KWM: Causal Calibration and Recovery Costs in Visual World Models

KineJing Research — internal working draft

15 October 2026

Research draft. This manuscript describes a local experimental study and a small trained calibrator around attributed upstream weights. It is not submitted or peer reviewed. Human authors, contributions and final scientific approval have not been confirmed. Reported prediction, control and transfer outcomes have separate independent audits; their scope limits are retained in the text.

Abstract

Correcting a frozen world model's action interface can improve prediction during a dynamics shift while harming recovery when native dynamics return. We test a common 25-expert gain bank around H-JEPA with 24 paired task seeds, three FourRoom checkpoints, randomized hidden switches, held-out gains and rotation stress. The locked reset-mixture candidate improves first-eight restoration MSE over legacy fixed-share by 7.68% (95% paired interval [4.98,11.17]%), missing the 10% point target, while regressing 24.02% versus frozen prediction ([16.21,36.28]%). The joint progression test fails. A separately frozen 5,273-parameter calibrator trained on past observed error surfaces improves recovery over legacy by 10.48% ([6.76,15.21]%), but still regresses 20.26% versus frozen and fails recovery and normal uncertainty guards. Post-feedback recovery remains worse than frozen for both candidates. An exploratory 16-seed local-control test increases fast-condition success from 0/16 frozen to 10/16 for both legacy and the selected controller; the trained controller succeeds in 9/16, without a clear advantage. Secondary PushT transfer shows recovery benefits over adaptive references accompanied by a shifted-error trade-off, using a different task-specific backbone. Independent bank arithmetic and CPU physical replay preserve all outcomes. Predictive-mean squared-error optimality does not establish calibrated weights, physical identification or control optimality. These findings characterize a constrained calibration interface rather than a new general foundation model or algorithmic priority claim.

1 Introduction

Visual world models can support planning by predicting how observations change under candidate actions. Their usefulness depends on both a learned representation and the action semantics used during training. A command that previously produced one displacement may later produce twice that displacement, half of it, or different changes along two axes. A representation can remain useful while the predictor's action interface becomes inaccurate.

H-JEPA learns a hierarchy of predictive representations for visual planning [1]. We reuse its lowest-level visual predictor and pretrained encoder. We ask a narrower question than training a new world model: can small, causal interface corrections improve prediction when actuation changes, and how much do they cost when the original actuation returns? Restoration is a useful stress

because continuing to apply a previously helpful correction can become harmful immediately after that correction is no longer needed.

An earlier local study supplied a concrete motivation. A classical fixed-share-inspired filter reduced shifted-phase latent error by 46.23% relative to frozen prediction, but its restoration error rose by 10.92%. The prespecified sensitivity-gating extension failed: relative improvement over the same ungated filter was -0.528%, with paired interval $[-1.072,-0.052]\%$. A separate short control probe improved fast-condition success from 2/16 to 15/16 but increased early recovery distance squared from 0.903 to 3.560 pixels squared. These earlier observations motivated the present protocol; they are not new held-out results of the randomized experiment.

We therefore replace fixed switch times with randomized hidden onset and duration, explicitly retain the first restored transition in the primary endpoint, and require several conditions for internal progression. Improving recovery relative to a legacy adaptive filter is insufficient if recovery still harms the frozen model substantially, if normal behavior degrades, or if the improvement is bought by surrendering shifted performance. These criteria make the intended trade-off testable rather than relying on a favorable average over selected phases.

We also separate prediction rules that are often conflated. Selecting the largest-weight expert, averaging expert embeddings, averaging gain coordinates before a nonlinear predictor, and averaging expert goal costs are different operations. Predictive means have a standard squared-error property under their assumed distribution. That property does not establish that a heuristically weighted bank describes the true distribution, nor that a planner should minimize distance of the mean embedding to a goal.

The study comprises a locked classical-candidate prediction test, an explicitly exploratory trained-calibrator follow-up, a separately defined local-control experiment, and secondary transfer to PushT. The trained component is real but small: 5,273 parameters operate on previous observed error surfaces. Pretrained encoder and predictor parameters are unchanged. We provide exact chronology, strong matched-bank controls and independent arithmetic and physical replay, while preserving negative outcomes and limitations. The scientific object is a constrained calibration interface, not a universal world-model capability claim.

2 Related work and attribution

Latent visual world models. H-JEPA provides the architecture, representations and pretrained weights used here [1]. We evaluate the lowest-level predictor rather than retrain the hierarchy or reproduce its full planning benchmark. Different FourRoom checkpoint seeds represent repeated upstream training in one environment family. PushT uses a different task-specific upstream checkpoint, so transfer of a calibration rule is distinct from use of one shared general backbone.

World-model adaptation. AdaJEPA uses execution feedback for adaptation of a latent world model [2]. Sandwich-Residuals keeps pretrained weights fixed while learning small surrounding corrections [3]. JEPA-TTT studies persistent test-time adaptation of latent dynamics [4]. These make clear that online feedback, frozen representations and small adaptation modules are existing directions. Our finite-bank controls are not full reproductions of those systems, and the experiment provides no empirical ranking against them. Estimators here reset between independent episodes, which also differs from persistent adaptation across episodes.

Expert tracking and recurrence. Fixed-share expert tracking was developed to remain competitive when a useful expert changes over time [5]. Mixing past posteriors explicitly addresses repeated switching among a small pool of previously useful experts [6]. Hidden-Markov formulations distinguish expert state learning from segment-local adaptation [7]. We use these as standard controls and conceptual precedents. A restoration-aware reset, shrinkage rule or reuse of expert weights is not, by itself, an original algorithmic principle.

Averaging and uncertainty. Bayesian model averaging combines model-specific predictive distributions using posterior probabilities [8]. Recent work also aggregates Bayesian online rules using sequential predictive losses [9]. Our exponential-MSE weights are expert weights with a generalized-Bayes interpretation; calibration and correctness of a generative likelihood are not assumed. Bayesian online change-point detection instead derives a posterior over run length under explicit assumptions [10]. The feedback-triggered heuristic tested here does not inherit that inference claim.

Informative feedback and representation checks. SIFt-RLS distinguishes information-bearing parameter directions from unexcited directions in recursive identification [11]; it does not establish guarantees for a nonlinear latent-error bank. Counterfactual Action Evaluation, Observation Bottlenecks, and Representation Geometry in Joint-Embedding Predictive World Models motivates auditing physical effects, rendered observations, embeddings and action sensitivity separately [12]. This matters because reduced latent MSE need not mean correct physical identification or improved task success.

3 Problem and causal interface

Let o_t be the current rendered observation and $z_t = E_\theta(o_t) \in \mathbb{R}^{192}$ its fixed embedding. A macro action $A_t \in \mathbb{R}^{5\times 2}$ repeats one issued command for five raw simulator steps. The model receives three observed embeddings and three aligned action blocks. A gain hypothesis $g \in [0.5, 2]^2$ changes only the latest outgoing block:

$$P_t(g) = F_\theta(z_{t-2:t}, C(A_{t-2}), C(A_{t-1}), C(A_t \,\mathrm{diag}(g))) \,. \tag{1}$$

Here C applies the supplied training action mean and standard deviation and flattens each block to ten inputs. We do not refit a normalizer on evaluation observations. Encoder, action encoder, action embedder and predictor remain fixed and run in evaluation mode.

Earlier issued actions remain unscaled even if executed under different physical gains. Equation (1) is the tested latest-only interface, not correction of the entire physical action history. This is an expressivity restriction. Even a known physical gain is not necessarily an upper bound on attainable latent accuracy under this interface.

The bank is the Cartesian grid

$$\mathcal{G} = \{0.5, 0.75, 1, 1.5, 2\}^2, \qquad K = 25. \tag{2}$$

Write $P_{t,k} = P_t(g_k)$ and let the identity expert have $g_{k_0} = (1, 1)$. Every bank method is evaluated on exactly these same forecasts within a teacher-forced episode. The observed target is released only after the current prediction is fixed:

$$\ell_{t,k} = \frac{1}{192}\|P_{t,k} - z_{t+1}\|^2. \tag{3}$$

True physical gain, phase, condition name, simulator position and change schedule belong to the generator and evaluator. They are excluded from estimator calls.

Cached features contain complete arrays, including later observations and evaluator metadata. Thus causality is established by the narrow sequential interface and independent replay of indexing, not by operating-system isolation from the cache. Stateless bank generation may be batched. No future target is supplied to a method before its forecast is recorded.

4 Classical calibration controls

4.1 Frozen and fixed-share controls

Frozen prediction always selects P_{t,k_0}. For the adaptive filter initialize $p_{0,k} = \mathrm{I}[k = k_0]$ and use an identity-biased prior

$$\pi_k = 0.8\,\mathrm{I}[k = k_0] + \frac{0.2}{K}. \tag{4}$$

Here $\mathrm{I}[B]$ is one if statement B is true and zero otherwise. After observing target z_{t+1}, update

$$q_{t,k} = (1-h)p_{t,k} + h\pi_k, \tag{5}$$

$$p_{t+1,k} = \frac{q_{t,k}\exp[-(\ell_{t,k} - \min_j \ell_{t,j})/T]}{\sum_j q_{t,j}\exp[-(\ell_{t,j} - \min_r \ell_{t,r})/T]}. \tag{6}$$

Subtracting the minimum loss improves numerical stability without changing the normalized weights. The registered legacy reference uses $h = 0.05$ and $T = 0.005$. Its MAP forecast is $P_{t,\arg\max_k p_{t,k}}$, with a fixed first-index tie rule.

The matched mixture uses $\bar{P}_t = \sum_k p_{t,k}P_{t,k}$ with the same posterior, hazard, temperature, prior and update order. The sharing step occurs in the feedback update, not as an additional hidden pre-forecast operation. The MAP and mixture controls therefore differ in their output mapping rather than the evidence received or the expert update.

4.2 Moving windows and shrinkage

The window-MAP control stores the last W observed expert loss vectors and selects the expert with lowest mean loss. It retains the current expert if that expert is within 10^{-12} of the minimum; otherwise it selects the first minimum. The standard reference has $W = 5$.

For window mixtures, let $\bar{\ell}_{t,k}$ be the mean of the most recent $\min(W, t+1)$ observed losses. Form a uniform-prior Gibbs vector

$$\widetilde{p}_{t+1,k} \propto \exp[-(\bar{\ell}_{t,k} - \min_j \bar{\ell}_{t,j})/T], \qquad p_{t+1} = (1-s)\widetilde{p}_{t+1} + se_{k_0}, \tag{7}$$

where s shrinks toward the frozen identity expert. The predeclared candidates use (W, s) equal to $(1, 0)$, $(1, 0.25)$, $(3, 0.25)$ and $(5, 0.25)$, all with $T = 0.005$. A matched MAP ablation of a selected window mixture uses its exact weights and applies an argmax only at forecast time. It is distinct from refitting a hard window-MAP estimator.

4.3 Observed-innovation reset heuristic

The reset candidate retains Equation (6) but uses

$$D_t = \mathrm{MSE}(\bar{P}_t, z_{t+1}) - \min_k \ell_{t,k}. \tag{8}$$

After the target has been observed, if $D_t > \kappa T$, set the effective hazard for that update to one; otherwise use $h = 0.05$. This restarts the update from π but still uses the observed expert losses. The finite candidates use $\kappa \in \{1, 3\}$. This is an attributed innovation-triggered engineering heuristic, not a calibrated change-point posterior or identifiability test.

The reset decision always uses the same pre-update mixture surprise, including in the matched MAP ablation. This preserves posterior equivalence under the same externally issued trajectory. It does not imply identical posterior paths between separately acting controllers, because their chosen actions and feedback can diverge. A reset triggered by the target at step t changes only subsequent predictions; it cannot revise the already scored forecast.

5 A separately frozen trained causal mixer

The exploratory v0.4 follow-up replaces the conventional expert update with a small offline-trained mapping from previous observed evidence to bank weights. It changes no upstream world-model parameter.

5.1 Observed evidence features

After target z_{t+1} is revealed, compute the loss surface $\ell_{t,1:25}$, bank spread

$$S_t = \frac{1}{192} \sum_{j=1}^{192} \mathrm{Var}_{k=1:25}(P_{t,k,j}), \tag{9}$$

using population variance over experts, and issued-command energy $U_t = \mathrm{mean}(A_t^2)$. The 28-dimensional evidence vector is

$$\phi_t = \mathrm{clip}_{[0,20]} \left[\left\{ \log\left(1 + \frac{\max(\ell_{t,k} - \min_r \ell_{t,r}, 0)}{\max(S_t, 10^{-6})}\right) \right\}_{k=1}^{25}, \log(1 + S_t/0.005), \log(1 + U_t/0.01), 1 \right]. \tag{10}$$

The last coordinate is a validity mask. Missing previous records are zero vectors. The next forecast uses $x_{t+1} = [\phi_t, \phi_{t-1}] \in \mathbb{R}^{56}$, newest first. No current candidate-action bank, future target, physical coordinate, condition or phase label enters this network.

5.2 Network and training

A $56 \to 64 \to 25$ MLP with GELU activation supplies

$$p_{t+1} = \mathrm{softmax}(f_\psi(x_{t+1}) + \log \pi). \tag{11}$$

Its two affine layers have $56 \cdot 64 + 64 + 64 \cdot 25 + 25 = 5,273$ learned parameters; the identity-prior buffers are not learned parameters. The last layer is initialized to zero. Before any valid evidence

the forecast is exactly the identity expert. The same weight vector applies to every currently considered action bank. At deployment ψ is fixed: feedback updates evidence history, not network parameters.

The four training task seeds are 50000–50003 and the four validation seeds are 50004–50007, using checkpoint 42 and the four development conditions. This yields 16 episodes and 1,280 transition examples per split. The training objective is next-latent mixture MSE, with equal sample count per task/condition episode and no phase-dependent loss weighting. A single predeclared AdamW run [13] uses learning rate 0.001, weight decay 0.001, batch size 128, 60 epochs and random seed 7304. Validation is evaluated after every epoch. The source, recipe, selected weights and evaluation source are frozen before held-out quality is read.

Selection prefers checkpoints satisfying the same normal, recovery and shifted point guards used for the classical candidates. Among eligible checkpoints it minimizes normalized recovery plus shifted error; if none is eligible, it selects the minimum objective and records failure. The actual selected epoch is 32, and no epoch satisfies all validation guards. This is a trained calibrator with an unsuccessful joint validation outcome, not evidence that training itself solves recovery.

The split also deserves a clear interpretation. These eight development seeds were already used for classical-model development. The internal four/four split supports selection of the trained follow-up, not a new confirmatory claim on validation. The independent 24-seed test remains the meaningful fresh prediction evaluation; v0.4 is explicitly exploratory relative to the original v0.3 primary hypothesis.

6 Prediction protocol and statistical decisions

6.1 Physical interventions and randomized timing

FourRoom uses one moving distractor at native speed 2.5. Agent native speed also remains 2.5. The generator applies $u_t^{\text{exec}} = M_t u_t$ to issued commands only inside a hidden interval. Issued two-axis commands are uniform in $[-0.45, 0.45]^2$, refreshed every four macro blocks, with five raw steps per block. Physical transformed components stay inside $[-0.9, 0.9]$, below the simulator action limit. The same task seed supplies matched commands to all branches.

Three unscored prefill transitions create history, followed by 80 scored transitions. Onset a is uniform on integers 15–30 and duration d is uniform on integers 15–25, from a schedule RNG independent of the command RNG. Offset is $b = a + d$. The intervention applies on $[a, b)$ and identity returns on $[b, 80)$. Schedules are matched across conditions and checkpoints. The normal branch retains scoring-window metadata but its physical matrix is always identity.

The rotation is a positive 20-degree matrix applied to a column command. Diagonal gain experts cannot represent it exactly. New gain values test interpolation within an intentionally known family; they are not new environments. The out-of-family row is a descriptive stress, excluded from the primary decision.

6.2 Splits, lock and primary endpoint

Classical development uses seeds 50000–50007 and checkpoint 42. Test uses seeds 51000–51023 and checkpoints 42, 43 and 44. None overlaps the earlier reserved 40000–43007 interval or the v0.2 development/test seeds. The test has 576 task-seed–checkpoint–condition episodes; the 24

Table 1: Prediction conditions. Every changed condition restores identity. The latter four conditions are absent from v0.3 development quality selection.

Group	Condition	Matrix during hidden interval
Development-seen	Normal	I
	Fast	$\mathrm{diag}(2, 2)$
	Slow	$\mathrm{diag}(0.5, 0.5)$
	Anisotropic	$\mathrm{diag}(2, 0.5)$
Held-out gain values	Fast	$\mathrm{diag}(1.7, 1.7)$
	Slow	$\mathrm{diag}(0.6, 0.6)$
	Anisotropic	$\mathrm{diag}(0.65, 1.8)$
Out-of-family stress	Rotation	$R(20°)$

task seeds are the independent units. Encoder and predictor remain the same supplied upstream networks throughout.

Development first evaluates the finite configuration menu. A candidate is eligible if its normal error is at most 1.02 times frozen error, its first-eight restoration error is at most 1.02 times frozen error, and its shifted error is at most 1.05 times legacy-filter error. Among eligible configurations choose the smallest sum of normalized recovery and shifted error, with a deterministic name tie break. If none is eligible, retain the minimum objective and record that all development joint guards failed. The actual selected fallback is the reset mixture with $\kappa = 1$. Its test MAP ablation and all standard baselines remain visible rather than being discarded after selection.

For task seed i, checkpoint j, primary changed condition c and method m, let

$$L_{i,j,c}^{m,\mathrm{rec}} = \frac{1}{8} \sum_{t=b_i}^{b_i+7} \mathrm{MSE}(\widehat{z}_{t+1}^{m}, z_{t+1}). \tag{12}$$

Average the three primary changed conditions and three checkpoints within each seed:

$$L_{i}^{m,\mathrm{rec}} = \frac{1}{9} \sum_{j \in \{42,43,44\}} \sum_{c \in \{\mathrm{fast,slow,anisotropic}\}} L_{i,j,c}^{m,\mathrm{rec}}. \tag{13}$$

The primary relative improvement is a ratio of means, $\Delta_{\mathrm{rec}} = 1 - \overline{L}^{\mathrm{candidate,rec}} / \overline{L}^{\mathrm{legacy,rec}}$. Shifted error first averages each episode over its own $[a_i, b_i)$ interval, then averages conditions/checkpoints within seed; longer shifts therefore do not acquire greater task weight. Normal uses all 80 no-change transitions.

6.3 Uncertainty and the progression conjunction

The prediction analysis uses 10,000 paired percentile bootstrap resamples of the 24 task-seed clusters, random seed 8304. Each resample retains every condition, checkpoint and method within a selected task seed. It recomputes ratios of paired means rather than averaging framewise ratios. Checkpoint repetitions and time steps do not increase the independent n.

The internal progression decision requires every condition in Table 2; it is not a journal acceptance rule. A 10% point estimate with an interval merely above zero does not establish at least 10% population improvement. A guardrail crossing its threshold is insufficient evidence of non-inferiority, not proof of equality. All requirements form one prespecified conjunction for one selected candidate.

Table 2: Locked v0.3 progression requirements. The recovery endpoint includes the first restored transition before its feedback has been observed.

Quantity	Requirement
Recovery improvement versus legacy	Point estimate at least 10% and lower 95% interval above zero
Recovery regression versus frozen	Upper 95% interval no more than 2%
Shift regression versus legacy	Upper 95% interval no more than 5%
Normal regression versus frozen	Upper 95% interval no more than 2%

Other method rankings, conditions, checkpoints and stress results are descriptive. The separately frozen learned mixer does not replace the original selected candidate after its outcomes are known.

6.4 Exploratory event-aligned recovery diagnosis

After seeing development failures but before inspecting held-out quality, we separately registered losses at offset b and at $b+1, \ldots, b+7$, their latter seven-block mean, and the complete eight-lag profile. The first forecast has not yet received a restored-transition target; later forecasts have at least one such feedback opportunity. Every method and condition group is reported, including unfavorable ones.

This diagnosis leaves the original eight-block primary and all margins unchanged. Favorable post-feedback behavior, if observed, can support a qualified statement about those later predictions. It cannot rescue a failed primary, establish an ideal causal baseline, or quantify a sharp irreducible error floor. The extra lag intervals are descriptive.

7 Exploratory local control

7.1 A goal activated only after adaptation warmup

The control protocol uses checkpoint 42, fresh seeds 52000–52015 and four before-to-after schedules: $1 \to 1$, $1 \to 2$, $1 \to 0.5$ and $2 \to 1$. Each episode starts at a seed-selected safe room center. Three zero blocks establish observation history. A seed-selected count of four to eight random command pairs in $[-0.3, 0.3]^2$, each followed by its negative, supplies 8–16 genuine feedback updates under the before gain. The agent must return to its center within 10^{-4} pixels.

Only then is a local goal activated, 18 pixels away in one of eight seed-selected compass directions, and the physical gain switches. The adapter state is retained. Its initial goal distance exceeds the 1.5-pixel scoring threshold; pre-warmup or pre-switch success cannot count. The environment's native coarse success flag is ignored. All 12 scored macro blocks execute without early stopping.

7.2 Expected cost versus the cost of a mean

For each current action u the planner constructs all 25 expert predictions. The 81 issued action candidates form the Cartesian nine-point grid $[-0.45, -0.3375, \ldots, 0.45]^2$. Coordinates, condition and true gain are not planner inputs. The goal image supplies an embedding z_{goal}.

For pre-feedback weights p_t, the expected-expert rule uses

$$C_{\text{expected}}(u) = \sum_k p_{t,k}\,\text{MSE}(P_{t,k}(u), z_{\text{goal}}), \tag{14}$$

while the predictive-mean rule uses

$$C_{\text{mean}}(u) = \text{MSE}\left(\sum_k p_{t,k} P_{t,k}(u), z_{\text{goal}}\right). \tag{15}$$

Both select the first minimum in the same candidate ordering. The six controllers are frozen, legacy MAP, legacy mixture with expected cost, candidate MAP with expected cost, candidate mixture with mean cost, and candidate mixture with expected cost. All compute the full action-by-expert bank, including frozen control. This matches bank-level compute while deliberately overcomputing the frozen reference; it is not a claim of an efficient frozen deployment.

Only the executed command produces an observed target and adaptation update. Its 25 counterfactual hypotheses can share that one feedback target. Unexecuted imagined branches are never treated as observations. Controllers can choose different commands and receive different trajectories; their planners and feedback opportunities match, not their realized data.

The main exploratory endpoint is mean physical goal distance squared over the first six scored blocks. Secondary endpoints are the mean over all 12, final distance and success below 1.5 pixels at two consecutive scored endpoints. The 384 runs equal 16 paired task seeds times four schedules times six controllers; the independent task count is 16. Exploratory paired intervals use 10,000 seed resamples and RNG 8305. The change remains tied to goal activation despite randomized warmup length, so this is a local adaptation probe rather than arbitrary change during long-horizon navigation. After prediction outcomes were available, a further exploratory control run evaluates the already frozen learned calibrator with both mean and expected cost on these same paired tasks. It adds 128 method episodes, not new independent task seeds. The network and hyperparameters are unchanged; these outcomes do not select a new model or alter the original prediction test. Together the two control runs contain 512 method episodes, all sharing the same 16 independent task seeds; 512 is not an independent sample count.

8 Secondary transfer to PushT

The transfer probe tests reuse of the same frozen calibration rules and, for the trained follow-up, the same learned evidence transform and weights. It uses a native PushT H-JEPA checkpoint and normalizer distinct from the FourRoom backbone. Thus it is estimator transfer across environment-specific world models, not a demonstration of shared general world-model weights.

The predeclared collection has 16 seeds 54000–54015, one pretrained model seed, eight conditions and 80 scored blocks after three prefill blocks. Timing follows the same randomized onset and duration construction. A fixed open-loop paired-pulse policy pushes with small horizontal jitter and increasing downward depth, followed by the exact negative command. Physical matrices modify raw relative commands before native PD control. Command components remain bounded without clipping.

Initial pusher and block states use small seeded perturbations. The native collision callback is preserved while an additional counter records only pusher–block contact points. Geometric visibility records pusher and block center, full bounding-box visibility and bounding-box area fraction at

every rendered frame. Visibility is a geometric diagnostic, not pixel segmentation. All 128 episodes remain, including no-contact or partly out-of-view cases; there is no quality-based filtering or PushT model-error tuning.

Shift and first-eight restoration latent error, full-episode error, pre-shift error, contact and visibility are secondary readouts. The probe supplies neither closed-loop PushT task success nor a manipulation benchmark result. New physical interactions can expose mismatch in a calibrator trained on FourRoom evidence even when its input dimension matches. A failure to transfer is an informative scope boundary.

9 Why averaging is not a control guarantee

9.1 A standard conditional-mean identity

Let $\mathcal{H}_t$ contain exactly the available history and current issued command. If the true next-target law has finite second moment, write $\mu_t = \mathbb{E}[Z_{t+1} \mid \mathcal{H}_t]$. For any history-measurable predictor a,

$$\mathbb{E}[\|Z_{t+1} - a\|^2 \mid \mathcal{H}_t] = \mathbb{E}[\|Z_{t+1} - \mu_t\|^2 \mid \mathcal{H}_t] + \|\mu_t - a\|^2. \tag{16}$$

The cross term vanishes by conditional centering. The true conditional mean minimizes squared prediction risk. If a mean instead belongs to a misspecified assumed distribution, the guarantee applies to that assumed distribution. Heuristic exponentiated losses or a trained softmax do not make the resulting weights true posterior model probabilities.

For a bank with weights p_k and mean $\bar{P}$, a separate algebraic identity is

$$\sum_k p_k \,\mathrm{MSE}(P_k, z) = \mathrm{MSE}(\bar{P}, z) + \sum_k p_k \,\mathrm{MSE}(P_k, \bar{P}). \tag{17}$$

Averaging prediction vectors therefore cannot exceed the weighted average of their losses on a fixed target. It need not beat the MAP expert or frozen expert. Lower error can arise from bias cancellation without more accurate physical gain identification. These standard facts do not establish novelty.

9.2 Three different aggregation operations

The cost of a predictive mean, the weighted expert cost and the cost after running the nonlinear predictor at an averaged gain differ:

$$C_{\mathrm{mean}}(u, z) = \mathrm{MSE}\left(\sum_k p_k P_k(u), z\right), \tag{18}$$

$$C_{\mathrm{expected}}(u, z) = \sum_k p_k \,\mathrm{MSE}(P_k(u), z), \tag{19}$$

$$C_{\mathrm{gain}}(u, z) = \mathrm{MSE}\left(F_\theta(\mathcal{H}_t, u \odot \sum_k p_k g_k), z\right). \tag{20}$$

Nonlinearity generally prevents moving the average through F_θ. Equation (17) also shows that expected expert cost includes a command-dependent disagreement penalty omitted by mean-goal

cost. These formulas concern expert point predictions; full predictive distributions would additionally require their within-component uncertainty.

Even a correct conditional mean does not resolve this distinction. Let a scalar physical gain be 0.5 or 2 with equal probability and let the next position be $Y = Gu$ with goal 1. The mean-position objective chooses $u = 0.8$ and predicts zero mean-goal error. Actual expected goal error is

$$\mathbb{E}[(Gu - 1)^2] = 2.125u^2 - 2.5u + 1, \tag{21}$$

whose minimizing command is 10/17. Expected error is 0.36 at 0.8 but $9/34 \approx 0.265$ at 10/17. This is a mathematical illustration, not an experiment with the FourRoom action bound. If the disagreement penalty were constant in the action, the one-step minimizers would coincide; that condition cannot be assumed.

Embedding aliasing, off-manifold averaged predictions, collision costs, multi-step uncertainty and action-dependent information introduce further gaps. A teacher-forced next-frame loss is therefore not a substitute for independent executed-action control evidence.

9.3 An information limit at hidden restoration

Consider two environments with identical observed history just before a command. One continues the shifted gain and the other restores identity without a visible cue. An estimator that receives no boundary label must issue the same forecast in both histories. If their next-target means differ, that one forecast cannot equal both. Feedback after execution can change the next prediction, not the already issued one.

This observation explains why the first restored transition deserves separate diagnosis. It does not prove that every observed first-step error is irreducible, that a specific reset is optimal, or that later predictions reach an ideal causal lower bound. A 2% average non-inferiority criterion remains a demanding empirical target, not a physical impossibility theorem.

10 Results and audit status

10.1 Verified development selection

The independent development preflight checks all 32 Cartesian episodes and 320 method records. It verifies source/feature hashes, 80-step shapes, randomized schedule indexing, matching commands and current/target alignment. Every stored loss, forecast weight and diagnostic gain is independently recomputed from saved banks with maximum observed difference zero. Selection independently reproduces the reset mixture with $\kappa = 1$, and reproduces the declaration that none of the seven candidate configurations passes all development guards.

The trained follow-up selects epoch 32 under its frozen recipe. Its validation normal, shifted and recovery means are approximately 0.007243, 0.009664 and 0.008315. Corresponding frozen normal and recovery means are 0.007186 and 0.007321, and legacy shifted mean is 0.009826. The trained model therefore also misses the joint validation requirement. These values describe development selection on four validation seeds, not held-out evidence of superiority.

10.2 Randomized held-out prediction

The independent audit passes all 576 task-seed–checkpoint–condition episodes and 3,456 classical-method records. Each has 80 scored forecasts. Exact Cartesian coverage, feature/source hashes, schedules and chronological replay pass; maximum loss, forecast-weight and diagnostic-gain discrepancies are zero. Table 3 reports every locked method, together with the separately audited trained follow-up. Its lag columns are descriptive.

Table 3: FourRoom held-out latent MSE means. Shift and recovery average the three primary changed conditions and three checkpoints within each of 24 task seeds. Normal uses all 80 no-change blocks. Recovery is all eight restoration blocks; lag 0 and the mean of lags 1–7 form the exploratory decomposition. The trained v0.4 row is a separate exploratory follow-up, not a replacement for the locked candidate.

Method	Shift	Recovery	Normal	Lag 0	Lags 1–7
Frozen	0.026236	0.008911	0.008589	0.007930	0.009051
Legacy MAP	0.012057	0.011971	0.008604	0.023541	0.010318
Legacy mixture	0.011790	0.011459	0.008578	0.023019	0.009808
Window MAP	0.011696	0.013283	0.009070	0.023269	0.011857
Candidate MAP	0.012830	0.011450	0.008629	0.023075	0.009789
Candidate mixture	0.012133	0.011051	0.008580	0.022371	0.009434
Trained v0.4 mixture	0.012113	0.010717	0.008760	0.019194	0.009506

The selected classical candidate has recovery improvement over legacy MAP of 7.6831%, with paired interval [4.9780,11.1666]%. This is positive but misses the registered 10% point requirement. More decisively, recovery error is 24.0167% above frozen, interval [16.2064,36.2818]%, far beyond the 2% guard. The original joint progression decision therefore fails.

The remaining guards pass. Candidate shifted error is 0.6367% above legacy. Its interval runs from −1.2902% to 2.2367%, below the 5% upper-bound tolerance. Normal regression versus frozen is −0.1062%, with an interval from −0.4140% to 0.1901%, satisfying its 2% guard. Reporting only the positive recovery interval versus legacy would conceal both the missed practical target and the substantial frozen-reference cost.

Standard calibration remains useful during the shift. Legacy MAP reduces shifted error by 54.05% relative to frozen, descriptive interval [48.62,59.28]%, but its recovery error is 34.34% higher than frozen. The window control has the smallest shifted point error in the table but has 49.06% recovery regression and about 5.59% normal regression. Matched mixture outputs reduce squared error without establishing a new mechanism: the candidate's MAP ablation has recovery 0.011450 versus mixture 0.011051, while the legacy mixture already improves on legacy MAP. All these comparisons are descriptive outside the registered candidate test.

10.3 A trained calibrator improves an adaptive reference, not the joint target

The independently audited learned follow-up uses the same 576 test episodes. Separate NumPy/SciPy linear, GELU and softmax operations replay its actual frozen tensors without importing the estimator implementation. Across 46,080 transitions the maximum learned-weight discrepancy is 4.94×10^{-7} and maximum loss discrepancy is 2.33×10^{-8}, within the declared numerical tolerances. The checkpoint contains 5,273 learned parameters with changed output weights; this is an actual trained artifact, not an untrained adapter interface.

Learned recovery MSE is 0.010716989, compared with legacy 0.011971119 and frozen 0.008911192. Its improvement versus legacy is 10.4763%, paired interval [6.7618,15.2121]%; versus the stronger legacy mixture it is 6.4785%, interval [3.9156,9.6421]%. Yet frozen-reference recovery regression is 20.2644%, interval [12.9310,30.5731]%. This is a smaller restoration cost than some adaptive controls, not its elimination.

Learned normal regression is 1.9919% versus frozen, interval [0.9836,3.0661]%. Although its point estimate is just below 2%, its upper uncertainty bound fails the guard. Shift regression versus legacy is 0.4654%, interval [−3.4695,3.6829]%, which passes the 5% tolerance. Applying the original criteria descriptively to this exploratory follow-up therefore still fails the recovery and normal guards. It does not change the original v0.3 outcome or become a new confirmatory primary.

10.4 The cost is not confined to the hidden first step

The lag decomposition shows a large first-step effect but does not justify attributing every restoration error to that information limit. At lag 0 the classical candidate's error is 0.022371489 versus frozen 0.007930016, a 182.11% regression, descriptive interval [107.25,299.24]%. After at least one restored-target feedback opportunity, its lags-1–7 mean is still 4.2297% above frozen, interval [1.4683,8.4613]%. It improves those later steps over legacy by 8.5684%, interval [5.4109,12.8125]%, while retaining a measurable frozen-reference cost.

The learned calibrator reduces lag-0 error relative to the classical candidate by 14.20%, interval [10.11,17.51]%, but does not show better later recovery than that candidate: its lags-1–7 point error is 0.76% higher and the interval spans benefit and harm. Relative to frozen, learned later error remains 5.0232% higher, interval [2.5618,8.2194]%. Thus its observed gain is concentrated in the first restored forecast rather than demonstrably faster reaction after feedback. Neither candidate reaches an ideal causal baseline or establishes post-feedback recovery equivalence.

10.5 Held-out gain values and rotation remain secondary

On the three held-out gain values, candidate shifted/recovery MSE is 0.010979/0.009843, compared with learned 0.012570/0.009776 and frozen 0.018445/0.008541. Candidate shift improvement over frozen is 40.48%, descriptive interval [34.68,46.02]%, while its recovery regression remains 15.25%, interval [9.86,23.09]%. Good shifted prediction therefore does not remove restoration cost even for these new gain values.

Under rotation, frozen shift/recovery MSE is 0.012505/0.008089, candidate 0.011945/0.008175 and learned 0.012357/0.008550. The candidate has a small 4.48% shifted reduction versus frozen, interval [1.39,8.26]%, while its recovery comparison includes benefit and harm. Since no bank expert represents the true rotation, a latent-error reduction does not identify that physical transformation. Complete per-condition, per-checkpoint and per-method values accompany the audits; no secondary row replaces the failed original conjunction.

10.6 Local control: fast benefit without a clear legacy advantage

The independent audit passes all 384 control episodes, including 8,928 observed-feedback transitions and 768 exact initial/goal RGB checks. Recomputed latent errors, weights, weighted forecasts and CPU physical distances match the stored evidence exactly; selected-branch cost error is at most 1.11×10^{-16}. Table 4 retains every controller and condition.

Table 4: Exploratory 18-pixel local control. Each cell gives first-six-block mean squared goal distance in pixels squared, followed by success count out of 16. All 12 blocks execute. MAP rows use expected cost with one-hot weights; "mean" and "expected" distinguish the candidate mixture's planner cost.

Controller	Normal	Fast	Slow	Recovery
Frozen	28.222 (16)	21.336 (0)	78.496 (15)	28.222 (16)
Legacy MAP	28.222 (16)	8.245 (10)	77.613 (15)	28.057 (16)
Legacy mix, expected	28.199 (16)	7.989 (10)	77.632 (15)	27.993 (16)
Candidate MAP	28.222 (16)	7.915 (9)	77.788 (15)	28.077 (16)
Candidate mix, mean	28.222 (16)	8.170 (9)	77.762 (15)	28.077 (16)
Candidate mix, expected	28.199 (16)	7.894 (10)	77.722 (15)	28.034 (16)

In the fast condition, legacy MAP improves early distance squared by 61.36% relative to frozen control, with descriptive paired interval [53.84,66.83]%, and increases success from 0/16 to 10/16. The candidate's expected-cost controller also succeeds in 10/16, reducing early distance squared by 63.00% versus frozen, interval [56.54,67.71]%. Its improvement over legacy MAP is only 4.25%, with interval [−2.30,11.64]%; this does not establish a candidate advantage over that standard controller.

For the candidate mixture, expected cost has lower fast-condition early distance than mean cost by 3.38%, interval [0.26,6.24]%, and success is 10/16 versus 9/16. This is exploratory evidence consistent with the cost distinction, not a corrected superiority claim or a new planning principle. Both forms have 15/16 slow-condition successes and 16/16 normal and recovery successes. All controller comparisons and endpoints remain available in the audit exports, including final distances and complete trajectories.

This longer local task does not reproduce the earlier 7.5-pixel task's recovery penalty. Candidate expected-cost early recovery distance squared is 28.0341 versus frozen 28.2216, a small 0.66% reduction with interval [0.23,1.16]%, and all methods succeed in 16/16. Mean final recovery distance is 0.7821 pixels for this candidate versus 0.7294 for frozen. The task-specific finding does not settle the randomized prediction recovery hypothesis or prove uniform recovery benefit. Goal distance, action saturation and the one-step precision grid change the control problem materially.

10.7 Trained-control follow-up: lower distance does not improve success

The separately audited trained-control follow-up retains both cost rules and all 128 episodes. Independent NumPy/SciPy weight replay has maximum error 2.52×10^{-7}, latent-loss error 2.75×10^{-8} and chosen-cost error 3.40×10^{-8}; CPU physical distances match exactly across 2,976 observed-feedback transitions. Table 5 reports all conditions, paired with the same 16 tasks as Table 4.

Table 5: Exploratory frozen trained-calibrator control. Cells use the same early distance-squared mean and success-count convention as Table 4. These runs were commissioned after prediction quality was known, with unchanged previously frozen weights and no selection.

Controller	Normal	Fast	Slow	Recovery
Learned, mean	28.137 (16)	7.601 (9)	77.560 (15)	28.018 (16)
Learned, expected	28.168 (16)	7.572 (9)	77.641 (15)	28.044 (16)

Relative to legacy MAP, fast early distance squared improves by 7.80% for mean cost, interval

[−1.09,16.48]%, and by 8.15% for expected cost, interval [−0.84,17.04]%. Both intervals include no improvement. Both learned controllers succeed in 9/16 fast episodes versus legacy 10/16 and frozen 0/16; the paired success difference versus legacy is −6.25 percentage points, interval [−18.75,0]. Recovery early-distance changes versus legacy are only 0.14% and 0.04% improvements, with intervals crossing zero. Normal and recovery success remain 16/16 and slow success 15/16 for both rules. Lower distance point estimates therefore do not establish a reliable advantage over the strong classical controller or better task success. The result remains a local, single-checkpoint exploratory follow-up.

10.8 PushT transfer: a narrower recovery trade-off

The secondary transfer audit retains all 128 episodes. It independently replays all 61,440 classical-method forecasts and 10,240 trained-calibrator forecasts with zero observed latent-loss discrepancy. Table 6 reports mean errors for the three primary gain conditions and the no-change branch. All eight condition results, including rotation, accompany the artifact; these comparisons are exploratory.

Table 6: Secondary PushT latent prediction means. Shift and first-eight recovery average fast, slow and anisotropic within each of 16 task seeds. Normal uses all 80 no-change transitions. A different native pretrained PushT backbone is used; no PushT fitting or error-based selection occurs.

Method	Shift MSE	Recovery MSE	Normal MSE
Frozen	0.029957	0.017699	0.021430
Legacy MAP	0.018813	0.019802	0.020003
Legacy mixture	0.018651	0.019000	0.019489
Window MAP	0.018801	0.020023	0.019361
Candidate mixture	0.019065	0.018794	0.019812
Candidate MAP	0.020167	0.019151	0.020602
Trained v0.4 mixture	0.019459	0.017879	0.019385

The trained calibrator improves recovery error by 9.71% versus legacy MAP, paired interval [7.73,11.71]%, and by 5.90% versus the matched legacy mixture, interval [4.55,7.23]%. Relative to frozen recovery it instead has improvement −1.02%, interval [−4.06,1.84]%; the interval includes a meaningful remaining restoration cost. This is not evidence of recovery equivalence to frozen prediction.

The shift trade-off must accompany that recovery comparison. Trained shifted error is 3.43% worse than legacy MAP, interval [2.06,4.93]%, although it is 35.04% lower than frozen, interval [32.69,37.32]%. Normal error is 9.54% lower than frozen, interval [7.32,12.20]%. The locked classical candidate improves recovery over legacy by 5.09%, interval [2.91,7.28]%, but remains 6.19% worse than frozen, interval [3.37,9.18]%. No single comparison supports uniformly best behavior.

All 10,752 rendered frames are finite and both pusher and block are fully visible under the geometric bounding-box diagnostic. Contact is not universal: 9/16 slow and 9/16 anisotropic shifted episodes have no pusher–block contact during that interval. They remain included. The probe therefore measures prediction under a contact-focused command policy with variable actual interaction, not manipulation success, universal physical understanding or one shared backbone across environments.

11 Independent audit and reproduction

The experiment separates source locking, feature generation, estimator evaluation, model training and analysis. The protocol, selection lock and source identities pin the generator, evaluator, estimator, menu and relevant upstream Python/YAML files. Upstream checkpoint and normalizer digests are checked before loading. Test caches retain the selection identity so that a new evaluator cannot silently reuse incompatible features.

The independent prediction audit imports no frozen estimator or evaluation implementation. It checks the exact task-seed–checkpoint–condition–method Cartesian set, duplicates, file hashes, schedules, finite values and dimensions. It independently implements the update equations and compares every stored loss, forecast weight and gain diagnostic. It then reconstructs seed-level endpoints and the paired bootstrap, including the full progression conjunction. Technical failure produces a separate audit artifact instead of deleting a failed episode or converting it into a quality result.

The control audit independently replays the native CPU simulator from its seed and executed commands. It checks each physical distance, warmup return, goal placement, initial/goal RGB, endpoint scoring and two-step success rule. Saved executed-command banks support independent posterior and forecast arithmetic. Saved 81-element cost vectors allow the selected argmin and tie rule to be checked; the chosen branch's cost is independently recomputed.

A material storage limit remains: the runner does not save all 81×25 latent predictions for every planner step. Thus the independent audit cannot regenerate every unchosen action's neural cost from stored banks. It also does not rerun the image encoder or full learned world-model forward. These limits distinguish reproducible decision arithmetic and physical execution from a complete independent neural regeneration claim.

Bank generation costs and cached CPU estimator costs must be reported separately. Skipping a weight update or using a small calibrator is not evidence that bank construction is cheap. The frozen reference can operate with one expert forward in an efficient implementation; the matched-compute control probe intentionally evaluates its full bank for comparison. Batch-amortized generation throughput is not single-stream closed-loop latency.

The local reproduction package retains protocols, source identities, selected configuration and trained weights, dependency information, complete per-step evidence, independent audit scripts and result artifacts. It attributes upstream code and checkpoints. Redistribution depends on the applicable source, weight and dependency licenses. No public repository publication or external submission is performed by this research draft.

12 Discussion and limitations

A small structural prior. The action family is known, positive and low dimensional. Twenty-five diagonal gain hypotheses are not a general physics model. They do not express delays, friction, inertia, actuator saturation, cross-axis coupling or contact-mode dynamics. A fixed rotation is one stress test rather than a comprehensive outside-family evaluation. Latest-only scaling further restricts the tested interface.

Three kinds of learning must stay separate. The upstream encoder and dynamics predictor were trained by H-JEPA's authors. The classical estimators fit feedback state without changing

those weights. The v0.4 follow-up genuinely trains 5,273 offline parameters and freezes them during deployment. Combining these components does not mean KineJing has pretrained a new general world model. API functionality and adapters are not evidence of trained backbone performance.

Latent scale and transfer. Raw latent MSE can differ across independently trained checkpoints and environments. The primary protocol averages supplied checkpoint losses as locked, but per-checkpoint results are required to expose scale differences. Dimension-preserving evidence features do not ensure semantic invariance across FourRoom and PushT. Training and selection on one environment can overfit its loss geometry despite a small parameter count.

Restoration and identification. A useful shifted correction can become a liability immediately after restoration. Fast forgetting or shrinkage may reduce that liability while weakening shifted correction. A mixture's coordinate-weighted diagnostic gain is not the gain used in a nonlinear forward and is not a calibrated physical estimate. Correct physical gains need not minimize a misspecified latent model's error. These are reasons to measure prediction, physical motion and task outcomes separately.

Statistical scope. The bootstrap conditions on a small set of supplied checkpoints, the command policy, schedule distribution and environment family. It does not quantify all upstream training uncertainty. Four validation task seeds cannot establish generalization of a learned calibrator. Secondary gain, rotation, lag and controller comparisons are exploratory, with no multiplicity-corrected superiority claim. Failed development eligibility, failed primary decisions and uncertainty crossing a guardrail must remain visible.

Control scope. Open-room local goals and a one-step grid planner do not establish long-horizon navigation, collision avoidance, task-directed exploration, robotic manipulation or hardware robustness. All controllers share a task generator and feedback opportunity budget but can induce different observation streams. PushT prediction transfer is not PushT task success, and environment-specific pretrained backbones are not shared foundation weights.

What would support stronger claims. A methodological contribution needs a distinct mechanism, faithful comparisons with the closest published systems, ablations beyond standard averaging, and independent reproduction. Broader model claims need genuinely new environments, observation processes and physical perturbations. Deployment claims additionally need relevant closed-loop and hardware tests, latency, resource usage and failure characterization. No expenditure, preprint, code release or numerical project gate guarantees journal acceptance or a permanent lead over other researchers.

13 Conclusion

KWM Research here examines a causal action-calibration interface around frozen attributed visual world models. The protocol makes restoration cost explicit, compares standard MAP and mixture controls fairly, and adds an actual small trained evidence aggregator without changing the upstream encoder or predictor. The locked candidate's 7.68% recovery improvement over legacy misses the 10% point target and leaves 24.02% regression versus frozen. The trained follow-up improves that

adaptive reference by 10.48% but still has 20.26% frozen-reference recovery regression and fails its normal uncertainty guard. Both retain excess error after restored feedback, so the first hidden step is not the whole explanation.

Local control and PushT transfer provide useful but narrower evidence: classic calibration raises fast local-control success, and transferred recovery benefits can coexist with shifted-error costs. The trained controller's 9/16 fast successes do not exceed the classical reference's 10/16, despite lower distance point estimates. Predictive averaging has a standard squared-error rationale whose assumptions and control implications must be checked. These results support a reproducible calibration study with explicit trade-offs, not a general foundation model, universal recovery guarantee or new algorithmic principle.

References

[1] Wancong Zhang, Basile Terver, Michael Rabbat, Yann LeCun, and Randall Balestriero. H-JEPA: End-to-End Learning of Hierarchical World Models for Visual Planning. arXiv:2610.06805, 2026. https://arxiv.org/abs/2610.06805.

[2] Ying Wang, Oumayma Bounou, Yann LeCun, and Mengye Ren. AdaJEPA: An Adaptive Latent World Model. arXiv:2606.32026, 2026. https://arxiv.org/abs/2606.32026.

[3] Krishnam Soni, Aditya Sehgal, Vedant Dave, and Elmar Rueckert. Sandwich-Residuals: Parameter-Efficient Test-time Adaptation of World Models. arXiv:2609.21740, 2026. https://arxiv.org/abs/2609.21740.

[4] Zheyuan Zhang, Suyu Ye, Nakul Agarwal, Hossein Nourkhiz Mahjoub, Ehsan Moradi Pari, Daniel Khashabi, Tianmin Shu, and Vaishnav Tadiparthi. JEPA-TTT: Persistent Test-Time Training of Latent World Models for Planning under Dynamics Shifts. arXiv:2610.00722, 2026. https://arxiv.org/abs/2610.00722.

[5] Mark Herbster and Manfred K. Warmuth. Tracking the Best Expert. *Machine Learning*, 32:151–178, 1998. https://doi.org/10.1023/A:1007424614876.

[6] Olivier Bousquet and Manfred K. Warmuth. Tracking a Small Set of Experts by Mixing Past Posteriors. *Journal of Machine Learning Research*, 3:363–396, 2002. https://www.jmlr.org/papers/v3/bousquet02b.html.

[7] Wouter M. Koolen and Tim van Erven. Switching between Hidden Markov Models using Fixed Share. arXiv:1008.4532, 2010. https://arxiv.org/abs/1008.4532.

[8] Jennifer A. Hoeting, David Madigan, Adrian E. Raftery, and Chris T. Volinsky. Bayesian Model Averaging: A Tutorial. *Statistical Science*, 14(4):382–417, 1999. https://sites.stat.washington.edu/www/research/online/hoeting1999.pdf.

[9] Jungbin Jun and Ilsang Ohn. Adaptive Bayesian Online Learning via Expert Aggregation. arXiv:2607.20239, 2026. https://arxiv.org/abs/2607.20239.

[10] Ryan Prescott Adams and David J. C. MacKay. Bayesian Online Changepoint Detection. arXiv:0710.3742, 2007. https://arxiv.org/abs/0710.3742.

[11] Brian Lai and Dennis S. Bernstein. SIFt-RLS: Subspace of Information Forgetting Recursive Least Squares. arXiv:2404.10844, 2024. https://arxiv.org/abs/2404.10844.

[12] Arjun Subramanian. Counterfactual Action Evaluation, Observation Bottlenecks, and Representation Geometry in Joint-Embedding Predictive World Models. arXiv:2610.02860, 2026. https://arxiv.org/abs/2610.02860.

[13] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. arXiv:1711.05101, 2017; conference version ICLR 2019. https://arxiv.org/abs/1711.05101.

A Indexing and evidence correspondence

The generator has 83 outgoing macro blocks and 84 native frames. Scored prediction index s corresponds to outgoing raw block $t = s + 3$, observed embeddings at raw indices $t-2, t-1, t$, and target at $t+1$. The intervention test is $a \leq t - 3 < b$, so scored transition a first changes physical actuation and scored transition b first restores identity. All eight registered recovery transitions exist because $b \leq 55 < 80 - 7$. Independent synthetic indexing checks cover all declared seed/condition combinations without inspecting model quality.

Prediction source correspondence is `generate.py`, `evaluate.py`, `estimators.py`, `menu.json` and `protocol.json`. The trained follow-up has its own `learned_estimator.py`, training source, protocol, `trained-mixer.pt` and freeze manifest. Control and PushT each have separate frozen task definitions. The manuscript is standalone: no external figures, bibliography database or included result files are required to compile it.

B Reporting discipline and authorship

The numerical narrative distinguishes each separate audit from a successful scientific hypothesis. The principal audit artifacts are

- `pilot-v03/test/independent-audit.json`
- `pilot-v03/control/independent-control-audit.json`
- `pilot-v04/independent-audit.json`
- `pilot-v04/control/numeric-audit.json`
- PushT `transfer-audit.json` under its transfer evidence directory.

A valid checkpoint, correct tensor shapes, an API round trip or an accurate CPU action replay does not establish task performance. Conversely, a scientifically negative result can be fully reproducible. Only actual held-out evidence may enter result tables.

All upstream intellectual and artifact contributions remain attributed. The work uses simulated imagery rather than newly collected personal or robot-camera data. Human authors remain responsible for scientific review, licenses, final claims and author contributions. AI assistance with implementation, checking, analysis and drafting should be disclosed according to the eventual venue's policy; it does not establish human authorship approval. This draft performs no external submission or public release.