al. [1]. All scores are normalized by DreamerV3's clean-environment performance on each task as a recovery ratio, then aggregated across the six DMC tasks and three random seeds (18 runs per method per setting). We report four point estimates, namely Median, Interquartile Mean (IQM), Mean, and Optimality Gap, each with 95% stratified bootstrap confidence intervals (Figure 10a); performance profiles giving the fraction of runs exceeding each score threshold (Figure 10b); and pairwise probability-of-improvement heatmaps $P(A > B)$ (Figure 10c). We provide each visualization separately for VH, VHR, and the combined VH+VHR pool.

Sequential video (VH). Under temporally coherent backgrounds, R2-Dreamer attains the highest aggregate scores across all four point estimates, with STARWM-R second and default STARWM third; the remaining baselines (DreamerV3, DenoisedMDP, Iso-Dream++) cluster at low IQM with large optimality gaps. The performance profile shows R2-Dreamer dominating STARWM across most score thresholds, while STARWM-R closes much of this gap. The pairwise heatmap reflects the same picture: $P(\text{STARWM-R} > \text{R2-Dreamer}) = 0.46$, statistically near the indifference threshold of 0.5, while $P(\text{STARWM} > \text{R2-Dreamer}) = 0.22$. As discussed in §4.2, the temporally coherent backgrounds in VH align well with R2-Dreamer's redundancy-reduction objective and partially co-opt STARWM's contrastive guidance; reward prediction in STARWM-R substantially mitigates this gap.

Random video (VHR). Destroying the temporal coherence of the background reverses the ranking. STARWM and STARWM-R lead on every aggregate metric, while R2-Dreamer collapses to scores comparable to DreamerV3 and the latent-decomposition baselines. The performance profile shows both STARWM variants stochastically dominating all other methods across the full range of τ. The

pairwise heatmap confirms strong dominance: $P(\text{STARWM-R} > B) = 1$ for every non-STARWM baseline B, and $P(\text{STARWM} > B) \geq 0.85$ for every baseline except STARWM-R. R2-Dreamer's redundancy-reduction objective offers little robustness when the temporal structure it relies on is destroyed.

Cross-condition pool (VH+VHR). Pooling all 36 runs per method gives the cleanest measure of cross-condition robustness, since methods specialized to one distractor type are penalized for failing on the other. STARWM-R attains the highest aggregate score on all four point estimates and the lowest optimality gap, and stochastically dominates every baseline in the performance profile. STARWM is second on IQM, Mean, and Optimality Gap, behind only STARWM-R. R2-Dreamer's pool-level performance reflects its asymmetry: strong VH offset by weak VHR places it close to DreamerV3 in the IQM and Mean panels. The pairwise heatmap quantifies this: $P(\text{STARWM-R} > B) \geq 0.73$ for every non-STARWM baseline, and $P(\text{STARWM-R} > B) \geq 0.54$ except STARWM-R. Together, these pool-level results support the claim made in §4.2 that STARWM and STARWM-R are the only methods robust across both distractor types.

E Ablation Studies & Core Mechanistic Analysis

E.1 Reward as Additional Attention Routing Signal

The default STARWM configuration uses inverse dynamics and contrastive learning to guide attention. Both signals target action-conditional or temporally predictable features. However, some task-relevant entities are neither directly action-conditional nor temporally distinctive: in `Ball-in-cup Catch`, the ball moves as a passive consequence of the cup's motion, and in `Cartpole Swingup`, the pole's angle is reward-critical but only indirectly controlled. Inverse dynamics provides weak gradients for such entities because their visual changes are not directly explained by the agent's action at a single timestep.

To address this, we augment the attention guidance with a reward prediction head that operates on the attention-pooled features e_t^{attn}:

$$\mathcal{L}_{\text{reward}}^{\text{attn}} = \|\text{MLP}_{\text{rwd}}(e_t^{\text{attn}}) - r_t\|^2, \tag{12}$$

with gradients flowing through the live attention map A_t into the attention parameters, analogously to the inverse dynamics and contrastive losses. This directly incentivizes attention to cover regions whose visual content is informative of reward.

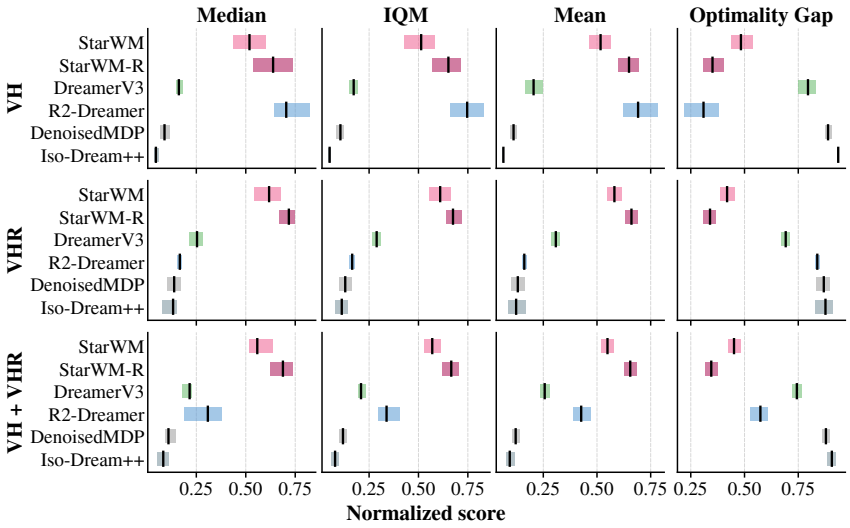
We report the results in the Figure 5 and the Table 6.

Reward prediction produces substantial improvements on tasks involving passive objects. On `Ball-in-cup Catch VHR`, the reward-augmented variant improves from 98 to 303, surpassing R2-Dreamer (239). On `Cartpole Swingup VH`, it improves from 308 to 771, also surpassing R2-Dreamer (652). In both cases, the passive object (ball, pole) is reward-critical but only indirectly action-conditional, and reward prediction provides the attention signal that inverse dynamics alone cannot.

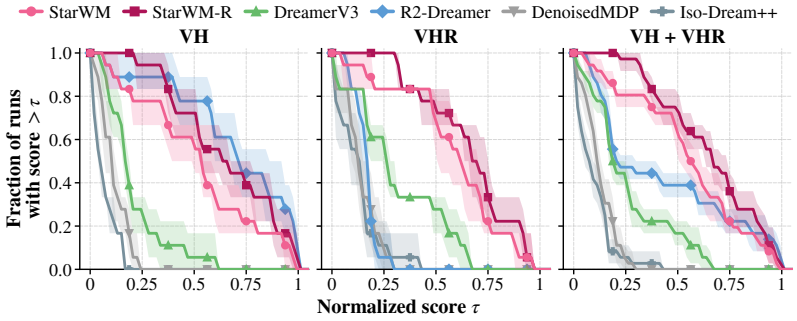
These results demonstrate that STARWM's architecture is agnostic to the specific attention loss: the gradient isolation mechanism ensures that any differentiable signal applied to the live attention path trains the routing without interfering with reconstruction. Reward prediction is one instance of a domain-specific signal that can be composed with or substituted for the default self-supervised objectives. The choice of attention loss thus becomes a modular design decision, selectable based on task characteristics: inverse dynamics for directly controlled entities, contrastive learning for temporally coherent entities, and reward prediction for passively task-relevant entities.

E.1.1 Case Analysis: Attention Drift on `Cartpole Swingup VH`

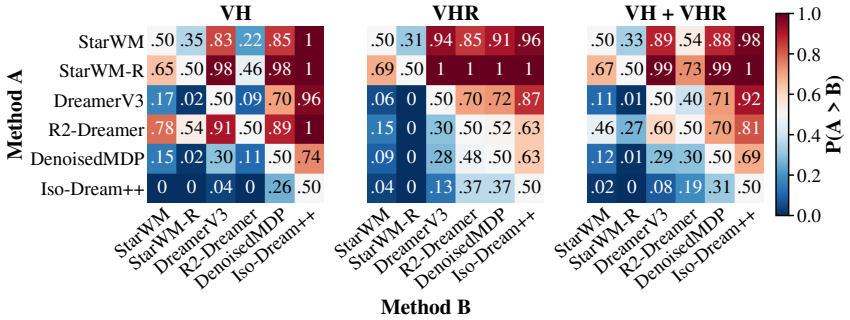
`Cartpole Swingup VH` is the task where STARWM most strongly underperforms STARWM-R. Figure 11 visualizes attention maps for default STARWM and STARWM-R on `Cartpole Swingup VH`. Default STARWM's four queries concentrate near the left and right endpoints of the track rather than on the cart itself, and exhibit little specialization across queries. STARWM-R locates the cart and pole accurately, with queries specializing to distinct object regions.



(a) Aggregate point estimates with 95% stratified bootstrap confidence intervals. Higher Median/IQM/Mean and lower Optimality Gap indicate stronger performance. Scores are normalized by DreamerV3 clean-environment performance.



(b) Performance profiles. Each curve plots the fraction of runs achieving normalized score above τ; curves further to the upper right indicate stronger and more reliable performance. Shaded regions show 95% stratified bootstrap confidence bands.



(c) Pairwise probability of improvement $P(A > B)$, computed as the fraction of (run-of-A, run-of-B) pairs in which A achieves a strictly higher score. Values above 0.5 indicate A is more likely to outperform B on a randomly drawn pair of runs.

Figure 10: **D.2.2: Aggregate statistical analysis on DMC under video distractors,** following Agarwal et al. [1]. Each row reports VH, VHR, and the combined VH+VHR pool, computed over six DMC tasks and three random seeds (18 runs per method per setting; 36 in the pool).

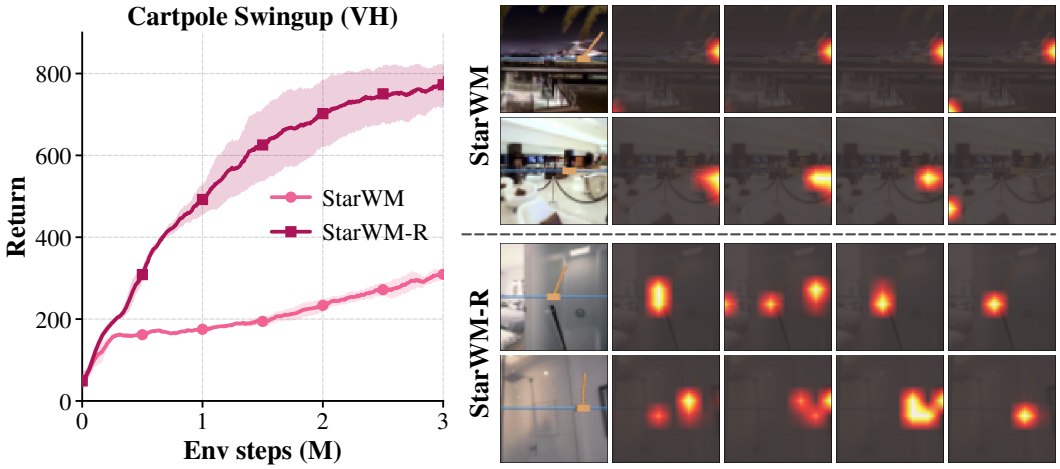


Figure 11: **E.1.1: Default STARWM's attention drifts on Cartpole VH; reward prediction recovers it.** Each row shows the input observation followed by the attention maps for the four queries. Default STARWM (top right) concentrates attention near the track endpoints without query specialization. STARWM-R (bottom right) localizes the cart and pole accurately, with queries specializing to distinct object regions.

Several factors plausibly contribute to this failure: the small size of the cart-and-pole assembly relative to the 8×8 attention resolution, the temporal coherence of the sequential video background that competes with the cart for the contrastive objective's attention, and the lack of corrective signal from inverse dynamics in this configuration. When self-supervised proxies (controllability, temporal coherence) do not point unambiguously at the task-relevant object, attention can drift to other plausible regions. This pattern motivates the reward-augmented variant STARWM-R (Appendix E.1): when self-supervised signals underdetermine the routing target, reward provides a direct task-relevance signal that breaks the ambiguity. The same mechanism explains the `Ball-in-cup` improvement, where the passive ball is weakly supervised by inverse dynamics until reward fills the gap. Practitioners can consider adding reward prediction whenever the default configuration's attention maps appear to drift or fail to specialize on a particular task.

E.2 Gradient Isolation Analysis

The stop-gradient operator in STARWM serves two roles: blocking reconstruction gradients from the decoder pathway (*detach B*: $\bar{A}_t$ used in inverse broadcasting and spatial composition) and blocking RSSM gradients from the dynamics pathway (*detach A*: $\bar{A}_t$ used in computing entity tokens e_t^{dyn}); see Figure 8 for where we apply detach. We evaluate configurations that remove these independently, with and without self-supervised attention guidance (SS), across environments of different distractor characteristics. The five configurations and the signals reaching the attention parameters in each are summarized in Table 7.

Table 7: **E.2: Configurations evaluated in the gradient isolation analysis.** Each row indicates which gradient pathways reach the attention parameters. detach A blocks RSSM gradients (via $\bar{A}_t$ in entity tokens); detach B blocks reconstruction gradients (via $\bar{A}_t$ in the dual-stream decoder); SS denotes self-supervised attention guidance (inverse dynamics + contrastive). A check mark indicates the gradient or signal reaches the attention parameters.

Symbol	Configuration	SS	RSSM grad.	Recon. grad.
★	STARWM (full)	✓	—	—
☆1	no detach A/B, with SS	✓	✓	✓
☆2	no detach A/B, no SS	—	✓	✓
☆3	only detach B, no SS	—	✓	—
☆4	only detach B, with SS	✓	✓	—

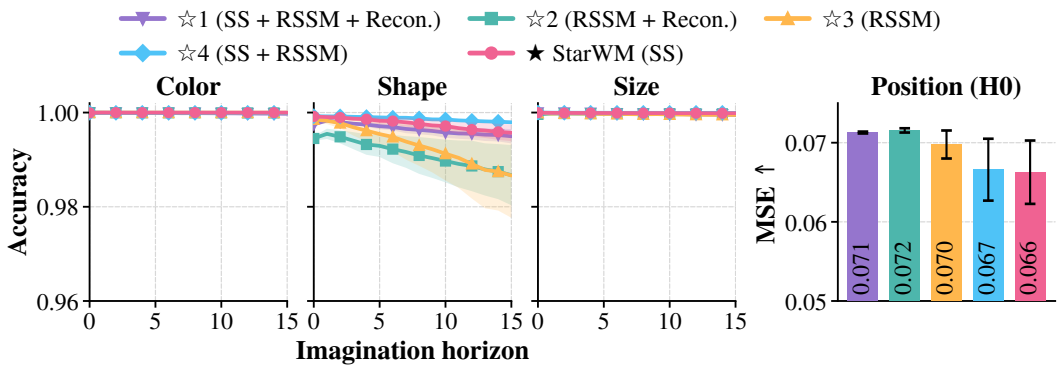


Figure 12: **E.2: Probe accuracy on NoisyShape under five gradient isolation configurations.** NoisyShape probe accuracy is invariant to gradient isolation choice.

Table 8: **E.2: Gradient isolation ablation on DMC.** Episode return (mean over 3 seeds) for Walker Walk and Cheetah Run under Clean and VHR settings. Configurations are defined in Table 7. Best per column in bold.

		Walker Walk		Cheetah Run	
Symbol	**Configuration**	Clean	VHR	Clean	VHR
★	StarWM (full)	**952.37** ±3.06	**872.94** ±50.99	752.37 ±34.84	**466.03** ±53.00
☆1	no detach A/B, with SS	810.66 ±157.94	181.08 ±17.72	641.66 ±52.40	181.75 ±31.91
☆3	only detach B, no SS	943.42 ±18.18	135.02 ±20.74	**880.96** ±6.10	199.37 ±94.74

All configurations succeed on NoisyShape. We first evaluate all five configurations in Table 7 on NoisyShape. Figure 12 shows that shape, color, and size retention at H_{15} remain near optimal across all configurations: whether attention is trained by self-supervised signals alone, by reconstruction and RSSM gradients alone, or by any combination, the resulting attention reliably localizes the foreground shape. This invariance is consistent with the analysis in §4.3: in NoisyShape, the easiest object to reconstruct (a geometric shape on a stochastic background) through RSSM pathway coincides with the action-conditional and temporally predictable object, so reconstruction gradients and self-supervised signals push attention toward the same target. The architecture's structural capacity for spatial routing is sufficient on its own; gradient isolation does not become necessary until reconstruction gradients and self-supervised signals start to disagree.

Detach B is critical under visual distractors. On DMC with video backgrounds, the alignment between reconstruction gradients and self-supervised signals breaks down: the visually dominant region (the video background) is no longer the same as the action-conditional region (the agent body). We evaluate three configurations from Table 7 that test the necessity of each component: ★: full StarWM (SS as the sole attention signal), ☆1: "no detach A/B with SS" (SS competing with both RSSM and reconstruction gradients on the attention path), and ☆3: "only detach B, no SS" (RSSM gradient as the sole attention signal). Table 8 reports evaluation returns on Walker Walk and Cheetah Run under Clean and VHR settings.

Removing detach B (no detach A/B with SS; ☆1) causes substantial degradation under VHR despite the presence of self-supervised signals. With reconstruction gradients reaching the attention parameters, the dense pixel loss pushes attention toward spatial coverage that minimizes reconstruction error, dragging attention toward the visually dominant background. SS gradients alone cannot counter this pull when reconstruction gradients are not blocked.

Removing SS while keeping detach B (only detach B, no SS; ☆3) also fails under VHR. Without explicit dynamics-grounded guidance, RSSM gradients are the sole attention signal; while RSSM does favor predictable features, this preference alone cannot reliably distinguish a controllable foreground from a temporally coherent background when both are reconstructable. Notably, this configuration performs comparably to or even better than full StarWM on Clean environments, where there is

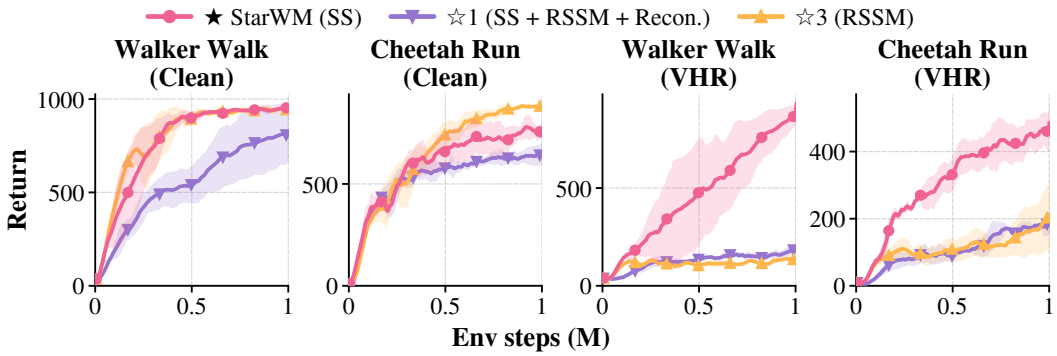


Figure 13: **E.2: Learning curves for the gradient isolation ablation on DMC.** Episode return over training for the three configurations of Table 8. Removing detach B causes attention to be co-opted by reconstruction gradients; removing SS leaves RSSM gradients as the only attention signal, insufficient to disambiguate foreground from background under video distractors. Full STARWM combines detach B with SS guidance.

Table 9: **E.2: Attention map statistics under three gradient isolation configurations on Cheetah Run.** Average entropy and maximum attention weight of query Q_0, computed over evaluation trajectories. Lower entropy and higher max weight indicate sharper, more focused attention. The maximum possible entropy is $\log(64) \approx 4.16$ (uniform over the 64 spatial tokens), and the uniform floor for max attention is $1/64 \approx 0.016$.

		Clean		VHR	
Symbol	**Configuration**	Entropy ↓	Max ↑	Entropy ↓	Max ↑
★	STARWM (full)	**0.14**	**0.92**	**0.15**	**0.94**
☆1	no detach A/B, with SS	3.90	0.04	4.15	0.02
☆3	only detach B, no SS	3.12	0.14	1.27	0.52

no background distractor to disambiguate, and adding SS introduces minor optimization overhead. Under distractors, however, the SS guidance becomes essential.

The two failure modes together establish that detach B and SS are jointly necessary, not redundant: detach B prevents reconstruction from co-opting the attention parameters, and SS provides the dynamics-grounded signal that correctly steers them. The cross-attention architecture provides the structural capacity for spatial routing, but the choice of what routes the attention requires both elements together when reconstruction and attention objectives compete.

Attention maps reveal the mechanism. To directly observe how each gradient pathway shapes attention behavior, we visualize the learned attention maps and quantify them via two statistics: the entropy of the attention distribution (lower = sharper) and the maximum attention weight (higher = more localized). Figure 14 shows attention maps for the four learned queries on Walker Walk, and Table 9 reports the corresponding quantitative statistics for Q_0 on Cheetah Run.

☆1 (no detach A/B with SS) produces nearly uniform attention. When all three gradient pathways (SS, RSSM, reconstruction) reach the attention parameters simultaneously, the dense pixel loss dominates: it pulls attention toward broad spatial coverage, defeating the routing mechanism entirely. Queries lose specialization, and the architecture's spatial bottleneck is bypassed.

☆3 (Only detach B, no SS) produces moderately structured attention. RSSM gradients alone favor temporally predictable regions, but this signal is too weak to specialize the four queries: under video distractors, only one of the four queries reliably localizes the agent, while the remaining three drift across various background regions. RSSM gradients identify some predictable region but cannot disambiguate among the multiple predictable regions present in a structured-video scene. On Clean Cheetah, though attention is even less focused, all of queries localize on the agent. This explains

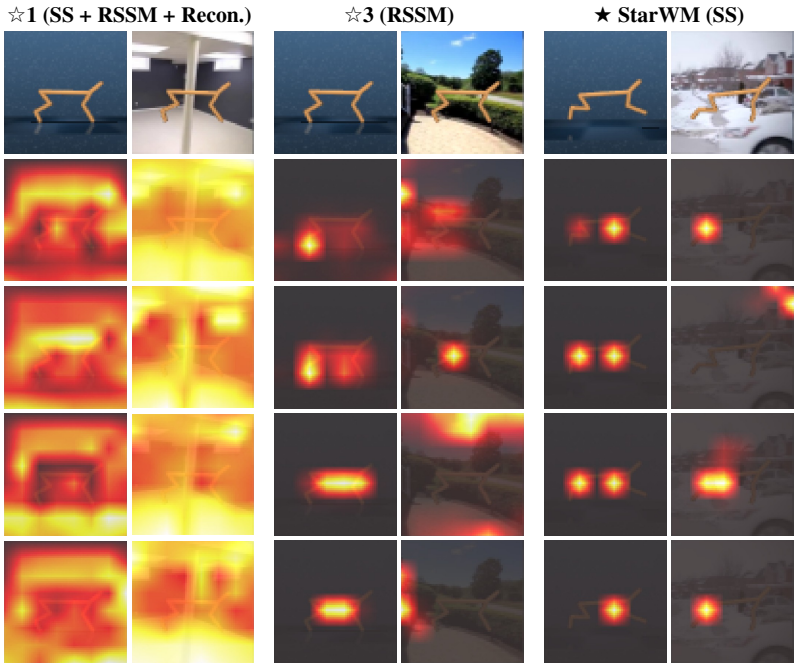


Figure 14: **E.2: Gradient isolation determines attention focus on DMC.** Attention maps from the four learned queries on Walker Walk under VHR, comparing the three configurations of Table 8. Top row: input observation. Rows 2–5: attention map for each of the four queries. Left (☆1: SS+RSSM+Recon., no detach): attention collapses to a nearly uniform distribution. Middle (☆3: RSSM only, detach B without SS): attention shows weak focus (1/4 queries specializing to agent body in VH). Right (★ STARWM: SS with detach A/B): attention is sharply localized on the agent's body parts (3/4 queries specializing to agent body in VH).

why this configuration's Clean Cheetah return exceeds STARWM. When there is nothing to filter, defocused attention loses no information and may benefit from broader feature coverage.

Full STARWM produces sharply focused attention. The attention sharpness on Clean and VHR is nearly identical, indicating that SS guidance with stop-gradient isolation produces a routing decision that is independent of the visual complexity of the background. Equally important, three of the four queries reliably specialize to distinct agent body regions under video distractors, with only one query attending to the background. Compared to 1:3 under RSSM-only guidance, this directly reflects the role of self-supervised signals.

E.3 Hyperparameter Sensitivity

Default configuration. All hyperparameters remain fixed across every evaluated DMC task and distractor regime without per-task tuning. We set self-supervised loss weights to $\lambda_{\text{inv}} = \lambda_{\text{ctr}} = 1.0$ with temperature $\tau = 0.1$, and apply a fixed β_{bg} annealing schedule across all experiments. The sweeps reported below evaluate stability under post-hoc variations rather than serving as a parameter search.

Loss weights and KL schedule perturbations. We evaluate single-parameter perturbations on Walker Walk VHR at 1M steps across 3 seeds.

Across all variations, performance remains solid between 787 and 925 (Table 10). Removing β_{bg} annealing entirely yields 918, showing that training does not depend strictly on schedule design. Higher penalties ($\lambda_{\text{inv}} = 3.0$ and $\beta_{\text{bg}}^{\text{final}} = 2.0$) slow convergence speed within the 1M budget, but all configurations remain functional without diverging.

Table 10: **E.3: One-at-a-time hyperparameter perturbations** (STARWM, Walker Walk VHR, matched 1M steps, mean ± SE, 3 seeds).

Perturbation	Return @ 1M
$\lambda_{\text{inv}} = 0.3$ (3× reduction)	**925** ±1
β_{bg}: annealing removed entirely	918 ±12
$\lambda_{\text{inv}} = 3.0$ (3× increase)	856 ±34
$\beta_{\text{bg}}^{\text{final}} = 2.0$ (4× increase)	787 ±77

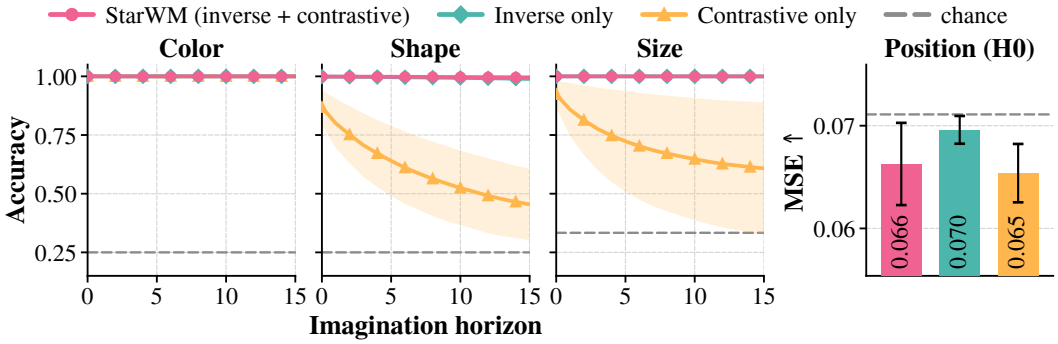


Figure 15: **E.3.1: Inverse dynamics is the primary attention signal.** Probe accuracy on the RSSM latent for full STARWM (inverse + contrastive), inverse-only, and contrastive-only variants on NoisyShape. Inverse dynamics alone is sufficient for near-perfect attribute retention; contrastive alone fails to localize the foreground.

E.3.1 Decomposition of Self-Supervised Objectives (Inv vs. Ctr)

STARWM's default attention guidance combines inverse dynamics and contrastive learning. While Appendix E.2 established that gradient isolation and self-supervised guidance are jointly necessary, this section asks a finer question: how do the two self-supervised signals divide the work? We characterize each signal's contribution on NoisyShape and across DMC, and identify a regime where contrastive learning introduces a partial conflict.

On NoisyShape, we ablate the two attention guidance signals independently: inverse dynamics alone, contrastive alone, and the full combination. Figure 15 reports probe accuracy on the RSSM latent. Full STARWM and the inverse-dynamics-only variant achieve indistinguishable performance across all attribute probes (color, shape, size at H_{15}), while the contrastive-only variant fails to guide attention toward the foreground: shape and size retention degrades through imagination. This contrast reveals the qualitative difference between the two signals. Inverse dynamics provides direct supervision: the gradient points to regions whose visual changes are explained by the agent's action, an unambiguous target. Temporal contrastive learning provides only indirect supervision: it rewards features that are temporally consistent. In a randomized-background environment like NoisyShape, this indirect signal is too weak to specify the correct target.

On DMC, we explore whether the contrastive signal is indeed helpful. Figure 16 compares full STARWM with the inverse-only variant across Clean, VH, and VHR settings.

Under VHR, contrastive learning is clearly beneficial: removing it substantially degrades performance. With independently sampled background frames at every step, the only feature with reliable temporal coherence is the agent itself. Contrastive learning exploits exactly this signal, providing dense temporal supervision in transitions where the agent's action is small or nearly action-independent (and thus where inverse dynamics gradient is weak). The two signals act complementarily: inverse dynamics identifies controllable regions, contrastive identifies temporally coherent regions, and under VHR these two sets coincide on the agent.

Under Clean and VH, the two configurations perform comparably.The inverse-only variant is even slightly better than full STARWM under VH. This is the regime where contrastive learning introduces a partial signal conflict: video frames played sequentially produce a temporally coherent background,

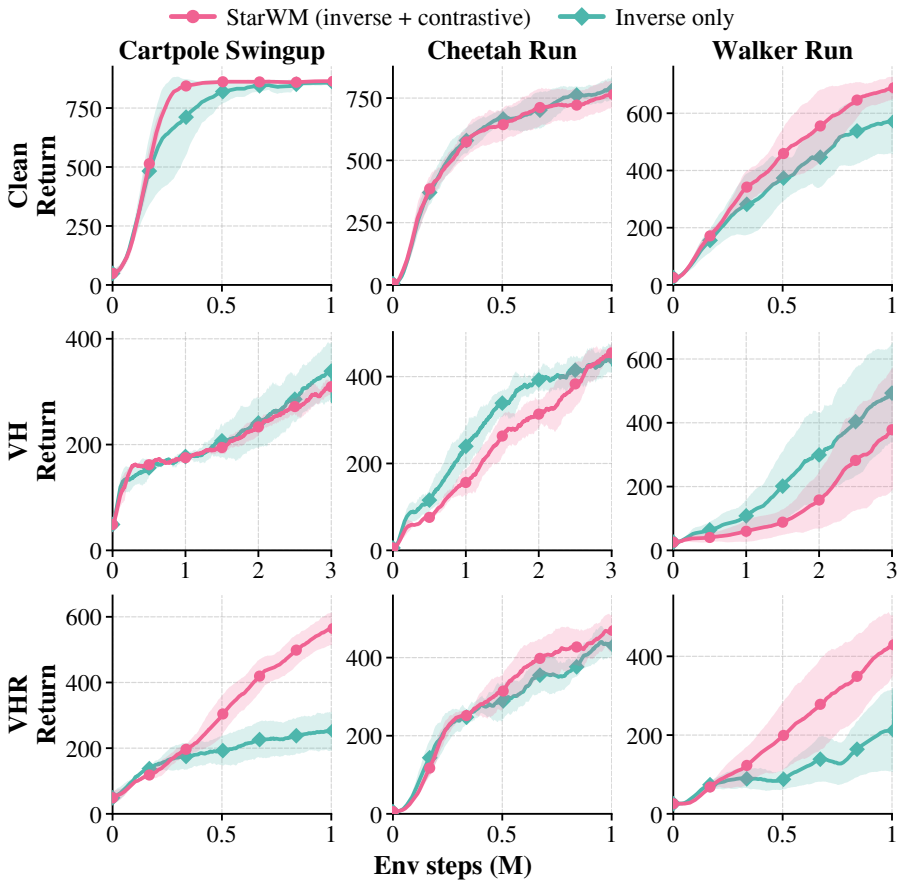


Figure 16: **E.3.1: Episode return for full STARWM (inverse + contrastive) versus inverse-only across Clean, VH (sequential video), and VHR (random video).** Contrastive supervision is essential under VHR, neutral under Clean and under VH.

satisfying the contrastive objective just as well as the agent does. Contrastive gradients then split ambiguously between agent and background, slightly diluting the otherwise correct routing established by inverse dynamics.

The fundamental tension: predictability without task information. The VH result reflects an epistemic limitation that is not specific to contrastive learning, but inherent to the self-supervised setting itself. When a background is temporally predictable, it is genuinely ambiguous whether the agent should encode it. In a task where reward depends on a moving landmark, the predictable background is task-relevant and should be retained; in a task where the agent must navigate around a distractor that happens to move predictably, it should be filtered. The agent has no way to tell these cases apart from self-supervised observation alone.

The weight of contrastive learning therefore encodes a prior held by the designer rather than a fact recoverable from the environment. A high weight reflects a prior that predictable structure should be preserved; a low weight reflects a prior that it should be filtered. This is a structural feature of self-supervision rather than a defect of the contrastive objective. Resolving it requires information beyond observation dynamics, such as a reward signal. We discuss in Appendix E.1 how reward prediction can be added as an additional attention-guidance signal that breaks the ambiguity.

E.3.2 Query count N.

We evaluate the number of routing queries $N \in \{1, 2, 4, 8\}$ on three DMC tasks under Video Hard Sequential (VH) distractors at 3M steps (Table 11 and Figure 17).

Table 11: **E.3: Effect of query count N on DMC VH** (episode return at 3M steps, mean ± std, 3 seeds).

N	Walker Walk	Cheetah Run	Cartpole Swingup
1	170.48 ±24.59	181.22 ±63.81	191.54 ±6.46
2	**928.73** ±34.39	**461.71** ±15.09	268.62 ±60.34
4	902.31 ±37.20	452.67 ±21.67	308.00 ±12.04
8	585.14 ±72.84	387.97 ±78.67	**394.90** ±98.07

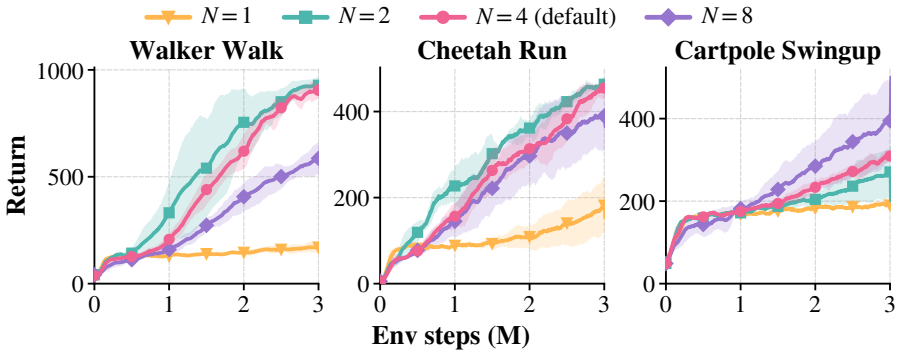


Figure 17: **E.3: Return curves across varying query counts N on DMC VH** (mean ± SE, 3 seeds).

A single query (N=1) underperforms across all tasks because a single attention mask cannot resolve distinct parts of an articulated body simultaneously. Increasing to N=2 and N=4 provides sufficient expressive capacity, achieving strong returns across tasks. Setting N=8 improves Cartpole Swingup but adds variance on Walker Walk. We use N=4 as a balanced default across all benchmarks.

E.4 Selective Spatial Encoding Analysis

The main NoisyShape experiments (Figure 3 and Table 5) randomize position independently at each step, making it a purely non-predictable attribute that an ideal world model should discard. A natural concern is whether STARWM's low position probe accuracy (MSE = 0.066, near chance) reflects a correct identification of position as an irrelevant feature, or an architectural inability to encode spatial information at all. To disambiguate, we construct a variant in which position follows a deterministic trajectory and evaluate whether STARWM's latent correctly encodes position in this setting. The results are shown in Figure 18 and Table 12.

Probing protocol. We follow the same probe protocol as Appendix C.2, extended with two additional probes: position (regression, MSE) and background ID (20-way classification, chance = 0.05). All numbers are mean ± standard deviation across 3 seeds.

STARWM encodes predictable position. In contrast to the random-position setting where STARWM retains position only coarsely (MSE ≈ 0.066, near chance), under predictable motion STARWM's position MSE drops by two orders of magnitude to ∼ 0.0008 at encoding and remains well below chance after 15 imagination steps. This confirms that the architecture does not blindly discard spatial information: when position is dynamically predictable, it is preserved by the RSSM latent. The slight gap to DreamerV3 at H15 reflects a reasonable trade-off, as STARWM routes spatial information through a bottleneck of $N = 4$ attention queries rather than a dense feature map, introducing mild quantization at the price of filtering objective-irrelevant content.

Background ID separates the two methods cleanly. The background ID probe provides a direct readout of objective-irrelevant contamination. DreamerV3's latent classifies the background with 82.4% accuracy over 20 classes, confirming that its dense reconstruction objective forces the encoder to preserve background identity even when the background carries no predictive information. STARWM's latent achieves 5.0% (exactly chance level), demonstrating that the dual-stream archi-

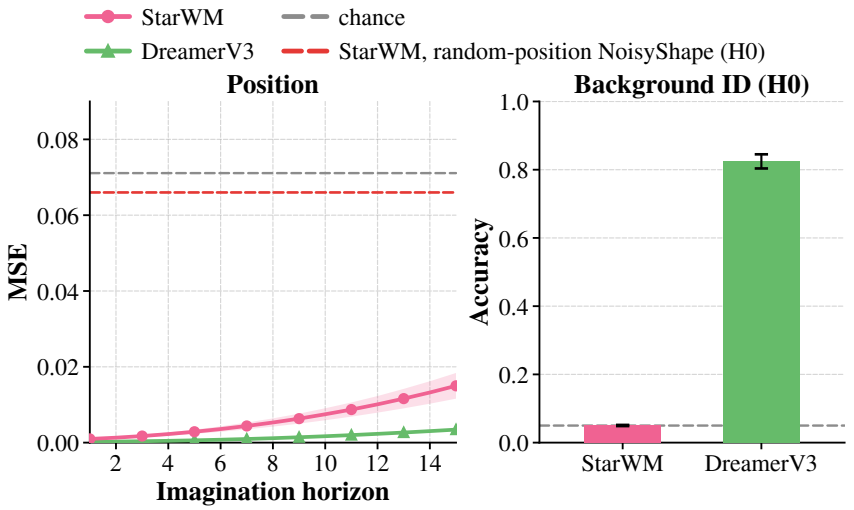


Figure 18: **E.4: Latent space probing under predictable object motion.** Left: Position MSE across a 15-step imagination horizon. When position follows a deterministic trajectory, STARWM successfully tracks and rolls out spatial coordinates well below chance level. Right: Background identification accuracy at the encoding step (H0). DreamerV3 suffers from severe objective-irrelevant contamination, while STARWM perfectly drops background identity. The grey dashed line in the left panel is the chance level of the random-position NoisyShape (0.0711), shown for comparison with the red dashed line (STARWM on random-position NoisyShape); the chance level of this variant is 0.135 (Table 12).

Table 12: **E.4: Probe accuracy under predictable position.** Position follows uniform linear motion with boundary reflection. Background ID is a 20-way classification over the 20 background frames of the pool. The position chance level is that of this variant (positions uniform on $[0.05, 0.95]^2$). H0 = encoding; H15 = 15 steps of open-loop imagination.

Stage	Model	Color	Shape	Size	Position MSE	BG ID ↓
H0 (Encoding)	DreamerV3	100.00 ±0.00	99.70 ±0.03	99.96 ±0.04	0.0003 ±0.0002	82.44 ±2.07
	STARWM	100.00 ±0.00	99.88 ±0.06	99.99 ±0.01	0.0008 ±0.0002	5.02 ±0.17
H15 (Imagination)	DreamerV3	99.98 ±0.03	97.46 ±1.56	99.99 ±0.00	0.0034 ±0.0002	—
	STARWM	100.00 ±0.00	99.71 ±0.18	99.99 ±0.01	0.0150 ±0.0046	—
Chance level		25.00	25.00	33.33	0.135	5.00

tecture successfully routes background content to the auxiliary VAE stream. This holds regardless of position predictability, indicating that STARWM's filtering is driven by the architectural routing rather than by the motion pattern of the specific entity.

Summary. The two NoisyShape variants together establish the correct dual behavior: STARWM discards position when it is irrelevant and encodes position when it is relevant. DreamerV3 encodes position in both settings, and encodes background identity in both settings.

E.5 Progressive Distraction Analysis across NoisyShape Variants

To understand how irrelevant visual complexity affects latent representations, we compare probe accuracy across NoisyShape variants of increasing difficulty, plus a distraction-free baseline. The results are shown in Figure 19 and Table 13.

DreamerV3 degrades progressively with distraction complexity. The Shape (fixed) baseline confirms that DreamerV3's architecture is fully capable of encoding objective-relevant attributes: with no position variation and no background, all probes achieve 100% accuracy through H15. However,

Table 13: **E.5: Probe accuracy across NoisyShape variants.** H0 = encoding step; H15 = 15-step open-loop imagination. All values: mean ± std over 3 seeds. Accuracies in %, position in MSE.

Environment	Model	Color	Shape	Size	Pos. MSE ↑
H0 (Encoding)					
Shape (fixed)	DreamerV3	100.00	100.00	100.00	—
Shape (clean)	DreamerV3	99.99 ±0.01	99.87 ±0.05	99.96 ±0.01	0.002
	STARWM	100.00 ±0.00	100.00 ±0.00	100.00 ±0.00	0.072
NoisyShape	DreamerV3	99.92 ±0.01	69.37 ±2.41	98.92 ±0.93	0.006
	STARWM	99.99 ±0.00	99.92 ±0.02	100.00 ±0.00	0.066
NoisyShape (Seq)	DreamerV3	99.88 ±0.05	78.02 ±5.63	98.94 ±0.18	0.005
	STARWM	100.00 ±0.00	99.82 ±0.04	99.99 ±0.00	0.070
Chance level		25.00	25.00	33.33	0.071
H15 (Imagination)					
Shape (fixed)	DreamerV3	100.00	100.00	100.00	—
Shape (clean)	DreamerV3	99.93 ±0.06	67.63 ±6.60	92.48 ±3.79	—
	STARWM	100.00 ±0.00	99.98 ±0.01	99.99 ±0.01	—
NoisyShape	DreamerV3	99.73 ±0.11	28.95 ±1.65	87.49 ±14.10	—
	STARWM	100.00 ±0.00	99.57 ±0.18	99.99 ±0.01	—
NoisyShape (Seq)	DreamerV3	93.88 ±5.11	27.00 ±0.87	81.25 ±3.36	—
	STARWM	100.00 ±0.00	99.63 ±0.15	99.99 ±0.00	—
Chance level		25.00	25.00	33.33	—

performance degrades as irrelevant complexity increases. In Shape (clean), where the only irrelevant factor is random position, shape H15 already drops to 67.6%. Adding random video backgrounds (NoisyShape) further reduces shape H0 from 99.9% to 69.4% and H15 to 29.0%. This progressive collapse demonstrates that DreamerV3's dense reconstruction objective forces the RSSM to allocate representational capacity to objective-irrelevant content, and this allocation scales with the visual complexity and temporal structure of the distraction.

STARWM is invariant to distraction complexity. Across all three distraction levels, STARWM maintains near-perfect probe accuracy: shape H15 $\geq 99.6\%$, size H15 $\geq 99.99\%$, and color H15 $= 100.0\%$. The consistency from Shape (clean) through NoisyShape (Seq) confirms that the dual-stream architecture with attention-based routing effectively isolates objective-relevant from objective-irrelevant content regardless of the distraction's visual complexity or temporal structure.

Position MSE confirms objective-irrelevant filtering. In all variants with random position, STARWM's position MSE remains near the chance level, while DreamerV3 encodes position with high precision.

F Stress Testing, Generalization & Boundary Cases

F.1 Spatial Granularity & Small Objects

Our default attention resolution of 8×8 spatial tokens (Appendix B.1) balances spatial granularity with the computational cost of cross-attention. As flagged in §6, this resolution may be insufficient when task-relevant objects occupy very few pixels. To test this directly, we rescale Cheetah Run and Walker Walk so that the agent's body occupies approximately one quarter of the pixel area in the default rendering (i.e., linear half-size), and re-evaluate STARWM, DreamerV3, and R2-Dreamer under Clean, VH, and VHR settings.

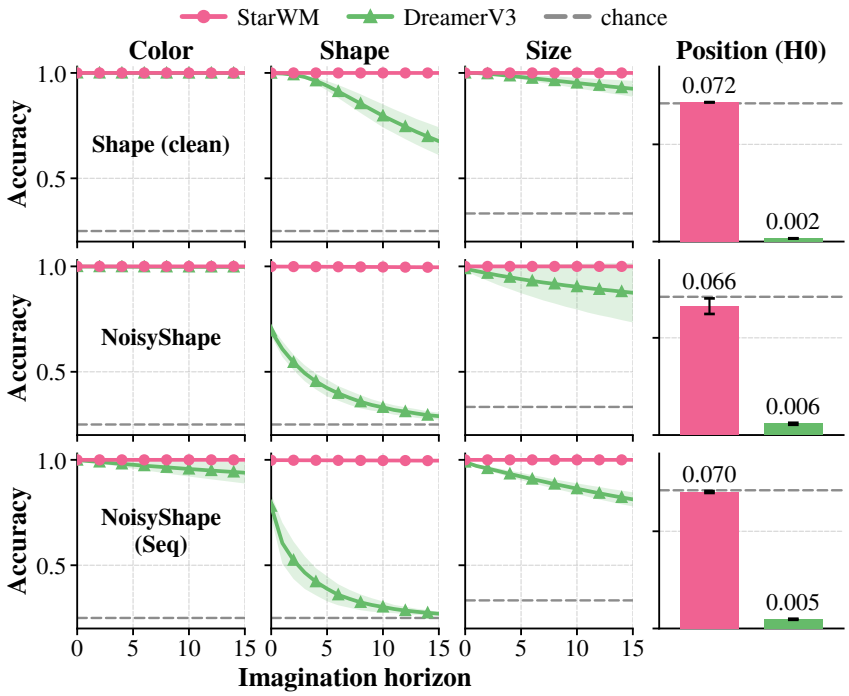


Figure 19: **E.5: Latent space probing across NoisyShape variants.** DreamerV3's performance degrades as irrelevant complexity increases, while STARWM is invariant to distraction complexity.

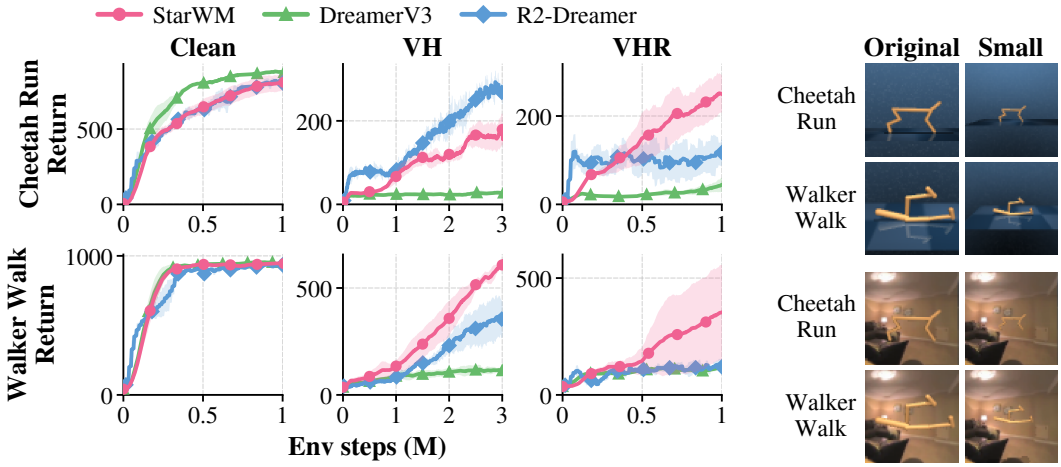


Figure 20: **F.1: Small-object stress test.** ***Left:*** Learning curves for Cheetah Run and Walker Walk with the agent rescaled to half its default linear size (one quarter pixel area), under Clean, VH (3M steps), and VHR (1M steps) settings. ***Right:*** Illustration of the environment setup.

In clean environments, STARWM and DreamerV3 reach returns comparable to the original-scale clean setting, indicating that smaller object size alone does not prevent learning. Under distractors, all three methods degrade relative to their original-scale numbers; for example, STARWM's VHR Walker Walk return drops from approximately 850 to 350. The shared degradation across methods suggests a regime where the visual signal-to-distractor ratio approaches the limits of these architectures, rather than a STARWM-specific failure mode.

The relative ordering between methods is largely preserved. Under VHR, STARWM continues to lead on both tasks despite the reduced absolute returns. Under VH, the result is split: STARWM

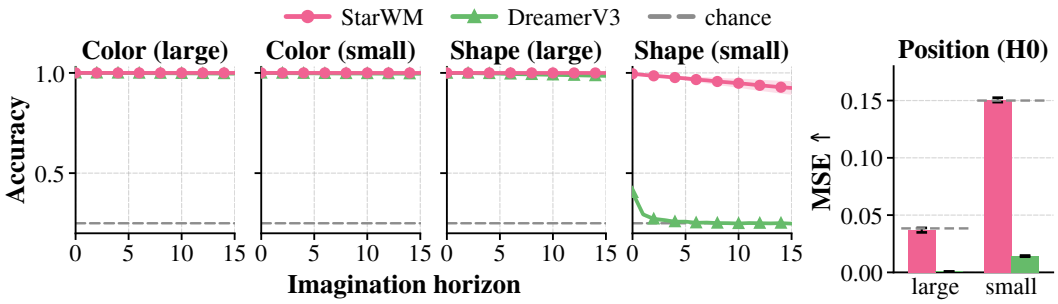


Figure 21: **F.2: Latent probe results on NoisyTwoShape.** Probe accuracy for color, shape of the large and small object across H0 to H15 imagination horizon, and position MSE at H0. Higher MSE on position indicates correct irrelevant treatment (positions are re-sampled each step). Dashed lines indicate per-object chance levels for position MSE, computed via Monte Carlo simulation over each object's position distribution under its size-dependent margin: 0.038 (large) and 0.150 (small).

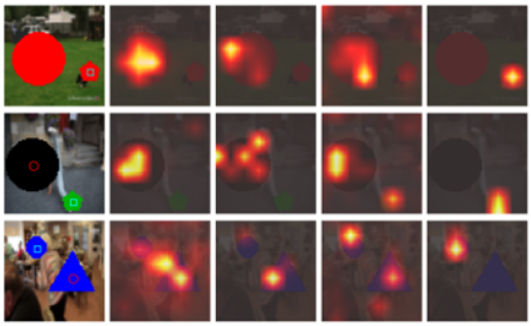


Figure 22: **F.2: StarWM attention maps on NoisyTwoShape.** Columns show the input image followed by the four query attention maps.

dominates on Walker Walk while R2-Dreamer leads on Cheetah Run. Based on the original-scale results in §4.2, we expect reward-augmented attention to mitigate the small-object VH degradation, but verification remains for future work.

These results empirically confirm the limitation discussed in §6: spatial granularity matters when objects shrink below what 8×8 attention can localize cleanly. A natural architectural extension is to intercept the encoder one stage earlier to expose 16×16 spatial tokens, trading compute for finer routing. We expect this together with reward-augmented attention to partially recover the small-object regime, with the relative contribution of each remaining an open empirical question.

F.2 Multi-Object Extension: NoisyTwoShape

We extend NoisyShape to a two-object setting to test (i) whether StarWM's attention queries spontaneously partition across multiple foreground entities without architectural modification, and (ii) whether the resulting representation cleanly retains both objects through long-horizon imagination under visual distractors. The NoisyTwoShape environment renders two non-overlapping geometric shapes, one large and one small, on DAVIS video backgrounds re-sampled each step, with positions also re-sampled each step. Each shape has its own shape and color attributes controlled by an 8-action discrete space. We use the same StarWM configuration as in NoisyShape (4 attention queries, no hyperparameter tuning) and probe color, shape, and position for both objects (probe sets of 200 episodes for H0 and 100 episodes for imagination).

Queries partition across objects. Figure 22 shows that StarWM's four attention queries distribute across the two foreground objects without per-object supervision, extending the emergent specialization observed in single-object NoisyShape to multi-object scenes. Across samples, queries

reliably cover both the larger and the smaller object, and none drift to the video background despite the changing distractor.

Both objects retained through long-horizon imagination. Figure 21 shows that this spatial partitioning translates into attribute retention through imagination. STARWM preserves color and shape of both objects with $\geq$99% accuracy at the encoding step (H0), and retains them through 15-step open-loop imagination: both colors and the larger object's shape remain near 100%, while the smaller object's shape degrades only modestly (from $\sim$99% to $\sim$92%). DreamerV3 by contrast fails on the smaller object from the encoding step itself: its small-shape probe reaches only 40% at encoding (chance 25%) and falls to chance during imagination, indicating that the encoder, which is forced by reconstruction to allocate capacity by pixel area, does not retain a usable representation of the smaller object even at the moment of observation. The reconstruction bias, mild for the larger object whose pixel signature dominates, becomes severe for the smaller one.

Position is correctly treated as objective irrelevant. The position MSE at H0 confirms that STARWM treats both object positions as objective-irrelevant, with MSE matching the chance reference for both the larger and the smaller object (chance levels differ across objects due to size-dependent position sampling ranges; computed by Monte Carlo simulation, see Figure 21 caption). DreamerV3 by contrast encodes both positions with high precision, reflecting the dense reconstruction objective's tendency to retain spatially-localized identifiers regardless of their predictive value.

F.3 Texture, Low Contrast, and Photorealistic Scenes

The main paper evaluates setups where the foreground visually stands out from the video background. Here, we test whether STARWM's routing relies on this visual contrast or truly learns from the underlying dynamics. We evaluate three settings with reduced foreground–background contrast: (i) a texture ladder that matches foreground and background visual distributions while keeping ground-truth labels; (ii) a desaturated Walker agent on DMC; and (iii) CALVIN, a photorealistic manipulation dataset where the robot, objects, table, and kitchen share the same visual render.

(i) Textured foreground ladder. We replace the solid color of the NoisyShape object with textures of increasing difficulty: **L1** adds procedural textures from an external domain; **L2** samples textures directly from the DAVIS background videos, sharing the same visual domain; **L3** alpha-blends the L2 texture into the background ($\alpha = 0.5$), removing clear boundaries. All other factors remain unchanged, allowing direct use of our probe protocol (Appendix C.2).

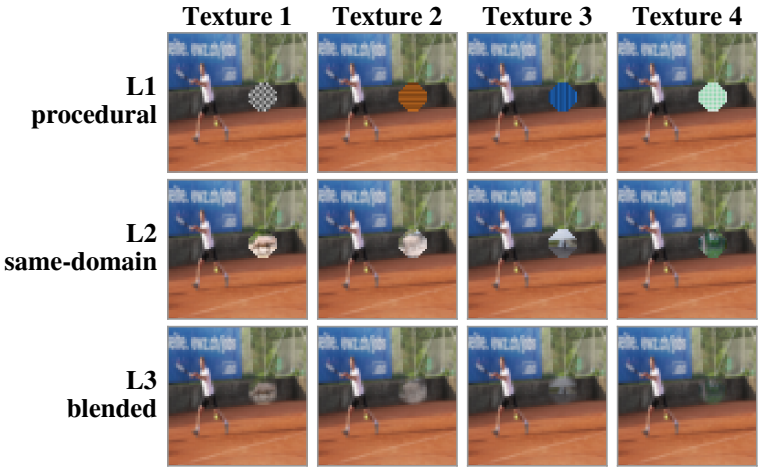


Figure 23: **F.3: Foreground texture ladder.** The same underlying state rendered across levels for the four texture IDs (replacing color). Only foreground appearance changes: L1 uses external procedural textures, L2 uses crops from DAVIS background videos, and L3 alpha-blends these crops into the background (α=0.5, feathered edges).

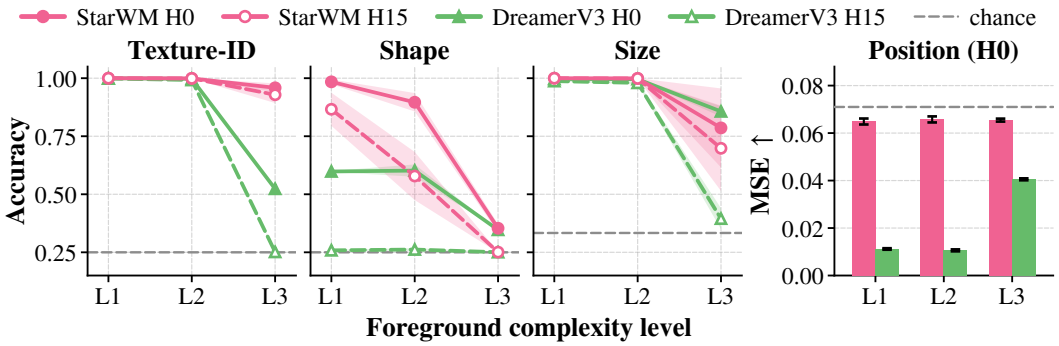


Figure 24: **F.3: Latent probe accuracy across the ladder** (mean ± SE, 3 seeds, 400k steps). Solid lines/filled markers denote the encoding step (H0); dashed lines/open markers denote 15-step open-loop rollouts (H15). Grey dashed lines show chance performance. For position MSE (right panel), higher values indicate that task-irrelevant position details are successfully filtered out. STARWM stays at chance, whereas DreamerV3 encodes exact position.

Table 14: **F.3: Probe accuracy on the texture ladder** (3 seeds, 400k steps). H0 = encoding step, H15 = 15-step rollout. Chance is 0.25 for texture and shape, 0.33 for size; chance position MSE is ≈0.071.

		texture-id		shape		size		pos. MSE
Level	Model	H0	H15	H0	H15	H0	H15	H0
L1	STARWM	**1.000**	**1.000**	**0.984**	**0.866**	1.000	1.000	0.065
	DreamerV3	0.999	0.999	0.598	0.259	0.996	0.987	0.011
L2	STARWM	**0.999**	**0.999**	**0.896**	**0.579**	0.999	0.999	0.067
	DreamerV3	0.997	0.993	0.600	0.262	0.996	0.980	0.010
L3	STARWM	**0.959**	**0.929**	0.354	0.251	**0.786**	**0.698**	0.065
	DreamerV3	0.523	0.252	0.347	0.250	0.857	0.395	0.041

As shown in Figure 23, Figure 24, and Table 14, both models maintain texture identity at L1 and L2 because the object still has defined boundaries. However, differences emerge on *shape*: STARWM retains shape features over 15 imagined steps (0.866 at L1, 0.579 at L2), whereas DreamerV3 drops to chance (≈0.25).

At **L3**, where the boundary is blended, the difference in texture tracking is pronounced: DreamerV3 falls to chance (0.252) at H15, while STARWM maintains 0.929. Blending removes sharp contours, so shape accuracy drops for both models. Across all levels, STARWM retains object size better (0.698 vs. 0.395 at L3, H15) and keeps position MSE near chance (0.065 vs. 0.071), confirming that it ignores task-irrelevant spatial information. In contrast, DreamerV3 tracks position closely (0.041).

Attention routing vs. representation. The attention mass on the object (≈12% of total pixels) shifts from 0.82 to 0.57 and 0.37 across L1–L3 (7.0×, 4.9×, and 3.2× above uniform attention). Because entity tokens use soft spatial weighting, STARWM tracks foreground identity reliably even when visual boundaries blur.

(ii) Low-contrast agent on DMC. We desaturate the Walker body colors to match the video background tone, evaluating performance across both distractor settings over 3 seeds.

Table 15: **F.3: Low-contrast Walker Walk return** (mean ± SE, 3 seeds). VH = coherent video background, VHR = per-step randomized frames.

Setting	STARWM	DreamerV3	R2-Dreamer
Walker VH (3M steps)	893 ±41	198 ±22	895 ±41
Walker VHR (1M steps)	**890** ±17	394 ±34	118 ±13

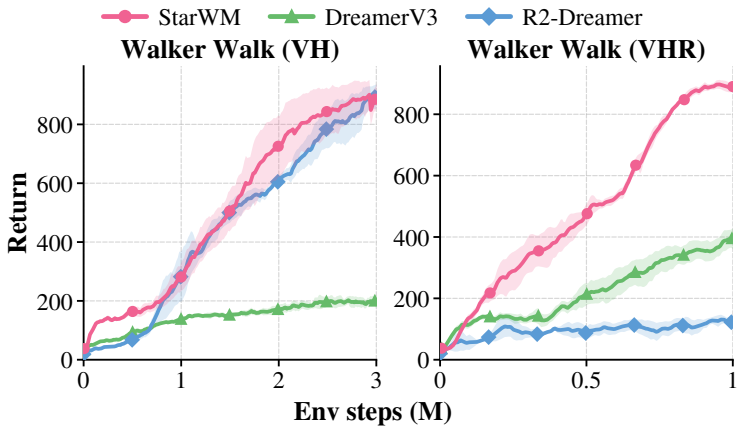


Figure 25: **F.3: Return curves for low-contrast Walker Walk** (mean ± SE, 3 seeds).

Under random frame distractors (VHR), STARWM reaches 890, substantially outperforming DreamerV3 (394) and R2-Dreamer (118). Under coherent video backgrounds (VH), STARWM (893) matches R2-Dreamer (895), while DreamerV3 reaches only 198. Removing color contrast does not impair STARWM's robustness under rapid visual shifts.

(iii) Photorealistic manipulation (CALVIN). We evaluate STARWM on the CALVIN benchmark (`task_D_D`, 512k frames of teleoperated play, static camera, 7-DOF actions) for 200k offline steps. Here, the robot, table, blocks, and environment are part of a unified photorealistic render. We project the gripper and block centers into image space using camera calibration and measure the attention allocated to each region. To account for workspace framing, we evaluate each region against a shuffled null baseline (the same attention map applied to a different frame).

Table 16: **F.3: Spatial attention concentration on CALVIN.** Measured on frames where a block moves ($n = 204$). Active blocks are identified by ground-truth movement.

Region	Frame share	Attention share	Concentration	Shuffled null
Gripper tip ($r = 5$px)	1.9%	16.6%	**8.69**×	2.83×
Active block	0.7%	4.2%	**6.09**×	2.56×
Static blocks	1.4%	2.1%	1.52×	2.03×
Gripper + all blocks	3.3%	18.8%	5.63×	2.48×

Attention focuses heavily on the gripper and the active block (8.69× and 6.09× concentration, compared to 2.83× and 2.56× nulls). In contrast, static blocks remain at chance levels (1.52× vs. 2.03× null), reflecting only spatial priors. Across all validation frames, gripper concentration averages 9.71× (2.84× null). Even without visual separation between foreground and background, STARWM routes attention based on controllability.

F.4 Passive and Non-Action-Driven Task-Relevant Objects

STARWM trains its router using inverse dynamics and a temporal contrastive loss. Because both objectives favor directly controlled entities, we investigate whether the model retains task-relevant objects that are *passive* (i.e., unaffected by agent actions). We evaluate two cases: a diagnostic environment with an explicitly static object, and DMC Reacher, where the target is a fixed goal rather than an actuated joint.

(i) Diagnostic environment: `PassiveShapeEnv`**.** We augment NoisyShape with a second object B alongside the controllable object A. Object B's color and position are sampled once per episode and remain fixed regardless of agent actions (Figure 27). Because actions never alter B, it provides no inverse-dynamics gradient. We evaluate whether B's properties are preserved in the latent state using the probing protocol in Appendix C.2, evaluating continuous position via regression (R^2).

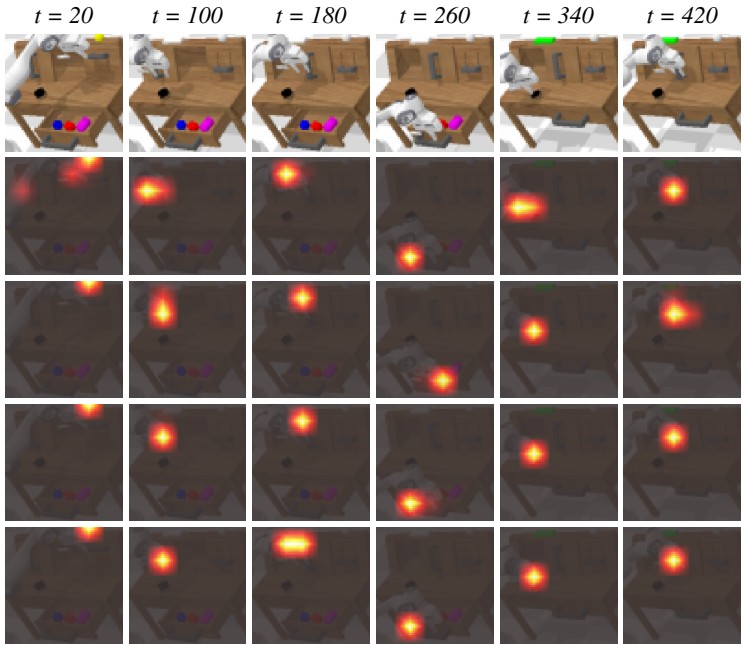


Figure 26: **F.3: Attention routing on CALVIN.** Input frame and corresponding attention maps from four queries. Attention targets the controllable objects and gripper over background structures.

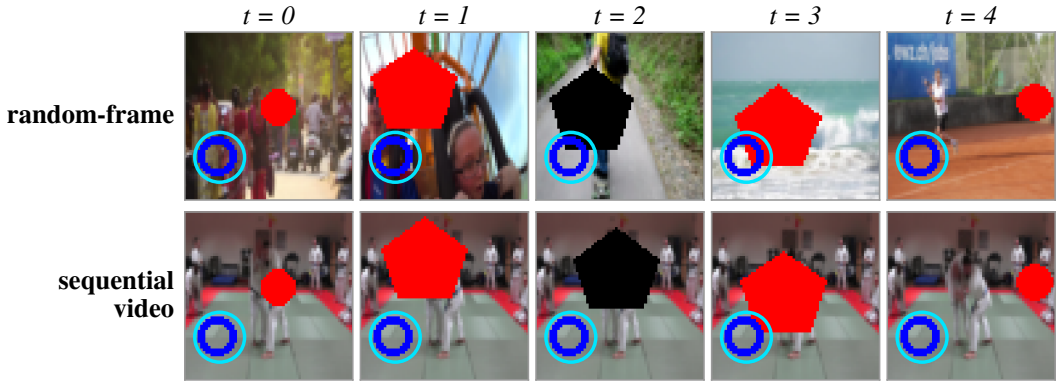


Figure 27: **F.4:** `PassiveShapeEnv`**.** Five consecutive steps under each background setting. The active object A changes attributes based on actions and moves at each step; the passive object B (cyan circle) maintains fixed attributes throughout the episode. Top: per-step randomized background frames. Bottom: sequential video background.

As shown in Table 17 and Figure 28, object identity is fully preserved under both backgrounds through 15 rollout steps. However, position recovery depends directly on the background regime: it is accurately decoded under random-frame backgrounds ($R^2 \approx 0.8$), but drops under sequential video ($R^2 \leq 0.13$ at H0, dropping below zero after imagination).

This difference stems from the temporal contrastive objective. Because InfoNCE samples negative pairs uniformly across frames, negatives typically come from different episodes. Encoding B lowers negative pair similarity because its attributes vary across episodes while remaining static within each episode. When the background is randomized, B serves as the primary cross-step identifier. On sequential video, however, the temporally coherent background already identifies the episode, diminishing the contrastive utility of encoding B's position.

Table 17: **F.4: Latent probe recovery for passive object** B (mean ± SE, 3 seeds, 212k steps). b_colour is 4-way classification accuracy (chance = 0.25); b_position reports held-out R^2.

Background	b_colour (acc) H0	b_colour (acc) H15	b_x (R^2) H0	b_x (R^2) H15	b_y (R^2) H0	b_y (R^2) H15
random-frame	**1.000** ±.000	**1.000** ±.000	**0.80** ±.09	**0.78** ±.09	**0.84** ±.04	**0.76** ±.08
sequential video	**0.998** ±.000	**0.994** ±.003	0.13 ±.03	−0.23 ±.02	0.08 ±.02	−0.18 ±.04

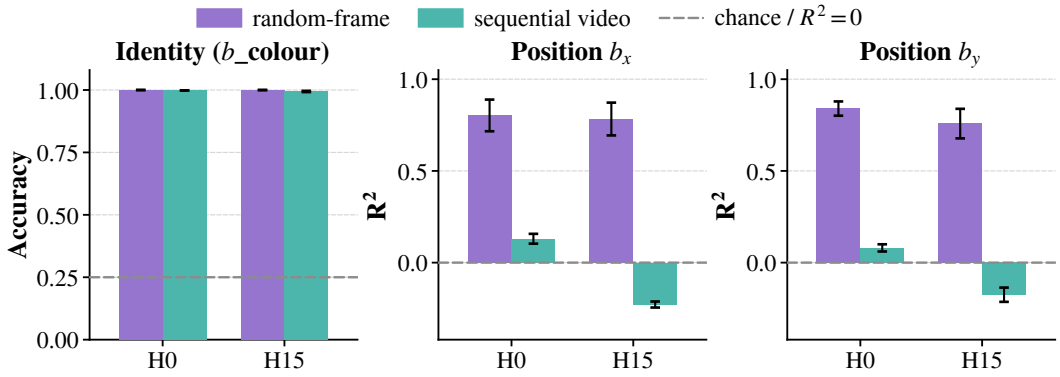


Figure 28: **F.4: Probe results for passive object** B (mean ± SE, 3 seeds, 212k steps). Object identity reaches ceiling accuracy across both regimes. Position is recovered on randomized backgrounds, but drops to or below R^2=0 under sequential video.

(ii) Fixed target control: DMC Reacher. In DMC Reacher, the target is a stationary disc that the agent must reach without directly manipulating it (Figure 29, left). We evaluate performance across all baselines under both distractor types.

Table 18: **F.4: DMC Reacher episode returns** (mean ± SE, 3 seeds). VH = temporally coherent video background, VHR = per-step randomized frames.

Setting	STARWM	STARWM-R	DreamerV3	R2-Dreamer
Reacher VH (3M steps)	391 ±136	789 ±33	881 ±34	**947** ±3
Reacher VHR (1M steps)	**690** ±70	657 ±1	450 ±37	66 ±9

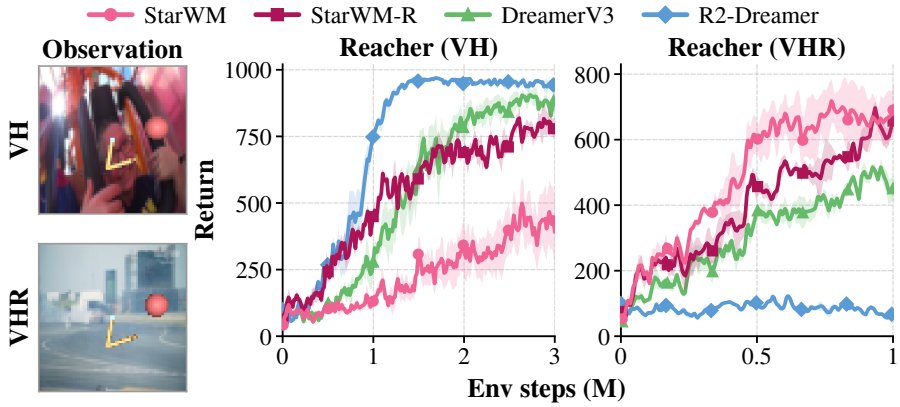


Figure 29: **F.4: DMC Reacher performance.** *Left:* observation frames showing the two-link arm and static target disc against DAVIS backgrounds (VH: coherent video, VHR: per-step random frames). *Right:* learning curves (mean ± SE, 3 seeds). On VH, reward augmentation (STARWM-R) recovers default STARWM's deficit. On VHR, STARWM and STARWM-R lead while R2-Dreamer drops.

Under coherent video (VH), standard STARWM scores 391 because the static target lacks both inverse-dynamics gradients and distinct contrastive signals. Adding reward supervision (STARWM-R; Appendix E.1) recovers this gap to 789. Under random backgrounds (VHR), STARWM and STARWM-R lead at 690 and 657, while R2-Dreamer drops to 66 (Table 18).

(iii) Conditions for object retention. Together with the Ball-in-Cup results (Appendix E.1), these experiments identify when self-supervised routing retains a task-relevant factor:

Table 19: **F.4: Determinants of task-relevant object retention.** Default routing retains features that remain static within an episode and distinct across episodes, provided no competing signal (such as coherent video) already satisfies that criterion.

Task-relevant factor	Episode-static & distinctive?	Competing signal?	Default STARWM outcome
Passive B position, random bg	✓	None	Encoded ($R^2 \approx 0.8$)
Passive B position, seq. bg	✓	Video	Suppressed ($R^2 \leq 0.13$)
Reacher target, VHR	✓	None	Solved (690)
Reacher target, VH	✓	Video	Low return (391); benefits from reward
Ball in flight (Ball-in-Cup)	✗	None	Requires reward routing

Default routing preserves features that are episode-static and cross-episode distinctive, unless a coherent video background already acts as a dominant episode identifier (Table 19). When self-supervised objectives do not isolate passive elements, reward-conditioned routing incorporates the missing task signals.

F.5 Delayed Action Effects

The inverse-dynamics loss in Eq. 6 relies on adjacent frames $(t, t{+}1)$, assuming actions take immediate visual effect. When action outcomes are delayed, this adjacent window forces the model to explain a transition before its true cause appears. Here, we show that this sensitivity stems strictly from the supervision window rather than the routing mechanism itself.

The `ActionDelay`(k) environment. We extend NoisyShape such that observations at step t reflect the delayed action a_{t-k} for $k \in \{1, 2, 4\}$, keeping all other task parameters unchanged. We evaluate two supervision windows at each delay k: **Adjacent** (mismatched) uses $(t, t{+}1)$, targeting an action a_t whose effects are not yet visible. **Shifted** (matched) aligns the window to $(t{+}k, t{+}k{+}1)$, capturing the frame pair directly modified by a_t (recovering the standard objective when $k{=}0$).

Evaluation metrics. We evaluate two held-out metrics on fresh rollouts: *attention mass on the object* (the fraction and uniform concentration of the top query's attention mask inside the object boundary) and *attribute decodability* from the latent state $[h; z]$ (Appendix C.2). We omit training accuracy of the inverse model, as memorizing static frame–action pairs on the replay buffer easily inflates this metric without reflecting genuine representation quality.

Table 20: **F.5: Delayed action effects** at 180k steps (mean ± SE, 3 seeds). Object coverage is ≈6.7% of the frame; ×uniform denotes attention concentration. Color decodability remains near ceiling (≈1.00) across all settings and is omitted. The $k{=}0$ row serves as an un-delayed reference point probed at its final checkpoint (≈145k steps).

k	Supervision window	Attn. mass	×Uniform	Shape	Size
0	Reference (no delay)	0.846	11.2×	0.999	1.000
1	Adjacent (mismatched)	0.439 ±.033	6.6×	0.385 ±.027	0.476 ±.029
1	**Shifted (matched)**	**0.824** ±.026	**12.3**×	**0.998** ±.001	**1.000** ±.000
2	Adjacent	0.352 ±.093	5.3×	0.654 ±.153	0.722 ±.143
2	**Shifted**	**0.755** ±.010	**11.3**×	**0.997** ±.000	**0.999** ±.000
4	Adjacent	0.357 ±.009	6.2×	0.448 ±.039	0.504 ±.028
4	**Shifted**	**0.725** ±.020	**12.5**×	**0.994** ±.001	**0.996** ±.001

Results. As shown in Table 20, mismatched supervision disperses attention (object mass drops to 0.35–0.44 vs. 0.85 without delay), significantly lowering shape and size decodability. Aligning the window to the actual action outcome **fully restores both metrics**: attention mass recovers to 0.73–0.82 (11–13$\times$ uniform, matching the un-delayed baseline), and probe accuracy for shape and size returns above 0.99 across all delays k.

While mismatched windows induce high variance across runs (e.g., shape decodability at $k{=}2$ is 0.654 ± 0.153), their performance consistently lags behind the matched setup. These results confirm that delayed effects do not hinder the routing mechanism itself; aligning the supervision window to the true transition dynamics fully preserves representation learning.

G Broader Impacts

This paper presents foundational research on visual representation learning for model-based reinforcement learning, evaluated entirely on simulated benchmarks (DeepMind Control Suite, DAVIS video backgrounds). The work does not target a specific application or deployed system.

Positive impacts. Distraction-robust world models could benefit beneficial applications such as robotic manipulation in cluttered environments, assistive autonomy in real-world settings with varying visual conditions, and sample-efficient learning where collecting clean data is expensive. The architectural separation between deciding *where* to route and *what* to extract may also support interpretability research, since the attention maps provide direct visualization of what the world model is attending to.

Potential negative impacts. As with any general-purpose advance in autonomous control, improvements in distraction robustness apply equally to applications with negative dual-use potential (e.g., autonomous weapons systems). We do not see specific harms unique to this work beyond those associated with the broader trajectory of visual RL research.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state our main claims and these claims are directly supported by results in §4.1 (probe analysis on NoisyShape and cross-sample image composition) and §4.2 (DMC policy performance).

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: A dedicated **Limitations** paragraph in §6 discusses the limitations of this work.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer:[N/A]

Justification: The paper presents an architectural and empirical contribution; it does not contain formal theorems or proofs.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: §3 describes the architecture (cross-attention module, dual-stream decoder, stop-gradient barriers, training objective) in full. Appendix B.1 provides pseudocode (Algorithm 1) and complete implementation details (encoder interception, multi-head attention, KL annealing schedule, etc.). Appendix B.2 lists all hyperparameters. Appendix C describes the NoisyShape environment design and DMC video distractor setup. Appendix B.3 describes baseline implementations.

Guidelines:

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

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

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

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

   Answer: [Yes]

   Justification: Anonymized code is provided in the supplementary material. Code will also be released on GitHub upon acceptance.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Appendix B.2 provides complete hyperparameters. Probe analysis method is described in §C.2.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: All DMC results report mean and standard error over 3 random seeds (Figure 5, Table 6 in Appendix D). Probe analysis on NoisyShape also reports mean and standard error over 3 seeds (Table 5). Following the reliable RL evaluation framework of Agarwal et al. [1], we provide *Aggregate metrics with 95% Stratified Bootstrap CI*, *Probability of Improvement Heatmap* and *Performance Profile* for DMC tasks in Appendix D.2.2

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Compute resources are described in Table 3 in Appendix B.3.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: A short broader-impact paragraph appears in Appendix G. The work is foundational research on visual representation learning for model-based RL.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: The paper does not release pre-trained generative models, scraped datasets, or other assets with foreseeable misuse potential. The released code trains world models on simulated control tasks.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: All external assets are properly cited and used in accordance with their licenses:

- DeepMind Control Suite [32] (Apache 2.0)
- DAVIS 2017 [26] (BSD 3-Clause License)

- DreamerV3[Original JAX] [11] (MIT License); DreamerV3 [PyTorch reimplementation; NM512/dreamerv3-torch on GitHub] (MIT License)
- DenoisedMDP [33] (Creative Commons)
- Iso-Dreamer++ [24] (MIT License)
- R2-Dreamer [21] (MIT License)

.

Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: We introduce the NoisyShape diagnostic environment (described in Appendix C.1) and the STARWM implementation. Both will be released with upon acceptance; anonymized versions are provided in supplementary material with documentation.

    Guidelines:

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

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

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

    Answer: [N/A]

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

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

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

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs were not used as a component of the core methods (training, evaluation, or analysis). LLMs were used only for writing assistance (editing, formatting) and coding assistance.

    Guidelines:

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