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

Anonymous Author(s)

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, 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 combines advantages of both paradigms through a cross-attention module gated by self-supervised dynamics signals that decides where reconstruction applies, while a dual-stream decoder lets reconstruction shape only what is extracted at attended locations, with stop-gradient barriers preventing the two objectives from interfering. This 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, STARWM outperforms reconstruction-based baselines while keeping competitive with the concurrent reconstruction-free approach, and is the only method that performs consistently well across clean, sequential-video, and random-video distractor settings. Mechanistic probing confirms STARWM preserves state attributes with near-perfect fidelity through long-horizon imagination while systematically discarding distractors.

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

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 [34], 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

Preprint.

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:

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

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.

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

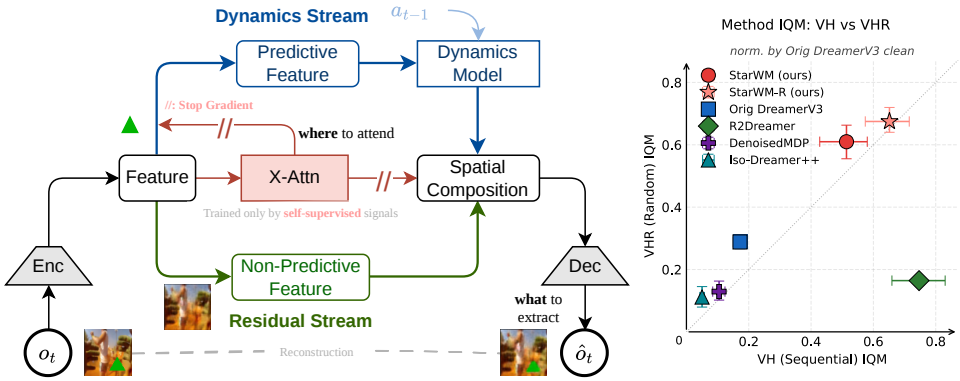


Figure 1: **StarWM disentangles *where-to-attend* (X-Attn) from *what-to-extract* (reconstruction $\hat{o}_t$). *Left:*** Self-supervised objective signals train a cross-attention module to localize predictable, action-conditional content, while the content of the attended regions is shaped by a reconstruction objective through a dual-stream decoder. Stop-gradient barriers (//) keep the two objectives from interfering. For detailed architecture diagram see Appendix C.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.

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 C.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:

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

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 C.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.4).

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 [37], 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 tests three claims about STARWM's design: that the latent representation cleanly separates objective-relevant from objective-irrelevant content (§4.1); that this separation translates into policy robustness across distractor types (§4.2); and that the cross-attention routing and self-supervised guidance, together with their gradient isolation, are each necessary for these properties (§4.3).

Baselines and implementation setup. We compare STARWM against four baselines representing distinct distraction-handling paradigms: **DreamerV3** [11], our reconstruction-based backbone; **DenoisedMDP** [34], 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 C.3); all results report mean and standard deviation across 3 random seeds.

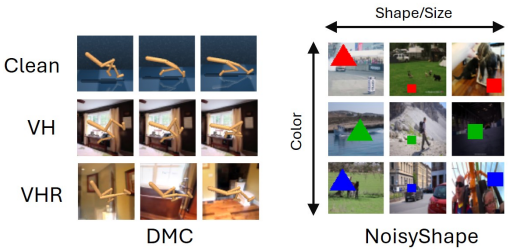


Figure 2: **Evaluation environments.** ***Left:*** DeepMind Control Suite with three background conditions: clean, Video Hard Sequential, and Video Hard Random. ***Right:*** NoisyShape: a geometric shape with controllable attributes (shape, color, size) on DAVIS video backgrounds, used for mechanistic 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 B.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.

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;

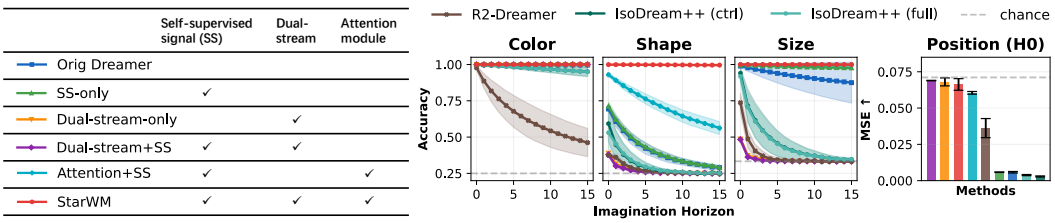


Figure 3: **Latent probe results on NoisyShape.** Probe accuracy for objective-relevant attributes (color, shape, size) from H0 (encoding) through H15 (15 steps of imagination), and objective-irrelevant attribute (position MSE) at H0 (H15 omitted as ground-truth position at $t+15$ is independent of the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

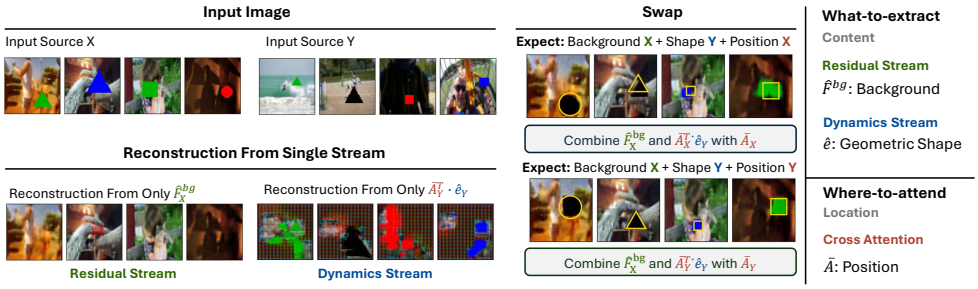


Figure 4: **Cross-sample composition reveals emergent factorization.** ***Left:*** Input frames from sources X and Y, with each stream's isolated reconstruction beneath. The residual stream renders only background context, while the dynamics stream renders entity content. ***Middle:*** Swapping components 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. ***Right:*** The three components specialize into distinct emergent roles, mapping onto the content and location separation of §3 without any explicit disentanglement objective.

policy comparisons are conducted on DMC (§4.2). Full evaluation protocol is in Appendix C.4 and evaluation on the variants of the NoisyShape is in Appendix E.2.

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 should encode position 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.1). 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

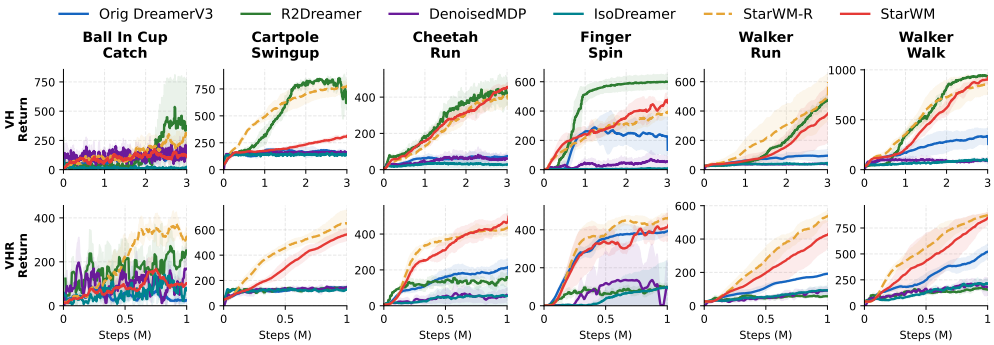


Figure 5: **DMC learning curves.** STARWM and STARWM-R are the only methods that perform strongly under both sequential (VH) and random (VHR) video backgrounds.

substitution cleanly swaps the corresponding visual factor while leaving the others intact, revealing 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.

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. See Appendix E.3 for ablation study of number of queries.

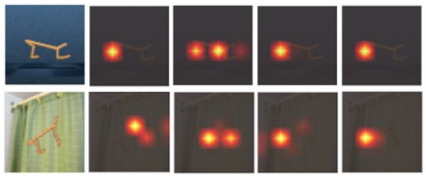


Figure 6: **Learned attention maps on DMC.** STARWM's cross-attention consistently localizes the agent through 4 queries under both clean and distracting settings.

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.6); 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 Component Analysis

The source of the attention signal. 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.5.

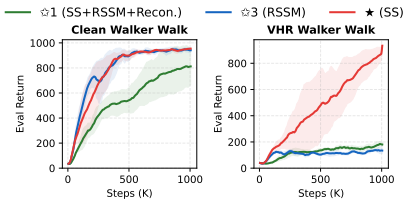


Figure 7: **Ablation study of sources of the attention signal.** Self-supervised guidance and gradient isolation are jointly necessary.

The role of each self-supervised signal. Decomposing the guidance into its two components, we find that inverse dynamics alone is sufficient for near-optimal attention routing on NoisyShape, while contrastive learning alone fails to localize the foreground. The two differ in what they supervise: inverse dynamics points attention to regions whose visual changes are explained by the agent's action, an unambiguous target; temporal contrastive learning only requires consecutive features to be more similar than random pairs, which fails to single out the agent from any other temporally smooth region of the scene. This is consistent with Schneider et al. [28], who find that purely temporal objectives approach supervised quality only under uniform backgrounds. On DMC, contrastive supervision helps under VHR, where the agent is the only temporally coherent feature, but is partially co-opted under VH. Full decomposition is in Appendix E.6.

5 Related Work

World models and distraction-robust representation learning. World models for visual control fall into two families: reconstruction-based methods such as the Dreamer family [9, 10, 11], which train RSSM dynamics via pixel reconstruction and KL-regularized latent prediction, and reconstruction-free methods such as TD-MPC [14, 13], which bypass pixel reconstruction entirely via reward, value, and latent-consistency losses. Reconstruction-based methods are vulnerable to the reconstruction bias discussed in §1; reconstruction-free methods avoid it by construction but forfeit the pixel-level visual grounding that reconstruction provides. STARWM sits between these two extremes: it retains reconstruction for its dense content supervision while using self-supervised signals to alleviate the reconstruction bias by redirecting attention away from distracting content.

Within the reconstruction-based family, prior work on distraction robustness modifies the learning objective, as summarized in §1: methods that replace pixel reconstruction with auxiliary objectives (DreamerPro [4], R2-Dreamer [21]) or latent decomposition (TIA [7], DenoisedMDP [34], IsoDream [23]). A parallel line operates spatially, localizing task-relevant regions via unsupervised keypoint detection [35], simulator masks or segmentation foundation models [16], or human-prompted SAM with vision foundation models [36]. These methods share STARWM's spatial routing intuition but depend on external sources of supervision. SMG [41] is closest to STARWM in pursuing a dual-branch architecture where a learned foreground mask separates task-relevant from task-irrelevant reconstruction. We do not include SMG as an empirical baseline because it is a model-free method built on DrQ-v2 [39] and relies on per-task augmentation design, placing it in a different category. STARWM differs in deriving the routing signal from environment dynamics, which are available to any RL agent without task-specific augmentation design, and in retaining reconstruction as the training signal for what the attended regions contain.

Object-centric learning and cross-attention. Slot Attention [19] decomposes scenes into object-level representations using iterative attention supervised by reconstruction, with a spatial broadcast decoder that uses the attention map as a composition mask. DETR [3] uses learnable queries for object detection via single-pass cross-attention. STARWM combines single-pass extraction (from the DETR tradition) with spatial broadcast composition (from Slot Attention). A key difference is the training signal for attention: Slot Attention's weights are learned through reconstruction alone via iterative refinement, while STARWM's attention is trained by self-supervised inverse dynamics and contrastive losses, with reconstruction shaping only the content of entity tokens through a designated pathway focusing on content. The single-pass design of STARWM also enables clean gradient isolation between attention routing and feature extracting, which iterative attention does not admit.

Self-supervised methods for RL. Self-supervised signals such as inverse dynamics [25], contrastive learning [17, 22], and self-predictive representations [29] are typically used to produce the final latent representation directly, applying global pooling that discards spatial structure. STARWM uses these signals differently: as supervision for a spatial attention mechanism that routes information into a subsequent world model, with reconstruction shaping the content of the routed representations. The attention gating mechanism is also supervision-agnostic. Any differentiable signal applied to the attention path trains the routing. 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 promote abstraction but risk losing fidelity to the underlying environment. We presented STARWM, a world model architecture that combines strengths of both methods by learned attentional gating to shield the latent dynamics model from irrelevant content. By limiting pixel grounding to objective-relevant content, STARWM mitigates the reconstruction bias: attention allocates representational capacity by objective relevance rather than pixel area, enabling temporally coherent prediction over long horizons. Our default attention objectives (inverse dynamics and contrastive learning) target controllable and predictable content respectively, and require no task supervision. Empirically, this configuration outperforms existing reconstruction-based baselines under visual distractors. The attentional gating mechanism is also modular: any differentiable objective can be added without modifying the rest of the architecture (e.g. adding reward prediction 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 evaluation is confined to single-agent continuous control with spatially structured image observations; multi-object, multi-agent, non-spatial inputs, and real-world settings with domain shift remain mostly untested, though preliminary multi-object experiments (Appendix E.8) suggest that attention queries spontaneously partition across objects without architectural modification. The 8×8 attention resolution may be too coarse for very small task-relevant objects (Appendix E.7). STARWM's default attention guidance combines inverse dynamics and contrastive losses, neither of which is universally sufficient: inverse dynamics provides little signal for passively moving task-relevant objects (e.g., the ball in Ball-in-Cup), while contrastive learning rewards temporal coherence indiscriminately and can be partially co-opted under temporally structured backgrounds. The modularity of the attention pathway lets reward prediction fill these gaps when task reward is available (STARWM-R), but identifying a single objective that is robust across all task types remains open (Appendix E.4.1). Finally, by retaining the image decoder, STARWM carries higher per-step compute and parameter cost than reconstruction-free alternatives such as R2-Dreamer, though it remains within the same order as the DreamerV3 backbone it builds on (Appendix C.3).

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] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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 Environment Design **17**

B.1 Shape Environments 17

B.2 DeepMind Control Suite Environments 19

C Algorithm and Implementation Details **20**

C.1 Algorithm 20

C.2 Hyperparameters 21

C.3 Implementation of Baseline Algorithms and Compute Resources 21

C.4 Probe Analysis Methodology for NoisyShape 23

D Full results **24**

D.1 Full results in NoisyShape Environment 24

D.2 Full results in DMC Environments 24

D.2.1 Per-task results 24

D.2.2 Aggregate Statistical Analysis 25

E Extended Ablation Studies **26**

E.1 Predictable Position Variant of NoisyShape 26

E.2 Progressive Distraction Analysis across NoisyShape Variants 28

E.3 Ablation Study of Number of Queries N 29

E.4 Reward as Additional Attention Routing Signal 30

E.4.1 Failure mode of STARWM on `Cartpole Swingup VH` 30

E.5 Gradient Isolation Analysis 31

E.6 Contrastive Signal Conflict Analysis 34

E.7 Sensitivity to Object Scale 36

E.8 Multi-Object Extension: NoisyTwoShape 36

F Broader Impacts **38**

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** [40]: 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** [34]: 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** [32]: 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.4). 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 [30]), 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 [38], SLATE [31]) 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 E.8) 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 Environment Design

B.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.

Figure 8: **B.1: Illustration of NoisyShape Environments.**

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 1.

Table 1: 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%)

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 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 1): $m = r/64 \in \{0.10, 0.18, 0.26\}$ 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 a pool of >10,000 frames). 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. 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 pool of 20 DAVIS clips 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.1 for results and discussion.

B.2 DeepMind Control Suite Environments

We evaluate on six continuous control tasks from the DeepMind Control Suite [33], standard in prior distraction-robust RL work [7, 34, 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 DAVIS [26] video clip that plays sequentially within each episode: each environment 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 from DAVIS videos, 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. We use the DAVIS 2017 training set. For VH, a video is selected uniformly at random at episode start and played from its first frame; if the episode exceeds the video length, the video loops. For VHR, at each step we first sample a video uniformly and then sample a frame uniformly from that video. All methods use the same video sampling sequence within each seed to ensure fair comparison.

C Algorithm and Implementation Details

C.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. $e_t^{dyn} = \bar{A}_t \cdot V_t$.

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.

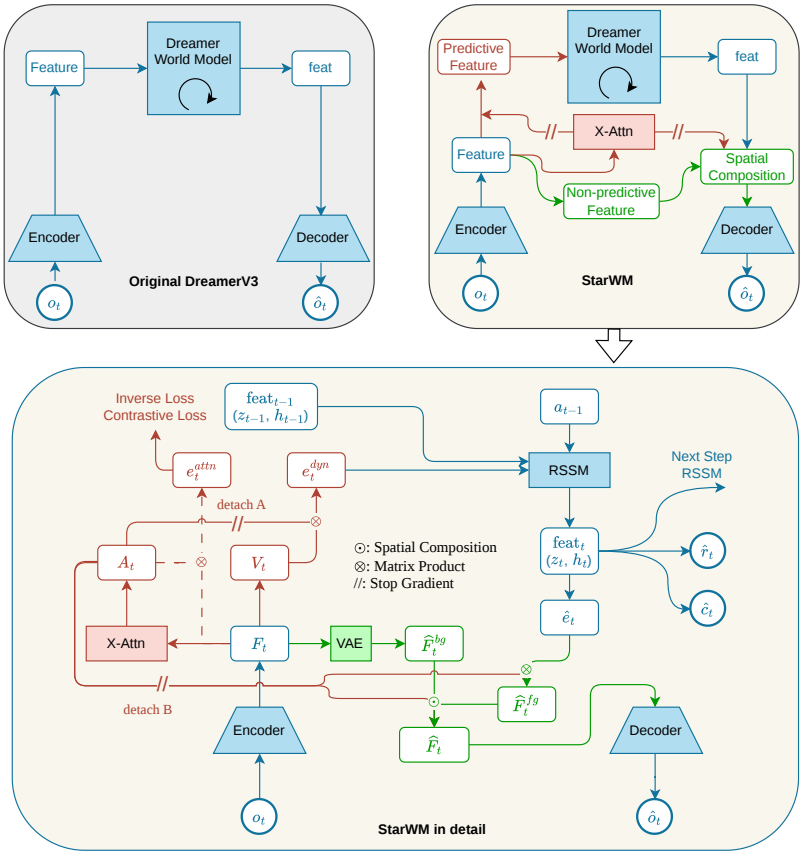


Figure 9: **StarWM architecture.** Top: comparison with DreamerV3. StarWM splits the encoder output via cross-attention into an attended stream (entering the RSSM) and a residual stream (auxiliary VAE, discarded during imagination). Bottom: detailed architecture. The cross-attention module (red) extracts 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.

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 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.

C.2 Hyperparameters

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

C.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 [34], 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 imple-

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

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

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

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

// Cross-attention spatial routing

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

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

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

$e_t^{\text{attn}} \leftarrow \mathrm{flatten}(A_t F_t)$ ▷ 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 $\bar{A}$)

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

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

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

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

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

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

// Dual-stream decoder

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

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

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

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

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

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

// 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 $\nabla\mathcal{L}$

mentation and adopt the recommended default hyperparameters. To ensure that all methods are evaluated under identical observation conditions, we make minimal modifications to integrate a 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.

STARWM runs at comparable speed to other reconstruction-based world models (DreamerV3, IsoDream++) 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

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

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

higher per-step compute. The cross-attention and self-supervised auxiliary heads add minor overhead relative to the DreamerV3 backbone STARWM builds on.

C.4 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 $\mathrm{proj}_{\mathrm{dyn}}$ 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: $\mathrm{Linear}(1536, 256) \rightarrow \mathrm{ReLU} \rightarrow \mathrm{Dropout}(0.2) \rightarrow \mathrm{Linear}(256, 256) \rightarrow \mathrm{ReLU} \rightarrow \mathrm{Dropout}(0.2) \rightarrow \mathrm{Linear}(256, C)$, 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.

D Full results

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 4 reports Probe accuracy on the NoisyShape diagnostic environment (§4.1) with standard errors across 3 seeds.

Table 4: **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** ±0.001	**99.98** ±0.00	25.04 ±0.10	33.31 ±0.29
+ SS only	**99.97** ±0.01	74.15 ±2.59	**99.65** ±0.12	0.006 ±0.000	**99.90** ±0.02	29.04 ±1.08	97.69 ±0.92
+ dual-stream + SS	**100.0** ±0.00	38.37 ±1.30	50.20±2.75	**0.069** ±0.000	**99.99** ±0.01	24.86 ±0.16	33.28 ±0.29
+ attention only	**99.97** ±0.01	92.95 ±1.21	**99.92** ±0.02	**0.061** ±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	97.75 ±1.42	37.46 ±1.36	73.67 ±6.63	0.036 ±0.007	46.20 ±9.72	24.82 ±0.24	33.17 ±0.15

D.2 Full results in DMC Environments

D.2.1 Per-task results

Table 5 reports full evaluation returns with standard errors 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 5: **D.2.1: DMC evaluation returns.** 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	984.54 $\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	147.53 $\pm$0.00	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

Table 6: **E.1: Probe accuracy under predictable position.** Position follows uniform linear motion with boundary reflection. Background ID is a 20-way classification over DAVIS clips. 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	0.0711	5.00

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 Extended Ablation Studies

E.1 Predictable Position Variant of NoisyShape

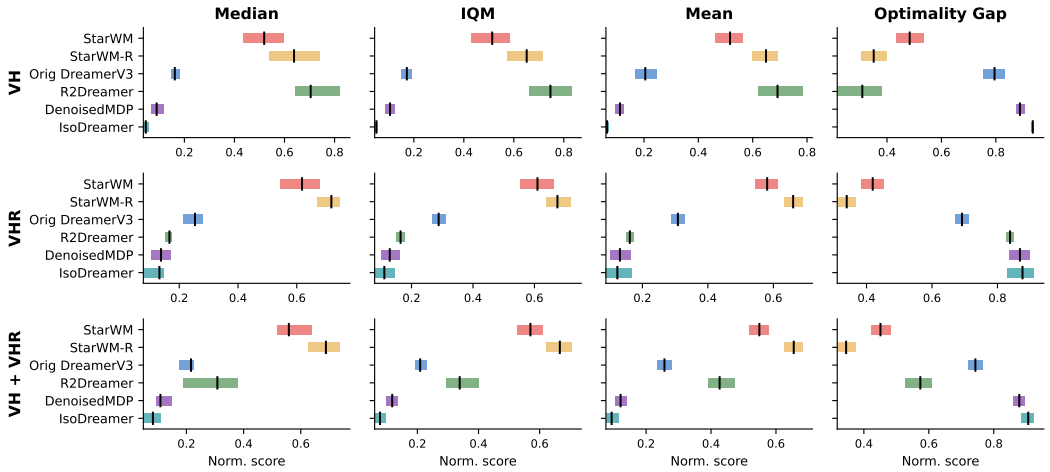
The main NoisyShape experiments (Figure 3 and Table 4) randomize position independently at each step, making it a purely non-predictable attribute that an ideal world model should discard. A natural concern is whether STARWM's low position probe accuracy (MSE $= 0.066$, near chance) reflects a correct 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 11 and Table 6.

Probing protocol. We follow the same probe protocol as Appendix C.4, extended with two additional probes: position (regression, MSE) and background ID (20-way classification, chance $= 0.05$). All numbers are mean $\pm$ standard error across 3 seeds.

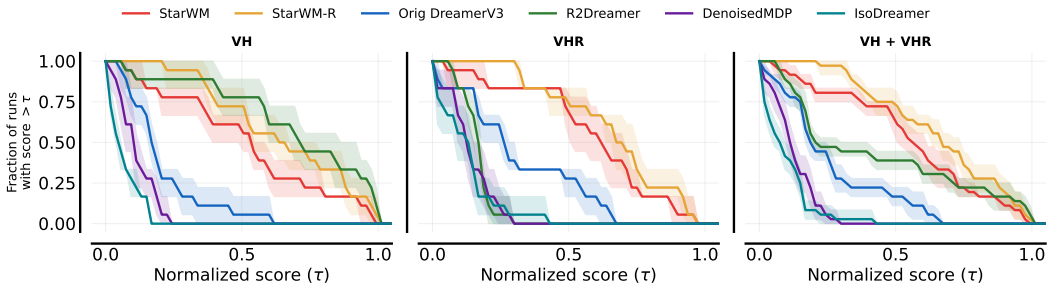
STARWM encodes predictable position. In contrast to the random-position setting where STARWM retains position only coarsely (MSE ≈ 0.066, near chance), under predictable motion STARWM's position MSE drops by two orders of magnitude to ~ 0.0008 at encoding and remains well below chance after 15 imagination 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 architecture 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.

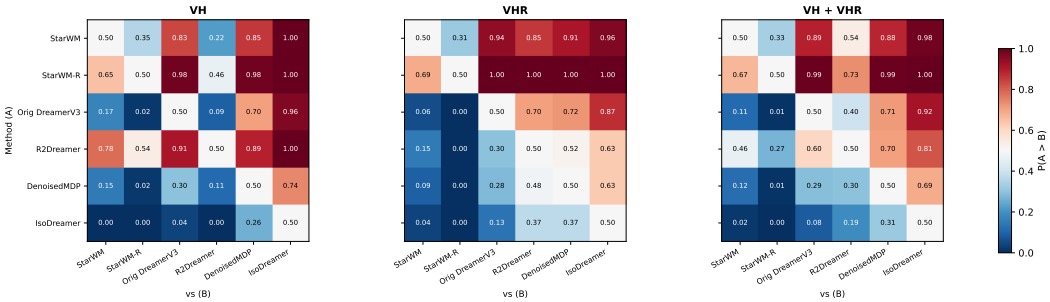
Summary. The two NoisyShape variants together establish the correct dual behavior: STARWM discards position when it is irrelevant and encodes position when it is relevant. DreamerV3 encodes position in both settings, and encodes background identity in both settings.



(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).

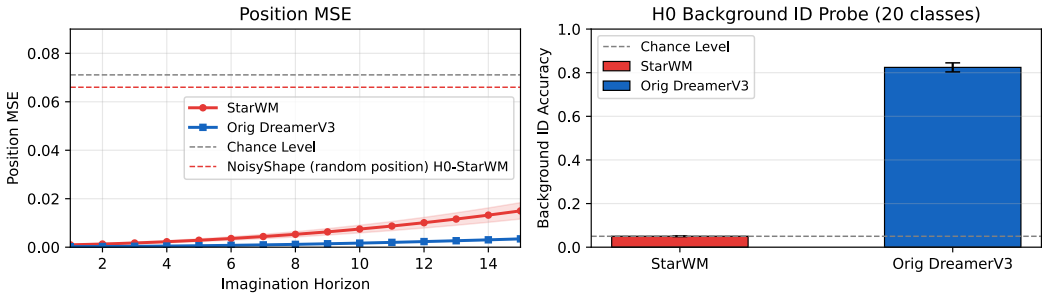


Figure 11: **E.1: Latent space probing under predictable object motion.** Left: Position MSE across a 15-step imagination horizon. When position follows a deterministic trajectory, STARWM successfully tracks and rolls out spatial coordinates well below chance level. Right: Background identification accuracy at the encoding step (H0). DreamerV3 suffers from severe objective-irrelevant contamination, while STARWM perfectly drops background identity.

E.2 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 12 and Table 7.

Table 7: **E.2: 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	100.00 ±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	—

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, 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

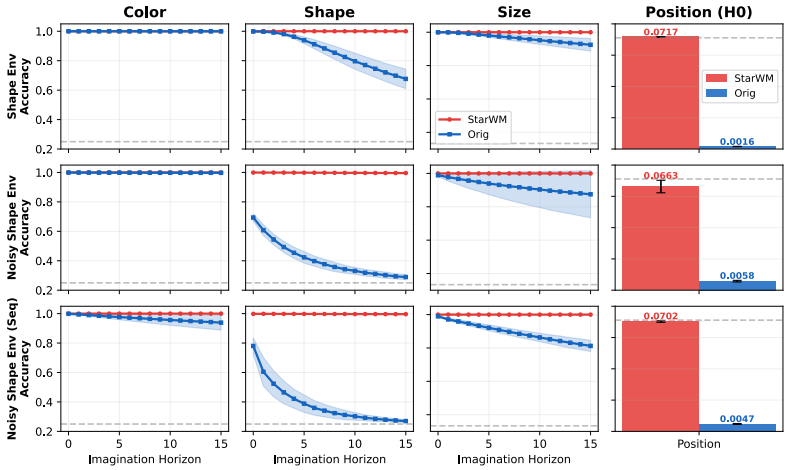


Figure 12: **E.2: Latent space probing across NoisyShape variants.** DreamerV3's performance degrades as irrelevant complexity increases, while STARWM is invariant to distraction complexity.

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.

E.3 Ablation Study of Number of Queries N

We evaluate STARWM with $N \in \{1, 2, 4, 8\}$ queries on three DMC tasks under the Video Hard Sequential (VH) setting. All other hyperparameters are held fixed; results are mean $\pm$ SE across 3 seeds. The results are shown in Figure 13 and Table 8.

Table 8: **E.3: Effect of query count N on DMC VH** (episode return, 3M steps, 3 seeds).

N	Walker Walk	Cheetah Run	Cartpole Swingup
1	170.48 $\pm$24.59	181.22 $\pm$63.81	191.54 $\pm$6.46
2	**928.73** $\pm$34.39	**461.71** $\pm$15.09	268.62 $\pm$60.34
4	902.31 $\pm$37.20	452.77 $\pm$22.76	308.00 $\pm$12.04
8	585.14 $\pm$72.84	387.97 $\pm$78.67	**394.90** $\pm$98.07

A single query (N=1) fails across all tasks: one soft attention mask cannot simultaneously localize multiple body parts of an articulated agent, and the resulting entity token averages over attended and unattended regions. Increasing to N=2 recovers strong performance, and N=4 (our default) performs comparably. At N=8, results are mixed: Cartpole Swingup improves, while Walker Walk degrades. We use N=4 throughout all experiments as a reasonable default that balances expressiveness and robustness across tasks without per-task tuning.

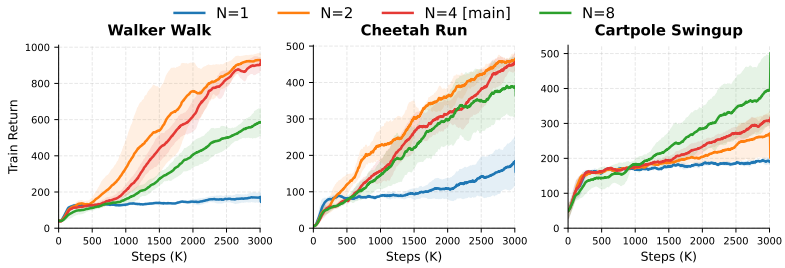


Figure 13: **E.3: Ablation Study of Number of Queries N.**

E.4 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 5.

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 756, 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.4.1 Failure mode of STARWM on `Cartpole Swingup VH`

`Cartpole Swingup VH` is the task where STARWM most strongly underperforms STARWM-R. Figure 14 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.

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.4): when self-

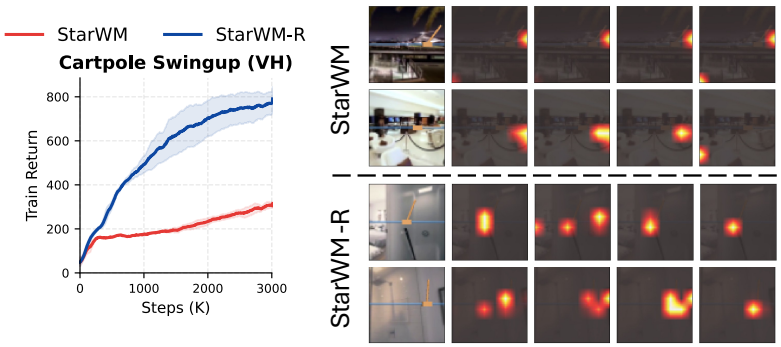


Figure 14: **E.4.1: Default STARWM's attention drifts on Cartpole VH; reward prediction recovers it.** Each row shows the input observation followed by the attention maps for the four queries. Default STARWM (top right) concentrates attention near the track endpoints without query specialization. STARWM-R (bottom right) localizes the cart and pole accurately, with queries specializing to distinct object regions.

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.5 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 9 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 9.

Table 9: **E.5: 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	✓	✓	—

All configurations succeed on NoisyShape. We first evaluate all five configurations in Table 9 on NoisyShape. Figure 15 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.

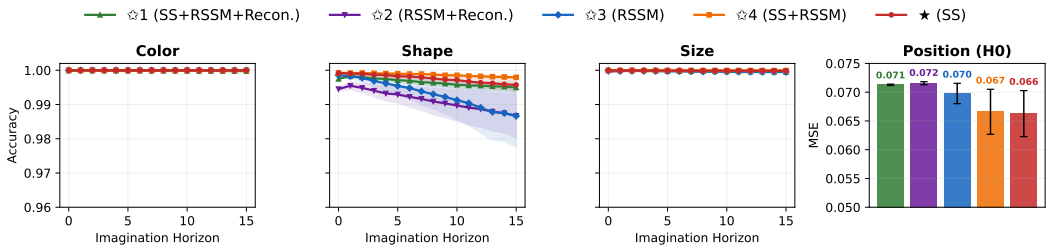


Figure 15: **E.5: Probe accuracy on NoisyShape under five gradient isolation configurations.** NoisyShape probe accuracy is invariant to gradient isolation choice.

Table 10: **E.5: 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 9. 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

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 9 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 10 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 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 17 shows attention maps for the four learned queries on Walker Walk, and Table 11 reports the corresponding quantitative statistics for Q_0 on Cheetah Run.

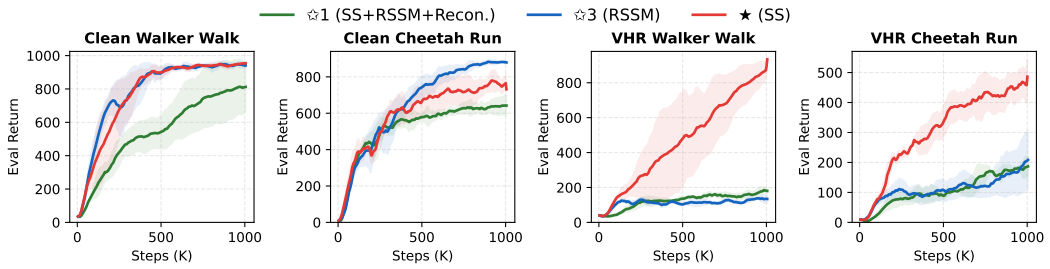


Figure 16: **E.5: Learning curves for the gradient isolation ablation on DMC.** Episode return over training for the three configurations of Table 10. Removing detach B causes attention to be co-opted by reconstruction gradients; removing SS leaves RSSM gradients as the only attention signal, insufficient to disambiguate foreground from background under video distractors. Full StarWM combines detach B with SS guidance.

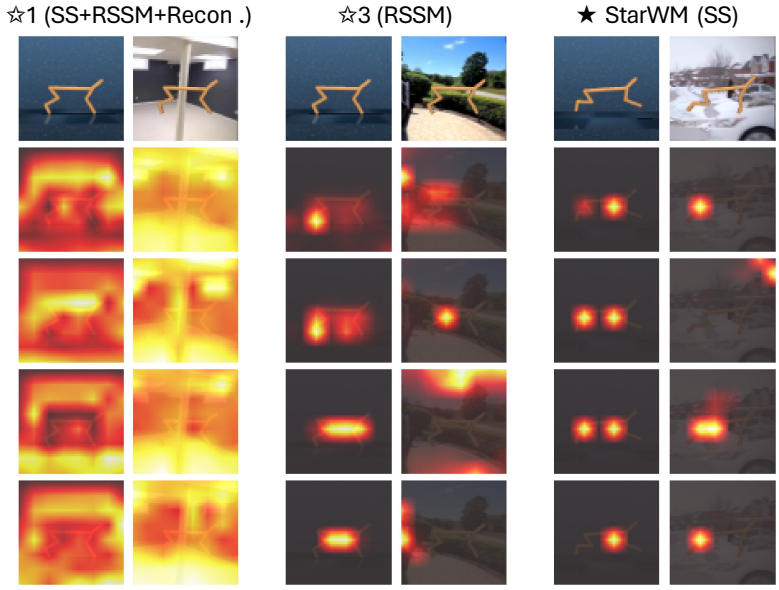


Figure 17: **E.5: Gradient isolation determines attention focus on DMC.** Attention maps from the four learned queries on Walker Walk under VHR, comparing the three configurations of Table 10. Top row: input observation. Rows 2–5: attention map for each of the four queries. Left (☆1: SS+RSSM+Recon., 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).

☆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 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.

Table 11: **E.5: 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

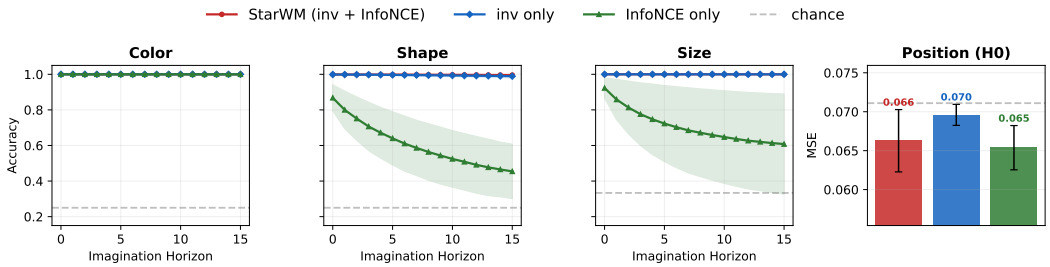


Figure 18: **E.6: Inverse dynamics is the primary attention signal.** Probe accuracy on the RSSM latent for full STARWM (inverse + contrastive), inverse-only, and contrastive-only variants on NoisyShape. Inverse dynamics alone is sufficient for near-perfect attribute retention; contrastive alone fails to localize the foreground.

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.6 Contrastive Signal Conflict Analysis

STARWM's default attention guidance combines inverse dynamics and contrastive learning. While Appendix E.5 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 18 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 19 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

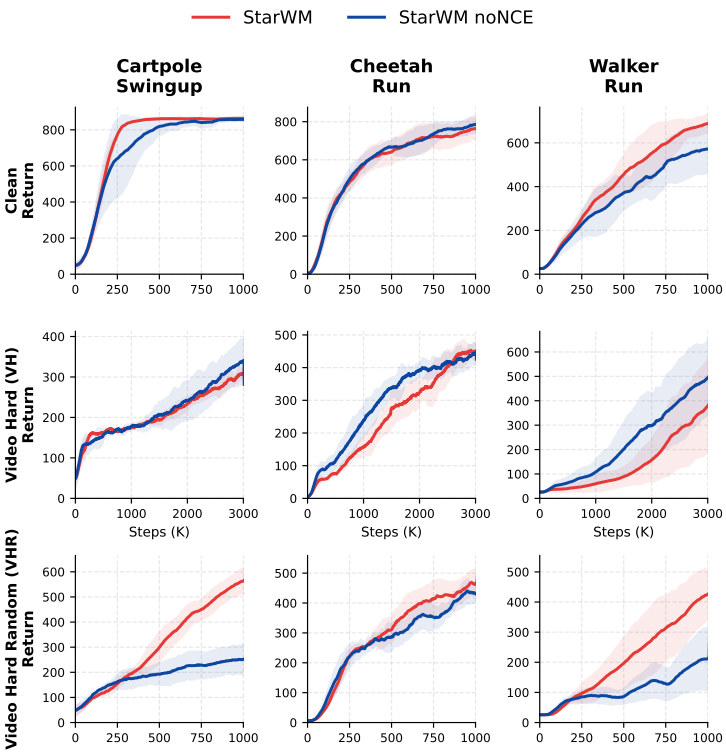


Figure 19: **E.6: Episode return for full StarWM (inverse + contrastive) versus inverse-only across Clean, VH (sequential video), and VHR (random video).** Contrastive supervision is essential under VHR, neutral under Clean and under VH.

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, 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.4 how reward prediction can be added as an additional attention-guidance signal that breaks the ambiguity.

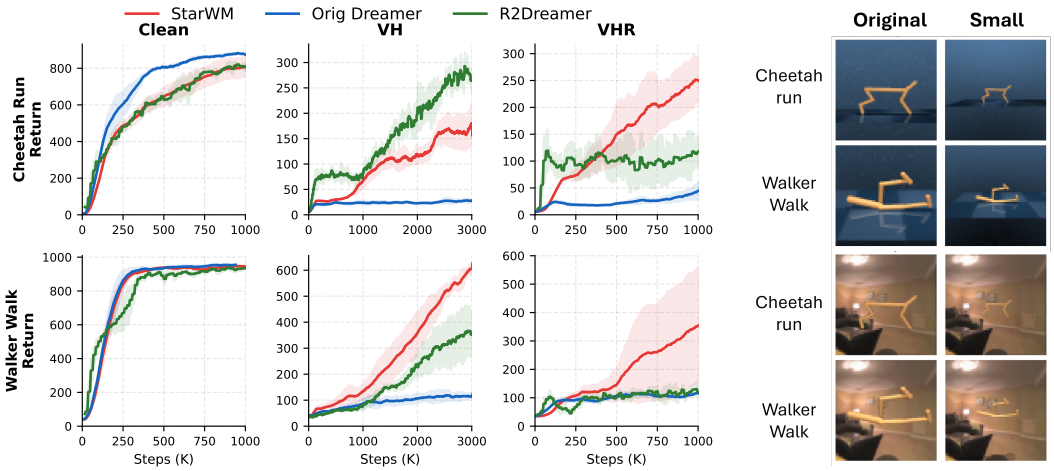


Figure 20: **E.7: Small-object stress test.** ***Left:*** Learning curves for Cheetah Run and Walker Walk with the agent rescaled to half its default linear size (one quarter pixel area), under Clean, VH (3M steps), and VHR (1M steps) settings. ***Right:*** Illustration of the environment setup.

E.7 Sensitivity to Object Scale

Our default attention resolution of 8×8 spatial tokens (Appendix C.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.

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 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.

E.8 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.

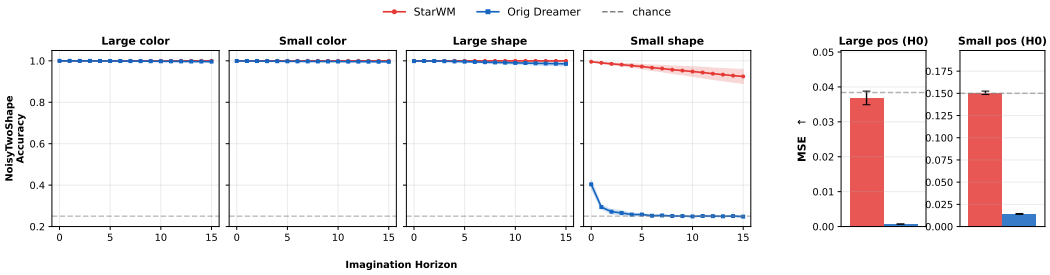


Figure 21: **E.8: Latent probe results on NoisyTwoShape.** Probe accuracy for color, shape of the large and small object across H0 to H15 imagination horizon, and position MSE at H0. Higher MSE on position indicates correct irrelevant treatment (positions are re-sampled each step). Dashed lines indicate 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).

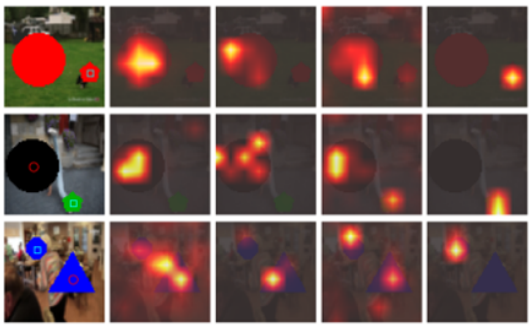


Figure 22: **E.8: StarWM attention maps on NoisyTwoShape.** Columns show the input image followed by the four query attention maps.

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$85%). DreamerV3 by contrast fails on the smaller object from the encoding step itself: its small-shape probe is at chance throughout, 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 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.