LATENT WORLD MODELS NEED TO ALSO SEE THE DETAILS: KEYFRAME-AWARE VISUAL ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Latent world models built on the Joint Embedding Predictive Architecture (JEPA), such as LeWorldModel, learn environment dynamics entirely in latent space and avoid the cost of pixel-level reconstruction. This purely latent paradigm, however, has two fundamental limitations. In fine-grained manipulation, the target object typically occupies only a tiny fraction of the frame, and the encoder's information bottleneck can suppress visual details critical to success. Moreover, the absence of any decoder that maps latents back to an observable space makes the long-horizon accuracy of latent rollouts unverifiable, allowing an overconfident yet physically implausible plan to pass unnoticed. We introduce **KAVA** (Keyframe-Aware Visual Alignment), a hybrid framework that augments a JEPA next-embedding objective with a temporal-saliency key-frame selector—combining the JEPA-loss gradient norm with an action-perturbation uncertainty score—and applies a pixel-level reconstruction loss only to a small subset of information-dense frames per trajectory. The reconstruction loss is decoupled from the predictor, preserving the latent signal the JEPA objective is designed to learn, and the decoder simultaneously serves as an online probe for the consistency of imagined rollouts. Across four image-based control benchmarks, KAVA achieves the best average planning success rate among latent world-model baselines. Selective, saliency-driven observation supervision is therefore a promising recipe for retaining the efficiency of reconstruction-free world models without sacrificing visual fidelity or rollout verifiability.

1 INTRODUCTION

A world model that internalises environmental dynamics is a cornerstone of data-efficient decision-making and a central building block for general-purpose robotic agents (Ha & Schmidhuber, 2018; Xing et al., 2025). Modern world models have converged on two principal routes (Jha et al., 2026). *Reconstruction-based* designs, exemplified by the Dreamer series (Hafner et al., 2020; 2021; 2023) and IRIS (Micheli et al., 2022), use an encoder–decoder architecture to compress high-dimensional observations into a latent state and supervise the encoder with pixel-level reconstruction. *Reconstruction-free* designs, notably TD-MPC2 (Hansen et al., 2024) and LeWorldModel (LeWM) (Maes et al., 2026), abandon the decoder entirely and learn dynamics in latent space, trading some pixel fidelity for stronger scalability and training stability. Among them, LeWM is the first JEPA (Dawid & LeCun, 2023; Assran et al., 2023) trained end-to-end from raw pixels without exponential moving averages, stop-gradients, or pretrained encoders, and has set the new efficiency bar for image-based control.

Both routes, however, share a common blind spot (Jha et al., 2026). In fine-grained manipulation, the task-critical object often occupies only a small fraction of the visual frame (Okada & Taniguchi, 2021; Fu et al., 2021). When such an observation is compressed into a few dozen latent dimensions, the object's distinctive features can be suppressed at the information bottleneck, even though minute pose changes may demand entirely different control strategies. More fundamentally, the central value of a world model lies in its look-ahead capability: rolling out candidate trajectories in latent space and selecting the highest-return path (Xing et al., 2025). Yet because reconstruction-free methods have no decoder to map latents back to an observable space, the long-horizon accuracy of these rollouts is unverifiable, and accumulating errors can yield overconfident yet physically implausible plans (Huang et al., 2026b).

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

To address both limitations, we introduce **KAVA** (Keyframe-Aware Visual Alignment). KAVA augments a JEPA world model with a sparse, saliency-driven reconstruction objective: only a small fraction of frames per trajectory—those on which the JEPA loss is locally sensitive and the predictor is locally uncertain—is decoded and supervised in pixel space, with the reconstruction loss decoupled from the predictor. The same decoder doubles as an online probe that lets us inspect, at inference time, whether an imagined rollout still corresponds to a physically plausible state.

The evaluation results on image-based control benchmarks show that KAVA matches or exceeds every JEPA-family baseline on all four environments, improving over LeWorldModel by 4.75 percentage points on average (up to 12 pp on OGB-Cube) and over the strong JEPA baseline (Sub-JEPA) by 8.75 percentage points on average, with a large absolute gain on the sparse-reward OGB-Cube task.

Our contributions are threefold: (1) We identify and characterise the dual limitations of pure latent world models—small-object information loss at the encoding bottleneck and the absence of rollout verification—and argue that both can be mitigated by selective, saliency-driven observation supervision. (2) We propose KAVA, a hybrid training objective that couples the JEPA next-embedding prediction loss with a selective key-frame reconstruction loss through a temporal-saliency selector, achieving an adjustable trade-off between computational efficiency and visual fidelity. (3) KAVA achieves the best average planning success rate among several representative latent world-model baselines across four image-based control benchmarks, and we show that the decoder probe serves as a useful online indicator of long-horizon rollout reliability.

2 Related Work

2.1 Latent-Space World Models

Latent-space world models compress observations into a compact state and learn dynamics in that space, avoiding planning directly in the high-dimensional pixel domain. The paradigm traces from PlaNet (Hafner et al., 2019), which introduced a convolutional encoder and a feed-forward dynamics model for gradient-based trajectory optimisation, through the Dreamer family (Hafner et al., 2020; 2021; 2023) whose Recurrent State-Space Model couples a deterministic recurrent state with stochastic latents for partial observability, to token-level designs such as IRIS (Micheli et al., 2022) that discretise observations with a VQ-VAE (Van Den Oord et al., 2017) and model transitions with an autoregressive Transformer (Vaswani et al., 2017). TD-MPC2 (Hansen et al., 2024) pushes the same line further by removing the decoder and scaling across several continuous-control tasks via task embeddings. Most recently, LeWorldModel (LeWM) (Maes et al., 2026) establishes JEPA as the first design trained stably end-to-end from raw pixels using only a next-embedding prediction loss and a SIGReg anti-collapse regulariser (Balestriero & LeCun, 2025), and TD-JEPA (Bagatella et al., 2026) couples policy-conditioned multi-step latent prediction with a temporal-difference objective for zero-shot reinforcement learning. This path has delivered strong efficiency, but the removal of the observation decoder also removes the most direct signal for preserving fine-grained visual structure.

2.2 Reconstruction-Based World Models

Reconstruction-based world models keep pixel-level supervision in the loop through a decoder that maps latents back to observations. The Dreamer series (Hafner et al., 2020; 2021) pioneered this design and has progressively balanced reconstruction fidelity against long-horizon dynamics accuracy; DIAMOND (Alonso et al., 2024) argues that discrete latent bottlenecks often discard visual details critical for control and shows that a diffusion decoder can preserve fine appearance at competitive cost, while large-scale action-conditioned simulators such as UniSim (Yang et al., 2024) and Cosmos (Agarwal et al., 2025; 2026) extend the same recipe to photorealistic video. The cost, however, is well known: pixel-level losses are easily dominated by low-level features such as texture, colour, and lighting that are only weakly correlated with task success (Pan et al., 2022; Wan et al., 2023; Hansen et al., 2024), so the encoder can spend capacity on task-irrelevant pixels. KAVA inherits the decoder's fidelity benefit but applies the reconstruction loss *only* to a small set of temporally salient key frames with a lightweight convolutional decoder, rather than reconstructing every frame as DIAMOND does, capturing the structural information that dense reconstruction dilutes.

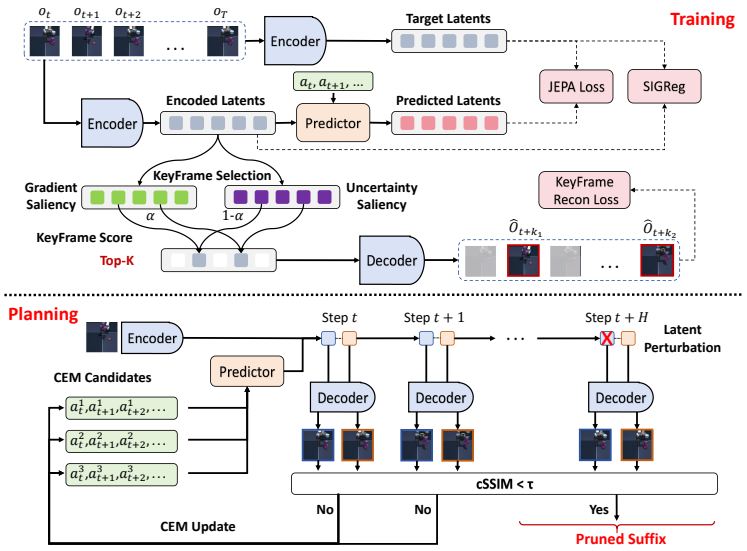


Figure 1: Overview of the KAVA framework. A shared ViT-Tiny encoder feeds a JEPA next-embedding predictor; KAVA augments this with a saliency-driven key-frame selector that combines the JEPA-loss gradient norm with action-perturbation uncertainty, then applies a pixel-level reconstruction loss only to the selected key frames, with the reconstruction loss decoupled from the predictor. The decoder is used only for the key-frame loss during training and as an observation-space diagnostic for latent rollouts at planning time.

2.3 Key-Frame Selection and Sparse Supervision

Classical key-frame selection identifies visually salient frames via inter-frame differences, optical flow, or clustering (Wolf, 1996; Dirfaux, 2000; Guan et al., 2012; Alexoudi et al., 2023), and high-masking-ratio self-supervised learners such as VideoMAE (Tong et al., 2022) have shown that extreme sparsification can still be informative. Masked World Models (Seo et al., 2023) sparsify supervision *spatially*, masking convolutional features within each frame; KAVA is complementary, selecting *which frames*—rather than which patches—deserve pixel supervision. In the world-model setting, a "critical" frame is one of greatest task progression or state-transition uncertainty, not merely one of maximal pixel change. KeyWorld (Li et al., 2025), SKIP (He et al., 2026) and EAWM (Peng et al., 2026) exploit this insight by selecting task-relevant events (approach, contact, grasp) and interpolating the gaps, but rely on offline trajectory simplification and pretrained diffusion backbones. KAVA differs in two ways: the saliency selector is applied *online* during training, so the criterion adapts as the world model matures, and it fuses two complementary signals—the JEPA-loss gradient norm and an action-perturbation variance—rather than relying on a single pretrained module.

3 Methodology

KAVA augments the latent world model above with two additions. Section 3.2 describes a saliency-driven key-frame selector coupled to a hybrid training objective, and Section 3.3 introduces a consistency-SSIM diagnostic that exploits the decoder to validate latent rollouts at inference time.

3.1 Problem Setup

As illustrated in Figure 1, a latent world model consists of an encoder and a predictor. Given a raw pixel observation $\mathbf{o}_t \in \mathbb{R}^{3\times H\times W}$, the encoder maps it to a compact latent representation

$$\mathbf{z}_t = f_\theta(\mathbf{o}_t) \in \mathbb{R}^{d_z}, \tag{1}$$

and the predictor estimates the next latent

$$\hat{\mathbf{z}}_{t+1} = g_\phi(\mathbf{z}_{t-h+1:t}, \mathbf{a}_{t-h+1:t}), \tag{2}$$

conditioned on a short history of h latents and actions, with $\mathbf{a}_t \in \mathbb{R}^{d_a}$. The predictor is trained to match the target latent $\mathbf{z}_{t+1} = f_\theta(\mathbf{o}_{t+1})$ obtained by encoding the subsequent observation.

We consider a fully offline and reward-free setting, in which the model is trained only from pre-collected trajectories of observations and actions, without access to rewards or task labels, following the JEPA line of work:

$$\mathcal{D} = \{(\mathbf{o}_{1:T}, \mathbf{a}_{1:T})\}, \tag{3}$$

where each trajectory of length T is partitioned into a short history of h context frames and $T - h$ prediction targets. KAVA is built on top of LeWM and inherits its three modules: a ViT-Tiny encoder f_θ (Dosovitskiy et al., 2020), an autoregressive Transformer predictor g_ϕ with h-frame history, and the SIGReg anti-collapse regulariser. The only architectural addition is a small convolutional decoder d_ψ that maps each latent back to pixel space; this decoder is the central handle by which KAVA injects selective observation supervision.

3.2 Keyframe Selection and Hybrid Training Objective

KAVA's central design is to invoke the decoder on a small, saliency-selected subset of frames per trajectory. We score every timestep t by combining two complementary signals.

Temporal-gradient saliency. The first signal captures how sensitive the JEPA objective is to the latent at t. After computing the JEPA loss on a batch, we back-propagate to the latent tensor and record the gradient norm:

$$s_t^{\text{grad}} = \left\| \nabla_{\mathbf{z}_t} \mathcal{L}_{\text{JEPA}} \right\|_2 . \tag{4}$$

A large s_t^{grad} indicates that the model finds this timestep "hard" or "surprising".

Action-perturbation uncertainty. The second signal captures the local action sensitivity of the learned dynamics. We perform N_{unc} forward passes through the predictor, each time adding Gaussian noise $\boldsymbol{\epsilon}^{(n)} \sim \mathcal{N}(\mathbf{0}, \sigma_a^2 \mathbf{I})$ to the actions in the context window, and measure the variance of the resulting predictions:

$$s_t^{\text{unc}} = \frac{1}{d_z} \sum_{j=1}^{d_z} \text{Var}_n \left[\left[g_\phi(\mathbf{z}_{t-h+1:t}, \mathbf{a}_{t-h+1:t} + \boldsymbol{\epsilon}^{(n)}) \right]_j \right] . \tag{5}$$

Strictly speaking, s_t^{unc} estimates the predictor's local action sensitivity—a Monte-Carlo proxy for the squared action-Jacobian magnitude—rather than Bayesian epistemic uncertainty; we use it to flag timesteps where the action choice matters most (e.g., contact events), so that an observation there carries the most information.

Combined keyframe score. The two signals are min-max normalised across the trajectory (the normalised scores are denoted by an overbar) and combined with a mixing coefficient $\alpha \in [0, 1]$:

$$s_t = \alpha \cdot \bar{s}_t^{\text{grad}} + (1 - \alpha) \cdot \bar{s}_t^{\text{unc}}. \tag{6}$$

The top-$k = \max(1, \lfloor \rho T \rfloor)$ scoring timesteps form the keyframe set $\mathcal{K}_\tau$, with $\rho \in (0, 1)$ controlling the selection ratio. We default to $\rho = 0.15$, $\alpha = 0.6$, $N_{\text{unc}} = 4$, and $\sigma_a = 0.1$.

Hybrid training objective. The total loss combines the JEPA-style dynamics prediction loss, the key-frame reconstruction loss, and the SIGReg regulariser:

$$\mathcal{L} = \mathcal{L}_{\text{JEPA}} + \lambda_{\text{SIGReg}} \cdot \mathcal{L}_{\text{SIGReg}} + \lambda_{\text{recon}} \cdot \underbrace{\frac{1}{|\mathcal{K}_\tau|} \sum_{t \in \mathcal{K}_\tau} \| d_\psi(\mathbf{z}_t) - \mathbf{o}_t \|_2^2}_{\mathcal{L}_{\text{recon}}^{\mathcal{K}}} . \tag{7}$$

The key-frame term $\mathcal{L}_{\text{recon}}^{\mathcal{K}}$ supervises the encoder and the decoder on the most informative frames. Crucially, the reconstruction branch shares no computational path with the predictor: gradients from $\mathcal{L}_{\text{recon}}^{\mathcal{K}}$ flow to f_θ and d_ψ but never to g_ϕ. The predictor therefore continues to receive a pure latent signal identical in spirit to a vanilla JEPA, while the encoder receives explicit pixel-level supervision only on the most informative frames. The decoder is thus a lightweight scaffold: during training it is invoked only on the selected key frames, and at planning time it is used solely as the cSSIM diagnostic probe (Section 3.3), never to generate observations for the rollout or the planner.

Table 1: Planning success rate (%, higher is better; mean ± std over five seeds) across methods. Bold marks the best result per method.

Environment	PLDM	DINO-WM	LeWM	Sub-JEPA	KAVA (w/o LRV)	KAVA (Ours)
Two-Room	$96.00_{\pm 1.50}$	$\mathbf{100.00}_{\pm 0.00}$	$92.00_{\pm 4.23}$	$94.00_{\pm 3.03}$	$94.00_{\pm 2.73}$	$98.00_{\pm 2.10}$
Reacher	$80.00_{\pm 3.40}$	$77.00_{\pm 3.84}$	$96.00_{\pm 4.42}$	$84.00_{\pm 3.78}$	$94.00_{\pm 1.14}$	$\mathbf{97.00}_{\pm 2.33}$
PushT	$76.00_{\pm 3.10}$	$76.00_{\pm 2.57}$	$\mathbf{98.00}_{\pm 6.53}$	$90.00_{\pm 4.72}$	$93.00_{\pm 4.19}$	$\mathbf{98.00}_{\pm 3.12}$
OGB-Cube	$63.00_{\pm 4.25}$	$87.00_{\pm 1.48}$	$76.00_{\pm 6.69}$	$78.00_{\pm 6.71}$	$79.00_{\pm 4.26}$	$\mathbf{88.00}_{\pm 2.50}$

3.3 Latent Rollout Validation via Consistency-SSIM

Because KAVA has a decoder, any latent state can be projected back into pixel space. We exploit this capability to introduce a *consistency-SSIM* (cSSIM) diagnostic that serves two purposes: (i) measuring the local smoothness of the decoder around a latent state, and (ii) providing an online signal for detecting and discarding unreliable steps during long-horizon latent rollouts.

For a latent state $\mathbf{z}_t$ (a predictor output $\hat{\mathbf{z}}_t$ at rollout time), we measure the structural similarity (SSIM) (Wang et al., 2004) between its decoded image and the image decoded from a perturbed version of the same latent:

$$\mathrm{cSSIM}_t \;=\; \mathrm{SSIM}(d_\psi(\mathbf{z}_t),\; d_\psi(\mathbf{z}_t + \eta))\,, \tag{8}$$

where $\eta \sim \mathcal{N}(0, \varepsilon^2 I)$ with $\varepsilon = 10^{-3}$ (see Appendix C for more details). A high cSSIM_t means that infinitesimal latent perturbations leave the decoded image essentially unchanged — the decoder is locally smooth at $\mathbf{z}_t$, which is exactly the behaviour one expects if $\mathbf{z}_t$ is to be treated as a meaningful state coordinate. Conversely, a low cSSIM_t indicates that the decoder is sensitive to small displacements in latent space, suggesting that $\mathbf{z}_t$ lies near a boundary of the learned manifold where the representation may be distorted or unreliable. We stress that cSSIM is a *one-sided, necessary-condition* signal: it certifies local smoothness, not physical correctness, since a smoothly wrong rollout could still score high. We therefore use it only to *prune* rollouts that drift into ill-conditioned regions of the latent manifold, never to verify correct ones.

Per-step rollout monitoring. During planning, the predictor g_ϕ unrolls a candidate trajectory $\{\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_H\}$ in latent space, where H is the planning horizon. We compute cSSIM_t for *every* step t along the rollout, forming a consistency profile

$$\mathbf{c}_H \;=\; \big[\mathrm{cSSIM}_1,\; \mathrm{cSSIM}_2,\; \ldots,\; \mathrm{cSSIM}_H\big]. \tag{9}$$

A healthy rollout exhibits a consistently high profile (all entries close to 1), indicating that the imagined trajectory remains on the well-structured region of the latent manifold. When the predictor drifts into an ill-conditioned region — e.g. due to error accumulation, out-of-distribution action sequences, or adversarial perturbations from the CEM search — cSSIM_t drops sharply at the step where the drift begins.

Step-level detection and pruning. We flag a rollout step as unreliable when its consistency score falls below a threshold τ:

$$\mathrm{unreliable}(t) \;\triangleq\; \big[\mathrm{cSSIM}_t < \tau\big]. \tag{10}$$

We calibrate τ separately for each environment as the 5th-percentile convergence value of cSSIM_t on held-out validation trajectories, so the vast majority of in-distribution latents score above it. Let $t^\star$ be the first step with $\mathrm{cSSIM}_{t^\star} < \tau$; we treat all subsequent steps $t \geq t^\star$ as contaminated by the same drift and discard the entire suffix:

$$\hat{z}_t \;\leftarrow\; \texttt{NULL}, \qquad \forall t \;\geq\; t^\star. \tag{11}$$

Candidate trajectories are ranked by the task's goal-reaching scoring function, identical across all compared methods; any trajectory with a detected failure is assigned the worst candidate score observed in the current CEM iteration and excluded from the top-k candidates passed to the next iteration, preventing overconfident but physically implausible plans from surviving the search.

4 Experiments

We design our experiments to answer four research questions: (1) **RQ1 (Effectiveness):** Does KAVA's selective key-frame reconstruction improve planning success rates over state-of-the-art

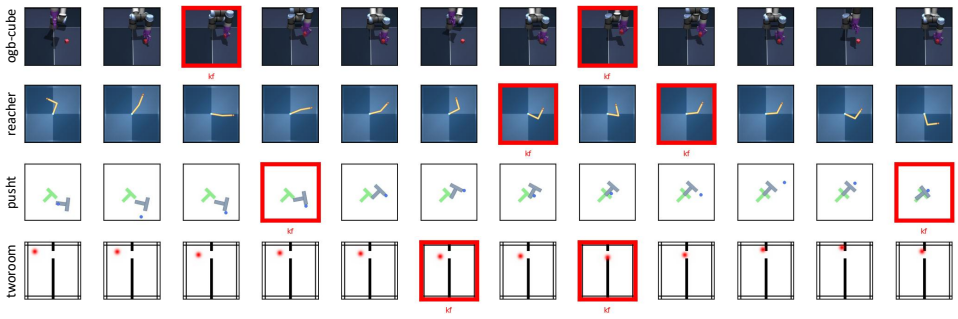


Figure 3: Saliency-driven key-frame selection on trajectories. For each environment, the red box shows the selection of key frames. The reconstructions remain visually faithful despite only 15% of frames being supervised.

latent world models (LeWM, Sub-JEPA) and zero-shot visual planners (DINO-WM, PLDM)? (2) **RQ2 (Robustness):** How sensitive is KAVA to the key hyperparameters that govern the key-frame selection? (3) **RQ3 (Mechanism):** Where do the gains come from? Does the key-frame selector pick task-relevant frames, and does the hybrid objective preserve both global latent geometry and small-object detail? (4) **RQ4 (Diagnostics):** Does the decoder-only rollout validation provide observation-space signals that pure latent baselines cannot produce?

4.1 Experimental Setup

Environments. We evaluate on four image-based continuous-control benchmarks (**Two-Room** (Zhou et al., 2024), **PushT** (Chi et al., 2025), **OGB-Cube** (Park et al., 2025), **Reacher** (Tassa et al., 2018)) that cover a range of dynamics, reward sparsity, and visual complexity. All observations are $3 \times 224 \times 224$ RGB frames with a frame skip of 5.

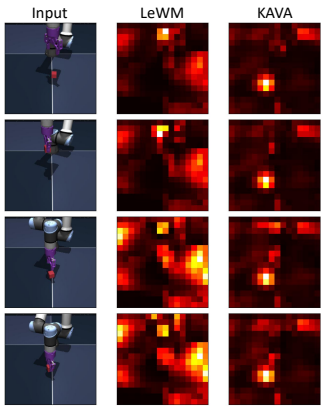


Figure 2: Encoder attention maps (OGB-Cube) for KAVA vs. LeWM. Each cell shows the attention row from the patch covering the moving block's centre, averaged over ViT heads. KAVA's attention is tightly localised near the block centre, while LeWM's is diffuse and biased toward corners.

Baselines. We compare KAVA against four representative latent world-model families that span the reconstruction–reconstruction-free spectrum: (1) **PLDM** (Sobal et al., 2025): a reward-free planning method that stress-tests latent dynamics on the same benchmarks; (2) **DINO-WM** (Zhou et al., 2024): a zero-shot visual planner that builds a world model on top of frozen DINO features; (3) **LeWM** (Maes et al., 2026): the reconstruction-free JEPA world model that KAVA directly extends; (4) **Sub-JEPA** (Zhao et al., 2026): the most recent JEPA-style baseline, which augments LeWM with subspace Gaussian regularization. We restrict the controlled comparison to latent world-model baselines: dense-reconstruction or reward-supervised methods (e.g., DreamerV3, TD-MPC2) follow different training regimes and are not directly comparable in our offline reward-free setting. For LeWM, Sub-JEPA, and KAVA we use matched compute (same encoder, predictor, optimiser, schedule, dataset splits, and seeds); the only differences across the three are the regulariser and the key-frame mechanism. KAVA adds ≈ 6.8M decoder parameters ($\approx 32\%$ of the ≈ 21.2M total), but because the decoder is invoked on only a $\rho = 0.15$ fraction of training frames, the training wall-clock overhead stays under 15%; at planning time, the cSSIM probe adds two lightweight decoder forward passes and one SSIM evaluation per candidate step. For PLDM and DINO-WM, we evaluated them under the same CEM planner and success criterion from the original releases (see Appendix B for more details).

4.2 Main Results

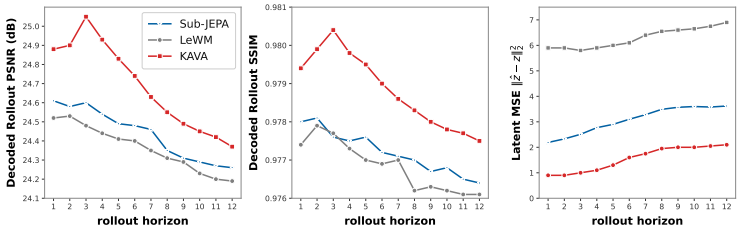


Figure 5: Latent rollout quality versus planning horizon on Two-Room. KAVA's rollouts degrade more slowly than those of LeWM and Sub-JEPA as the horizon grows, consistent with selective reconstruction and cSSIM-based pruning suppressing error accumulation. (see Appendix C)

Table 1 reports the planning success rate of KAVA and baseline methods on four benchmarks. KAVA achieves the best or tied-best result among JEPA-family methods on all tasks, addressing **RQ1**.

Overall performance. KAVA ranks first or tied-first on three of four environments (Reacher, PushT, OGB-Cube) and second on Two-Room, achieving the strongest aggregate performance among all latent world-model baselines. Compared with Sub-JEPA, the strongest prior JEPA baseline, KAVA improves by $+4$ to $+13$ percentage points across all environments. Relative to LeWM, our direct base model, KAVA improves by up to $+12$ pp on OGB-Cube while matching it on Reacher (97% vs. 96%). KAVA also attains a lower variance than LeWM on Reacher (2.33 vs. 4.42), suggesting that selective reconstruction stabilises training across seeds.

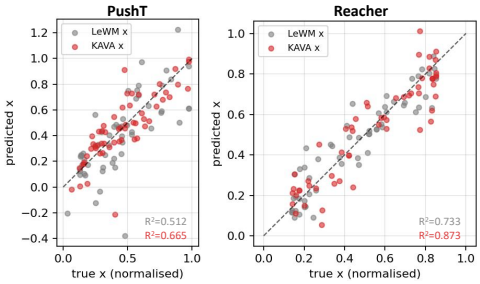


Figure 4: State-probe analysis on PushT and Reacher: object position predicted from a linear regression on top of a frozen KAVA encoder, plotted against ground-truth object position.

Sparse-reward advantage. The most substantial gain appears on OGB-Cube, the only task with a sparse binary reward. Here, pixel-level supervision on salient frames provides the clearest signal for locating the small target cube: KAVA reaches 88% of planning success rate, outperforming LeWM (76%) and Sub-JEPA (78%) by $+12$ and $+10$ pp respectively. This validates our core hypothesis—that selective reconstruction anchors the encoder on task-critical visual details precisely when the reward signal cannot.

Comparison with zero-shot planners. KAVA matches or exceeds DINO-WM on PushT (98% vs. 76%) and Reacher (97% vs. 77%), and remains competitive on Two-Room and OGB-Cube, despite being trained end-to-end from raw pixels without pretrained features.

4.3 Ablation Study and Hyperparameter Sensitivity

This section ablates the latent rollout validation component and sweeps the key-frame selector hyper-parameters, addressing **RQ2** and **RQ4**.

Effect of latent rollout validation. We evaluate a KAVA variant that retains the key-frame reconstruction objective but disables the consistency-SSIM (cSSIM) rollout-pruning mechanism at planning time (Table 1, w/o LVR). Removing rollout validation causes consistent degradation across all environments. The gap is largest on OGB-Cube, where sparse rewards make it hardest to recover from a physically implausible rollout through reward guidance alone. This suggests that without cSSIM-based pruning, the predictor's accumulated errors in long-horizon imagination survive the CEM search and produce overconfident yet invalid plans. Consistent with this, Figure 5 shows that KAVA's latent rollouts degrade more slowly than those of LeWM and Sub-JEPA as the planning horizon grows. The ablated variant itself is mixed relative to LeWM—better on Two-Room and OGB-Cube, but worse on PushT and Reacher—indicating that key-frame reconstruction and cSSIM-based validation contribute complementary gains rather than either one dominating.

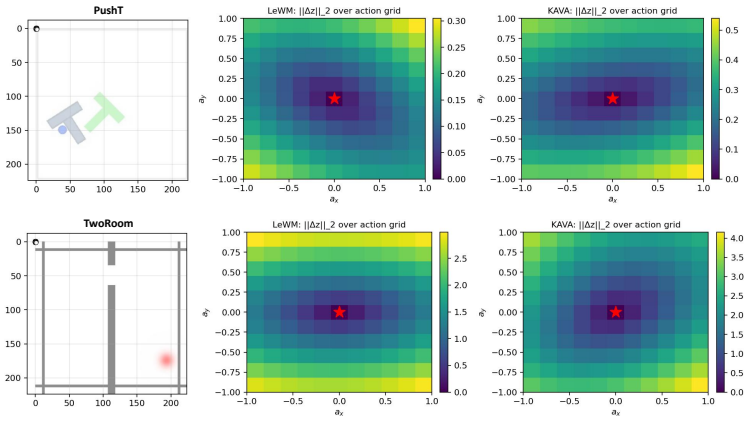


Figure 6: Action-sensitivity probe of the predictor on a fixed context frame (PushT and Two-Room). We sweep an 11×11 action grid over $[-1, 1]^2$ and plot the resulting predicted next-latent for KAVA and LeWM. KAVA's response cloud is broader and more isotropic (mean $\|\Delta \mathbf{z}\|_2 = 0.219$ vs. 0.121), with peak displacement in the four corners.

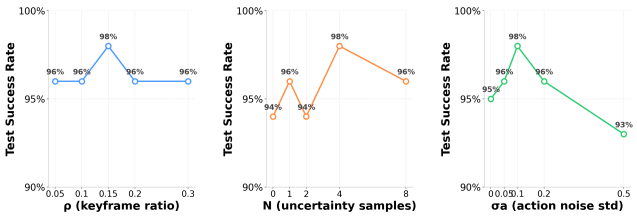


Figure 7: Hyperparameter sensitivity on PushT. *Left:* key-frame ratio ρ (all values $\geq 96\%$). *Centre:* uncertainty sample count N_{unc} (peak at 4). *Right:* action-perturbation scale σ_a (peak at 0.1, robust across $10\times$ range).

Key-frame ratio ρ. Success remains $\geq 96\%$ across all five values $\rho \in \{0.05, 0.10, 0.15, 0.20, 0.30\}$, peaking at the default $\rho = 0.15$ (98%). The method is therefore robust to ρ over a $6\times$ range: even 5% frame reconstruction captures most of the gain, and larger ratios yield diminishing returns.

Uncertainty sample count N_{unc}. The curve is non-monotonic (Figure 7, centre): $N_{\text{unc}} = 4$ achieves the peak at 98%, while $N_{\text{unc}} = 0$ (gradient-only selection) drops to 94%. The gap confirms that action-perturbation uncertainty contributes complementary information beyond gradient saliency.

Action-perturbation scale σ_a. Success stays within 93%–98% across a $10\times$ range of σ_a, with the peak at the default 0.1 (Figure 7, right). Both extremes degrade performance: too little noise ($\sigma_a = 0.0$) loses 3 pp, too much ($\sigma_a = 0.5$) loses 5 pp. The wide tolerance means practitioners need not finely tune σ_a per environment.

4.4 Qualitative Analysis

This subsection probes *why* KAVA works, addressing **RQ3** and **RQ4**.

Key-Frame Selection. Figure 3 visualises the saliency-driven selector on representative trajectories from all four environments. The selector consistently identifies semantically meaningful frames: scene transitions (the Two-Room doorway crossing), physical contact events (PushT block touching the wall), and motion reversals (Reacher arm changing direction, OGB-Cube corner rotating into view). The decoded reconstructions of the selected frames remain visually faithful despite only 15% of the trajectory being supervised—evidence that the encoder retains sufficient information density at the selected timesteps.

Encoder Attention. We extract the attention matrix from the last ViT layer, averaged over heads, and visualise the row corresponding to the patch covering the centre of the moving object (Figure 2).

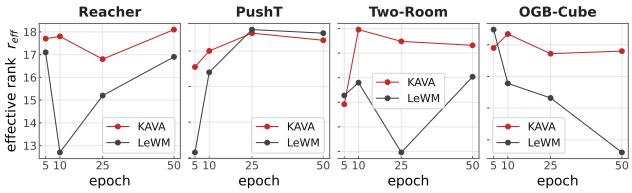


Figure 8: Effective rank of latents versus training epochs on four tasks. KAVA achieves higher effective rank than LeWM through most of training, indicating richer latent utilisation.

KAVA's attention map shows concentrated, bright activations tightly localised around the block centre, whereas LeWM's attention is more diffuse and skewed toward object corners and background edges. This demonstrates that the hybrid objective steers the encoder's representational capacity toward the task-relevant object rather than dispersing it over irrelevant regions.

Action Sensitivity of the Predictor. A well-conditioned world model must respond sensitively to action inputs; otherwise the planner cannot distinguish beneficial actions. We probe the predictor's response to a dense 11×11 action grid spanning $[-1, 1]^2$ on a fixed context frame (Figure 6). KAVA produces a markedly broader and more isotropic response cloud: mean $\|\Delta \mathbf{z}\|_2 = 0.219$ vs. 0.121 for LeWM, with peak displacement in the four action-grid corners. We treat this comparison qualitatively, since the two latent spaces are trained with different objectives and need not share a common scale. The trend is consistent with a second benefit of the uncertainty-driven key-frame signal: by selecting frames where the predictor is uncertain, it also trains the predictor to be globally more action-conditioned. Combined with the attention analysis, KAVA's latents are simultaneously more *localised* (encoder) and more *action-responsive* (predictor), enabling more precise planning.

Effective Rank. We sample $N = 2000$ observations per environment from the evaluation set and compute the effective rank r_{eff} of the encoder latent matrix following Sub-JEPA (Zhao et al., 2026):

$$r_{\text{eff}} = \exp\left(-\sum_{i=1}^{d_z} p_i \log p_i \right), \quad p_i = \frac{\sigma_i}{\sum_j \sigma_j},$$

where $\{\sigma_i\}$ are the singular values of the centred latent matrix $Z \in \mathbb{R}^{N \times D}$. In Figure 8, KAVA generally achieves higher effective rank than LeWM during training. A higher r_{eff} means the latent utilises more of its dimensions, preventing collapse onto a low-rank sub-manifold that discards the visual detail needed for key-frame reconstruction. This aligns with Sub-JEPA's observation that effective rank correlates with downstream control performance on low-intrinsic-dimension tasks.

State-Probe Decodability. To verify that the hybrid objective preserves linear probeability, we train a linear regressor on top of a frozen KAVA encoder to predict ground-truth object positions from latents (Figure 4). On both PushT (block position) and Reacher (end-effector coordinates), the predictions align closely with ground truth along the diagonal over 24×12 frame windows. This confirms that the hybrid objective retains task-relevant geometric structure in the latent space—a non-trivial property when adding a high-capacity decoder.

5 Conclusions

JEPA-style latent world models are efficient but forfeit two signals: fine-grained visual structure, and rollout verifiability. KAVA restores both with a small decoder applied only to saliency-selected key frames, with the reconstruction loss decoupled from the predictor. On four image-based control benchmarks it attains the highest average planning success rate among latent world-model baselines, while sharpening encoder attention on task-relevant objects, strengthening action conditioning, and providing an observation-space monitor for imagined rollouts. The saliency signals presume a sufficiently trained predictor, so the selector may be unreliable early in training; annealing the key-frame ratio ρ and extending the decoder scaffolding to language-conditioned (Lin et al., 2024) or multi-modal settings (Mazzaglia et al., 2024; Chen et al., 2026; Huang et al., 2026a) are natural next steps, as is evaluation on longer-horizon and occlusion-heavy manipulation suites that would further stress-test the motivating failure mode. Selective observation supervision thus preserves reconstruction-free efficiency without sacrificing visual fidelity or verifiability.

AI USE STATEMENT

In preparing this work, we utilized a Large Language Model (LLM) solely for limited, auxiliary purposes. Specifically, the LLM was employed to assist with refining the English phrasing of certain paragraphs, correcting grammatical errors, and, in the Related Work section, for generating ideas and searching for potentially relevant literature. The model did not contribute to the core research ideation, mathematical formulation, theoretical analysis, algorithm development, experimental design, or interpretation of results. All authors take full responsibility for the entire content of this work, including any text generated with the aid of the LLM.

REFERENCES

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

Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, et al. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026.

Panagiota Alexoudi, Ioannis Mademlis, and Ioannis Pitas. Escaping local minima in deep reinforcement learning for video summarization. In *Proceedings of the 2023 ACM International Conference on Multimedia Retrieval, (ICMR)*, pp. 530–534. ACM, 2023.

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

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

Marco Bagatella, Matteo Pirotta, Ahmed Touati, Alessandro Lazaric, and Andrea Tirinzoni. Td-jepa: Latent-predictive representations for zero-shot reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 36211–36244, 2026.

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Yabo Chen, Yuanzhi Liang, Jiepeng Wang, Tingxi Chen, Junfei Cheng, Zixiao Gu, Yuyang Huang, Zicheng Jiang, Wei Li, Tian Li, Weichen Li, Zuoxin Li, Guangce Liu, Jialun Liu, Junqi Liu, Haoyuan Wang, Qizhen Weng, Xuan'er Wu, Xunzhi Xiang, Xiaoyan Yang, Xin Zhang, Shiwen Zhang, Junyu Zhou, Chengcheng Zhou, Haibin Huang, Chi Zhang, and Xuelong Li. Teleworld: Towards dynamic multimodal synthesis with a 4d world model. *arXiv preprint arXiv:2601.00051*, 2026.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Anna Dawid and Yann LeCun. Introduction to latent variable energy-based models: A path towards autonomous machine intelligence. *arXiv preprint arXiv:2306.02572*, 2023.

Frédéric Dirfaux. Key frame selection to represent a video. In *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*, volume 2, pp. 275–278. IEEE, 2000.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Xiang Fu, Ge Yang, Pulkit Agrawal, and Tommi Jaakkola. Learning task informed abstractions. In *International Conference on Machine Learning*, pp. 3480–3491. PMLR, 2021.

Genliang Guan, Zhiyong Wang, Shiyang Lu, Jeremiah Da Deng, and David Dagan Feng. Keypoint-based keyframe selection. *IEEE Transactions on circuits and systems for video technology*, 23(4): 729–734, 2012.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

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

Danijar Hafner, Timothy P. Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Danijar Hafner, Timothy P. Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy P. Lillicrap. Mastering diverse domains through world models. *CoRR*, abs/2301.04104, 2023.

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

Ziheng He, Yixiang Chen, Ning Yang, Zhanqian Wu, Qisen Ma, Yuan Xu, Jiabing Yang, Peiyan Li, Xiangnan Wu, Xiaofeng Wang, et al. Skip: Sparse keyframe interpolation paradigm for efficient embodied world models. *arXiv preprint arXiv:2606.00664*, 2026.

Changxin Huang, Lv Tang, Zhaohuan Zhan, Lisha Yu, Runhao Zeng, Zun Liu, Zhengjie Wang, and Jianqiang Li. Unemo: Collaborative visual-language reasoning and navigation via a multimodal world model. In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI*, pp. 18315–18323. AAAI Press, 2026a.

Siqiao Huang, Partha Kaushik, Michael Chen, Hengkai Pan, Kaiwen Geng, Omar Chehab, Fernando Moreno-Pino, and Max Simchowitz. Nano world models: A minimalist implementation of future video prediction. *arXiv preprint arXiv:2605.23993*, 2026b.

Saurav Jha, Artem Zholus, Sarath Chandar, et al. Reconstruction or semantics? what makes a latent space useful for robotic world models. *arXiv preprint arXiv:2605.06388*, 2026.

Sibo Li, Qianyue Hao, Yu Shang, and Yong Li. Keyworld: Key frame reasoning enables effective and efficient world models. *arXiv preprint arXiv:2509.21027*, 2025.

Jessy Lin, Yuqing Du, Olivia Watkins, Danijar Hafner, Pieter Abbeel, Dan Klein, and Anca D. Dragan. Learning to model the world with language. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 29992–30017. PMLR / OpenReview.net, 2024.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworld-model: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

Pietro Mazzaglia, Tim Verbelen, Bart Dhoedt, Aaron C. Courville, and Sai Rajeswar Mudumba. Genrl: Multimodal-foundation world models for generalization in embodied agents. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS*, 2024.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. *arXiv preprint arXiv:2209.00588*, 2022.

Masashi Okada and Tadahiro Taniguchi. Dreaming: Model-based reinforcement learning by latent imagination without reconstruction. In *2021 ieee international conference on robotics and automation (icra)*, pp. 4209–4215. IEEE, 2021.

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.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations*, volume 2025, pp. 94937–94982, 2025.

Zhao-Han Peng, Shaohui Li, Zhi Li, Shulan Ruan, Yu Liu, and You He. From observations to events: Event-aware world model for reinforcement learning. *arXiv preprint arXiv:2601.19336*, 2026.

Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control. In *Conference on Robot Learning*, pp. 1332–1344. PMLR, 2023.

Vlad Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim GJ Rudner, and Yann LeCun. Stress-testing offline reward-free reinforcement learning: A case for planning with latent dynamics models. In *7th Robot Learning Workshop: Towards Robots with Human-Level Abilities*, 2025.

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

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Annual Conference on Neural Information Processing Systems 2022, NeurIPS*, 2022.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Shenghua Wan, Yucen Wang, Minghao Shao, Ruying Chen, and De-Chuan Zhan. Semail: eliminating distractors in visual imitation via separated models. In *International Conference on Machine Learning*, pp. 35426–35443. PMLR, 2023.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Wayne Wolf. Key frame selection by motion analysis. In *1996 IEEE international conference on acoustics, speech, and signal processing conference proceedings*, volume 2, pp. 1228–1231. IEEE, 1996.

Eric Xing, Mingkai Deng, Jinyu Hou, and Zhiting Hu. Critiques of world models. *arXiv preprint arXiv:2507.05169*, 2025.

Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Leslie Pack Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *The Twelfth International Conference on Learning Representations, (ICLR)*. OpenReview.net, 2024.

Kai Zhao, Dongliang Nie, Yuchen Lin, Zhehan Luo, Yixiao Gu, Deng-Ping Fan, and Dan Zeng. Sub-jepa: Subspace gaussian regularization for stable end-to-end world models. *arXiv preprint arXiv:2605.09241*, 2026.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

Table 2: Image-based control benchmarks used in our evaluation. All observations are $3 \times 224 \times 224$ RGB frames at a frame skip of 5.

Task	$\mathcal{A}$	Reward type	Success criterion
Two-Room	2	dense	reach goal within ϵ
PushT	2	dense	block within target pose
OGB-Cube	5	*sparse*	cube within target pose
Reacher	2	dense	end-effector within ϵ

A Algorithm

We show the detailed pseudocode of KAVA in Algorithm 1.

Algorithm 1 The KAVA training step on a single mini-batch.

Input: trajectory $\mathbf{o}_{1:T}$, actions $\mathbf{a}_{1:T}$; current parameters (θ, ϕ, ψ) of encoder, predictor and decoder; KAVA hyper-parameters $(\rho, \alpha, N_{\text{unc}}, \sigma_a, \lambda_{\text{recon}}, \lambda_{\text{SIGReg}})$. **Output**: updated parameters (θ, ϕ, ψ).

1: Encode the trajectory: $\mathbf{z}_{1:T} \leftarrow f_\theta(\mathbf{o}_{1:T})$.
2: Roll out the predictor one step: $\hat{\mathbf{z}}_{t+1} \leftarrow g_\phi(\mathbf{z}_{1:t}, \mathbf{a}_{1:t})$.
3: **Gradient saliency.** Compute $\mathcal{L}_{\text{JEPA}} = \frac{1}{T-1}\sum_t \|\hat{\mathbf{z}}_{t+1} - \mathbf{z}_{t+1}\|_2^2$ and back-propagate to the latents; set $s_t^{\text{grad}} \leftarrow \|\nabla_{\mathbf{z}_t} \mathcal{L}_{\text{JEPA}}\|_2$.
4: **Uncertainty saliency.** For $n = 1, \ldots, N_{\text{unc}}$ draw $\boldsymbol{\epsilon}^{(n)} \sim \mathcal{N}(\mathbf{0}, \sigma_a^2 \mathbf{I})$; form $\hat{\mathbf{z}}_{t+1}^{(n)} \leftarrow g_\phi(\mathbf{z}_{1:t}, \mathbf{a}_{1:t} + \boldsymbol{\epsilon}^{(n)})$ and set $s_t^{\text{unc}} \leftarrow \frac{1}{d_z}\sum_j \text{Var}_n[\hat{z}_{t+1,j}^{(n)}]$.
5: **Select keyframes.** Combine the two saliencies as $s_t \leftarrow \alpha\, \bar{s}_t^{\text{grad}} + (1-\alpha)\, \bar{s}_t^{\text{unc}}$ (min-max normalised) and pick $\mathcal{K} \leftarrow \text{argtop-}k\ s$ with $k = \max(1, \lfloor \rho T \rfloor)$.
6: **Key-frame reconstruction** (*decoupled from the predictor*; gradient $\rightarrow f_\theta, d_\psi$, *not* g_ϕ): $\mathcal{L}_{\text{recon}}^{\mathcal{K}} \leftarrow \frac{1}{|\mathcal{K}|}\sum_{t \in \mathcal{K}} \|d_\psi(\mathbf{z}_t) - \mathbf{o}_t\|_2^2$.
7: **Anti-collapse regulariser**: $\mathcal{L}_{\text{SIGReg}} \leftarrow \text{SIGReg}(\mathbf{z}_{1:T})$.
8: **Total loss**: $\mathcal{L} \leftarrow \mathcal{L}_{\text{JEPA}} + \lambda_{\text{SIGReg}}\mathcal{L}_{\text{SIGReg}} + \lambda_{\text{recon}}\mathcal{L}_{\text{recon}}^{\mathcal{K}}$.
9: Update $(\theta, \phi, \psi) \leftarrow (\theta, \phi, \psi) - \eta\, \nabla\mathcal{L}$.

B Environments and Baselines

Task descriptions. We evaluate on four image-based continuous-control benchmarks that span a range of dynamics, reward sparsity, and visual complexity. All observations are $3 \times 224 \times 224$ RGB frames at a frame skip of 5 (Table 2, Figure 9).

- **Two-Room** (Zhou et al., 2024). A 2-DoF point-mass agent must navigate between two rooms connected by a narrow doorway. The dense reward is the negative distance to the goal. The trajectory contains one abrupt visual change at the doorway and otherwise smooth motion, which exercises the saliency selector.
- **PushT** (Chi et al., 2025). A 2-DoF agent must push a T-shaped block to a target pose using dense contact-based rewards. The task exercises manipulation of a small object against a cluttered background.
- **OGB-Cube** (Park et al., 2025). A 5-DoF dexterous cube-manipulation task with a *sparse* binary success reward, drawn from the OGBench manipulation suite. The cube is small relative to the frame and the success signal does not identify which pixel drove the success—a deliberate stress test for the small-object information loss that motivates KAVA.
- **Reacher** (Tassa et al., 2018). A 2-DoF reacher with a dense negative-distance reward, included as a stress test for over-reconstruction in the dense-reward regime.

Baseline configurations. All baselines are evaluated under the same CEM planner and success criterion as KAVA, so any performance gap is attributable to the world model rather than to the



Figure 9: Visualisation of the four evaluation environments: Two-Room (top-left), PushT (top-right), OGB-Cube (bottom-left), and Reacher (bottom-right). Each environment presents distinct challenges in dynamics, reward sparsity, and visual complexity.

planner. For the two JEPA-family baselines we re-train from scratch under the matched-compute configuration of Section 4.1; for the two pretrained visual planners we use the released checkpoints and the default CEM planner, with no fine-tuning, so the comparison reflects a zero-shot regime.

- **LeWM** and **Sub-JEPA** are re-trained from scratch with the same encoder, predictor, optimiser, schedule, and dataset splits as KAVA, varying only the loss (LeWM: $\mathcal{L}_{\text{JEPA}} + \lambda_{\text{SIGReg}} \mathcal{L}_{\text{SIGReg}}$; Sub-JEPA: the official subspace objective with $K = 32$). Reported numbers are mean $\pm$ std over 5 random seeds (default seed 1 plus four additional seeds). The matched-compute setup makes the LeWM comparison a clean ablation of the decoder.
- **DINO-WM** is used as released, with frozen DINO features and the default CEM planner. No fine-tuning is performed; the baseline is evaluated zero-shot.
- **PLDM** is used as released, with the default CEM planner. PLDM is reward-free and is evaluated on its ability to track a goal state in latent space; we report success rate under the same criterion as KAVA.

C Implementation Details

All models share the same backbone and training infrastructure; KAVA only modifies the loss and adds a small decoder.

Backbone. ViT-Tiny encoder f_θ (patch size 16, depth 12, 3 heads) and an autoregressive Transformer predictor g_ϕ with a 3-frame context history. Latent dimensionality $d_z = 192$, action dimensionality d_a per environment (Table 2).

Decoder. The KAVA decoder d_ψ is a 4-layer transposed-convolutional network that mirrors the ViT-Tiny encoder: a linear projection first reshapes the 192-dimensional latent into a spatial feature map, which the transposed convolutions then upsample to the observation resolution. It contains $\approx$ 6.8M parameters ($\approx 32\%$ of the $\approx$ 21.2M total). The decoder is invoked on the selected key frames during training and, at evaluation time, only as the cSSIM probe; it never participates in the latent rollout itself. In Figure 5, decoded metrics for the decoder-free baselines are produced by auxiliary decoders trained post hoc on the frozen encoders; since the latent spaces of independently trained models are not scale-aligned, latent MSE is compared in terms of growth with the horizon rather than absolute magnitude.

cSSIM diagnostic. SSIM is computed with the standard 11×11 Gaussian window (Wang et al., 2004). The latent perturbation scale is $\varepsilon = 10^{-3}$, and the threshold τ is calibrated per environment as the 5th percentile of the converged cSSIM_t values on held-out validation trajectories. During planning, the probe evaluates two decoder forward passes and one SSIM computation per candidate step.

Optimisation. AdamW with learning rate 5×10^{-5}, weight decay 10^{-3}, batch size 64, gradient clip 1.0, and bf16 mixed precision on a single NVIDIA 4090. We use a cosine warmup schedule

and run for up to 20 epochs ($\approx$ 60k updates) on PushT, OGB-Cube, and Reacher, and 34 epochs on Two-Room. Every 5 epochs we re-evaluate with the CEM planner.

KAVA-specific hyper-parameters. Defaults are $\rho = 0.15$, $\alpha = 0.6$, $N_{\text{unc}} = 4$, $\sigma_a = 0.1$, $\lambda_{\text{recon}} = 1.0$, and $\lambda_{\text{SIGReg}} = 0.09$. The first four are varied in the hyperparameter sensitivity study (Section 4.3); the last two are kept at the LeWM and Sub-JEPA recommended values, respectively.

Randomness. We report all main results with mean $\pm$ std over 5 random seeds.