AVL-JEPA: PREVENTING CAUSAL DYNAMICS INFORMATION COLLAPSE IN JOINT EMBEDDING PREDICTIVE ARCHITECTURE WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Joint embedding predictive architectures (JEPAs) predict future latent representations without reconstructing observations, enabling world models to focus on high-level semantic dynamics. However, a JEPA can preserve high dimensional visual information while discarding information about the physical consequences of actions. We call this failure mode *causal dynamics information collapse* and propose action-grounded vision-invariance latent (AVL) to prevent this collapse. We first use the executed action as an auxiliary dynamics anchor that encourages the model to preserve dynamics information, and then use a vision-invariance pathway which aligns perturbed and clean latent predictions without discarding dynamics information, forcing the model to fully understand and utilize causal dynamics information. We validate AVL on four robotic control tasks (TwoRoom, PushT, OGBench Cube, and Reacher), showing that it substantially improves success rates under visual perturbations while preserving clean-environment performance. We further evaluate physical consequence alignment, clean–noisy dynamics consistency, and the causal effect of targeted transition subspace erasure. Collectively, these results indicate that dynamic information causally relevant to planning is preserved from collapse under AVL.

1 INTRODUCTION

World models learn to predict how an environment evolves, providing a compact substrate for planning and control (Ha & Schmidhuber, 2018). Joint-embedding predictive architecture (JEPA) offers one way to achieve this goal. Instead of reconstructing future observations, a JEPA model predicts future latent representations directly (Assran et al., 2023). This design sidesteps pixel-level reconstruction and focuses on learning abstract predictive structure, making it attractive for world modeling.

A central challenge in JEPA training is preventing representation collapse, where all inputs map to a narrow region of latent space. Existing work addresses this at the representation level: variance, covariance, and distributional regularizers penalize degenerate latent statistics (Bardes et al., 2022); EMA-based teacher networks provide a slowly moving prediction target that resists instantaneous collapse (Assran et al., 2023; Bardes et al., 2024); JEPA world models such as LEWM combine joint-embedding prediction with SIGReg regularization to prevent geometry-degenerate latent embeddings (Maes et al., 2026; Balestriero & LeCun, 2025). These mechanisms are effective at preventing global collapse and make the learned representations remain high dimensional, non-constant, and predictive on average.

However, preventing global collapse does not guarantee that the latent preserves the information needed to distinguish how different actions causally change the physical state. Two distinct actions that produce different physical state changes may yield nearly identical predicted latent transitions. We refer to this failure as *causal dynamics information collapse*.

We introduce action-grounded vision-invariance latent (AVL) to mitigate this collapse. The action-grounded pathway treats the real action as an auxiliary dynamic anchor, forcing the model to preserve enough dynamic information. The vision-invariance pathway then exposes the model to clean and visually perturbed observations while freezing the action head. The perturbed latent is aligned with

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.

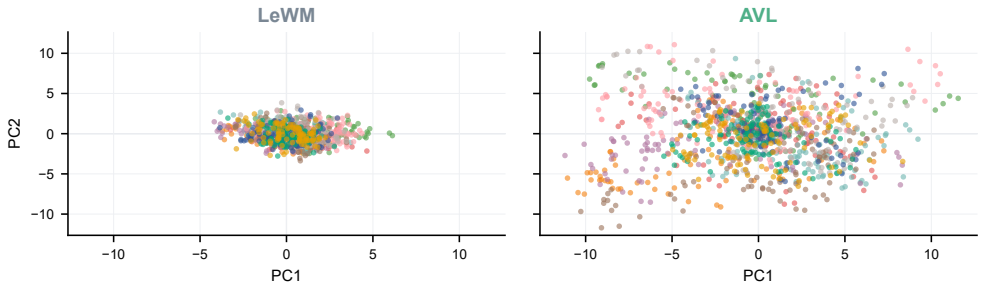

Figure 1: Measured PCA projection of predicted Δz transition responses for LEWM and AVL (ours). Both panels use one shared transform and identical action candidates; different colors denote different candidate action categories.

the clean latent, but the action loss prevents this alignment from being achieved by simply discarding information, making the model fully use causal dynamics information to predict the future latent.

Our method substantially mitigates causal dynamics information collapse and improves closed-loop planning performance. On TwoRoom, we lift predicted-to-physical consequence alignment (Spearman ρ) from 0.35 to 0.53 and reduce normalized regret from 0.152 to 0.061, while clean–noisy ranking consistency increases from 0.89 to 0.98. In closed-loop planning, success rate under Gaussian policy input noise (σ=0.10) improves from 15.3% to 81.5% on PushT and from 65.7% to 88.7% on TwoRoom. Under saturation shift ($+0.35$), it increases from 14.0% to 62.0% on PushT and from 54.8% to 86.7% on TwoRoom. We also improve OGBench Cube (from 73.2% to 78.2% under clean) and remain competitive on Reacher. These results suggest that global non-collapse is only a starting point, a useful world-model latent should preserve causal dynamics-relevant physical structure.

Our main contributions are:

- We formulate *causal dynamics information collapse*, a failure mode in which high dimensional visual latents insufficiently preserve action-conditioned physical change, and operationalize it with controlled experiments tied to physical outcomes.
- We introduce action-grounded vision-invariance latent (AVL) that targets the collapse: the action-grounded pathway grounds state-conditioned latent transitions with executed actions to preserve dynamic information in latent space, and the vision-invariance pathway aligns perturbed and clean latents while freezing the action head, forcing the model to understand and utilize causal dynamics information to predict future latent.
- We validate AVL across TwoRoom, PushT, Reacher, and OGBench Cube, showing that it substantially reduces success-rate degradation under Gaussian, brightness, and saturation perturbations while preserving clean environment performance, and provide mechanism-level experiments showing that the preserved dynamic information is causally meaningful.

2 Related Work

Latent world models. World models learn a compressed simulation of the environment for planning and control (Ha & Schmidhuber, 2018). Early latent dynamics models reconstruct observations from learned states and train policies in latent imagination (Hafner et al., 2019; 2020; 2025; Wu et al., 2023). Reconstruction-free alternatives predict future latent features directly: DINO-WM plans on top of frozen DINOv2 features (Zhou et al., 2025; Oquab et al., 2024), PLDM combines joint-embedding prediction with VICReg-style collapse prevention and inverse dynamics (Sobal et al., 2025), and LEWM introduces SIGReg regularization for stable end to end training from pixels (Maes et al., 2026; Balestriero & LeCun, 2025).

Joint embedding predictive architecture. JEPA predicts target embeddings from context embeddings without pixel reconstruction and has become a prominent framework for self supervised representation learning (LeCun, 2022; Assran et al., 2023). A central challenge in JEPA training is representation collapse, where all inputs map to a narrow latent region. Early preventions rely on architectural asymmetry: I-JEPA uses an EMA target encoder and masked context prediction so that

the context encoder cannot trivially copy the target (Assran et al., 2023). V-JEPA extends this to video with spatiotemporal masking (Bardes et al., 2024), and V-JEPA 2 scales the same recipe to large-scale video understanding and planning (Assran et al., 2025). Others use explicit statistical regularizers: VICReg penalizes latent variance, invariance, and covariance (Bardes et al., 2022), and LeJEPA provides a provable non-collapse guarantee through spectral regularization (Balestriero & LeCun, 2025). LEWM uses SIGReg to maintain non-degenerate latent statistics in world model (Maes et al., 2026), Sub-JEPA adds subspace Gaussian regularization (Zhao et al., 2026), and SD-JEPA decomposes the latent into progression and content subspaces (Thil et al., 2026). Concurrent works such as SMWM and Delta-JEPA expect inverse-dynamics method alone to prevent collapse. (Ivashkov et al., 2026; Zhang et al., 2026). However, these methods mostly address collapse at the level of global latent statistics: variance, rank, covariance, or distributional non-degeneracy. Our work targets this gap: we study how to prevent the selective loss of causal dynamics information in JEPA world models, an information collapse directly limits model performance.

3 METHOD

3.1 CAUSAL DYNAMICS INFORMATION COLLAPSE

The JEPA prediction objective rewards information according to its predictability, which need not coincide with its causal relevance for control. Let X_t denote the information available to the predictor, let u_t be a candidate component of the current representation, and let Y_{t+H} denote the vector-valued future latent target. Under squared prediction loss, define the Bayes risks with and without retaining u_t as

$$\mathcal{R}(X_t) = \mathbb{E}\Big[\|Y_{t+H} - \mathbb{E}[Y_{t+H} \mid X_t]\|_2^2\Big]. \tag{1}$$

$$\mathcal{R}(X_t, u_t) = \mathbb{E}\Big[\|Y_{t+H} - \mathbb{E}[Y_{t+H} \mid X_t, u_t]\|_2^2\Big]. \tag{2}$$

The predictability gain from retaining u_t is therefore

$$\begin{aligned}\Delta\mathcal{R}(u_t) &= \mathcal{R}(X_t) - \mathcal{R}(X_t, u_t) \\ &= \mathbb{E}[\operatorname{tr}\operatorname{Var}(\mathbb{E}[Y_{t+H} \mid X_t, u_t] \mid X_t)] \geq 0.\end{aligned} \tag{3}$$

Thus, temporally persistent visual factors can be favored because they directly reduce future-prediction uncertainty, whereas action-induced physical changes may require nonlinear state–action interactions and provide a weaker or less direct optimization signal. This creates a route by which visual information is retained while action-relevant dynamics are underrepresented.

The JEPA prediction objective also identifies inputs only up to predictive equivalence. For an action set $\mathcal{A}$, define

$$o \sim_{\mathcal{A}} o' \quad \Longleftrightarrow \quad p(z^+_{t+H} \mid o, a) = p(z^+_{t+H} \mid o', a), \qquad \forall a \in \mathcal{A}. \tag{4}$$

JEPA prediction does not require states within one such equivalence class to remain distinguishable, even if they have different physical consequences under actions outside the observed training support. Consequently, generic non-collapse constraints can maintain non-degenerate latent geometry without guaranteeing causal dynamics representations.

3.2 AVL OVERVIEW

Our method intervenes at two points in training. The action-grounded pathway establishes a clean dynamics-aware latent space using an action anchor. The vision-invariance pathway uses visual invariance as an alignment constraint: the latent transition predicted from a perturbed observation is aligned with the corresponding clean transition. Because the two views naturally differ, this alignment can be achieved either by preserving and utilizing the relevant causal dynamics or by discarding information until the two latent become similar. The frozen action head and its active loss rule out the latter shortcut, so the model is driven to infer state changes from causal dynamics.

3.3 ACTION-GROUNDED: TRANSITION-ATTRIBUTED ACTION ANCHORING

Let o_t be the current observation, o_{t+H} the observation after H low level action steps, $a_{t:t+H-1}$ the executed action sequence, $z_t = f_\theta(o_{\leq t})$ the current latent produced by the encoder, z^+_{t+H} the

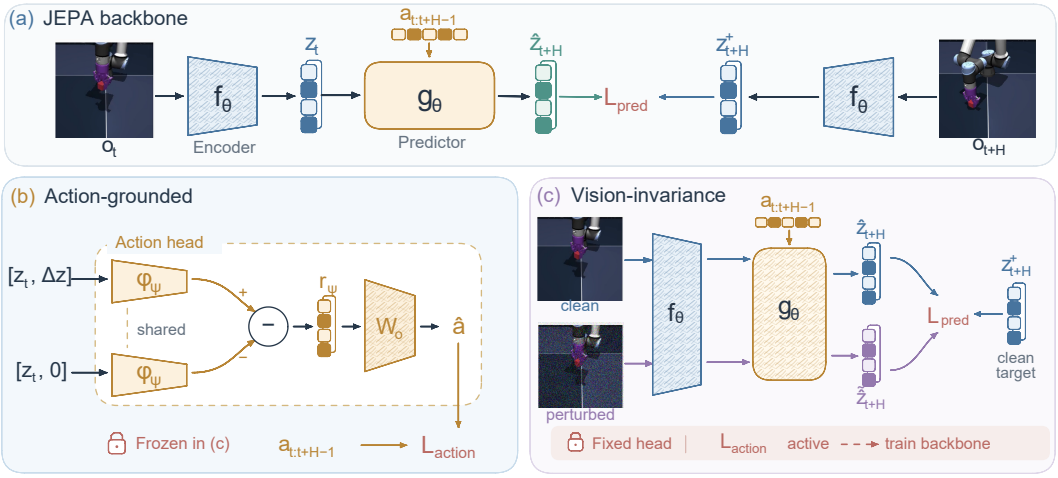

Figure 2: Overview of the proposed AVL method. The action-grounded pathway anchors clean latent transitions to executed actions through a transition-attributed action head that subtracts a state-only zero-transition baseline. The vision-invariance pathway aligns noisy and clean latent predictions while freezing the action head. The alignment creates a potential collapse shortcut, but the frozen action loss requires the latent to retain action-relevant dynamics, thereby targeting causal dynamics information collapse.

target future latent produced by the same shared encoder, and $\Delta z = z^{+}_{t+H} - z_t$. Here H is the prediction horizon and the superscript $+$ denotes the target branch. The action-conditioned predictor g_θ produces $\hat{z}_{t+H} = g_\theta(z_t, a_{t:t+H-1})$.

The latent may contain task-irrelevant information, so a raw latent difference does not necessarily isolate the dynamics relevant to action prediction. The shared extractor provides feature space in which the transition response can be attributed relative to the same state, while the zero-transition branch removes the state-only component. Specifically, the action head uses a three-layer Linear–LayerNorm–GELU trunk, with input dimension $2d_z$, followed by a bias-free output projection. We construct its state-conditioned transition response as

$$r_\psi(z_t, \Delta z) = \phi_\psi([z_t, \Delta z]) - \phi_\psi([z_t, \mathbf{0}]), \tag{5}$$

where ϕ_ψ is the shared trainable trunk, $[z_t, \Delta z]$ denotes concatenation, and $\mathbf{0}$ is a zero transition of the same dimension as Δz. So it removes the state-only component and makes r_ψ the transition-attributed response. We then predict the executed action sequence with

$$\hat{a}_{t:t+H-1} = W_o r_\psi(z_t, \Delta z), \tag{6}$$

where $W_o \in \mathbb{R}^{d_a \times h}$ is the bias-free output projection from the transition-attributed hidden response to the task-specific H-step action target. In the implementation, the trunk is $\mathrm{Linear}(2d_z, h) \to \mathrm{LayerNorm}(h) \to \mathrm{GELU}$ followed by two $\mathrm{Linear}(h, h) \to \mathrm{LayerNorm}(h) \to \mathrm{GELU}$ blocks, with d_z=192 and h=256, while d_a is task-specific. Because the identical zero-transition branch is subtracted, $r_\psi(z_t, \mathbf{0}) = \mathbf{0}$ exactly; this property does not require the hidden trunk itself to be bias-free.

The action-grounded objective is

$$\mathcal{L}_{\mathrm{S1}} = \mathcal{L}_{\mathrm{pred}} + \lambda_1 \mathcal{L}_{\mathrm{SIGReg}} + \lambda_2 \mathcal{L}_{\mathrm{action}}, \tag{7}$$

where $\mathcal{L}_{\mathrm{pred}}$ is the latent prediction MSE, $\mathcal{L}_{\mathrm{SIGReg}}$ is the non-collapse regularizer, and

$$\mathcal{L}_{\mathrm{action}} = \mathrm{SmoothL1}(\hat{a}_{t:t+H-1}, a_{t:t+H-1})\,. \tag{8}$$

H corresponds to low level action steps after action blocking. The output dimension d_a is task-specific and is determined by the environment action dimension multiplied by the action-block length. The auxiliary branch is trainable together with the encoder, projector, action encoder, and predictor during the action-grounded pathway. It is saved separately and excluded from the exported planning checkpoint.

The action-grounded pathway is evaluated by asking whether its predicted candidate-transition geometry preserves physical consequences. For state s, let $\Delta\hat{z}_s^{(k)}$ be the predicted transition response for candidate k, $\Delta x_s^{(k)}$ its task-relevant physical response, $p_s^{(k)} = \|\Delta\hat{z}_s^{(k)}\|_2$ the model score, and $q_s^{(k)} = \|\Delta x_s^{(k)}\|_2$ the physical score. Physical consequence alignment compares all unordered candidate pairs:

$$\rho_{\text{phys}}(s) = \rho_{\text{S}}\Big(\{\|\Delta\hat{z}_s^{(i)} - \Delta\hat{z}_s^{(j)}\|_2\}_{i<j}, \{\|\Delta x_s^{(i)} - \Delta x_s^{(j)}\|_2\}_{i<j}\Big), \tag{9}$$

where ρ_{S} is Spearman's rank correlation. With $k_s^* = \arg\max_k q_s^{(k)}$ and $\hat{k}_s = \arg\max_k p_s^{(k)}$, normalized regret and Top-K oracle inclusion are

$$R_{\text{norm}}(s) = \frac{q_s^{(k_s^*)} - q_s^{(\hat{k}_s)}}{q_s^{(k_s^*)} + \epsilon}, \qquad A_K(s) = \mathbf{1}\Big[k_s^* \in \text{TopK}\{p_s^{(k)}\}_{k=1}^M\Big]. \tag{10}$$

We report state means for ρ_{phys}, R_{norm}, and A_K, with $K \in \{1, 3\}$. To test whether consequence-ranking information is concentrated in the predicted response, let U_k contain the leading k global principal directions of centered candidate-transition responses. Targeted erasure produces

$$\Delta\hat{z}_{s,-k}^{(i)} = \Delta\hat{z}_s^{(i)} - U_k U_k^\top \Delta\hat{z}_s^{(i)}, \quad \Delta\rho(k) = \bar{\rho}(0) - \bar{\rho}(k), \quad \Delta R(k) = \bar{R}(k) - \bar{R}(0), \tag{11}$$

where bars denote means over matched states; positive $\Delta\rho$ and ΔR indicate worse physical ordering after erasure. Finally, for seed j with E_j closed-loop episodes, $S_j = E_j^{-1}\sum_{e=1}^{E_j}\mathbf{1}[\text{success}_{j,e}]$, and we report

$$\bar{S} = \frac{1}{J}\sum_{j=1}^{J} S_j, \qquad \sigma_S = \sqrt{\frac{1}{J}\sum_{j=1}^{J}(S_j - \bar{S})^2}, \tag{12}$$

with $J = 10$ and population standard deviation ($\text{ddof} = 0$).

3.4 Vision-invariance: clean–perturbed latent alignment with frozen dynamics anchor

The vision-invariance pathway applies visual perturbation to a single branch during training: each current input is randomly drawn, with roughly equal probability ($1/3$ each), from {clean, Gaussian noise with $\sigma = 0.01$, Gaussian noise with $\sigma = 0.03$}. The perturbation is applied only to the current observation o_t; the future target observation o_{t+H} remains clean. Let $z_t = f_\theta(o_{\le t})$ and $\tilde{z}_t = f_\theta(\tilde{o}_{\le t})$ denote the clean and perturbed context latents for the same underlying state. Their predicted future latents are $\hat{z}_{t+H} = g_\theta(z_t, a_{t:t+H-1})$ and $\hat{\tilde{z}}_{t+H} = g_\theta(\tilde{z}_t, a_{t:t+H-1})$. Both are trained toward the same clean target z_{t+H}^+, which induces the model to better utilize causal dynamics information.

The vision-invariance pathway uses the same loss terms and hyperparameters as the action-grounded pathway:

$$\mathcal{L}_{\text{S2}} = \mathcal{L}_{\text{pred}} + \lambda_1\mathcal{L}_{\text{SIGReg}} + \lambda_2\mathcal{L}_{\text{action}}, \tag{13}$$

The action-grounded action-head parameters are frozen in the vision-invariance pathway, but the action loss remains active and back-propagates through the model.

The vision-invariance pathway is evaluated on matched clean and perturbed predictions. For state s, candidate k, and perturbation repeat r, let $\Delta\hat{z}_{s,\text{c}}^{(k)}$ and $\Delta\hat{z}_{s,r,\eta}^{(k)}$ denote the clean and noisy predicted transitions. The clean action spread and fixed-action nuisance drift are

$$D_{\text{act}}(s) = \frac{2}{M(M-1)}\sum_{i<j}\Big\|\Delta\hat{z}_{s,\text{c}}^{(i)} - \Delta\hat{z}_{s,\text{c}}^{(j)}\Big\|_2, \tag{14}$$

$$D_{\text{nuis}}(s,r) = \frac{1}{M}\sum_{k=1}^{M}\Big\|\Delta\hat{z}_{s,r,\eta}^{(k)} - \Delta\hat{z}_{s,\text{c}}^{(k)}\Big\|_2, \qquad Q_{\text{A/N}}(s,r) = \frac{D_{\text{act}}(s)}{D_{\text{nuis}}(s,r) + \epsilon}. \tag{15}$$

Let $d_{s,\text{c}}^{(ij)} = \|\Delta\hat{z}_{s,\text{c}}^{(i)} - \Delta\hat{z}_{s,\text{c}}^{(j)}\|_2$ and $d_{s,r,\eta}^{(ij)} = \|\Delta\hat{z}_{s,r,\eta}^{(i)} - \Delta\hat{z}_{s,r,\eta}^{(j)}\|_2$ for $i < j$. Clean–noisy pairwise-geometry consistency and Top-1 stability are

$$C_\rho(s,r) = \rho_{\text{S}}\Big(\{d_{s,\text{c}}^{(ij)}\}_{i<j}, \{d_{s,r,\eta}^{(ij)}\}_{i<j}\Big), \qquad C_{\text{Top1}}(s,r) = \mathbf{1}\Big[\arg\max_k p_{s,\text{c}}^{(k)} = \arg\max_k p_{s,r,\eta}^{(k)}\Big], \tag{16}$$

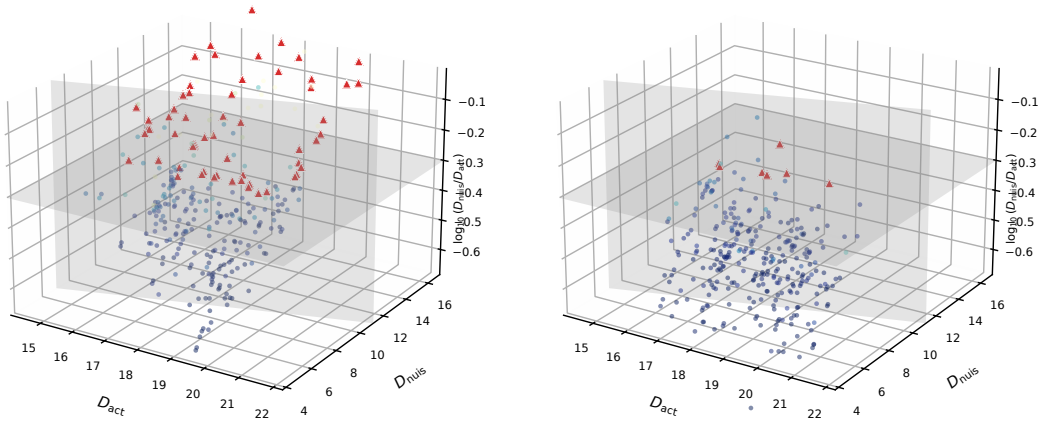

Figure 3: Clean–noisy candidate-transition geometry on TwoRoom for LEWM and AVL (ours). Red triangles indicate states satisfying $D_{\text{nuis}}/D_{\text{act}} > 0.5$ together with a clean–noisy Top-1 candidate flip. The transparent planes show the same ratio threshold in the $(D_{\text{act}}, D_{\text{nuis}})$ and $\log_{10}(D_{\text{nuis}}/D_{\text{act}})$ views.

Table 1: Clean–noisy dynamics consistency on TwoRoom and PushT (100 matched states; three noisy repeats; Gaussian $\sigma = 0.05$). Higher clean–noisy Spearman, Top-1 agreement, and action-to-nuisance ratio are better; lower nuisance drift and noisy regret are better. Best values within each task and metric are bolded.

Task	Model	Clean–noisy Spearman ↑	Top-1 agreement (%) ↑	Action-to nuisance ratio ↑	Nuisance drift ↓	Noisy regret ↓
TwoRoom	LEWM	0.89	58	2.2	8.9	0.162
	Action-grounded	0.90	60	2.0	8.7	0.088
	AVL (ours)	**0.98**	**76**	**2.9**	**6.8**	**0.069**
PushT	LEWM	0.809	45.3	2.35	6.15	0.387
	Action-grounded	0.970	76.0	7.21	3.60	0.313
	AVL (ours)	**0.998**	**93.0**	**19.23**	**1.07**	**0.272**

where $p_{s,\text{c}}^{(k)} = \|\Delta\hat{z}_{s,\text{c}}^{(k)}\|_2$ and $p_{s,r,\eta}^{(k)} = \|\Delta\hat{z}_{s,r,\eta}^{(k)}\|_2$. These quantities separate retained action-conditioned variation from perturbation-induced drift and directly test whether the frozen dynamics anchor preserves candidate geometry under visual corruption.

4 EXPERIMENTS

4.1 TASKS AND METRICS

Tasks. We organize the study into four complementary experiments. **Counterfactual physical consequence alignment** (Figure 4) asks whether action anchoring improves alignment between latent transitions and physical consequences. **Clean–noisy dynamics consistency** (Table 1) tests whether clean–perturbed alignment with the frozen dynamics anchor preserves causal transition structure rather than collapsing it on both TwoRoom and PushT. **Candidate-transition geometry and targeted erasure** (Figure 5) examines the geometry of predicted candidate transitions and tests whether the directions used for consequence ranking are causally important by selectively removing them. Finally, **closed-loop planning** (Table 2) evaluates MPC performance to determine whether our model retains and utilizes causal dynamics information. We evaluate mechanism-level consistency on TwoRoom and PushT, and closed-loop planning on TwoRoom, PushT, Reacher, and OGBench

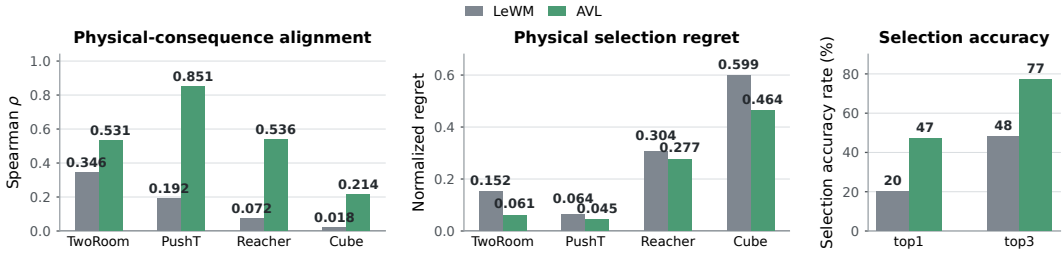

Figure 4: Counterfactual physical consequence alignment, selection quality, and TwoRoom selection accuracy. The first panel reports predicted-to-physical Spearman correlation, the second reports normalized regret, and the third reports Top-1 and Top-3 selection accuracy. Higher Spearman and selection accuracy are better, whereas lower regret is better.



Figure 5: Targeted erasure sweep on TwoRoom and PushT. Each panel reports the absolute difference between the erased and corresponding non-erased baseline values across $k \in \{1, 2, 3, 5, 8\}$ for LEWM and AVL.

Cube, following standard latent-world-model benchmarks and continuous-control suites (Park et al., 2025; Tassa et al., 2018; Tunyasuvunakool et al., 2020).

Experiment evaluation protocol. For closed-loop planning, each model is evaluated independently over 10 random seeds, with 50 episodes per seed. The action-grounded auxiliary action head uses a 256-dimensional hidden response and a task-specific output dimension d_a spanning $H = 5$ low level action steps. The auxiliary branch is saved separately and filtered from the weights used at evaluation, so planning uses the original LEWM architecture and interface with no additional modules. The remaining experiment evaluation protocols and metrics are detailed in Appendix B.1.

4.2 Implementation Details

Training configurations. We train on an NVIDIA RTX 5090 GPU. The action-grounded pathway is trained for 10 epochs with AdamW, learning rate 5×10^{-5}, weight decay 10^{-3}, bf16 mixed precision, and eight data loader workers. The vision-invariance pathway is initialized from action-grounded weights and trained for 2 epochs with AdamW, learning rate 10^{-5}, the same weight decay and precision, and batch size 128. Both pathways use $\lambda_1 = 0.09$ for SIGReg, matching the original LeWM configuration, and $\lambda_2 = 5$ for the action loss. On TwoRoom, the action-grounded sweep gives success rates of 85.2%, 87.2%, 92.8%, and 92.6% for $\lambda_2 \in \{0.05, 0.5, 5, 10\}$, respectively. This sweep supports the choice $\lambda_2 = 5$: increasing the action weight from 0.05 to 5 improves success by 7.6 percentage points, while $\lambda_2 = 10$ does not improve further. In the vision-invariance pathway the action head is frozen while its loss remains active. The measured training time of both pathways is 41.7 minutes per epoch on PushT, 15.3 minutes per epoch on TwoRoom, 53.4 minutes per epoch on Reacher, and 61.9 minutes per epoch on Cube. Detailed training conditions are provided in Appendix B.6.

Planning configuration. Closed-loop planning uses the LeWorldModel sampling-based planner with the default cross-task interface. The planner uses the cross-entropy method (CEM) with 300 candidate action sequences per iteration, 30 optimization iterations, and the top 30 candidates as elites. The prediction horizon, receding horizon, and action-block size are all 5; thus the planner predicts five action blocks and executes five low level actions before replanning. Goals are offset by 25 environment steps. Each seed is evaluated over 50 episodes, with a closed-loop execution budget

Table 2: Clean closed-loop results.

Model	TwoRoom	PushT	Reacher	Cube
PLDM	91.7 ± 1.3	78.1 ± 1.7	78.3 ± 2.4	65.2 ± 1.5
SD-JEPA	86.3 ± 6.2	84.6 ± 4.7	85.0 ± 5.2	69.6 ± 5.3
LEWM	88.4 ± 2.3	$\underline{96.0 \pm 3.5}$	$\mathbf{86.4 \pm 4.4}$	$\underline{73.2 \pm 6.8}$
Action-grounded	$\underline{92.8 \pm 2.4}$	95.5 ± 1.9	82.4 ± 3.9	69.1 ± 5.6
Vision-invariance	85.2 ± 3.2	91.6 ± 3.7	84.0 ± 2.8	72.4 ± 7.8
AVL (ours)	$\mathbf{93.2 \pm 1.6}$	$\mathbf{96.0 \pm 1.3}$	$\underline{85.2 \pm 4.4}$	$\mathbf{78.2 \pm 5.3}$

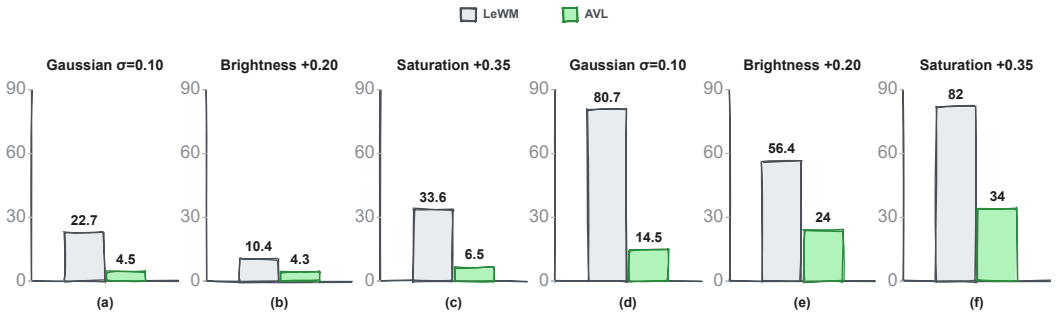

Figure 6: Success-rate degradation under Gaussian policy input noise. Panels (a)–(c) report TwoRoom and panels (d)–(f) report PushT.

of 50 environment steps per episode. The planning configuration is used for all model comparisons unless stated otherwise. The closed-loop evaluation time is 1.8 minutes per seed on PushT, 1.3 minutes per seed on TwoRoom, 1.9 minutes per seed on Reacher, and 2.3 minutes per seed on Cube.

4.3 CLEAN–NOISY DYNAMICS CONSISTENCY

We analyze the collapse using the clean–perturbed quantities defined in Equations 14–16. All quantities are first averaged across the three perturbation repeats for each state and then across states. Figure 3 visualizes the same-state nuisance/action geometry; red markers identify $D_{\text{nuis}}/D_{\text{act}} > 0.5$ together with a Top-1 flip.

On TwoRoom, the vision-invariance pathway supplies the clean–perturbed alignment constraint while the frozen action loss prevents the model from discarding action-relevant variation. Relative to Action-grounded, clean–noisy Spearman increases by 0.0808 (95% CI $[0.0646, 0.0987]$), same-top-1 frequency increases by 16.3 percentage points (95% CI $[9.7, 23.0]$), nuisance drift decreases by 3.318 (95% CI $[-3.579, -3.074]$), and action-to-nuisance ratio increases by 0.899 (95% CI $[0.840, 0.960]$). The independently repeated PushT protocol shows the same stability pattern: AVL has the highest clean–noisy Spearman and Top-1 agreement, the lowest nuisance drift, and the highest action-to-nuisance ratio. The simultaneous reduction in nuisance drift and retention of action separation is consistent with mitigating causal dynamics information collapse rather than uniformly suppressing latent variation.

4.4 COUNTERFACTUAL PHYSICAL CONSEQUENCE ALIGNMENT

Figure 4 summarizes counterfactual physical consequence alignment across all four tasks. We report predicted-to-physical Spearman correlation and normalized regret, which directly measure whether the model orders candidate transitions according to their physical consequences and whether its selected candidate is close to the physical oracle. AVL improves both measures on TwoRoom, PushT, Reacher, and Cube in the aggregate view.

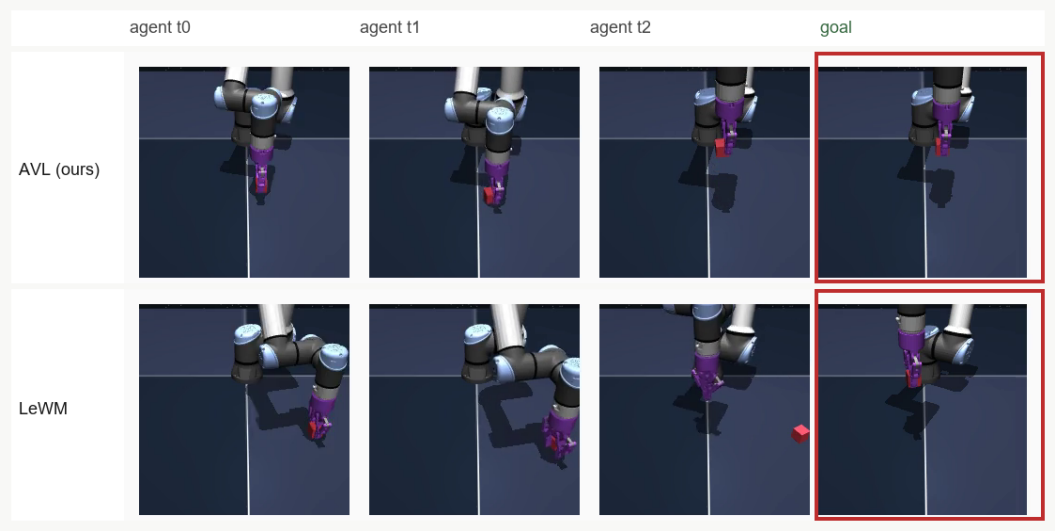

Figure 7: Closed-loop OGBench Cube rollouts. The top row shows a successful AVL (ours) rollout, whereas the bottom row shows a failed LEWM rollout under the same task. Each row contains three temporally ordered agent observations and the corresponding goal observation.

4.5 CANDIDATE-TRANSITION GEOMETRY AND TARGETED ERASURE

We apply the same k-dimensional leading-subspace erasure procedure to both models following Equation 11 and compare the resulting changes in physical consequence ranking; a larger performance degradation for AVL indicates that its leading transition-response directions concentrate more information relevant to causal consequence. Figure 5 reports the complete $k \in \{1, 2, 3, 5, 8\}$ sweep on TwoRoom and PushT. We report the absolute difference from each model's corresponding non-erased baseline.

4.6 MAIN CLOSED-LOOP RESULTS

Table 2 reports clean closed-loop results; the corresponding visual-perturbation results are reported in Appendix C.1. Each table cell reports the mean success rate and population standard deviation defined in Equation 12 across 10 seeds, in percentage points. In Table 2, the best result in each task column is bold and the second-best is underlined.

The full AVL (ours) pathway attains the highest success rate on TwoRoom and PushT across all four conditions and on Cube across all but the brightness shift. Figure 6 reports degradation relative to each model's own clean performance.

Another observation concerns generalization. The vision-invariance pathway is trained only with Gaussian noise on the policy input observation, yet the model also improves performance on brightness and saturation shifts, indicating the transfer is consistent with learning and exploiting causal dynamics information rather than merely fitting a specific augmentation.

5 CONCLUSION

We studied how to prevent causal dynamics information collapse in JEPA world models. Our AVL method uses action anchoring and constrained clean–perturbed alignment to preserve and exploit action conditioned physical structure. Counterfactual consequence ranking, clean–noisy consistency, targeted subspace erasure, and closed-loop transfer across unseen perturbations provide complementary evidence that the method effectively prevents this collapse.

Limitations and future work. AVL builds on the SIGReg regularizer of LEWM rather than replacing it, and we evaluate it only in simulation. We plan to remove the dependence on regularizers and to conduct real robot experiments in future work.

AI USE STATEMENT

Generative AI tools were used to aid or polish the writing of this paper. All AI-assisted text was reviewed by the authors, who take responsibility for the final content.

REPRODUCIBILITY STATEMENT

An anonymized version of the code is provided in the supplementary material. It contains the training entry points for both AVL pathways and for the LeWM baseline, the closed-loop and visual-perturbation evaluation scripts, the Hydra configuration files for all four tasks, and the implementations of the action head, the losses, and the perturbation. Appendix B.6 lists the hardware, optimizer settings, training schedule, data splits, and random seeds; Appendix B.1 documents the mechanism experiment protocols; and Appendix C.1 specifies the perturbation settings and the additional closed-loop results.

REFERENCES

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

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

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

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 2455–2467, 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 (ICML)*, pp. 2555–2565, 2019.

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

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

Petr Ivashkov, Randall Balestriero, and Bernhard Schölkopf. Sensorimotor world models: Perception for action via inverse dynamics, 2026.

Yann LeCun. A path towards autonomous machine intelligence version 0.9.2, 2022-06-27. *OpenReview*, 2022.

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

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

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *International Conference on Learning Representations (ICLR)*, 2025.

Uladzislau Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 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.

Lucas Thil, Jesse Read, Rim Kaddah, and Guillaume Doquet. Subspace-Decomposed JEPAs: Disentangling progression and content in latent world models. *arXiv preprint arXiv:2605.31111*, 2026.

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.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. DayDreamer: World models for physical robot learning. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 2226–2240, 2023.

Zhenghao Zhang, Yuanxiang Wang, Zhenyu Guan, Yujia Yang, Bingkang Shi, Tianyu Zong, Hongzhu Yi, Guoqing Chao, Xingchen Chen, Tiankun Yang, Chenxi Bao, Tao Yu, Jingjing Zhou, and Jungang Xu. Delta-JEPA: Learning action-sensitive world models via latent difference decoding, 2026.

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. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79115–79135, 2025.

A NOTATION

Table 3: Summary of the notation used throughout the paper.

Notation	Description
Observations and latents	
o_t	Observation at time step t
$\tilde{o}_t$	Visually perturbed observation at time step t
o_{t+H}	Observation after H low level action steps
$z_t = f_\theta(o_{\leq t})$	Current latent produced by the encoder
$\tilde{z}_t$	Latent of the perturbed observation
z^+_{t+H}	Target future latent from the shared encoder
$\Delta z = z^+_{t+H} - z_t$	Latent transition
X_t	Information available to the JEPA predictor at time t
u_t	Candidate component of the current representation
Y_{t+H}	Vector-valued future latent target
$\mathcal{A}$	Action set used to define predictive equivalence
$o \sim_{\mathcal{A}} o'$	Predictive equivalence under all actions in $\mathcal{A}$
$\Delta\mathcal{R}(u_t)$	Reduction in squared-prediction Bayes risk from retaining u_t
$\hat{z}_{t+H}$	Predicted future latent
$\hat{\tilde{z}}_{t+H}$	Predicted future latent from the perturbed input
Actions and predictor	
$a_{t:t+H-1}$	Executed action sequence over the horizon
$\hat{a}_{t:t+H-1}$	Predicted action sequence
g_θ	Action-conditioned predictor
ϕ_ψ	Shared transition-attribution trunk
$r_\psi(z_t, \Delta z)$	Transition-attributed response
W_o	Bias-free output projection
Dimensions and hyperparameters	
H	Prediction horizon (action-block length)
d_z	Latent embedding dimension
h	Hidden trunk dimension
d_a	Task-specific action output dimension
Training objectives	
$\mathcal{L}_{\text{pred}}$	Latent prediction MSE
$\mathcal{L}_{\text{SIGReg}}$	SIGReg non-collapse regularizer
$\mathcal{L}_{\text{action}}$	Action prediction loss
$\mathcal{L}_{\text{S1}}$, $\mathcal{L}_{\text{S2}}$	Action-grounded and vision-invariance objectives
λ_1, λ_2	SIGReg and action loss weights
Evaluation metrics	
ρ_{phys}	Predicted-to-physical Spearman correlation
R_{norm}	Normalized regret
A_K	Top-K oracle inclusion
D_{act}	Clean action spread
D_{nuis}	Fixed-action nuisance drift
$Q_{\text{A/N}}$	Action-to-nuisance ratio
C_ρ	Clean–noisy pairwise-geometry consistency
C_{Top1}	Clean–noisy Top-1 stability
U_k	Leading k principal directions of transition responses
$\bar{S}$	Mean success rate across seeds
σ_S	Population standard deviation of the success rate
J	Number of evaluation seeds
E_j	Number of closed-loop episodes for seed j

B Additional Experimental Details

B.1 Mechanism-level protocols

For the counterfactual experiment, we sample 100 matched simulator states and evaluate the same 12 candidate action sequences at every state, with each candidate spanning 5 action blocks. The simulator executes the shared candidate bank to obtain physical consequences. Spearman correlation and normalized regret measure ordering agreement and the physical cost of model selection; top-k inclusion is reported as a secondary selection metric. For clean–noisy consistency, Gaussian noise with $\sigma = 0.05$ is applied only to the current policy observation, while the physical rollout, goal, and success criterion remain clean; three noisy repeats are evaluated per state. Clean–noisy ranking consistency measures whether candidate ordering is preserved; nuisance drift measures perturbation-induced transition change at fixed action, and the action-to-nuisance ratio compares action-induced separation with that drift. Candidate-transition erasure uses the same predicted candidate responses and compares targeted removal of leading principal directions with random orthogonal removal.

B.2 Counterfactual protocol

The archived mechanism experiment uses 100 matched simulator states and 12 shared candidate action sequences per state. Each candidate spans five action blocks, and each sequence covers 25 physical low level steps. Predicted latent responses, target latent responses, and two dimensional physical responses are saved for every state–candidate pair. Simulator restoration was audited over 1,200 renders per model with zero non-exact restorations and zero pixel difference. States are sampled from the held-out TwoRoom trajectory pool with a fixed seed; the shared action bank is reused for both models and consists of 12 sequences of five action blocks. Images use the model's standard ImageNet normalization after any evaluation-time uint8 corruption. The physical oracle is the candidate with the largest physical response norm; regret is the oracle-minus-chosen response difference. All erasure branches rescore the same cached candidate responses.

B.3 PCA projection analysis

Figure 1 is computed from archived TwoRoom counterfactual responses rather than from a conceptual drawing. For each of 100 matched simulator states, the same bank of 12 candidate action sequences is evaluated by LeWM, Action-grounded, and AVL. Each candidate contains five action blocks, corresponding to 25 low level environment steps. The plotted quantity is the predicted transition response $\Delta\hat{z} = \hat{z}_{t+H} - z_t \in \mathbb{R}^{192}$, yielding an array of size $100 \times 12 \times 192$ for each model. The action bank contains zero, cardinal, diagonal, and three smooth random candidate categories. Different colors in Figure 1 denote different candidate action categories; the individual category names are omitted from the main figure to keep the comparison visually compact.

To make the two displayed panels directly comparable, we first concatenate all predicted transition responses from LeWM, Action-grounded, and AVL, standardize every latent dimension using one shared `StandardScaler`, and fit one two-component PCA transform to the resulting pooled matrix. We then project the LeWM and AVL responses through this same fitted scaler and PCA transform and display those two panels with identical coordinate limits. Action-grounded responses participate in fitting the shared transform to preserve the original three-model analysis protocol, but its panel is omitted from Figure 1 for clarity. The first two components explain 5.80% and 4.94% of the standardized variance, respectively. Because the projection retains only two of 192 dimensions, it is used as a qualitative visualization of transition-response organization, not as standalone quantitative evidence of physical consequence alignment or planning performance; those claims are evaluated using the counterfactual ranking, regret, erasure, and closed-loop results.

B.4 Perturbation protocol

The clean–noisy consistency experiment applies Gaussian noise with $\sigma = 0.05$ to the current policy observation only. The goal observation, simulator dynamics, physical consequence, and success criterion remain clean. Each state is evaluated with three noise repeats. The archived smoke audit contains paired clean/noisy policy input images, confirming that the perturbation is applied at the policy input. Noise is added to the policy input image in uint8 range $[0, 255]$ before ImageNet



Figure 8: Input-level visual perturbation examples on an archived real PushT RGB frame. The clean image is shown alongside the Gaussian, brightness, and saturation settings used in the evaluation protocol. These examples illustrate the appearance transformation only; they are not episode-level performance measurements.

normalization, with independent per-pixel IID sampling and a fixed corruption seed of 123; the goal observation is kept clean. A strict corruption audit over all conditions reported zero corruption failures. The additional evaluation settings use pure additive brightness shifts of $+0.10$ and $+0.20$, and luminance-preserving saturation shifts of $+0.15$ and $+0.35$, all applied to the current policy input while leaving the goal and physical rollout clean. Figure 8 visualizes these input-level transformations on one archived real PushT frame; it is illustrative and does not encode closed-loop success.

B.5 Candidate-transition subspace erasure

For each model, an operational PCA basis is fitted to centered predicted candidate-transition responses on 100 TwoRoom states, with 12 candidate action sequences per state and 5 action blocks per candidate. Before candidate scoring, the leading k-dimensional subspace is removed and the candidates are rescored by the norm of the erased predicted transition. Figure 9 reports the complete, directly computed sweep $k \in \{1, 2, 3, 5, 8, 12, 16, 20, 30, 40\}$ for LeWM and AVL, with 10 same-dimensional random-subspace repeats as control. The targeted and random branches are evaluated from the same candidate responses; no interpolation or reconstructed aggregate values are used.

B.6 Main experiment training conditions

The main LeWM training run was performed on AutoDL with an NVIDIA RTX 5090 (32 GB), a 14-vCPU Intel Xeon Gold 6348 CPU at 2.60 GHz, and the project's '.venv' environment. The implementation used PyTorch 2.12.1+cu130 with its bundled CUDA 13.0 runtime, Lightning/stable-pretraining, bf16 mixed precision, Weights & Biases, and a local CSV logger. The data were split into 90% training and 10% held-out samples; image resolution was 224, embedding dimension was 192, history size was 3, and one future prediction was trained.

The optimizer was AdamW with learning rate 5×10^{-5} and weight decay 10^{-3}. Training used 11 epochs, batch size 128, 8 loader workers as specified by the training YAML, prefetch factor 3, pinned memory, persistent workers, bf16 precision, and gradient clipping at 1.0. The run summary recorded on the host additionally lists 12 workers; this is retained as an environment note rather than silently overwriting the versioned YAML setting. The SIGReg loss weight was 0.09, with 17 knots and 1024 projections. The checkpoint selected for evaluation was the final exported 'lewm' checkpoint from the run; exact run identifiers and file hashes are retained in the accompanying experiment archive. These settings describe training-time configuration only; evaluation-time visual perturbations are applied separately at the policy input as specified in the perturbation protocol.

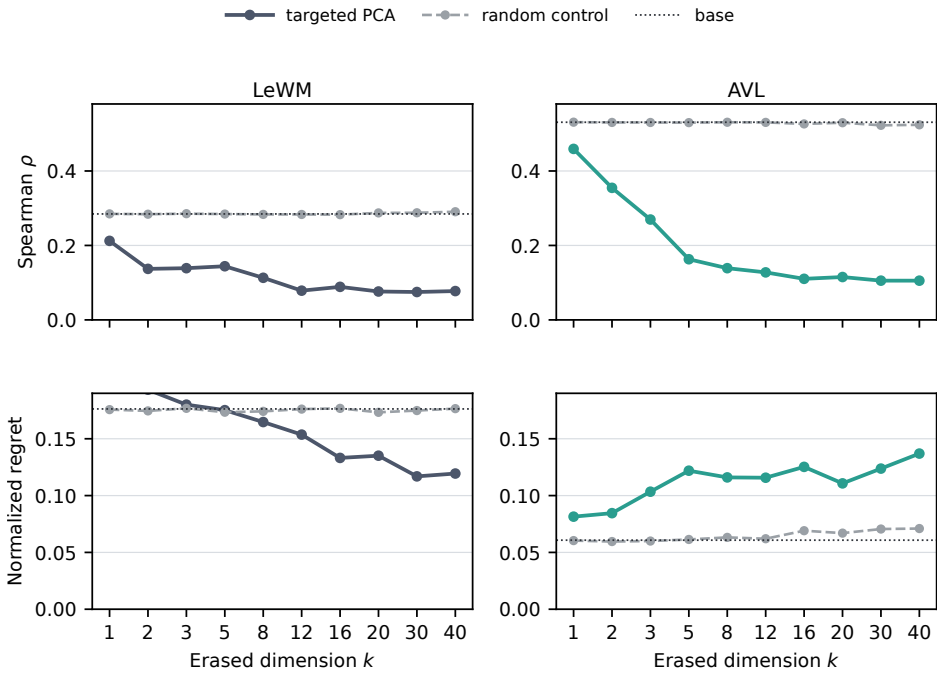

Figure 9: Candidate-transition subspace erasure sweep on TwoRoom for the directly computed dimensions $k \in \{1, 2, 3, 5, 8, 12, 16, 20, 30, 40\}$. Targeted PCA removal is compared with same-dimensional random-subspace removal and the unmodified base.

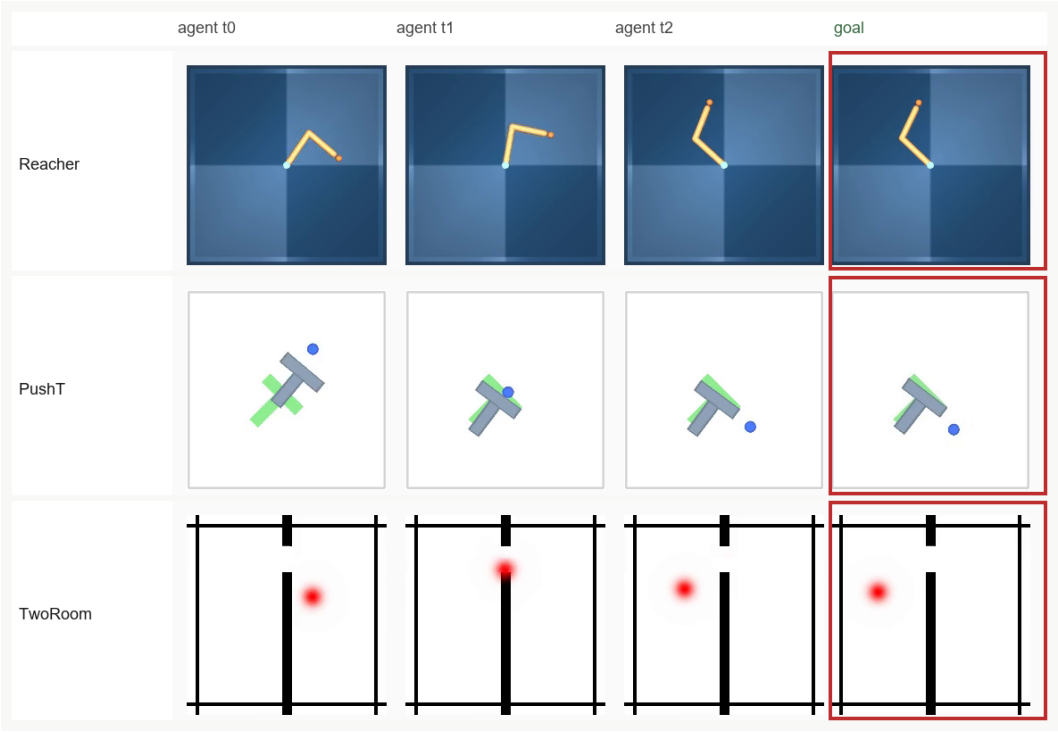

Figure 10: Additional representative closed-loop rollouts for Reacher, PushT, and TwoRoom. Each row shows three temporally ordered agent observations and the corresponding goal observation.

C Additional Results

C.1 Complete experiment results

The remaining experimental results (the candidate-transition subspace erasure sweep and the additional visual-perturbation conditions) are reported below.

C.2 Additional closed-loop rollout visualizations

Figure 10 shows representative closed-loop rollouts from Reacher, PushT, and TwoRoom. These examples are provided as qualitative context; they are not matched head-to-head comparisons and do not constitute aggregate performance evidence.

D Hyperparameters and Compute

The main training configuration is summarized above. For reproducibility, the model uses an image size of 224, a 192 dimensional latent embedding, history size 3, and one prediction target. The run uses a 0.9 training split, AdamW with learning rate 5×10^{-5} and weight decay 10^{-3}, bf16 precision, gradient clipping at 1.0, batch size 128, prefetch factor 3, pinned memory, and persistent workers. The configuration-level worker count is 8; a separate runtime summary reports 12 workers. The planning evaluation uses the task-specific checkpoint exports and the CEM settings given in Section 4.2, with no training performed during evaluation.

Table 4: Gaussian policy input noise, $\sigma = 0.10$.

Model	TwoRoom	PushT	Reacher	Cube
LeWM	65.7 ± 3.9	15.3 ± 5.8	42.6 ± 3.4	52.6 ± 7.2
Action-grounded	76.0 ± 1.6	10.0 ± 2.8	82.6 ± 3.3	64.2 ± 6.5
Vision-invariance	82.6 ± 2.9	31.3 ± 4.1	$\mathbf{84.0 \pm 2.8}$	68.0 ± 9.2
AVL (ours)	$\mathbf{88.7 \pm 1.9}$	$\mathbf{81.5 \pm 4.1}$	78.2 ± 4.4	$\mathbf{72.7 \pm 7.4}$

Table 5: Brightness shift, $+0.20$.

Model	TwoRoom	PushT	Reacher	Cube
LeWM	78.0 ± 4.8	39.6 ± 3.3	11.6 ± 6.9	45.4 ± 3.7
Action-grounded	86.0 ± 1.4	30.5 ± 2.6	7.3 ± 0.9	42.0 ± 1.6
Vision-invariance	80.7 ± 3.7	62.6 ± 2.4	14.1 ± 6.6	48.5 ± 4.9
AVL (ours)	$\mathbf{88.9 \pm 2.4}$	$\mathbf{72.0 \pm 0.7}$	$\mathbf{14.7 \pm 8.1}$	$\mathbf{48.7 \pm 5.7}$

Table 6: Saturation shift, $+0.35$.

Model	TwoRoom	PushT	Reacher	Cube
LeWM	54.8 ± 4.5	14.0 ± 3.9	24.4 ± 2.4	70.8 ± 5.2
Action-grounded	36.0 ± 1.1	31.3 ± 1.8	54.0 ± 4.3	69.3 ± 7.4
Vision-invariance	85.3 ± 3.4	22.4 ± 1.6	49.5 ± 2.4	71.4 ± 3.8
AVL (ours)	$\mathbf{86.7 \pm 3.4}$	$\mathbf{62.0 \pm 4.2}$	$\mathbf{65.3 \pm 0.9}$	$\mathbf{74.0 \pm 1.6}$

Table 7: Gaussian policy input noise, $\sigma = 0.03$.

Model	TwoRoom	PushT	Reacher	Cube
LEWM	83.5 ± 2.6	85.6 ± 1.7	80.2 ± 4.3	72.5 ± 8.2
Action-grounded	90.7 ± 2.9	92.0 ± 1.6	**83.1 ± 5.2**	71.3 ± 8.9
Vision-invariance	81.4 ± 3.4	92.6 ± 3.4	82.8 ± 4.1	70.9 ± 9.8
AVL (ours)	**92.0 ± 1.6**	**96.7 ± 0.9**	80.5 ± 5.8	**74.6 ± 7.3**

Table 8: Gaussian policy input noise, $\sigma = 0.05$.

Model	TwoRoom	PushT	Reacher	Cube
LEWM	83.3 ± 1.8	65.3 ± 4.8	73.9 ± 2.7	71.3 ± 9.2
Action-grounded	90.8 ± 1.6	74.7 ± 5.5	82.0 ± 2.4	68.7 ± 8.9
Vision-invariance	83.1 ± 1.8	91.2 ± 0.9	**87.4 ± 2.0**	69.3 ± 8.2
AVL (ours)	**92.0 ± 1.6**	**95.3 ± 0.9**	82.5 ± 3.9	**74.4 ± 9.4**

Table 9: Brightness shift, -0.10.

Model	TwoRoom	PushT	Reacher	Cube
LEWM	83.6 ± 4.6	87.6 ± 3.8	17.6 ± 3.5	44.0 ± 7.4
Action-grounded	92.0 ± 4.3	93.0 ± 2.9	**31.3 ± 0.9**	44.7 ± 6.8
Vision-invariance	81.3 ± 2.9	92.0 ± 2.8	21.4 ± 4.2	44.1 ± 5.5
AVL (ours)	**92.7 ± 2.4**	**96.6 ± 1.7**	30.0 ± 5.7	**46.0 ± 7.1**

Table 10: Brightness shift, $+0.10$.

Model	TwoRoom	PushT	Reacher	Cube
LEWM	82.8 ± 1.3	91.2 ± 3.8	15.6 ± 4.7	56.8 ± 4.3
Action-grounded	90.6 ± 0.9	92.0 ± 3.4	20.0 ± 4.3	**63.6 ± 6.8**
Vision-invariance	81.3 ± 4.7	91.3 ± 2.4	18.1 ± 2.8	55.2 ± 5.2
AVL (ours)	**92.7 ± 0.9**	**92.0 ± 2.2**	**24.7 ± 1.9**	54.1 ± 1.6

Table 11: Brightness shift, -0.20.

Model	TwoRoom	PushT	Reacher	Cube
LEWM	54.0 ± 4.8	12.0 ± 3.6	14.8 ± 8.2	37.6 ± 5.6
Action-grounded	43.2 ± 4.9	30.5 ± 3.5	20.3 ± 9.0	44.7 ± 4.1
Vision-invariance	82.0 ± 3.2	22.6 ± 0.9	22.5 ± 6.2	42.9 ± 4.3
AVL (ours)	**91.3 ± 3.4**	**40.7 ± 1.3**	**24.7 ± 5.3**	**47.3 ± 4.7**

Table 12: Saturation shift, $+0.15$.

Model	TwoRoom	PushT	Reacher	Cube
LEWM	83.2 ± 1.8	91.6 ± 3.2	56.4 ± 6.5	70.2 ± 6.1
Action-grounded	92.0 ± 1.6	78.0 ± 1.8	**85.3 ± 5.3**	72.0 ± 5.9
Vision-invariance	83.1 ± 1.9	85.3 ± 5.2	81.6 ± 5.7	69.6 ± 5.4
AVL (ours)	**94.0 ± 1.6**	**93.6 ± 3.7**	82.7 ± 3.8	**73.7 ± 5.7**