# PARM-WM: PERSISTENT ACTION–RESPONSE MEMORY FOR LONG-HORIZON VISUAL WORLD MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Long-horizon visual prediction must retain how a scene responds to earlier actions as real observations leave the rolling context. We introduce PARM-WM, a world model with Persistent Action–Response Memory (PARM). An observable response encoder combines visual feature changes with aligned actions to form a compact state; a zero-preserving, low-rank residual then reads this state throughout the rollout. A validation-selected latent interpolation combines the response branch with a JEPA continuation using one mixing weight across sources. We evaluate the complete predictor against seven action-conditioned baselines on three robotic trajectory sources, using seven observed frames to predict twenty future frames. PARM-WM achieves the best values on all four metrics for RoboMimic Lift, LIBERO New, and LanguageTable, reducing pixel MSE relative to JEPA-WM by 49.3%, 3.8%, and 12.1%, respectively. Globally Holm-corrected paired tests confirm improvements over every baseline on all four metrics on all three sources. These results establish persistent action–response memory as a useful interface for sustained visual forecasting.

## 1 INTRODUCTION

An action specifies a command; an observed transition reveals how the scene responded. This distinction matters for visual world models used to anticipate robot interactions, where image-conditioned dynamics connect actions to their consequences (Finn et al., 2016; Ebert et al., 2018; Zhou et al., 2025). During an open-loop forecast, real observations progressively leave the model's rolling context. Recent world models address long-range consistency through recurrent state, retrieved history, and explicit memory (Samsami et al., 2024; Guo et al., 2026; Stapf et al., 2026). We address the complementary problem of retaining observed action–response evidence as prediction moves beyond the observed prefix.

Representation learning has substantially improved the space in which world models predict. Latent dynamics support imagination-based control (Ha & Schmidhuber, 2018; Hafner et al., 2020; 2021; 2025), while joint-embedding objectives and pretrained visual features provide reusable representations for forecasting and planning (Assran et al., 2023; Bardes et al., 2024; Zhou et al., 2025; Assran et al., 2025). Studies of predictive representations, temporal objectives, and context diversity further show that modeling choices beyond visual reconstruction affect learned dynamics (Burchi & Timofte, 2025; Pang et al., 2025; Terver et al., 2026; Wang et al., 2026). These advances motivate an explicit distinction between the evolving scene representation and the observed interaction evidence used to condition its future evolution.

Consider a robot pushing an object. Its observed displacement depends on the current scene and interaction dynamics, making recent responses useful context for adaptation (Kumar et al., 2021; Lyu et al., 2025). A rolling-context predictor can directly access an observed action–response event while those frames remain in its input; afterward, its influence must be carried through generated representations. Longer histories and recurrent architectures supply temporal capacity (Wang et al., 2017; Chang et al., 2021; Tang et al., 2023), and memory retrieval preserves access to selected observations (Guo et al., 2026; Chun et al., 2026). Our design instead summarizes completed action–response pairs once and exposes this summary to every subsequent prediction step.

We introduce PARM-WM, a world model with Persistent Action–Response Memory. Its writer encodes aligned actions and visual feature changes before pooling them into a bounded state. Its



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.

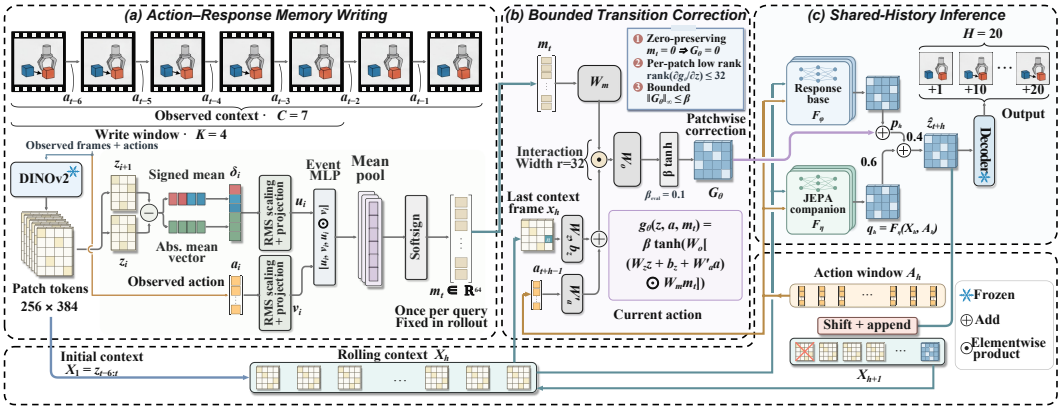



Figure 1: Persistent action–response conditioning. An observed prefix writes a bounded state; the state conditions subsequent latent transitions through a low-rank residual. The response-conditioned and JEPA branches share an autoregressive context and a fixed interpolation weight.

reader uses that state to modulate both the visual and action dependence of a bounded, low-rank transition correction. The resulting interface separates evidence acquisition from predictive evolution: observations write the state, and future predictions read it without rewriting it. Structural analysis characterizes zero-state recovery, local Jacobian rank, and finite-horizon influence. The evaluated predictor combines the response branch with a separately fitted JEPA continuation through a fixed shared-history interpolation.

We evaluate seven-frame-to-twenty-frame prediction on RoboMimic Lift, LIBERO New, and LanguageTable (Mandlekar et al., 2022; Liu et al., 2023; Lynch et al., 2022). The complete predictor achieves the best point estimates on all four metrics for all three sources, with pixel-MSE reductions of 49.3%, 3.8%, and 12.1% over JEPA-WM. Globally corrected paired tests establish all-baseline, all-metric improvements on all three sources. Controlled state ablations examine what the conditioning state contributes and how its effect changes with prediction length.

Our contributions are threefold:

- We separate observed action–response evidence from the evolving prediction context, establishing an explicit information path for retaining interaction evidence in long-horizon visual world models.
- We instantiate this path with paired memory writing and bounded transition reading, deriving exact local properties and finite-horizon perturbation bounds under explicit stability assumptions.
- We evaluate the complete predictor across three trajectory sources and seven baseline families, connecting broad forecasting comparisons with controlled state interventions to clarify the predictive value of response-conditioned dynamics.

## 2 Related Work

**Latent world models and joint-embedding prediction.** World Models and PlaNet established compact predictive states as a basis for imagination and planning, while Dreamer and DreamerV2 connected latent dynamics to value learning (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2020; 2021). DreamerV3 and TD-MPC2 extend this recipe across broader task and control regimes (Hafner et al., 2025; Hansen et al., 2024). Joint-embedding prediction provides a complementary route: I-JEPA and V-JEPA learn representations by predicting features rather than reconstructing every pixel (Assran et al., 2023; Bardes et al., 2024); V-JEPA 2 extends this family to action-conditioned robotic planning (Assran et al., 2025). DINOv2 and DINO-WM show how frozen visual features can support transferable representation and transition learning from offline trajectories (Oquab et al., 2024; Zhou et al., 2025). Generative interactive environments such as Genie learn latent action controls from unlabeled videos, while diffusion world models such as DIAMOND preserve pixel

details in interactive simulation (Bruce et al., 2024; Alonso et al., 2024). Recent analyses of JEPA world models and context diversity further separate representation quality, transition learning, and context design (Terver et al., 2026; Wang et al., 2026). PARM-WM addresses the transition interface by retaining a compact summary of realized responses alongside the evolving latent context.

**History-based adaptation and persistent memory.** Adaptive robotics commonly infers hidden context from interaction history. RMA learns a latent environment context for rapid locomotion adaptation, and DyWA extends history-based adaptation to manipulation with future-state prediction and feature modulation (Kumar et al., 2021; Lyu et al., 2025). Memory-oriented world models preserve or retrieve observations across longer horizons: R2I studies memory tasks in world-model control, Ctrl-World retrieves sparse history for controllable robot prediction, and CoME composes memory experts for diffusion world models (Samsami et al., 2024; Guo et al., 2026; Stapf et al., 2026). Sparse imagination reduces token processing during visual rollout, while WorldGym evaluates policies with action-conditioned rollouts (Chun et al., 2026; Quevedo et al., 2026). These methods use different memory objects, generators, and control objectives. PARM-WM introduces a fixed state written from completed action–response pairs and read through a bounded latent residual.

**Temporal prediction and response-conditioned dynamics.** Recurrent video predictors propagate temporal state through spatial and temporal memory, while MAU and SwinLSTM use motion-aware or windowed attention mechanisms (Wang et al., 2017; Chang et al., 2021; Tang et al., 2023). SimVP and TAU demonstrate efficient convolutional and attention-based alternatives for multi-step video prediction (Gao et al., 2022; Tan et al., 2023). Recent ICLR work such as TWISTER, ReViWo, and FLIP studies contrastive, representation, and long-horizon objectives for world-model forecasting (Burchi & Timofte, 2025; Pang et al., 2025; Gao et al., 2025). UWM and AVID study action-conditioned visual generation, while VideoWorld learns latent video dynamics from unlabeled videos (Zhu et al., 2025; Rigter et al., 2025; Ren et al., 2025). Astra extends interactive video prediction with noisy-history memory and action-aware adapters, while explicit digital-twin construction and video-generated motion offer complementary robot-world-model routes (Zhu et al., 2026; Barcellona et al., 2025; Bharadhwaj et al., 2025). These generative systems provide context for the design space; our experiments compare the action-conditioned latent and video-prediction families listed in Section 5.1.

**Positioning.** The writer draws on permutation-invariant set aggregation, while the reader is related in spirit to feature-wise conditioning and low-rank adaptation (Zaheer et al., 2017; Perez et al., 2018; Hu et al., 2022). Those works motivate useful design primitives but do not specify a persistent action–response channel for visual forecasting. PARM-WM combines the primitives into a write-once/read-through interface, analyzes its local response geometry, and evaluates the resulting complete predictor across held-out robotic trajectory sources.

## 3 Problem formulation

**Action-conditioned visual forecasting.** At time $t$, an agent has an observed image prefix $o_{t-C+1:t}$, aligned actions $a_{t-C+1:t-1}$, and a known future action sequence. A visual encoder $S$ maps observations to latent tokens $z_t = S(o_t) \in \mathbb{R}^{N\times d}$. An action-conditioned predictor receives a rolling context $X_h$ and predicts future tokens $\hat{z}_{t+h}$, which are appended to the context and decoded for visual evaluation. This latent forecasting formulation follows the predictive-state and action-conditioned world-model literature (Ha & Schmidhuber, 2018; Hafner et al., 2019; Bardes et al., 2024; Zhou et al., 2025). The goal is to forecast the future under the supplied actions while retaining information that is available in the observed prefix.

**Preserving observed response evidence.** Actions specify interventions; observations reveal their realized effects. In rolling-context prediction, executed actions remain available while the observations that reveal their effects are progressively replaced by predicted frames. Longer contexts and recurrent state increase capacity, but do not by themselves require an observed action to remain paired with its realized response. We therefore separate observed response evidence from the evolving visual context, retaining it in a compact state that remains available throughout the forecast.

Our design encodes action–response pairs before aggregation, writes their summary once per observed query, and retains the resulting state throughout the open-loop horizon. A bounded reader controls the transition correction and exactly recovers the base predictor at zero state. Section 4 formalizes the paired writer, low-rank reader, and shared-history rollout. This separation distinguishes persistent context from recurrent state propagation and retrieved history (Wang et al., 2017; Guo et al., 2026; Stapf et al., 2026).

# 4 Persistent Action–Response Memory

## 4.1 Interface overview

PARM-WM is a write–read–rollout interface. At the start of each prediction query, a writer maps completed action–response events to a compact state $m_t$; a reader uses that state and the current visual–action input to modify each latent transition. The state is then held fixed while the visual context evolves. The evaluated system fits the response branch and a companion JEPA branch separately, and at inference interpolates their predictions before advancing one shared history.

## 4.2 Writing paired action–response memory

Let $\tau = t - C + 1$ and $\mathcal{I}_t = \{\tau, \dots, \tau + K - 1\}$ index $K \le C - 1$ completed transitions. All experiments fix $K = 4$ at the start of the seven-frame context: the first five frames write memory, and all seven initialize prediction. For $i \in \mathcal{I}_t$,

$$\delta_i = \left[ \frac{1}{N} \sum_{n=1}^{N} (z_{i+1,n} - z_{i,n}),\ \frac{1}{N} \sum_{n=1}^{N} |z_{i+1,n} - z_{i,n}| \right] \in \mathbb{R}^{2d}. \tag{1}$$

These terms capture signed feature-space change and changes that cancel under signed spatial averaging. Training-prefix coordinatewise RMS scales $s_\delta, s_a$ normalize the writer inputs; their estimation is specified in Appendix D.4.

With $u_i = W_\delta(\delta_i \oslash s_\delta)$ and $v_i = W_a(a_i \oslash s_a)$, the writer computes

$$r_i = W_2 \,\mathrm{SiLU}(W_1[u_i, v_i, u_i \odot v_i]), \qquad m_t = \mathrm{softsign}\left( \frac{1}{K} \sum_{i \in \mathcal{I}_t} r_i \right). \tag{2}$$

The product exposes within-event interactions before pooling: different pairings can have identical marginal means and different product means (Appendix C). Separate response and action paths also enter the event map directly. This explicitly supplies pairing features to a learned 64-dimensional summary.

Bias-free projections and mean pooling implement a permutation-invariant set summary of complete events (Zaheer et al., 2017) and are zero-preserving for zero model-space inputs. Here $a_i = 0$ denotes the training-mean raw action. Softsign bounds each memory coordinate in $(-1, 1)$. The pooled state summarizes global response statistics; spatial layout and recent temporal order remain in the visual context. The fixed prefix is the writing window of the evaluated architecture. Within each query, predicted frames update $X_h$ while $m_t$ continues to represent the observed events.

## 4.3 Reading memory through a bounded transition correction

For a patch $z = x_{h,n}$ in the last context frame and current standardized action $a$, the reader produces

$$g_\theta(z, a, m) = \beta \tanh\big(W_o\big[(W_z z + b_z + W'_a a) \odot W_m m\big]\big), \qquad \beta \ge 0. \tag{3}$$

The projections share width $r = 32$, and $W_o$ maps back to $d$ features. Only the visual projection has a bias. State and action are shared across patches, while $z$ determines each patch's correction. Initializing $W_o = 0$ starts the branch at its base predictor.

The memory modulates both visual and action dependence. Let $D_m = \mathrm{diag}(W_m m)$. The pre-activation is

$$\begin{aligned} s(z, a, m) &= R_z(m) z + R_a(m) a + b_m(m), \\ R_z(m) &= W_o D_m W_z, \quad R_a(m) = W_o D_m W'_a, \quad b_m(m) = W_o D_m b_z. \end{aligned} \tag{4}$$

Thus past responses reweight how current visual features and commands enter the correction. This combines feature-wise conditioning with a low-dimensional adaptation path, related to general conditioning layers and low-rank parameterizations (Perez et al., 2018; Hu et al., 2022). Each linear map factors through the same $r$ channels; Appendix C gives the corresponding rank-one expansion.

**Proposition 1** (Controlled response conditioning)**.** *For any patch $z$, action $a$, and state $m$, the reader satisfies*

$$g_\theta(z, a, 0) = 0, \qquad \|g_\theta(z, a, m)\|_\infty \le \beta, \qquad \operatorname{rank}(J_z g_\theta) \le r. \tag{5}$$

The multiplicative path gives exact zero-state recovery, output tanh controls the injected amplitude, and the factorization bounds the local Jacobian rank for each patch. These roles are distinct from softsign's control of memory scale. Appendix C proves the properties and bounds sensitivity to state changes.

### 4.4 Learning with persistent response context

Let $x_h$ be the last frame of $X_h$ and $A_h = (a_{t+h-C}, \dots, a_{t+h-1})$. Stacking the patch corrections as $G_\theta$, the response branch predicts

$$p_h = F_\phi(X_h, A_h) + G_\theta(x_h, a_{t+h-1}, m_t). \tag{6}$$

During fitting, $p_h^{\mathrm{tr}}$ denotes Equation (6) evaluated on the branch's own context $X_h^p$, which advances as $X_{h+1}^p = \mathrm{shift}(X_h^p, p_h^{\mathrm{tr}})$. The shift removes the oldest frame and appends the prediction. With target $z_{t+h}^* = S(o_{t+h})$ and source-shared image decoder $D$, the objective is

$$\mathcal{L} = \frac{1}{H_{\mathrm{tr}}} \sum_{h=1}^{H_{\mathrm{tr}}} \left[ \frac{\|p_h^{\mathrm{tr}} - z_{t+h}^*\|_F^2}{Nd} + \gamma \left( \ell_{\mathrm{RGB}}(D(p_h^{\mathrm{tr}}), o_{t+h}) + \omega \ell_{\mathrm{mov}}(D(p_h^{\mathrm{tr}}), o_{t+h}; o_t) \right) \right]. \tag{7}$$

The image losses measure full-frame and dynamic-region error (Section 5.1). The encoder and decoder parameters are frozen. The writer and reader learn with frozen $F_\phi$ for three sources and joint continuation on RoboMimic. Gradients propagate through all $H_{\mathrm{tr}}$ prediction steps, including the fixed memory, predicted context, and decoder inputs. The companion $F_\eta$ is fitted separately with its own predicted history and forecasting loss. Appendix D.4 specifies the training stages and normalization.

### 4.5 Shared-history inference

After fitting, the two branches read one common context. The full predictor is

$$\begin{aligned} \hat{z}_{t+h} &= \mathcal{B}_h(X_h) + \lambda G_\theta(x_h, a_{t+h-1}, m_t), \\ \mathcal{B}_h(X) &= \lambda F_\phi(X, A_h) + (1 - \lambda) F_\eta(X, A_h). \end{aligned} \tag{8}$$

This expression separates the two fitted predictors' combination from the explicit memory correction. Setting $m_t = 0$ recovers $\mathcal{B}_h$; $F_\phi$ and $F_\eta$ denote the response base and companion, respectively. The JEPA-WM table row is selected by the external-baseline protocol, with checkpoint identities detailed in Appendix D.4. Only the mixed prediction advances inference:

$$X_{h+1} = \mathrm{shift}(X_h, \hat{z}_{t+h}), \qquad \hat{o}_{t+h} = D(\hat{z}_{t+h}), \qquad X_1 = z_{\tau:t}. \tag{9}$$

The common weight $\lambda = 0.4$ is selected on RoboMimic validation and retained across sources; evaluation uses $\beta = 0.1$. Both settings are fixed before testing. Appendix D.4 records RoboMimic's training/evaluation amplitude distinction. Proposition 5 characterizes finite-horizon latent displacement under a state intervention, relating bounded injection to propagation through the fitted mixture.

## 5 Experiments

The experiments evaluate cross-source forecasting accuracy, the contribution of response-state information, and its dependence on prediction length. Held-out tests assess the complete predictor; controlled state replacements isolate conditioning effects on the validation split. Our code is provided as an attachment in the submission system.

### 5.1 EVALUATION PROTOCOL

**Sources and observations.** We evaluate episode-held-out subsets of RoboMimic Lift, four LIBERO tasks, and LanguageTable block-to-block simulation (Mandlekar et al., 2022; Liu et al., 2023; Lynch et al., 2022). RoboMimic and LIBERO use 120/40/40 train/validation/test episodes. LanguageTable has 144/48/48 episodes, of which 111/38/36 support seven observed and 20 future frames. The 55 shorter episodes stay in their original splits but contribute no H20 windows. This manuscript evaluates these three trajectory sources.

All models receive seven $64 \times 64$ RGB frames and aligned past and known future actions, standardized from training transitions. Actions align with the transitions they cause; language, proprioception, reward, and simulator state are excluded. Latent models resize images to the frozen DINOv2-S/14 encoder's $224 \times 224$ input. Source frame rates are 20, 20, and 10 Hz respectively, so H20 spans different physical durations.

**Metrics and aggregation.** For future step $h$ and pixel location $u$, the dynamic mask is $M_h(u) = \mathbf{1}\{\max_c |o_{t+h}(u,c) - o_t(u,c)| > 5/255\}$. With $e_h = \hat{o}_{t+h} - o_{t+h}$, the two error metrics are

$$\ell_{\mathrm{RGB}} = \frac{\sum_{u,c} e_h(u,c)^2}{3 \cdot 64^2}, \qquad \ell_{\mathrm{mov}} = \frac{\sum_{u,c} M_h(u) e_h(u,c)^2}{\max(3 \sum_u M_h(u), 1)}. \tag{10}$$

Motion MSE measures image error within this changing-region mask. PSNR is $-10 \log_{10} \max(\ell_{\mathrm{RGB}}, 10^{-12})$ per frame; SSIM uses an $11 \times 11$ Gaussian window, $\sigma = 1.5$, unit data range, and population moments on valid spatial support. Outputs are scored without clipping. H20 averages all 20 predicted frames. We average up to three unique, equally spaced valid queries within each episode, then episodes within tasks, and finally weight tasks equally.

**Comparators and model selection.** Seven comparators cover DINO-WM, JEPA-WM, and local action-conditioned adaptations of SimVP, TAU, PredRNNv2, MAU, and SwinLSTM-B (Zhou et al., 2025; Terver et al., 2026; Gao et al., 2022; Tan et al., 2023; Wang et al., 2023; Chang et al., 2021; Tang et al., 2023). The five video architectures receive broadcast action channels; recurrent cores retain their states during rollout. DINO-WM, JEPA-WM, and our branches share frozen visual features and a source-specific image decoder; the video models directly predict RGB. The metrics compare these complete prediction pipelines.

Learning rates, training horizons, continuations, and decoder fits are selected on training and validation data. Each main row uses one seed-zero fit per method and source, including both fitted branches of our ensemble. Training compute varies across pipelines; source-specific budgets and the additional RoboMimic continuation are documented with the configurations. The blend weight is fixed before testing.

**Paired uncertainty.** Matched held-out episodes are compared using 10,000 paired bootstrap resamples and a one-sided sign-permutation test with 20,000 draws. Holm correction covers all $3 \times 7 \times 4 = 84$ source–baseline–metric tests. Boldface denotes the best point estimate; significance requires adjusted $q < 0.05$. Intervals estimate between-episode variation for the fixed fitted checkpoints.

### 5.2 LONG-HORIZON PREDICTION ACROSS SOURCES

Tables 1–3 compare all baselines on all three sources. PARM-WM uses $\lambda = 0.4$ across separately fitted source predictors. A dagger on a baseline cell denotes a significant improvement by PARM-WM after global Holm correction.

**RoboMimic Lift: the largest improvement.** PARM-WM improves every baseline in all four metrics after correction. Relative to JEPA-WM, pixel and motion MSE fall by 49.3% and 47.8%; PSNR increases by 2.776 dB and SSIM by 0.0323. Paired 95% intervals are $[0.000825, 0.001030]$ for pixel-MSE reduction and $[2.501, 3.044]$ dB for PSNR gain. The complete two-branch predictor improves both full-frame and changing-region accuracy.

Table 1: **RoboMimic Lift.** H20 test prediction over 40 held-out episodes. Bold denotes the best point estimate; † marks a baseline cell improved by PARM-WM with global Holm $q < 0.05$ across 84 tests. All 28 comparisons favor PARM-WM after correction.

| Method | Pixel MSE ↓ | Motion MSE ↓ | PSNR ↑ | SSIM ↑ |
|---|---|---|---|---|
| DINO-WM | 0.003554† | 0.016815† | 25.9208† | 0.8825† |
| JEPA-WM | 0.001881† | 0.008518† | 27.9785† | 0.9194† |
| SimVP + actions | 0.002635† | 0.010764† | 26.3183† | 0.8664† |
| TAU + actions | 0.002412† | 0.010868† | 27.0596† | 0.8912† |
| PredRNNv2 + actions | 0.004930† | 0.020608† | 23.6599† | 0.7517† |
| MAU + actions | 0.004296† | 0.015748† | 24.1776† | 0.8252† |
| SwinLSTM-B + actions | 0.007798† | 0.037025† | 21.6956† | 0.7474† |
| PARM-WM (ours) | **0.000954** | **0.004450** | **30.7548** | **0.9517** |

Table 2: **LIBERO New.** H20 test prediction over 40 held-out episodes. Bold denotes the best point estimate; † marks a baseline cell improved by PARM-WM with global Holm $q < 0.05$ across 84 tests. All 28 comparisons favor PARM-WM after correction.

| Method | Pixel MSE ↓ | Motion MSE ↓ | PSNR ↑ | SSIM ↑ |
|---|---|---|---|---|
| DINO-WM | 0.005026† | 0.022401† | 24.0720† | 0.8070† |
| JEPA-WM | 0.003952† | 0.017070† | 24.9904† | 0.8278† |
| SimVP + actions | 0.005327† | 0.020088† | 23.7462† | 0.7503† |
| TAU + actions | 0.004818† | 0.019448† | 24.2219† | 0.7703† |
| PredRNNv2 + actions | 0.006288† | 0.025491† | 22.7645† | 0.6463† |
| MAU + actions | 0.006633† | 0.024444† | 22.7606† | 0.7248† |
| SwinLSTM-B + actions | 0.009638† | 0.041263† | 21.2992† | 0.6600† |
| PARM-WM (ours) | **0.003803** | **0.016780** | **25.1707** | **0.8322** |

**LIBERO New: gains across all four metrics.** PARM-WM leads all four point estimates. Against JEPA-WM, pixel and motion MSE decrease by 3.8% and 1.7%, and PSNR increases by 0.180 dB; the paired 95% pixel-MSE reduction interval is $[0.000106, 0.000194]$. All 28 comparisons pass global correction over the 84-test family. The JEPA-WM motion-MSE comparison has $q = 0.0097$.

**LanguageTable: prediction in planar manipulation.** PARM-WM improves all seven baselines across all four metrics after correction. Relative to JEPA-WM, pixel and motion MSE decrease by 12.1% and 8.5%; PSNR rises by 0.445 dB and SSIM by 0.0081. Paired 95% intervals are $[0.000280, 0.000547]$ for pixel-MSE reduction and $[0.357, 0.537]$ dB for PSNR gain.

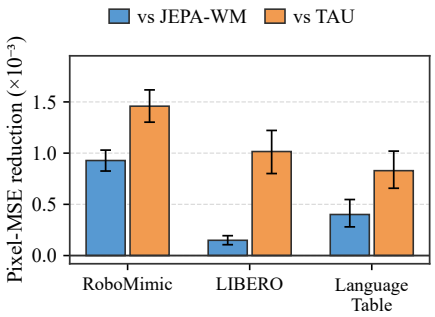



Figure 2: H20 pixel-MSE reduction against JEPA-WM and TAU; positive favors PARM-WM. Error bars: recorded paired 95% bootstrap intervals.

Figure 2 contrasts the representation-aligned JEPA reference with the direct-RGB TAU comparator across the three sources.

The agreement between full-frame and motion-region metrics links overall reconstruction quality to the parts of the scene affected by interaction. A single interpolation weight supports this ordering across sources with different error scales and gain magnitudes. Together, the JEPA-WM and TAU comparisons establish the complete predictor's advantage against both latent-space and direct-RGB forecasting baselines.

### 5.3 Information carried by the persistent state

Table 3: **LanguageTable block-to-block simulation.** H20 test prediction over 36 held-out episodes. Bold denotes the best point estimate; † marks a baseline cell improved by PARM-WM with global Holm $q < 0.05$ across 84 tests. All 28 comparisons favor PARM-WM after correction.

| Method | Pixel MSE ↓ | Motion MSE ↓ | PSNR ↑ | SSIM ↑ |
|---|---|---|---|---|
| DINO-WM | 0.005182† | 0.025708† | 24.1494† | 0.8450† |
| JEPA-WM | 0.003312† | 0.014198† | 25.6078† | 0.8790† |
| SimVP + actions | 0.004826† | 0.024693† | 24.3730† | 0.8285† |
| TAU + actions | 0.003740† | 0.019667† | 25.3315† | 0.8649† |
| PredRNNv2 + actions | 0.004274† | 0.021374† | 25.0666† | 0.8418† |
| MAU + actions | 0.010691† | 0.036349† | 20.8831† | 0.7786† |
| SwinLSTM-B + actions | 0.005657† | 0.035845† | 24.0296† | 0.8612† |
| PARM-WM (ours) | **0.002911** | **0.012988** | **26.0533** | **0.8871** |

Table 4: **Persistent-state interventions.** H20 RoboMimic validation: 40 episodes, one fitted seed, frozen-base response model. Full uses the correct action–response state. Same-task shuffling, action-only inputs, and zeroing reuse its checkpoint; static states are trained separately (LR: learning rate). Table 1 reports the joint-trained ensemble. Bold denotes the best point estimate.

| Method | Pixel MSE ↓ | Motion MSE ↓ | PSNR ↑ | SSIM ↑ |
|---|---|---|---|---|
| Full | **0.001685** | **0.007654** | **28.1984** | **0.9225** |
| Shuffled state | 0.001736 | 0.007900 | 28.1008 | 0.9206 |
| Action-only state | 0.001947 | 0.008771 | 27.6186 | 0.9132 |
| Zero state | 0.002739 | 0.011832 | 26.3506 | 0.8912 |
| Static state (LR $10^{-4}$) | 0.002152 | 0.009601 | 27.2259 | 0.9088 |
| Static state (LR $3 \times 10^{-4}$) | 0.001895 | 0.008520 | 27.7334 | 0.9158 |
| Static state (LR $10^{-3}$) | 0.001740 | 0.007910 | 28.0847 | 0.9210 |

Table 4 isolates the contribution of persistent-state information on the RoboMimic validation split. Inference interventions keep the fitted model parameters fixed and use zero, another same-task episode's state, or action-only writer inputs. Three static controls learn constant states with a context-dependent reader. Correct state leads all H20 metrics: motion error is 3.1% below shuffled, 12.7% below action-only, and 3.2% below the strongest static control. Its paired 95% motion-error interval relative to shuffled state is $[-0.000355, -0.000144]$. Shuffling retains 94.1% of the correct-versus-zero improvement, indicating a large shared correction plus an additional query-matching gain.

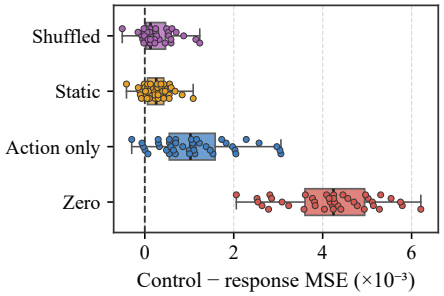


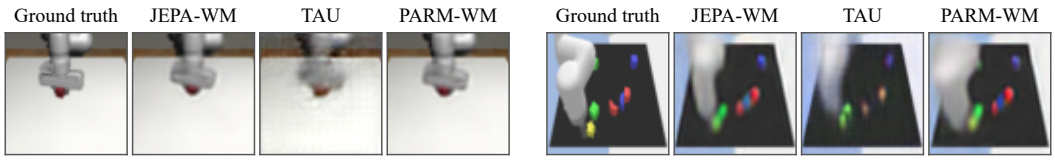

Figure 3: H20 motion-MSE penalties, 40 episodes per control. Boxes: quartiles and median; whiskers: full range. Points show every paired episode. Positive favors response.

These controls distinguish response evidence from a shared correction and from query matching. The action-only comparison tests the value of pairing commands with their observed consequences; static and shuffled states test what is added by conditioning on the current episode. The correct state improves on each control under the same evaluation protocol.

Figure 3 shows lower response-state motion error on 32/40 episodes against shuffled state, 34/40 against static state, 37/40 against action-only state, and 40/40 against zero state.

### 5.4 State effects across prediction horizons

Figure 4 shows expanding motion-error differences from H1 to H20. The response state gains an advantage over the strongest static control as the forecast horizon increases. Static and shuffled states capture the shared correction, while matching the state to the observed episode adds a further gain. The memory remains fixed throughout these rollouts, while the visual context progressively




(a) RoboMimic R1: lift continuation (b) LanguageTable T1: arm continuation

Figure 5: Qualitative comparison at $t + 20$: RoboMimic R1 and LanguageTable T1. Each comparison fixes the observed prefix and future actions. PARM-WM lowers both pixel and motion errors over H20 and at the endpoint relative to both displayed baselines. Full frames share the same spatial scale.

replaces observations with predictions. The increasing gaps show that the information written from the observed prefix remains useful during later autoregressive transitions.

Curves average the first $H$ prediction errors; static uses the best of three validation learning rates. These ablations characterize the contribution of state information across prediction horizons, complementing the source-wise comparisons of overall forecasting performance.

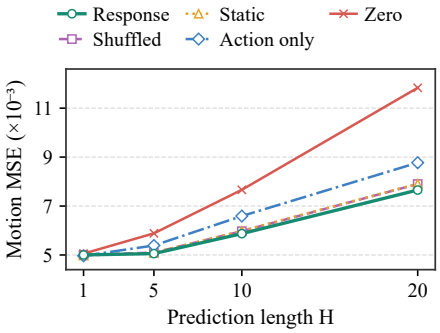



Figure 4: Mean motion error over the first $H$ predictions. Markers denote measured horizons; lines connect them. Same fitted model and 40 validation episodes as Figure 3.

### 5.5 Qualitative comparisons

Figure 5 compares the twentieth predicted frame for RoboMimic R1 and LanguageTable T1. In R1, PARM-WM better preserves the lifted gripper/object location; in T1, it better follows the changing arm configuration than the straighter predictions of JEPA-WM and TAU. H20 motion MSE decreases by 62.3%/63.4% in R1 and 53.7%/54.6% in T1 relative to JEPA-WM/TAU. The predictions capture sustained motion beyond the observed prefix. Appendix G.1 provides the complete rollouts.

## 6 Conclusion and Limitation

**Conclusion.** We introduced PARM-WM, a visual world model that separates observed action–response evidence from the evolving prediction context. A paired writer summarizes completed interactions in a bounded state, and a zero-preserving, low-rank reader makes that evidence available at every future transition. This write-once/read-through design retains the conditioning signal as autoregressive predictions progressively replace the original observations.

Across RoboMimic Lift, LIBERO New, and LanguageTable, the complete predictor combines response-conditioned dynamics with a JEPA continuation through one fixed interpolation weight and improves all seven baselines on all four metrics after global correction. Controlled state interventions distinguish a shared correction from the additional value of response evidence and query matching, with gains extending across the prediction horizon. Together, these results establish an explicit memory interface for preserving realized interaction experience in long-horizon visual forecasting.

**Limitations and future work.** This work studies persistent action–response memory in the context of visual forecasting for robotic manipulation. A natural direction for future work is to extend the framework to a broader range of tasks, embodiments, and environments, and to explore its use with richer multimodal observations. Beyond forecasting, integrating the proposed memory interface with closed-loop planning and control would allow future studies to examine how retained interaction experience supports sequential decision-making. Longer-term interaction also offers opportunities to study how such memory can be accumulated and reused across tasks. Building on the separation between observed interaction evidence and predictive dynamics, these extensions provide avenues for developing persistent memory mechanisms in more general embodied world models.

## AI USE STATEMENT

We used AI tools to assist with programming and to polish the language of the manuscript.

## RESPONSIBLE RESEARCH AND REPRODUCIBILITY

The study uses existing public or simulated robot trajectories. Section 5.1 defines the evaluation units and statistical procedure. The appendix specifies data eligibility, checkpoint selection, held-out tests, and validation interventions. Table-generation and plotting scripts read the recorded experiment summaries. Our code is provided as an attachment in the submission system.

## REFERENCES

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

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, June 2023.

Mahmoud 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.

Leonardo Barcellona, Andrii Zadaianchuk, Davide Allegro, Samuele Papa, Stefano Ghidoni, and Efstratios Gavves. Dream to manipulate: Compositional world models empowering robot imitation learning with imagination. In *International Conference on Learning Representations*, volume 2025, pp. 56729–56763, 2025.

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.

Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 3936–3951. PMLR, 27–30 Sep 2025.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Maria Elisabeth Bechtle, Feryal Behbahani, Stephanie C.Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando De Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4603–4623. PMLR, 21–27 Jul 2024.

Maxime Burchi and Radu Timofte. Learning transformer-based world models with contrastive predictive coding. In *International Conference on Learning Representations*, volume 2025, pp. 6949–6970, 2025.

Zheng Chang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, Yan Ye, Xinguang Xiang, and Wen Gao. MAU: A motion-aware unit for video prediction and beyond. In *Advances in Neural Information Processing Systems*, volume 34, pp. 26950–26962. Curran Associates, Inc., 2021.

Junha Chun, Youngjoon Jeong, and Taesup Kim. Sparse imagination for efficient visual world model planning. In *International Conference on Learning Representations*, volume 2026, pp. 102160–102188, 2026.

Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018.

B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, January 1979.

Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Chongkai Gao, Haozhuo Zhang, Zhixuan Xu, Zhehao Cai, and Lin Shao. FLIP: Flow-centric generative planning as general-purpose manipulation world model. In *International Conference on Learning Representations*, volume 2025, pp. 21927–21948, 2025.

Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z. Li. SimVP: Simpler yet better video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3170–3180, June 2022.

Yanjiang Guo, Lucy Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A controllable generative world model for robot manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 6121–6138, 2026.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31, pp. 2451–2463. Curran Associates, Inc., 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 09–15 Jun 2019.

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

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering Atari with discrete world models. In *International Conference on Learning Representations*, 2021.

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

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, volume 2024, pp. 47376–47405, 2024.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: Rapid motor adaptation for legged robots. In *Proceedings of Robotics: Science and Systems*, Virtual, July 2021.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 44776–44791. Curran Associates, Inc., 2023.

Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time. *arXiv preprint arXiv:2210.06407*, 2022.

Jiangran Lyu, Ziming Li, Xuesong Shi, Chaoyi Xu, Yizhou Wang, and He Wang. DyWA: Dynamics-adaptive world action model for generalizable non-prehensile manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11058–11068, October 2025.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1678–1690. PMLR, 08–11 Nov 2022.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. 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 Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Jing-Cheng Pang, Nan Tang, Kaiyuan Li, Yuting Tang, Xin-Qiang Cai, Zhen-Yu Zhang, Gang Niu, Masashi Sugiyama, and Yang Yu. Learning view-invariant world models for visual robotic manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 54853–54876, 2025.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1):3942–3951, April 2018.

Julian Quevedo, Ansh Kumar Sharma, Yixiang Sun, Varad Suryavanshi, Percy Liang, and Sherry Yang. Worldgym: World model as an environment for policy evaluation. In *International Conference on Learning Representations*, volume 2026, pp. 78932–78957, 2026.

Zhongwei Ren, Yunchao Wei, Xun Guo, Yao Zhao, Bingyi Kang, Jiashi Feng, and Xiaojie Jin. Videoworld: Exploring knowledge learning from unlabeled videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29029–29039, June 2025.

Marc Rigter, Tarun Gupta, Agrin Hilmkil, and Chao Ma. AVID: Adapting video diffusion models to world models. *Reinforcement Learning Journal*, 6:737–764, 2025.

Mohammad Reza Samsami, Artem Zholus, Janarthanan Rajendran, and Sarath Chandar. Mastering memory tasks with world models. In *International Conference on Learning Representations*, volume 2024, pp. 2312–2348, 2024.

Sebastian Stapf, Pablo Acuaviva, Aram Davtyan, and Paolo Favaro. Composition of memory experts for diffusion world models. In *International Conference on Learning Representations*, volume 2026, pp. 15377–15403, 2026.

Cheng Tan, Zhangyang Gao, Lirong Wu, Yongjie Xu, Jun Xia, Siyuan Li, and Stan Z. Li. Temporal attention unit: Towards efficient spatiotemporal predictive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18770–18782, June 2023.

Song Tang, Chuang Li, Pu Zhang, and RongNian Tang. SwinLSTM: Improving spatiotemporal prediction accuracy using Swin Transformer and LSTM. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13470–13479, October 2023.

Basile Terver, Tsung-Yen Yang, Jean Ponce, Adrien Bardes, and Yann LeCun. What drives success in physical planning with joint-embedding predictive world models? *Transactions on Machine Learning Research*, 2026.

Fan Wang, Zhiyuan Chen, Yuxuan Zhong, Sunjian Zheng, Pengtao Shao, Bo Yu, Shaoshan Liu, Jianan Wang, Ning Ding, Yang Cao, and Yu Kang. Context and diversity matter: The emergence of in-context learning in world models. In *International Conference on Learning Representations*, volume 2026, pp. 100857–100880, 2026.

Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S. Yu. PredRNN: Recurrent neural networks for predictive learning using spatiotemporal LSTMs. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Yunbo Wang, Haixu Wu, Jianjin Zhang, Zhifeng Gao, Jianmin Wang, Philip S. Yu, and Mingsheng Long. PredRNN: A recurrent neural network for spatiotemporal predictive learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2208–2225, February 2023. ISSN 1939-3539.

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

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Russ R Salakhutdinov, and Alexander Smola. Deep sets. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

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. PMLR, 13–19 Jul 2025.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025.

Yixuan Zhu, Jiaqi Feng, Wenzhao Zheng, Yuan Gao, Xin Tao, Pengfei Wan, Jiwen Lu, and Jie Zhou. Astra: General interactive world model with autoregressive denoising. In *International Conference on Learning Representations*, volume 2026, pp. 79167–79184, 2026.

# A APPENDIX OVERVIEW

This appendix develops the theory and evaluation of PARM-WM. Appendix B unifies the symbols and dimensions used by the model, rollout analysis, and evaluation. Appendix C expands the write–read interface with two formal algorithms, paired-event moments, summary equivalence, local reader geometry, gradient propagation, same-context mixture risk, parameter accounting, and finite-horizon intervention bounds. Appendix D specifies the datasets, training procedures, baselines, and evaluation protocol. Appendix E gives complete cross-source results and additional quantitative analyses. Appendix F examines state interventions, episode-level gains, endpoint accuracy, and validation-based mixture selection. Appendix G compares decoded rollouts under identical observed prefixes and future actions, including regional error decompositions and complete per-frame trajectories. Dataset-level results and matched-query visualizations characterize aggregate performance and its spatial and temporal structure, respectively.

# B NOTATION

This appendix uses one notation system for the model, its rollout analysis, and the experimental summaries. Dimensions distinguish vectors, token matrices, and their linear maps; calligraphic letters denote collections or operators. A subscript $t$ identifies the last observed frame of a query, whereas $h$ indexes a predicted step. Unless a superscript is shown, $a_i$ denotes the standardized action supplied to the model. The corresponding raw action is written $a_i^{\mathrm{raw}}$; hence $a_i = 0$ means the training mean of that action coordinate. Table 5 collects the symbols used throughout the main text and this appendix.

Table 5: Notation for PARM-WM. Dimensions are included where they fix the meaning of an operator.

| Symbol | Meaning |
|---|---|
| $H_{\mathrm{im}}, W_{\mathrm{im}}, N, d, d_a, r$ | Image height/width before encoder preprocessing, patch count, token width, action dimension, and reader width; the reported values are $64, 64, 256, 384, d_a, 32$. |
| $d_m, d_w$ | Memory dimension and writer hidden width, respectively 64 and 128, used for symbolic interface-complexity accounting. |
| $o_t \in [0,1]^{H_{\mathrm{im}} \times W_{\mathrm{im}} \times 3}$ | RGB observation at time $t$; the stored evaluation frame is $64 \times 64$, and $S$ internally resizes it to $224 \times 224$ for DINOv2. |
| $S, D$ | Frozen visual encoder and source-specific image decoder. |
| $z_t = S(o_t) \in \mathbb{R}^{N \times d}$ | Patch-token frame; $z_{t,n} \in \mathbb{R}^d$ is patch $n$. In the experiments $N = 256$ and $d = 384$. |
| $C, K, H, H_{\mathrm{tr}}$ | Number of observed frames, written action–response events, prediction steps, and training rollout steps. The reported setting is $(C, K, H) = (7, 4, 20)$. |
| $t, \tau, h, n, i$ | Last observed index, $\tau = t - C + 1$, predicted-step index ($1 \le h \le H$), patch index, and event index ($i \in \{\tau, \dots, \tau + K - 1\}$). |
| $\mathcal{I}_t$ | Writing-window event indices $\{\tau, \dots, \tau + K - 1\}$, where $K \le C - 1$. |
| $a_i^{\mathrm{raw}} \in \mathbb{R}^{d_a}$, $a_i \in \mathbb{R}^{d_a}$ | Raw and standardized actions. With training statistics $\mu_a, \sigma_a$, $a_i = (a_i^{\mathrm{raw}} - \mu_a) \oslash \max(\sigma_a, 0.01)$ coordinatewise. |
| $X_h, A_h$ | Rolling latent context and its aligned action window at prediction step $h$; $X_1 = (z_\tau, \dots, z_t)$ and $A_h = (a_{t+h-C}, \dots, a_{t+h-1})$. |
| $x_h$ | Last token frame in $X_h$; the reader evaluates its patch tokens while the base predictors consume the full context $X_h$. |
| $\delta_i \in \mathbb{R}^{2d}$ | Signed and absolute spatial means of the visual feature change for event $i$: $\delta_i = [N^{-1} \sum_n (z_{i+1,n} - z_{i,n}),\ N^{-1} \sum_n \lvert z_{i+1,n} - z_{i,n} \rvert]$. |
| $s_\delta \in \mathbb{R}^{2d}_+$, $s_a \in \mathbb{R}^{d_a}_+$ | Training-prefix RMS scales for response and standardized-action coordinates; $\oslash$ is coordinatewise division. |
| $W_\delta, W_a$ | Bias-free writer projections in $\mathbb{R}^{64 \times 2d}$ and $\mathbb{R}^{64 \times d_a}$. |
| $u_i, v_i, r_i, m_t$ | Projected response, projected action, event feature, and persistent memory; each of $u_i, v_i, r_i, m_t$ lies in $\mathbb{R}^{64}$. |
| $E_{\theta_{\mathrm{w}}}, \bar{u}, \bar{v}, \pi$ | Paired-event writer, coordinatewise means of projected responses and actions, and an event-index permutation. Complete pairs or actions alone are permuted as explicitly specified in each statement. |

Continued on the next page

| Symbol | Meaning (continued) |
|---|---|
| $W_1, W_2$ | Writer MLP matrices with widths $192 \to 128 \to 64$; the input is $[u_i, v_i, u_i \odot v_i] \in \mathbb{R}^{192}$. |
| $\mathrm{SiLU}, \sigma, \mathrm{softsign}$ | Writer activation $\mathrm{SiLU}(x) = x\,\sigma(x)$, logistic function $\sigma(x) = (1+e^{-x})^{-1}$, and coordinatewise bounded pooling map, $\mathrm{softsign}(x) = x/(1+\lvert x\rvert)$. |
| $F_\phi, F_\eta$ | Response-branch base predictor and independently continued companion predictor; both map a context and action window to $\mathbb{R}^{N\times d}$. |
| $\phi, \eta, \theta_{\mathrm{w}}, \theta_{\mathrm{r}}, \theta$ | Parameters of the response base, companion predictor, writer, and reader, respectively; $\theta$ is the generic reader notation in the main text and equals $\theta_{\mathrm{r}}$ in the appendix. The predictor branches have separate training histories. |
| $g_\theta(z,a,m), g_{\theta_{\mathrm{r}}}(z,a,m) \in \mathbb{R}^d$, $G_{\theta_{\mathrm{r}}}$ | Single-patch bounded residual and the matrix obtained by stacking it over all patches of the last frame. |
| $W_z, b_z, W'_a, W_m, W_o$ | Reader parameters with $W_z \in \mathbb{R}^{r\times d}$, $b_z \in \mathbb{R}^r$, $W'_a \in \mathbb{R}^{r\times d_a}$, $W_m \in \mathbb{R}^{r\times 64}$, and $W_o \in \mathbb{R}^{d\times r}$; $r = 32$. |
| $w_{o,j}, w_{z,j}^{\mathsf{T}}, w_{a,j}'^{\mathsf{T}}$ | Column $j$ of $W_o$ and rows $j$ of $W_z, W'_a$, used in the rank-one expansion of the visual and action maps. |
| $\beta, \beta_{\mathrm{train}}, \beta_{\mathrm{eval}}$ | Residual amplitude and its training/evaluation configurations. The final RoboMimic response stage uses $\beta_{\mathrm{train}} = 0.05$ and evaluation reconstructs $\beta_{\mathrm{eval}} = 0.1$. |
| $\lambda \in [0,1]$ | Shared-history mixture weight; the reported value is $\lambda = 0.4$. |
| $p_h, q_h, \hat{z}_{t+h}, \hat{o}_{t+h}$ | Response-branch token prediction, companion prediction, mixed token prediction, and decoded image prediction. |
| $p_h^{\mathrm{tr}}, q_h^{\mathrm{tr}}, y_h$ | Branch outputs on their own training histories and the generic branch output used in Algorithm 2. |
| $b \in \{p,q\}, X_h^b, x_h^b, \mathcal{L}_b$ | Branch indicator, branch-specific rolling context, its last token frame, and branch fitting objective. The superscript $b$ is a label, not a power. |
| $\mathcal{B}_h(X)$ | Zero-state mixture $\lambda F_\phi(X, A_h) + (1-\lambda)F_\eta(X, A_h)$ used in the finite-horizon intervention analysis. |
| $D_m = \mathrm{diag}(W_m m)$ | State-dependent diagonal reader gate. The symbols $R_z(m), R_a(m), b_m(m)$ denote the induced visual, action, and bias maps; $b_m$ is distinct from the RGB channel index $c$. |
| $b(z,a), s(z,a,m), D_s$ | Reader input $W_z z + b_z + W'_a a \in \mathbb{R}^r$, pre-activation $W_o D_m b(z,a) \in \mathbb{R}^d$, and its tanh derivative $D_s = \mathrm{diag}(1-\tanh^2(s)) \in \mathbb{R}^{d\times d}$. |
| $\kappa(z,a)$ | Fixed-input memory sensitivity coefficient $\beta\lVert W_o\rVert_2 \lVert b(z,a)\rVert_\infty \lVert W_m\rVert_2$. |
| $\gamma, \omega, \mathcal{L}$ | Image-loss weight, motion-region-loss weight, and the multi-step training objective. |
| $z_{t+h}^*$, $o_{t+h}^*$ (= $o_{t+h}$ in the main text) | Encoded target token frame and target RGB observation at future step $h$. |
| $M_h(u), \ell_{\mathrm{RGB}}, \ell_{\mathrm{mov}}$ | Dynamic-region mask, full-image squared error, and motion-region squared error used for training and evaluation. |
| $e_h = \hat{o}_{t+h} - o_{t+h}^*$ | RGB residual image used in Equation (10). |
| $e_h^{\mathrm{pix}}, e_h^{\mathrm{mot}}$ | Frame-level image errors; these are distinct from the latent intervention distances $\varepsilon_h$ and $\Delta_h$ below. |
| $\lVert\cdot\rVert_F, \lVert\cdot\rVert_2, \lVert\cdot\rVert_\infty$ | Frobenius, Euclidean (or spectral for matrices), and coordinatewise maximum norms. For a token frame, $\lVert z\rVert_{\mathrm{rms}} = \lVert z\rVert_F/\sqrt{Nd}$. |
| $\lVert X\rVert_{\mathrm{ctx}}$ | Context maximum-RMS norm, $\lVert X\rVert_{\mathrm{ctx}} = \max_{1\le k\le C} \lVert X_k\rVert_{\mathrm{rms}}$. |
| $\varepsilon_h$ | Latent frame-RMS distance between stateful and zero-state executions. |
| $\Delta_h$ | Latent frame-RMS distance between two executions that use different fixed memories under the same actions. |
| $L, J_z, \mathrm{rank}$ | Context-to-frame Lipschitz constant, local Jacobian with respect to a patch, and matrix rank. |
| $\oslash, \odot, [\cdot,\cdot], \mathrm{diag}, \mathrm{shift}$ | Coordinatewise division, coordinatewise product, concatenation, diagonal operator, and context update that drops the oldest frame and appends a token. |
| $\mathcal{D}_{\mathrm{tr}}, \mathcal{D}_{\mathrm{val}}, \mathcal{D}_{\mathrm{te}}$ | Episode-level training, validation, and frozen-test collections. |
| $\mathcal{U},\ u,\ c$ | Pixel-location set, pixel index, and RGB channel index. The writer's $u_i$ is a projected response vector and is distinct from pixel index $u$. |
| $\mu_a, \sigma_a, \mathcal{S}$ | Training-transition action mean, coordinatewise action scale, and fixed set of eligible training prefixes used to estimate $s_\delta$ and $s_a$. |
| $\mathbf{1}\{\cdot\}, \{R,G,B\}, j, k$ | Indicator, RGB channel set, output/rank index, and context-position index used in metric and proof definitions. |

Continued on the next page

| Symbol | Meaning (continued) |
|---|---|
| $\mathrm{PSNR}, \mathrm{SSIM}, \mathrm{H20}$ | Frame-level image metrics and their mean over the first $H = 20$ predictions. |
| $T, T_\pi, \mathbb{E}_\pi$ | Mean coordinatewise response–action product, its action-permuted version, and expectation over a uniform action permutation. |
| $\xi_i, f_{\mathrm{e}}, L_{\mathrm{e}}$ | Normalized event $[\delta_i \oslash s_\delta, a_i \oslash s_a]$, learned event map, and its Lipschitz constant on the compared event domain. |
| $q, B_{\mathrm{e}}$ | Number of changed events and maximum event-feature perturbation in Proposition 3; $q$ is distinct from branch output $q_h$. |
| $L_G, \kappa_G, L_* = L + \lambda L_G$ | Stacked-reader sensitivity to a token frame, sensitivity to memory, and the combined propagation coefficient in Corollary 7. |
| $L_D, \Vert \cdot \Vert_{\mathrm{im}}$ | Conditional decoder Lipschitz constant and image RMS norm, $\Vert o \Vert_F / \sqrt{3 H_{\mathrm{im}} W_{\mathrm{im}}}$. |
| $b_h$ | Geometric upper envelope $\lambda\beta \sum_{j=0}^{h-1} L^j$ used in the induction for Proposition 5. |
| $\Delta_{\mathrm{act}}$ | Source-specific transition-action offset; distinct from rollout distance $\Delta_h$. |
| $s \in \mathcal{S}, \delta_{s,i,j}, a_{s,i,j}$ | Training-prefix index and coordinate $j$ of event $i$ in that prefix, used to estimate writer normalization scales. |
| $\mathrm{H}\, h$ | Mean score over predictions $1, \ldots, h$; an endpoint score at $t + h$ is explicitly labeled as such. |
| $\widetilde{u}_i, \widetilde{v}_i, \Sigma_u, \Sigma_v$ | Centered projected event vectors and empirical covariance matrices with divisor $K$; $\mathrm{Cov}_\pi$ denotes covariance over action permutations. |
| $\mathcal{E}, \mathcal{W}, \sim_{\mathcal{W}}$ | Completed event collection, writer map, and equality of written memories. |
| $U, V, M_W$ | Local Euclidean bounds for projected response and action vectors, and $\max(\Vert W_\delta \Vert_2, \Vert W_a \Vert_2)$. |
| $\mathcal{T}_h, Q_h, \mathsf{A}_h, \mathsf{B}_h$ | Response-branch context update, its total state-to-context Jacobian, and its partial context and memory Jacobians. $\mathsf{A}_h$ is distinct from the action window $A_h$. |
| $P, v_h, \psi, \ell_h$ | Newest-frame selector, per-step loss gradient with respect to the predicted token frame, writer parameters ($\psi = \theta_{\mathrm{w}}$), and per-step fitting loss. |
| $e_p, e_q, \langle \cdot, \cdot \rangle_{\mathrm{rms}}$ | Same-context latent branch errors and the Frobenius inner product divided by $Nd$. These errors are distinct from the RGB residual $e_h$. |
| $a_{\mathrm{mix}}, b_{\mathrm{mix}}, c_{\mathrm{mix}}, d_{\mathrm{mix}}, R(\lambda)$ | Two branch quadratic risks, their uncentered error cross-moment, prediction disagreement energy, and the same-context mixture risk. |
| $\lambda_{\mathrm{oracle}}, \mathrm{clip}_{[0,1]}$ | Constrained same-context risk minimizer and projection onto $[0, 1]$. The oracle weight is not the empirically selected deployment weight. |
| $P_{\mathrm{write}}, P_{\mathrm{read}}$ | Numbers of trainable writer and reader parameters, excluding both predictors, the encoder, and the decoder. |
| $\rho_h, e_h^{\mathrm{stat}}, c_h^{\mathrm{dyn}}, c_h^{\mathrm{stat}}$ | Fraction of changing pixels, complement-region MSE, and the dynamic and complement contributions to full-image MSE. |
| $\overline{x}, \mathrm{Cov}_H$ | Uniform finite-horizon average and covariance with divisor $H$. |
| $\bar{e}_h, \widetilde{e}_h$ | Prefix-average error and frame pixel MSE clamped below at $10^{-12}$ for computing PSNR, respectively. |
| $\mathcal{J}, E_j, Q_{j,e}, s_{j,e,q,h}$ | Evaluated task set, eligible episode set for task $j$, query set for episode $e$, and a frame-level evaluation score. Query index $q$ is local to this aggregation formula, not the number of perturbed events. |
| $g_e^{\mathrm{pix}}, g_e^{\mathrm{mot}}$ | Episode-level control-minus-correct MSE differences; positive favors the correct state in the validation distribution analysis. |

The quantities $\varepsilon_h$ and $\Delta_h$ measure latent prediction distances under controlled state interventions, whereas $e_h$ denotes the RGB prediction residual. The response and companion branches are trained with separate histories and objectives. Algorithm 2 fits the branches; Algorithm 1 combines them during shared-history inference.

## C Method Details

This appendix expands the method in the order in which information flows through the model: completed action–response events are written once, the resulting state is read by a bounded residual, and the mixed prediction is then rolled forward through one shared history. The two training branches

**Algorithm 1** Persistent-memory rollout with a shared history

**Require:** Observed frames $o_{\tau:t}$, aligned past and future standardized actions, fitted $S, D, F_\phi, F_\eta, E_{\theta_\mathrm{w}}, g_{\theta_\mathrm{r}}$, mixture weight $\lambda$, evaluation amplitude $\beta_{\mathrm{eval}}$, event count $K$, horizon $H$
**Ensure:** Predicted frames $\{\hat{o}_{t+h}\}_{h=1}^{H}$

Encode $z_{\tau:t} \leftarrow S(o_{\tau:t})$ and set $X_1 \leftarrow z_{\tau:t}$.
**for** $i = \tau, \ldots, \tau + K - 1$ **do**
    Compute $\delta_i$ from $(z_i, z_{i+1})$ by Equation (1).
**end for**
$m_t \leftarrow E_{\theta_\mathrm{w}}(\{(\delta_i, a_i)\}_{i \in \mathcal{I}_t})$ using the fixed scales $s_\delta, s_a$.
**for** $h = 1, \ldots, H$ **do**
    $x_h \leftarrow \mathrm{last}(X_h)$ and $A_h \leftarrow (a_{t+h-C}, \ldots, a_{t+h-1})$.
    $p_h \leftarrow F_\phi(X_h, A_h) + G_{\theta_\mathrm{r}}(x_h, a_{t+h-1}, m_t; \beta_{\mathrm{eval}})$.
    $q_h \leftarrow F_\eta(X_h, A_h)$.
    $\hat{z}_{t+h} \leftarrow \lambda p_h + (1 - \lambda) q_h$.
    $\hat{o}_{t+h} \leftarrow D(\hat{z}_{t+h})$.
    $X_{h+1} \leftarrow \mathrm{shift}(X_h, \hat{z}_{t+h})$.
**end for**
**return** $\{\hat{o}_{t+h}\}_{h=1}^{H}$.

**Algorithm 2** Independent autoregressive fitting with full rollout gradients

**Require:** Training queries and targets from $\mathcal{D}_{\mathrm{tr}}$, context length $C$, event count $K$, frozen $S, D$, warm-started $F_\phi, F_\eta$, writer and reader, ordered branch-specific stages $(H_{\mathrm{tr}}, \gamma, \omega, \beta_{\mathrm{train}})$
**Ensure:** Fitted response and companion branches

**for** each branch $b \in \{p, q\}$ independently **do**
    **for** each fitting stage of branch $b$ **do**
        **if** $b = p$ **then**
            Estimate $s_\delta, s_a$ from training prefixes by Equation (45).
        **end if**
        **for** each minibatch in the stage **do**
            Encode the observations and targets; set $X_1^b \leftarrow z_{\tau:t}$ and $\mathcal{L}_b \leftarrow 0$.
            **if** $b = p$ **then**
                Write $m_t$ from the $K$ completed observed events.
            **end if**
            **for** $h = 1, \ldots, H_{\mathrm{tr}}$ **do**
                $x_h^b \leftarrow \mathrm{last}(X_h^b)$ and $A_h \leftarrow (a_{t+h-C}, \ldots, a_{t+h-1})$.
                **if** $b = p$ **then**
                    $y_h \leftarrow F_\phi(X_h^p, A_h) + G_{\theta_\mathrm{r}}(x_h^p, a_{t+h-1}, m_t; \beta_{\mathrm{train}})$.
                **else**
                    $y_h \leftarrow F_\eta(X_h^q, A_h)$.
                **end if**
                Add the per-step latent, RGB, and motion-region loss summand in Equation (7) for $y_h$ to $\mathcal{L}_b$.
                $X_{h+1}^b \leftarrow \mathrm{shift}(X_h^b, y_h)$.
            **end for**
            Average $\mathcal{L}_b$ over predicted steps and minibatch queries; differentiate through all predicted contexts and, for $b = p$, the reused memory.
            Update only the parameters trainable in this branch and stage.
        **end for**
    **end for**
**end for**
**return** the two fitted branches and the writer/reader parameters.

are fitted independently; their shared-history mixture is an inference operation. All actions in the derivations below are standardized model coordinates as defined in Appendix B.

### C.1 A PERSISTENT WRITE–READ–ROLLOUT INTERFACE

Let $\tau = t - C + 1$ be the first frame in an observed context and let $\mathcal{I}_t = \{\tau, \dots, \tau + K - 1\}$ index completed transitions. A query contains the frames $z_{\tau:t}$ and the aligned actions needed for the first $K$ events. The writer computes one state $m_t = E_{\theta_w}(\{(\delta_i, a_i)\}_{i \in \mathcal{I}_t})$. The state is not recomputed from predicted frames. At rollout step $h$, the reader evaluates the current last-frame patches and current action, while the base predictors read the evolving context $X_h$:

$$\hat{z}_{t+h} = \lambda[F_\phi(X_h, A_h) + G_{\theta_r}(x_h, a_{t+h-1}, m_t)] + (1 - \lambda)F_\eta(X_h, A_h). \tag{11}$$

Only $\hat{z}_{t+h}$ is appended to the history. This makes the causal separation explicit: $m_t$ carries observed response evidence, whereas $X_h$ carries the predictions generated after the query begins.

The state is computed before the first prediction and remains fixed throughout the query. The inference map is specified by its fitted weights and the amplitude $\beta_{\text{eval}}$; source-specific training amplitudes are reported separately in Appendix D.4.

The writer state is reused without detaching it, and the predicted context is also retained in the computation graph during training. Consequently, the multi-step objective assigns credit through the response state, the reader, and the autoregressive context. The response trainable set comprises the writer and reader, plus $\phi$ during RoboMimic joint continuation; the companion trainable set is $\eta$. Frozen encoder and decoder parameters stay fixed, while derivatives through decoder inputs remain active. At inference, the fitted branches are evaluated on the same $X_h$ before the mixed token advances that history.

### C.2 PAIRED EVENT WRITING

For each completed event, the response statistic is

$$\delta_i = \left[ \frac{1}{N} \sum_{n=1}^{N} (z_{i+1,n} - z_{i,n}),\ \frac{1}{N} \sum_{n=1}^{N} |z_{i+1,n} - z_{i,n}| \right] \in \mathbb{R}^{2d}. \tag{12}$$

The two halves retain signed direction and changes that cancel under signed spatial averaging. With bias-free projections $u_i = W_\delta(\delta_i \oslash s_\delta)$ and $v_i = W_a(a_i \oslash s_a)$, the event network is

$$r_i = W_2 \,\mathrm{SiLU}(W_1[u_i, v_i, u_i \odot v_i]), \qquad m_t = \mathrm{softsign}\left( \frac{1}{K} \sum_{i \in \mathcal{I}_t} r_i \right). \tag{13}$$

The product channel supplies an explicit second-order pairing statistic to the nonlinear event map before pooling.

Denote the writer by $E_{\theta_w}(\{(\delta_i, a_i)\}_{i=1}^K)$ and let $\bar{u} = K^{-1} \sum_i u_i$ and $\bar{v} = K^{-1} \sum_i v_i$. The product average decomposes as

$$\frac{1}{K} \sum_i u_i \odot v_i = \bar{u} \odot \bar{v} + \frac{1}{K} \sum_i (u_i - \bar{u}) \odot (v_i - \bar{v}). \tag{14}$$

The second term is a centered cross-moment and is absent from the two marginal means. The identity follows by expanding the centered products and using $\sum_i (u_i - \bar{u}) = \sum_i (v_i - \bar{v}) = 0$.

**Proposition 2** (Pairing information and independent permutation)**.** *Let $T = K^{-1} \sum_i u_i \odot v_i$ and let $\pi$ be uniform over the permutations of $K$ events. If only the actions are permuted, then*

$$\mathbb{E}_\pi \left[ \frac{1}{K} \sum_i u_i \odot v_{\pi(i)} \right] = \bar{u} \odot \bar{v}, \qquad T - \mathbb{E}_\pi[T_\pi] = \frac{1}{K} \sum_i (u_i - \bar{u}) \odot (v_i - \bar{v}). \tag{15}$$

*Consequently, the two marginal means do not determine $T$.*

*Proof.* For each $i$, $v_{\pi(i)}$ is uniform over $v_1, \ldots, v_K$, so $\mathbb{E}_\pi[v_{\pi(i)}] = \bar{v}$. Linearity of expectation proves the first identity, and Equation (14) proves the second. For a scalar example, the paired events $(1, 1), (-1, -1)$ have zero marginal means and $T = 1$; exchanging their actions preserves both means and gives $T = -1$. Thus a summary of the marginal means alone cannot recover $T$. □

Proposition 2 characterizes the writer's input cross-moment under within-query action permutation. Appendix F evaluates a separate intervention: replacing the complete state with that of another same-task episode.

**Writer invariances.** For a permutation $\pi$ of complete event pairs, the writer satisfies

$$\begin{aligned} E_{\theta_w}(\{(\delta_{\pi(i)}, a_{\pi(i)})\}_{i=1}^K) &= E_{\theta_w}(\{(\delta_i, a_i)\}_{i=1}^K), \\ E_{\theta_w}(\{(0,0)\}_{i=1}^K) &= 0, \qquad \|m_t\|_\infty < 1. \end{aligned} \tag{16}$$

Every writer projection is bias-free and $\mathrm{SiLU}(0) = 0$, so zero model-space inputs give $u_i = v_i = r_i = 0$ before the softsign. Complete pair permutations only permute summands before their mean, preserving the action–response association within each event. The strict coordinate bound is the range of softsign for finite inputs.

**Proposition 3** (Stability of pooled event writing)**.** *Let $\xi_i = [\delta_i \oslash s_\delta, a_i \oslash s_a]$ denote a normalized event and let $f_e(\xi_i) = r_i$ be its learned event map. Assume $f_e$ is $L_e$-Lipschitz on the compared events. For two aligned collections of $K$ events and their memories $m, m'$,*

$$\|m - m'\|_2 \le \frac{1}{K} \sum_{i=1}^K \|r_i - r_i'\|_2 \le \frac{L_e}{K} \sum_{i=1}^K \|\xi_i - \xi_i'\|_2. \tag{17}$$

*Proof.* For a scalar $v$, the derivative of $v/(1 + |v|)$ is $(1 + |v|)^{-2}$, including its continuous value at zero. Coordinatewise softsign is therefore 1-Lipschitz in Euclidean norm. Apply this fact to the two event means, use the triangle inequality, and then apply the assumed event-map Lipschitz bound to each summand. □

The Lipschitz assumption applies on the bounded event domain. If $q$ events change and each event-feature change has norm at most $B_e$, the first bound gives $\|m - m'\|_2 \le qB_e/K$, quantifying the dilution of bounded event perturbations by mean pooling.

### C.3 Bounded low-rank reading

For one patch $z$ and the current standardized action $a$, define

$$g_{\theta_r}(z, a, m) = \beta \tanh\big(W_o\big[(W_z z + b_z + W_a' a) \odot W_m m\big]\big). \tag{18}$$

Let $D_m = \mathrm{diag}(W_m m)$ and define

$$\begin{aligned} s(z, a, m) &= R_z(m) z + R_a(m) a + b_m(m), \\ R_z(m) &= W_o D_m W_z, \qquad R_a(m) = W_o D_m W_a', \qquad b_m(m) = W_o D_m b_z. \end{aligned} \tag{19}$$

The state therefore reweights the same $r$ output directions for the current visual feature, action, and visual bias. Writing $w_{o,j}$ for column $j$ of $W_o$, and $w_{z,j}^\mathsf{T}$ and $w_{a,j}'^\mathsf{T}$ for rows of $W_z$ and $W_a'$, gives

$$\begin{aligned} R_z(m) &= \sum_{j=1}^r (W_m m)_j \, w_{o,j} w_{z,j}^\mathsf{T}, \\ R_a(m) &= \sum_{j=1}^r (W_m m)_j \, w_{o,j} w_{a,j}'^\mathsf{T}. \end{aligned} \tag{20}$$

This is a state-selected family of rank-one operators with a shared output basis, rather than a separate full-rank transition for each memory.

*Proof of Proposition 1.* Set $m = 0$. Then $D_m = 0$ and the pre-activation of (18) is zero, so the zero-state identity follows. For every output coordinate, $|\tanh(\cdot)| \le 1$, which gives $\|g_{\theta_r}(z, a, m)\|_\infty \le \beta$. For the Jacobian, let $b(z, a) = W_z z + b_z + W'_a a$, let $s = W_o D_m b(z, a)$, and let $D_s = \mathrm{diag}(1 - \tanh^2(s))$. The chain rule yields

$$J_z g_{\theta_r} = \beta D_s W_o \,\mathrm{diag}(W_m m) W_z, \qquad \mathrm{rank}(J_z g_{\theta_r}) \le r. \tag{21}$$

The factorization passes through $r$ channels, so its rank cannot exceed $r$. The rank claim is local to one patch; $D_s$ can vary with the patch and action. □

For fixed $z, a$, the state sensitivity follows from the 1-Lipschitz property of coordinatewise $\tanh$:

$$\begin{aligned} \|g_{\theta_r}(z, a, m) - g_{\theta_r}(z, a, m')\|_2 &\le \kappa(z, a)\|m - m'\|_2, \\ \kappa(z, a) &= \beta\|W_o\|_2 \|b(z, a)\|_\infty \|W_m\|_2. \end{aligned} \tag{22}$$

Indeed, the difference is bounded by $\beta\|W_o \,\mathrm{diag}(b(z, a)) W_m (m - m')\|_2$. Submultiplicativity and $\|\,\mathrm{diag}(b)\|_2 = \|b\|_\infty$ give (22); matrix $\|\cdot\|_2$ denotes the spectral norm when its argument is a matrix.

**Proposition 4** (Same-input memory gap)**.** *For any two finite memories $m, m'$ and the same patch/action pair $(z, a)$,*

$$\|g_{\theta_r}(z, a, m) - g_{\theta_r}(z, a, m')\|_\infty \le 2\beta. \tag{23}$$

*For the stacked residuals of one frame, $\|G_{\theta_r}(x, a, m) - G_{\theta_r}(x, a, m')\|_{\mathrm{rms}} \le 2\beta$.*

*Proof.* Each coordinate of either residual lies in $[-\beta, \beta]$ by the bounded output in Proposition 1. Their coordinatewise difference is therefore at most $2\beta$ in absolute value. Averaging the squared coordinate differences over $Nd$ entries gives the frame-RMS bound. □

### C.4 Shared-history interpolation

Let $p$ and $q$ be branch predictions evaluated on the same context and let $z^*$ be a target. For $0 \le \lambda \le 1$, expanding the squared norms gives

$$\begin{aligned} \|\lambda p + (1 - \lambda)q - z^*\|_F^2 = \lambda\|p - z^*\|_F^2 + (1 - \lambda)\|q - z^*\|_F^2 \\ - \lambda(1 - \lambda)\|p - q\|_F^2. \end{aligned} \tag{24}$$

The disagreement term is nonnegative, so on this common input the interpolation error is no larger than the weighted average of the two branch errors. The common-history premise holds for the two branch predictions evaluated at each step of Algorithm 1.

### C.5 Finite-horizon influence of persistent memory

For fixed future actions, define the zero-state base mixture

$$\mathcal{B}_h(X) = \lambda F_\phi(X, A_h) + (1 - \lambda) F_\eta(X, A_h). \tag{25}$$

The same fitted parameters are used in the stateful and zero-state executions. Assume that $\mathcal{B}_h$ is $L$-Lipschitz from $\|\cdot\|_{\mathrm{ctx}}$ to frame RMS on the compared contexts, with $L \ge 0$. Let $\hat{z}^m_{t+h}$ and $\hat{z}^0_{t+h}$ be the executions with memory $m_t$ and with zero memory, respectively, and define $\varepsilon_h = \|\hat{z}^m_{t+h} - \hat{z}^0_{t+h}\|_{\mathrm{rms}}$.

**Proposition 5** (Finite-horizon influence of a persistent state)**.** *Under identical initial observations, future actions, and fitted weights, with $0 \le \lambda \le 1$ and $\beta \ge 0$, the intervention distance satisfies*

$$\varepsilon_h \le \lambda\beta \sum_{j=0}^{h-1} L^j, \qquad 1 \le h \le H. \tag{26}$$

*Proof.* At step $h$, the zero-state execution is $\mathcal{B}_h(X^0_h)$ and the stateful execution is $\mathcal{B}_h(X^m_h) + \lambda G_h(m_t)$. Proposition 1 implies $\|G_h(m_t)\|_{\mathrm{rms}} \le \beta$. The Lipschitz assumption and the triangle inequality therefore give

$$\varepsilon_h \le L\|X^m_h - X^0_h\|_{\mathrm{ctx}} + \lambda\beta \le L \max_{1 \le k \le C} \varepsilon_{h-k} + \lambda\beta,$$

where $\varepsilon_j = 0$ for $j \le 0$. Let $b_h = \lambda\beta \sum_{j=0}^{h-1} L^j$ and $b_j = 0$ for $j \le 0$. Inductively, $\varepsilon_j \le b_j$ for $j < h$. Since $b_h$ is nondecreasing for $L \ge 0$, the context maximum is at most $b_{h-1}$, and

$$\varepsilon_h \le L b_{h-1} + \lambda\beta = b_h.$$

This proves (26). When $L < 1$ the geometric bound is at most $\lambda\beta/(1-L)$; when $L = 1$ it is $h\lambda\beta$. □

**Corollary 6** (Two-memory rollout separation)**.** *Let $\hat{z}^m$ and $\hat{z}^{m'}$ use two fixed memories $m, m'$ with identical initial observations, future actions, and fitted weights. Suppose $0 \le \lambda \le 1$, $\beta \ge 0$, and $\mathcal{B}_h$ is $L$-Lipschitz between the two compared contexts at each $1 \le h \le H$. Then*

$$\Delta_h := \|\hat{z}^m_{t+h} - \hat{z}^{m'}_{t+h}\|_{\mathrm{rms}} \le 2\lambda\beta \sum_{j=0}^{h-1} L^j. \tag{27}$$

*Proof.* Use the same base decomposition (25) for both executions. Regardless of their current patch and action inputs, each residual stack has all coordinates in $[-\beta, \beta]$ by Proposition 1; hence their frame-RMS difference is at most $2\beta$. Thus $\Delta_h \le L \max_{1\le k\le C} \Delta_{h-k} + 2\lambda\beta$ with $\Delta_j = 0$ for $j \le 0$. Induction with $2\lambda\beta \sum_{j=0}^{h-1} L^j$ proves the claim. □

### C.6 From observed-event perturbations to rollout sensitivity

The amplitude bound applies uniformly to arbitrary states. A complementary local bound captures the distance between their memory representations. Assume, on the compared trajectories and states, that

$$\begin{aligned} \|G(x,a,m) - G(x',a,m)\|_{\mathrm{rms}} &\le L_G \|x - x'\|_{\mathrm{rms}}, \\ \|G(x,a,m) - G(x,a,m')\|_{\mathrm{rms}} &\le \kappa_G \|m - m'\|_2. \end{aligned} \tag{28}$$

For example, one may take a uniform upper bound on $\beta\|W_o\|_2\|\operatorname{diag}(W_m m)\|_2\|W_z\|_2$ for $L_G$, and on $[\sum_n \kappa(x_n,a)^2/(Nd)]^{1/2}$ for $\kappa_G$. These constants follow from Equation (21) and Equation (22), respectively.

**Corollary 7** (Memory-sensitive finite-horizon bound)**.** *Under the premises of Corollary 6 and Equation* (28)*, put $L_* = L + \lambda L_G$. Then*

$$\Delta_h \le \min\left\{ 2\lambda\beta \sum_{j=0}^{h-1} L^j,\ \lambda\kappa_G \|m - m'\|_2 \sum_{j=0}^{h-1} L_*^j \right\}. \tag{29}$$

*If the memories are written from the two collections in Proposition 3, $\|m - m'\|_2$ in the second term can be replaced by $L_{\mathrm{e}} K^{-1} \sum_i \|\xi_i - \xi_i'\|_2$.*

*Proof.* Add and subtract $G(x_h^{m'}, a, m)$ when comparing the two residuals. Equation (28) and the base Lipschitz bound give

$$\Delta_h \le (L + \lambda L_G) \max_{1\le k\le C} \Delta_{h-k} + \lambda\kappa_G \|m - m'\|_2.$$

The geometric-sum induction from Proposition 5 proves the second bound. The first is Corollary 6; both hold simultaneously. Substitute Equation (17) for the last assertion. □

This composition isolates the memory channel by perturbing the observed events while holding the initial rollout context fixed.

**Decoded sensitivity.** If the fixed decoder is $L_D$-Lipschitz from latent frame RMS to image RMS on the compared predictions, then $\|D(\hat{z}^m_{t+h}) - D(\hat{z}^{m'}_{t+h})\|_{\mathrm{im}} \le L_D \Delta_h$, where $\|o\|_{\mathrm{im}} = \|o\|_F / \sqrt{3 H_{\mathrm{im}} W_{\mathrm{im}}}$. Lipschitz continuity transfers the latent intervention bound to image space with factor $L_D$. Together, the writer, reader, rollout, and decoder bounds characterize the propagation of an observed-event perturbation through the prediction pipeline under the stated sensitivity assumptions.

### C.7 FINITE-SAMPLE VARIABILITY OF INDEPENDENTLY REASSIGNED ACTIONS

Proposition 2 separates a paired cross-moment from its expectation under independent action reassignment. The exact finite-sample covariance characterizes its variability over permutations of the fixed observed events.

Let $\widetilde{u}_i = u_i - \bar{u}$ and $\widetilde{v}_i = v_i - \bar{v}$, and define

$$\Sigma_u = K^{-1} \sum_i \widetilde{u}_i \widetilde{u}_i^\top, \qquad \Sigma_v = K^{-1} \sum_i \widetilde{v}_i \widetilde{v}_i^\top. \tag{30}$$

These empirical second moments use divisor $K$. Let $T_\pi = K^{-1} \sum_i u_i \odot v_{\pi(i)}$, where $\pi$ is uniform over all permutations of $K$ indices.

**Proposition 8** (Conditional permutation covariance)**.** *For $K > 1$,*

$$\mathrm{Cov}_\pi(T_\pi) = \frac{\Sigma_u \odot \Sigma_v}{K-1}, \qquad \mathbb{E}_\pi \|T_\pi - \bar{u} \odot \bar{v}\|_2^2 = \frac{\mathrm{tr}(\Sigma_u \odot \Sigma_v)}{K-1}. \tag{31}$$

*Proof.* Centering gives $T_\pi - \bar{u} \odot \bar{v} = K^{-1} \sum_i \widetilde{u}_i \odot \widetilde{v}_{\pi(i)}$. For coordinates $j, k$, uniform sampling without replacement gives

$$\mathbb{E}_\pi[\widetilde{v}_{\pi(i),j} \widetilde{v}_{\pi(l),k}] = \begin{cases} (\Sigma_v)_{jk}, & i = l, \\ -(\Sigma_v)_{jk}/(K-1), & i \neq l. \end{cases}$$

The second equality follows because $\sum_i \widetilde{v}_i = 0$. Similarly, $\sum_i \widetilde{u}_{i,j} \widetilde{u}_{i,k} = K(\Sigma_u)_{jk}$ and $\sum_{i \neq l} \widetilde{u}_{i,j} \widetilde{u}_{l,k} = -K(\Sigma_u)_{jk}$. Substituting both sums into the covariance expansion gives

$$K^{-2} \left[ K(\Sigma_u)_{jk}(\Sigma_v)_{jk} + \frac{K(\Sigma_u)_{jk}(\Sigma_v)_{jk}}{K-1} \right] = \frac{(\Sigma_u)_{jk}(\Sigma_v)_{jk}}{K-1}.$$

Taking the trace proves the mean-squared fluctuation identity. □

For $K = 1$, reassignment is the identity and its covariance is zero. For $K > 1$, each diagonal entry quantifies the fluctuation of an input pairing coordinate; constant response or action coordinates have zero variance. At the evaluated $K = 4$, the denominator is three. This covariance describes the projected product features supplied to the nonlinear event map.

### C.8 SUMMARY EQUIVALENCE AND THE INFORMATION RETAINED BY MEMORY

Let $\mathcal{E}$ denote a collection of completed response–action events and $\mathcal{W}(\mathcal{E})$ the writer in Equation 2. Define $\mathcal{E} \sim_{\mathcal{W}} \mathcal{E}'$ when $\mathcal{W}(\mathcal{E}) = \mathcal{W}(\mathcal{E}')$. This relation makes explicit the event collections represented by the same conditioning state.

**Proposition 9** (Indistinguishability through the memory channel)**.** *Fix model weights, the mixture weight, initial visual context, and the future action sequence. If $\mathcal{E} \sim_{\mathcal{W}} \mathcal{E}'$, the two deterministic executions produce identical latent predictions and identical decoded frames at every rollout step.*

*Proof.* The first-step predictor inputs agree, including memory. Hence both branch outputs and their mixture agree. Shifting in the same predicted token frame preserves equality of the contexts. Induction proves equality at every step, and applying the same decoder preserves it. □

The result isolates memory equivalence at fixed visual context and actions. In the complete predictor, the memory and rolling context jointly determine each transition.

Two equivalences are built into the writer. First, permuting complete events preserves their mean-pooled representation. Second, for each event, applying the same permutation to the patch indices of its token differences preserves both the signed and absolute averages in Equation 1. These permutations act on already-computed token differences. The writer retains within-event action–response pairing and aggregates across event order and patch arrangement. Ordered spatial token maps remain available in the seven-frame prediction context.

### C.9 AN EXPLICIT LOCAL BOUND FOR THE EVENT MAP

The event-stability result can be related directly to the writer weights. Use normalized event coordinates $\xi = [\delta \oslash s_\delta, a \oslash s_a]$, and consider a domain on which projected vectors satisfy $\|u\|_2 \le U$ and $\|v\|_2 \le V$. Write $M_W = \max(\|W_\delta\|_2, \|W_a\|_2)$. One admissible constant is

$$L_{\mathrm{e}} = \|W_2\|_2 \|W_1\|_2 (1 + e^{-1}) \sqrt{1 + U^2 + V^2}\, M_W. \tag{32}$$

The constant applies on the specified bounded projected-input domain.

*Proof.* For two pairs,

$$\|u \odot v - u' \odot v'\|_2 \le V\|u - u'\|_2 + U\|v - v'\|_2 \le \sqrt{U^2 + V^2}\sqrt{\|u - u'\|_2^2 + \|v - v'\|_2^2}.$$

Combining this with the two direct paths bounds the Lipschitz constant of $[u, v, u \odot v]$ by $\sqrt{1 + U^2 + V^2}$ in the concatenated pair norm. For the scalar SiLU function, $\mathrm{SiLU}'(x) = \sigma(x) + x\sigma'(x)$. Since $0 < \sigma(x) < 1$, $\sigma'(x) \le e^{-|x|}$, and $|x|e^{-|x|} \le e^{-1}$, its absolute derivative is at most $1 + e^{-1}$. Coordinatewise application has the same Euclidean Lipschitz bound. Composition with the three linear maps proves the claim. □

The RMS scales are fixed positive quantities during a fitting stage. If perturbations are instead expressed in unnormalized inputs, the bound also contains the spectral norm of the diagonal inverse-scale operator. Thus the coordinatewise RMS scales determine how raw event perturbations enter the worst-case sensitivity bound.

### C.10 GRADIENT PATHS THROUGH A REUSED RESPONSE STATE

Memory is written once, but receives learning signals from multiple future steps. Consider response-branch fitting, flatten the context into a vector, and write its update as $X_{h+1}^p = \mathcal{T}_h(X_h^p, m)$. The action sequence and observed initial context are fixed. At differentiable points, define

$$Q_h = \frac{\partial X_h^p}{\partial m}, \qquad \mathsf{A}_h = \frac{\partial \mathcal{T}_h}{\partial X}, \qquad \mathsf{B}_h = \frac{\partial \mathcal{T}_h}{\partial m}. \tag{33}$$

Here $\mathsf{A}_h$ is a Jacobian, distinct from the action window $A_h$. Since the observed context does not depend on $m$, $Q_1 = 0$.

**Proposition 10** (Repeated-state gradient decomposition)**.** *The state-to-context Jacobian satisfies*

$$Q_{h+1} = \mathsf{A}_h Q_h + \mathsf{B}_h = \sum_{j=1}^{h} (\mathsf{A}_h \mathsf{A}_{h-1} \cdots \mathsf{A}_{j+1}) \mathsf{B}_j, \tag{34}$$

*where an empty product is the identity. Let $P$ select the newest token frame from a context, and let $v_h = \nabla_{p_h^{\mathrm{tr}}} \ell_h$. For $\mathcal{L} = H_{\mathrm{tr}}^{-1} \sum_h \ell_h$,*

$$\nabla_m \mathcal{L} = \frac{1}{H_{\mathrm{tr}}} \sum_{h=1}^{H_{\mathrm{tr}}} Q_{h+1}^\top P^\top v_h. \tag{35}$$

*Proof.* Differentiate the context update by the chain rule to obtain the recursion. Substitution proves its product-sum expansion by induction. The predicted frame is $p_h^{\mathrm{tr}} = P X_{h+1}^p$, so differentiating each loss term gives $Q_{h+1}^\top P^\top v_h$; summing gives the result. □

For writer parameters $\psi$, the memory-mediated contribution is $(\partial m / \partial \psi)^\top \nabla_m \mathcal{L}$. The term $\mathsf{B}_h$ is the direct influence of rereading memory at step $h$, while the products of $\mathsf{A}$ propagate effects from earlier reads through the evolving context. This separates persistence of the supplied state from accumulation of its consequences.

Freezing predictor or decoder parameters does not remove their derivatives with respect to inputs. In response-only fitting, later losses can therefore train earlier corrections through the fixed base

predictor, and RGB losses can train latent outputs through the fixed decoder. By contrast, at the exact initialization $W_o = 0$, the derivative of the reader with respect to $m$ is zero. The output matrix can still receive a gradient when its gated input and the loss derivative are nonzero; after it changes, memory-mediated gradients become available through the updated output map. This separates zero-residual initialization from subsequent memory-mediated learning. For the static-state control, a nonzero initial state avoids setting both sides of the multiplicative adaptation path to zero.

### C.11 When a same-context mixture reduces quadratic risk

The interpolation identity can be extended to expected latent risk without assuming independent branch errors. Consider two predictions evaluated on the same fixed context and actions, with errors $e_p = p - y$ and $e_q = q - y$. Use the normalized inner product $\langle x, y\rangle_{\mathrm{rms}} = \langle x, y\rangle_F/(Nd)$ and assume finite second moments. Define

$$a_{\mathrm{mix}} = \mathbb{E}\|e_p\|^2_{\mathrm{rms}}, \quad b_{\mathrm{mix}} = \mathbb{E}\|e_q\|^2_{\mathrm{rms}}, \quad c_{\mathrm{mix}} = \mathbb{E}\langle e_p, e_q\rangle_{\mathrm{rms}},$$

and $d_{\mathrm{mix}} = a_{\mathrm{mix}} + b_{\mathrm{mix}} - 2c_{\mathrm{mix}} = \mathbb{E}\|p - q\|^2_{\mathrm{rms}} \geq 0$.

**Proposition 11** (Constrained same-context oracle)**.** *If $d_{\mathrm{mix}} > 0$, the minimizing weight on $[0, 1]$ is*

$$\lambda_{\mathrm{oracle}} = \mathrm{clip}_{[0,1]}\left(\frac{b_{\mathrm{mix}} - c_{\mathrm{mix}}}{d_{\mathrm{mix}}}\right). \tag{36}$$

*The optimum is interior if and only if $c_{\mathrm{mix}} < \min(a_{\mathrm{mix}}, b_{\mathrm{mix}})$; in that case its expected squared error is strictly smaller than both endpoint risks. If $d_{\mathrm{mix}} = 0$, the predictions agree almost surely and all weights have the same risk.*

*Proof.* Expanding the square gives

$$R(\lambda) = b_{\mathrm{mix}} + 2\lambda(c_{\mathrm{mix}} - b_{\mathrm{mix}}) + \lambda^2 d_{\mathrm{mix}}.$$

For positive $d_{\mathrm{mix}}$ this is strictly convex. Its stationary point is $(b_{\mathrm{mix}} - c_{\mathrm{mix}})/d_{\mathrm{mix}}$, and projection onto the interval gives the constrained minimizer. Positivity requires $c_{\mathrm{mix}} < b_{\mathrm{mix}}$ and being smaller than one requires $c_{\mathrm{mix}} < a_{\mathrm{mix}}$. Strict convexity then yields strict improvement over both endpoints. If $d_{\mathrm{mix}} = 0$, the nonnegative random variable $\|p - q\|^2_{\mathrm{rms}}$ is zero almost surely. □

The uncentered cross-moment $c_{\mathrm{mix}}$ captures dependence between branch errors, so the criterion accounts for both individual accuracy and error complementarity. It accommodates biased and dependent branches. The oracle holds at fixed context; during shared-history rollout, $\lambda$ also changes subsequent contexts and branch predictions. Appendix F.7 therefore selects the deployment weight using the complete recursive predictor and nonlinear RGB decoder on validation trajectories.

### C.12 Parameter and computation accounting for the interface

Let the memory dimension be $d_m$, the writer hidden width $d_w$, and the reader width $r$. All writer maps are bias-free. Counting the response projection, action projection, and two-layer event map gives

$$P_{\mathrm{write}} = d_m(2d + d_a) + 4d_m d_w. \tag{37}$$

The reader contains the visual projection and its bias, the action and state projections, and the output projection:

$$P_{\mathrm{read}} = r(2d + d_a + d_m + 1). \tag{38}$$

RMS scales are fitted buffers, not trainable parameters. RoboMimic and LIBERO use $d_a = 7$, while LanguageTable uses $d_a = 2$. With $(d, d_m, d_w, r) = (384, 64, 128, 32)$, the interface counts are:

| Action dimension | Writer | Reader | Interface total |
|---|---|---|---|
| $d_a = 7$ | 82,368 | 26,880 | 109,248 |
| $d_a = 2$ | 82,048 | 26,720 | 108,768 |

Table 6: Dataset subsets and temporal alignment. Counts are whole episodes in train/validation/test; $d_a$ is the action dimension and $\Delta_{\mathrm{act}}$ is the transition-action offset.

| Source | Episodes | FPS | $d_a$ | $\Delta_{\mathrm{act}}$ | Test used |
|---|---|---|---|---|---|
| RoboMimic Lift | 120/40/40 | 20 | 7 | 0 | 40 |
| LIBERO New | 120/40/40 | 20 | 7 | 1 | 40 |
| LanguageTable B2B | 144/48/48 | 10 | 2 | 0 | 36 |

These counts describe the writer–reader interface. The complete system additionally contains the encoder, decoder, two latent predictors, and training optimizer state.

For already-encoded support frames, constructing pooled signed and absolute differences costs $O(KNd)$. Event processing costs $O(K[d_m(2d+d_a)+d_m d_w])$, and pooling stores a $d_m$-dimensional state. At a rollout step, the patchwise visual and output projections cost $O(Ndr)$; the action and state projections can be shared across patches. The written state has constant size with respect to rollout length, alongside the $C \times N \times d$ rolling visual context. Training through an unrolled sequence additionally stores activations unless recomputation is used. These operation and storage counts characterize the architecture; source-specific training budgets are reported in Appendix D.4.

# D Experimental details

## D.1 Datasets, observations, and action alignment

We study open-loop, action-conditioned visual prediction on episode-held-out subsets of RoboMimic Lift, four LIBERO tasks, and LanguageTable block-to-block simulation (Mandlekar et al., 2022; Liu et al., 2023; Lynch et al., 2022). Each query contains seven observed $64 \times 64$ RGB frames and twenty future targets. Actions are aligned to the transitions they cause, with source offset $\Delta_{\mathrm{act}}$ in Table 6: the stored action at index $i + \Delta_{\mathrm{act}}$ is assigned to the transition $o_i \to o_{i+1}$ before standardization. Images and aligned actions are the only inputs: language, proprioception, rewards, and simulator state are excluded. Future ground-truth frames provide supervision and scores, never autoregressive inputs at evaluation.

RoboMimic and LIBERO use seven operational-space pose/gripper coordinates; LanguageTable uses two Cartesian-setpoint deltas. H20 corresponds to source-specific physical durations of 1.0, 1.0, and 2.0 seconds, respectively. All splits are fixed at the episode level before model fitting or score inspection. LanguageTable retains all complete episodes in the released subset; its H20-eligible train/validation/test counts are 111/38/36. The 55 shorter episodes remain in their original splits and are not reassigned. Table 6 reports the sizes of the evaluated subsets. The four LIBERO tasks are opening the top cabinet drawer (Kitchen Scene 2), placing chocolate pudding to the right of the plate (Living Room Scene 6), putting the bowl on the plate (LIBERO Goal), and placing the right-hand book on the cabinet shelf (Study Scene 4). Training queries are sampled uniformly over eligible tasks, then episodes within tasks, then valid query positions. Evaluation uses up to three unique, equally spaced valid queries per eligible episode. Each query stays within one episode and has all seven observations and twenty targets.

## D.2 Evaluation metrics and aggregation

Let $o^*_{t+h}$ be a target image, $\hat{o}_{t+h}$ its prediction, and $o_t$ the last observation. RGB values have unit data range. The dynamic mask is

$$M_h(u) = \mathbf{1}\Big\{\max_c |o^*_{t+h}(u,c) - o_t(u,c)| > 5/255\Big\}. \tag{39}$$

Pixel and motion-region MSE are defined in Equation (10). The latter averages squared RGB error only over changing pixels, with its denominator clamped to one when the mask is empty. It measures RGB reconstruction accuracy in dynamic regions. PSNR is computed per frame as $-10\log_{10}\max(\ell_{\mathrm{RGB}}, 10^{-12})$ before hierarchical averaging. SSIM uses an $11\times11$ Gaussian window, $\sigma = 1.5$, unit dynamic range, population moments, valid spatial support, and equal averaging over RGB channels (Wang et al., 2004). Raw predictions are scored without clipping; clipping is only used to display images in Appendix G.

H$h$ denotes the cumulative mean over the first $h$ predicted frames. We average queries within episodes, episodes within tasks, and finally tasks with equal weight, yielding a task-macro, episode-macro mean. All compared methods use identical episodes, query positions, actions, and target horizons.

### D.2.1 THE UNIT OF AVERAGING

For task set $\mathcal{J}$, episode sets $E_j$, query sets $Q_{j,e}$, and frame-level score $s_{j,e,q,h}$, the reported H$H$ score is

$$\frac{1}{|\mathcal{J}|}\sum_{j\in\mathcal{J}}\frac{1}{|E_j|}\sum_{e\in E_j}\frac{1}{|Q_{j,e}|}\sum_{q\in Q_{j,e}}\frac{1}{H}\sum_{h=1}^{H}s_{j,e,q,h}. \tag{40}$$

Each task has equal weight, and episodes receive equal weight within their task irrespective of duration. The comparison is paired at the episode level after within-query and within-episode aggregation. Episode-level resampling preserves the grouping of correlated predictions within each rollout. The distribution plots in Appendix F.4 use all forty validation episodes from the single RoboMimic Lift task, where an unweighted episode average agrees with the task-macro mean.

### D.2.2 CHANGING-REGION ERROR VERSUS FULL-IMAGE ERROR

The motion metric weights every selected pixel equally within its frame, whereas full-image MSE also reflects how much of the frame changes. These metrics capture complementary aspects of prediction quality. Define the changing-region fraction $\rho_h = |\mathcal{U}|^{-1}\sum_{u\in\mathcal{U}} M_h(u)$ and let $e_h^{\mathrm{stat}}$ be the mean squared RGB error on the complementary pixels. For $0 < \rho_h < 1$,

$$e_h^{\mathrm{pix}} = \underbrace{\rho_h e_h^{\mathrm{mot}}}_{c_h^{\mathrm{dyn}}} + \underbrace{(1-\rho_h)e_h^{\mathrm{stat}}}_{c_h^{\mathrm{stat}}}. \tag{41}$$

This identity follows by partitioning the sum of squared RGB residuals. At an empty region, its contribution is zero regardless of a convention for its conditional mean. The plotted decomposition computes each contribution directly as a masked sum divided by the full number of RGB entries, avoiding an undefined regional average. All six exported queries have nonempty changing regions and nonempty complements.

Time-averaged regional contributions depend on both conditional errors and their covariance with mask size. With bars denoting horizon averages and covariance using divisor $H$,

$$\overline{e^{\mathrm{pix}}} = \bar{\rho}\,\overline{e^{\mathrm{mot}}} + (1-\bar{\rho})\overline{e^{\mathrm{stat}}} + \mathrm{Cov}_H(\rho, e^{\mathrm{mot}} - e^{\mathrm{stat}}). \tag{42}$$

The covariance term accounts for temporal co-variation between mask size and regional error. The visualization in Appendix G.4 uses the exact per-frame products in Equation (41).

The mask depends only on target and last observed images and is identical for all compared methods. It defines an image-change region that can include visible displacement, illumination differences, and camera changes. Its complement consists of pixels below the specified change threshold. Accordingly, motion MSE measures RGB prediction accuracy on this target-defined region.

### D.2.3 PREFIX MEANS, ENDPOINT ERRORS, AND PSNR

An H20 score measures average quality over the entire forecast interval; endpoint error measures quality at the final prediction step. If $\bar{e}_h = h^{-1}\sum_{s=1}^{h} e_s$, then

$$\frac{1}{h_2-h_1}\sum_{s=h_1+1}^{h_2} e_s = \frac{h_2\bar{e}_{h_2} - h_1\bar{e}_{h_1}}{h_2-h_1}. \tag{43}$$

Thus two prefix means determine the average error over the intervening block. The endpoint plots use separately recorded frame-level metrics with the same query and episode weighting.

PSNR adds a separate nonlinear averaging distinction. For positive clamped frame errors $\widetilde{e}_h = \max(e_h^{\mathrm{pix}}, 10^{-12})$, convexity of $-\log$ gives

$$\frac{1}{H}\sum_h -10\log_{10}\widetilde{e}_h \geq -10\log_{10}\left(\frac{1}{H}\sum_h \widetilde{e}_h\right). \tag{44}$$

Table 7: Comparator representations and use of the supplied action sequence.

| Method | Predictive representation | Action interface / rollout |
|---|---|---|
| DINO-WM (Zhou et al., 2025) | Frozen DINOv2 tokens; causal transformer | Concatenated action features; autoregressive tokens. |
| JEPA-WM (Terver et al., 2026) | Same tokens; adaptive-normalization transformer | Action modulation in each block; autoregressive tokens. |
| SimVP (Gao et al., 2022) | RGB encoder, temporal translator, decoder | Broadcast action channels; rolling RGB context. |
| TAU (Tan et al., 2023) | RGB temporal-attention translator | Broadcast action channels; rolling RGB context. |
| PredRNNv2 (Wang et al., 2023) | Recurrent spatiotemporal memory | Broadcast actions; persistent recurrent states. |
| MAU (Chang et al., 2021) | Motion-aware recurrent units | Broadcast actions; recurrent visual prediction. |
| SwinLSTM-B (Tang et al., 2023) | Window-attention LSTM | Broadcast actions; persistent recurrent state. |

The left side is the frame-averaged PSNR convention used here. Equality holds only when the clamped errors are constant. The gap between the two sides reflects the nonlinear effect of variation in frame-level error. SSIM is likewise computed and averaged as a separate frame-level score. Together, the four metrics characterize pixel accuracy, changing-region accuracy, logarithmic reconstruction quality, and local structural similarity.

### D.3 Comparator families and representation controls

Table 7 identifies the seven comparator families. We fit and evaluate action-conditioned adaptations of all seven families under the common data and evaluation protocol.

SimVP, TAU, PredRNNv2, and MAU use their OpenSTL implementations; SwinLSTM-B uses the authors' single-cell model. The video architectures directly predict RGB, retaining recurrent states where applicable. DINO-WM, JEPA-WM, and both PARM-WM branches use the same frozen visual features and a source-specific image decoder fitted only on training frames. Shared features and decoding control representation differences among latent methods; direct-RGB pipelines retain their native image representations.

SimVP and TAU use spatial/temporal hidden widths $16/256$ and four spatial/four temporal blocks. PredRNNv2 uses four width-128 recurrent layers, $4 \times 4$ patching, and $5 \times 5$ filters, without reverse scheduled sampling. MAU uses four width-64 layers, $5 \times 5$ filters, spatial reduction factor 4, and temporal history length 5. SwinLSTM-B uses embedding width 128, depth 12, four attention heads, $2 \times 2$ patches, window size 4, and drop-path rate 0.1. TAU retains its sequence regularizer with coefficient 0.1, and PredRNNv2 retains memory-decoupling loss with coefficient 0.1; SimVP, MAU, and SwinLSTM-B use RGB MSE alone. Evaluation uses autoregressive recurrent inputs after the observed context.

### D.4 Architecture, normalization, and optimization

**Representation and dimensions.** The frozen DINOv2-S/14 encoder receives images resized to $224 \times 224$ by antialiased bilinear interpolation, followed by channelwise mean/standard-deviation normalization of $0.5/0.5$, and produces $N = 256$ tokens of width $d = 384$. Both latent predictor families use six transformer blocks and sixteen attention heads; the JEPA predictor width is 384. The memory dimension is 64, response/action projections have width 64, and the writer MLP has widths $192 \to 128 \to 64$. Reader width is $r = 32$. AdamW uses weight decay $10^{-4}$ and gradient-norm clipping at 10. Main results use one seed-zero fit per method and source.

**Shared image decoder.** The decoder reshapes the $256 \times 384$ token frame into a $16 \times 16$ feature map. Three $3 \times 3$ convolutions map channels $384 \to 128 \to 64 \to 3$, with eight-group normalization after the first convolution, GELU after the first two convolutions, and two bilinear $2\times$ upsampling stages.

Table 8: Source-specific PARM-WM settings. Updates are optimizer steps. RoboMimic response settings describe its final continuation stage; preceding stages are specified in the text.

| Setting | RoboMimic Lift | LIBERO / LanguageTable |
|---|---|---|
| JEPA initialization updates | 6,144 | 2,048 |
| Response trainable parameters | Writer, reader, base | Writer and reader; base frozen |
| Final response-stage updates | 512 | 512 |
| Training horizon $H_{\mathrm{tr}}$ | 20 | 10 |
| Writer/reader learning rate | $3 \times 10^{-4}$ | $10^{-3}$ |
| Base learning rate | $10^{-5}$ | Not updated |
| RGB / motion weights $(\gamma, \omega)$ | $(100, 0)$ | $(30, 0)$ |
| $\beta_{\mathrm{train}}/\beta_{\mathrm{eval}}$ | $0.05/0.1$ | $0.1/0.1$ |
| Memory / reader width | $64/32$ | $64/32$ |
| Observed / written / predicted $(C, K, H)$ | $(7, 4, 20)$ | $(7, 4, 20)$ |
| Inference mixture $\lambda$ | 0.4 | 0.4 (transferred) |

The decoder uses a linear RGB output layer. Each source-specific decoder is fitted on training frames with RGB MSE, AdamW, batch size 32, and 4,096 updates at learning rate $3 \times 10^{-4}$. RoboMimic uses a further 4,096-update continuation at $10^{-4}$, for 8,192 updates total. Frame sampling is uniform over tasks, then training episodes, then frames. The selected decoder is frozen for all latent predictors on that source, preserving a common rendering map.

**Training-only normalization.** Standardized actions are $a_i = (a_i^{\mathrm{raw}} - \mu_a) \oslash \max(\sigma_a, 0.01)$, with statistics computed over valid training transitions only. At the start of each memory-fitting stage, we estimate coordinatewise RMS scales from $|\mathcal{S}| = 256$ sampled training prefixes, each containing $K = 4$ events:

$$s_{\delta,j} = \max\left(10^{-6}, \sqrt{\frac{1}{K|\mathcal{S}|}\sum_{s\in\mathcal{S}}\sum_{i=1}^{K}\delta_{s,i,j}^2}\right),$$
$$s_{a,j} = \max\left(10^{-6}, \sqrt{\frac{1}{K|\mathcal{S}|}\sum_{s\in\mathcal{S}}\sum_{i=1}^{K}a_{s,i,j}^2}\right). \tag{45}$$

These scales stay fixed during that stage and inference; continuation stages refit them exclusively on training prefixes. Shared scales preserve comparable response magnitudes across events and queries.

**Objectives and stages.** The latent objective is Equation (7): native-token MSE plus $\gamma$ times RGB MSE and $\gamma\omega$ times motion MSE. The response and companion branches are optimized independently on their own autoregressive contexts with full rollout gradients. RGB models optimize RGB MSE plus architecture-specific auxiliary terms where applicable. The shared-history mixture is constructed after fitting.

For LIBERO and LanguageTable, JEPA initialization uses 2,048 latent-only updates at learning rate $3 \times 10^{-4}$, batch size 4, and $H_{\mathrm{tr}} = 2$; the 512-update response stage uses batch size 2. Selected external baselines use 512-update, batch-size-2, H10 stages after their respective initializations. Their selected learning rate is $10^{-4}$, except MAU on LanguageTable ($3 \times 10^{-4}$), from the evaluated set $\{10^{-4}, 3 \times 10^{-4}, 10^{-3}\}$. RoboMimic's response path consists of 6,144 JEPA initialization updates, 512 response-refit updates, 1,024 joint-continuation updates, and the final 512-update RGB-weight-100 continuation. The comparison uses pipeline-specific training budgets, and the complete predictor evaluates two branches per step.

**Selection and branch identities.** All fitting uses training episodes; learning rates, continuation choices, and model selection use validation episodes. The evaluated mixture grid is $\lambda \in \{0, 0.1, \ldots, 1\}$ on RoboMimic validation. Its selected $0.4$ minimizes H20 pixel and motion MSE and is fixed across all sources and queries before testing. For RoboMimic, validation and test both use $\beta_{\mathrm{eval}} = 0.1$, whereas the final response stage uses $\beta_{\mathrm{train}} = 0.05$. The evaluation amplitude is fixed during model construction; validation selection varies the mixture coefficient alone.

Table 9: PARM-WM relative to JEPA-WM on the three frozen sources. Pixel and motion entries are percentage reductions; PSNR and SSIM entries are absolute increases.

| Source | Pixel MSE | Motion MSE | PSNR | SSIM |
|---|---|---|---|---|
| RoboMimic Lift | 49.3% | 47.8% | +2.776 | +0.0323 |
| LIBERO New | 3.8% | 1.7% | +0.180 | +0.0044 |
| LanguageTable | 12.1% | 8.5% | +0.445 | +0.0081 |

The companion $F_\eta$ is also the external JEPA-WM checkpoint on LIBERO and LanguageTable. On RoboMimic it is the separately continued RGB-weight-100 fit, whereas the external JEPA-WM row uses the RGB-weight-30 refit. Zero memory therefore recovers the fitted base mixture, whose RoboMimic components differ from the external JEPA-WM row. Appendix F examines component interventions separately.

### D.5 Paired uncertainty and interpretation

The frozen test has three sources, seven external methods, and four metrics: $3 \times 7 \times 4 = 84$ paired hypotheses. Confidence intervals are paired episode-level percentile bootstrap intervals with 10,000 draws (Efron, 1979). One-sided paired sign-permutation tests use 20,000 draws, with Holm correction over the full 84-test family (Holm, 1979). All 24 method–source evaluations retain matched episode/query sets. Intervals quantify between-episode variation conditional on the fitted checkpoints. The validation state study uses its own two-sided tests and multiplicity families (Appendix F), separately from the frozen-test family.

## E Additional quantitative analyses

### E.1 Cross-source metric summary

Table 9 summarizes PARM-WM's improvements over JEPA-WM. Effect magnitudes are reported here, with globally Holm-corrected significance in the source-wise comparison tables.

PARM-WM improves the JEPA-WM point estimates in all four metrics on every source. RoboMimic provides the largest improvement, and LanguageTable shows gains in both full-frame and changing-region accuracy. LIBERO extends these gains to additional manipulation scenes.

### E.2 Paired effects beyond pixel error

Figure 6 extends the main paper's pixel-error comparisons to all four metrics for JEPA-WM, the representation-aligned reference. The error axes use baseline minus PARM-WM, whereas PSNR and SSIM use PARM-WM minus baseline, so positive always favors PARM-WM. Each interval is the recorded paired episode-bootstrap interval. All displayed comparisons pass Holm correction over the three-source family, including LIBERO motion MSE ($q = 0.0097$). The intervals quantify paired effect uncertainty; marker fill encodes family-corrected significance.

### E.3 Full-image and changing-region tradeoffs

Figure 7 places all eight evaluated methods in the same two-error coordinate system for each source. Lower-left indicates smaller errors on both the entire image and the target-defined changing region. Source-specific axis ranges resolve differences among methods under each source's camera, scene, and action rate. All eight methods are shown in every panel.

PARM-WM occupies the lower-left extreme on RoboMimic, LIBERO, and LanguageTable in these two point-estimate metrics. The small separation from JEPA-WM on LIBERO is visible alongside the much larger separation on RoboMimic. The paired tests provide the corresponding inferential assessment, including the corrected LIBERO motion-MSE result described above.

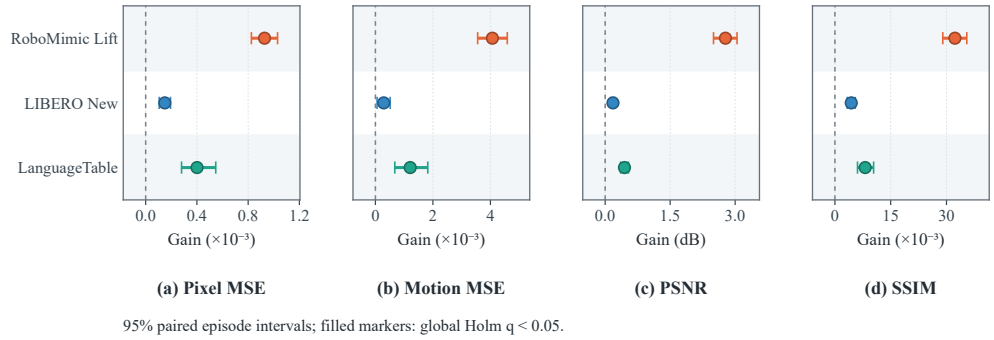



Figure 6: Four-metric paired effects relative to JEPA-WM. Bars are 95% paired episode-bootstrap intervals (10,000 resamples); source sample sizes are 40, 40, and 36 episodes in row order. Filled markers indicate $q < 0.05$ after Holm correction across all 84 frozen-test hypotheses. Pixel MSE, motion MSE, and SSIM differences are expressed in $10^{-3}$ units; PSNR differences are in dB. Intervals condition on the fitted checkpoints.

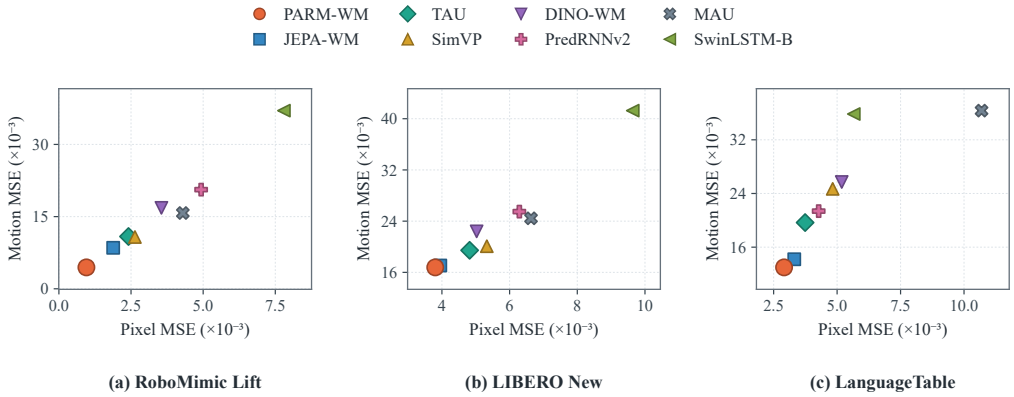



Figure 7: Two-metric geometry of the complete frozen comparison. Each marker is one method's task-macro H20 point estimate; all 24 method–source combinations are included. Both axes show MSE in $10^{-3}$ units and lower is better. Panel-specific axis ranges preserve within-source detail.

### E.4 Cross-source synthesis

The source-wise comparisons establish improvements over the representation-aligned JEPA-WM reference, with the largest gains on RoboMimic. Comparisons against direct-RGB methods characterize the source-dependent ranking of complete predictive pipelines. The state interventions in Appendix F complement these system-level results by isolating the effect of the prefix-derived memory.

## F Ablation and sensitivity analyses

### F.1 State-intervention design

The state study isolates the contribution of persistent-state information on the 40-episode RoboMimic validation split. Table 4 reports all H20 state-ablation scores. Inference interventions keep the fitted model parameters fixed and vary only the supplied state information, while separately trained static controls test input-dependent versus constant conditioning.

The action-only condition repeats the first support frame before computing writer descriptors, making both signed and absolute changes zero while retaining the actions and all fitted model parameters. The three static controls use learning rates $10^{-4}$, $3 \times 10^{-4}$, and $10^{-3}$, with their other matched

Table 10: Meaning of the validation state controls. Input interventions and separately trained controls have different inferential roles.

| Condition | Intervention | Property examined |
|---|---|---|
| Correct | Own observed action–response state | Reference conditional predictor. |
| Shuffled | Another episode's state within the same task | Query-specific state content. |
| Action-only | Zero observed feature changes; preserve actions | Contribution of observed responses. |
| Zero | Set $m = 0$ at inference | State-conditioned correction of the fitted base. |
| Static | Separately learn one input-independent state | Input-dependent versus constant conditioning. |

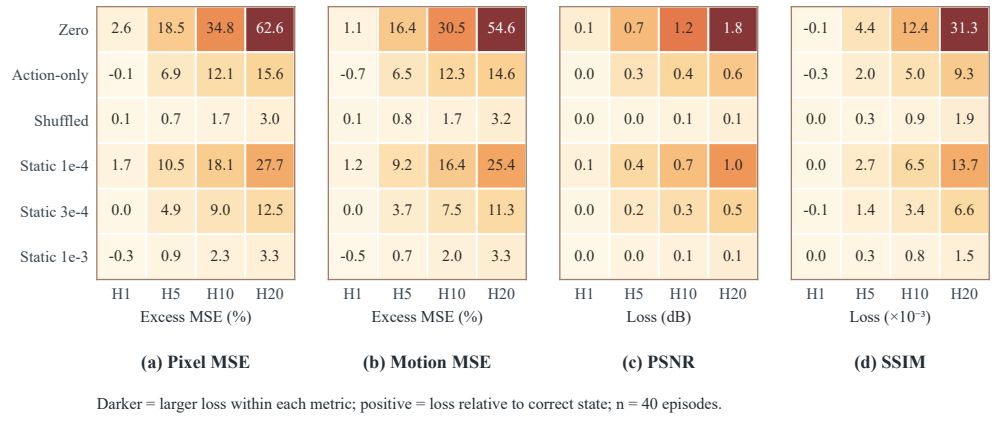



Figure 8: Validation state sensitivity across H1/H5/H10/H20. Each panel reports a control's score relative to the correctly matched state. Positive values indicate worse performance for the control; pixel and motion MSE are percentage increases, PSNR losses are in dB, and SSIM losses are in $10^{-3}$ units. Values are rounded to one decimal. All six controls use the same 40 validation episodes and fixed fitted checkpoints.

settings retained. A static state starts at 0.1, whereas the reader output map starts at zero, enabling initial gradients through the multiplicative reader path. The state-shuffle intervention replaces the whole state across episodes; Proposition 2 instead analyzes action permutation within a query.

### F.2 Response matching across horizons and metrics

The correct state has the best H20 point estimates in all four metrics. Its motion MSE is 3.1% below the shuffled state and 3.2% below the strongest static control. The shuffled state retains 94.1% of the correct-versus-zero motion-MSE improvement. Query matching therefore adds a further improvement to the substantial shared conditioning effect.

Figure 8 expands the horizon comparison to all four metrics and all static controls. Values are descriptive within this validation study; the main paper's paired-episode plot gives the corresponding uncertainty for H20.

### F.3 Paired validation analysis

Validation intervals use 100,000 paired episode bootstrap draws, and two-sided sign-flip tests use 100,000 draws. The recorded H20 family contains four controls (zero, action-only, shuffled, and the strongest static fit) times four metrics, for 16 comparisons. The all-horizon family additionally includes H1/H5/H10/H20, for 64 comparisons. These families are separate from the 84 frozen-test comparisons. Both model selection and the state analyses use this validation split; their intervals quantify between-episode variation conditional on the fitted models.

Table 11: Episode-level H20 gains of the correct state over each validation control. All gains are control minus correct, in $\times 10^{-3}$ units. Brackets enclose the first and third quartiles. Wins count strictly positive paired gains among all 40 episodes.

| Control | Metric | Median [Q1, Q3] | Wins |
|---|---|---|---|
| Zero state | Pixel MSE | 1.081 [0.928, 1.189] | 40/40 |
| Zero state | Motion MSE | 4.243 [3.603, 4.943] | 40/40 |
| Action-only | Pixel MSE | 0.226 [0.140, 0.381] | 39/40 |
| Action-only | Motion MSE | 1.028 [0.551, 1.583] | 37/40 |
| Shuffled state | Pixel MSE | 0.033 [0.006, 0.099] | 31/40 |
| Shuffled state | Motion MSE | 0.128 [0.025, 0.466] | 32/40 |
| Static state | Pixel MSE | 0.056 [0.008, 0.102] | 34/40 |
| Static state | Motion MSE | 0.257 [0.061, 0.426] | 34/40 |

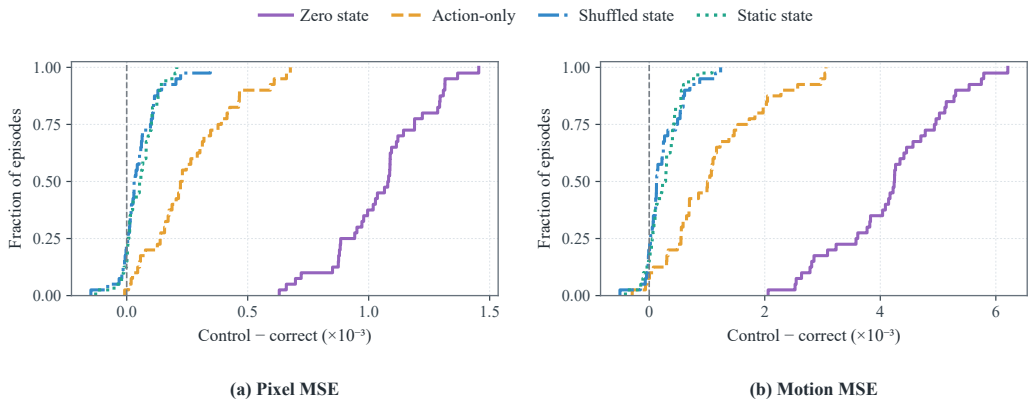



Figure 9: Empirical distributions of H20 paired gains for all forty RoboMimic validation episodes. Gains are control minus correct; values right of zero favor the matched response state. Static denotes the validation-selected learning rate $10^{-3}$.

### F.4 Episode-level distribution of state benefits

To characterize variation in intervention effects across episodes, define $g_e^{\text{pix}} = e_{e,\text{control}}^{\text{pix}} - e_{e,\text{correct}}^{\text{pix}}$ and analogously $g_e^{\text{mot}}$ after averaging each episode's matched queries. Figure 9 gives their empirical cumulative distributions, and Table 11 reports medians, interquartile ranges, and strictly positive episode counts. All forty episodes are retained for every control. Each curve's value at zero is the fraction of episodes without a strictly positive gain; the rightward separation of a curve indicates generally larger gains.

Correct state improves both errors over zero state on all 40 episodes. Against action-only state, the counts are 39/40 for pixel error and 37/40 for motion error. Against shuffled state, the counts are 31/40 and 32/40; against the strongest static state, they are 34/40 for each metric. For example, the median motion gain over zero is $4.243 \times 10^{-3}$, and over shuffled state it is $0.128 \times 10^{-3}$. These gains distinguish the shared state-conditioned correction from the episode-specific benefit of matching the state to the observed prefix.

### F.5 Concordance between full-image and motion improvements

Figure 10 pairs the two gains within each episode to show their joint distribution. The upper-right quadrant denotes a simultaneous improvement in image and changing-region fidelity; opposite-sign quadrants identify metric tradeoffs. Axis ranges vary between controls because the zero-state effect is substantially larger than the matched-versus-shuffled effect.

Both errors improve in 40/40 episodes against zero, 36/40 against action-only, 30/40 against shuffled, and 33/40 against static state. The remaining episodes comprise three single-metric gains and seven

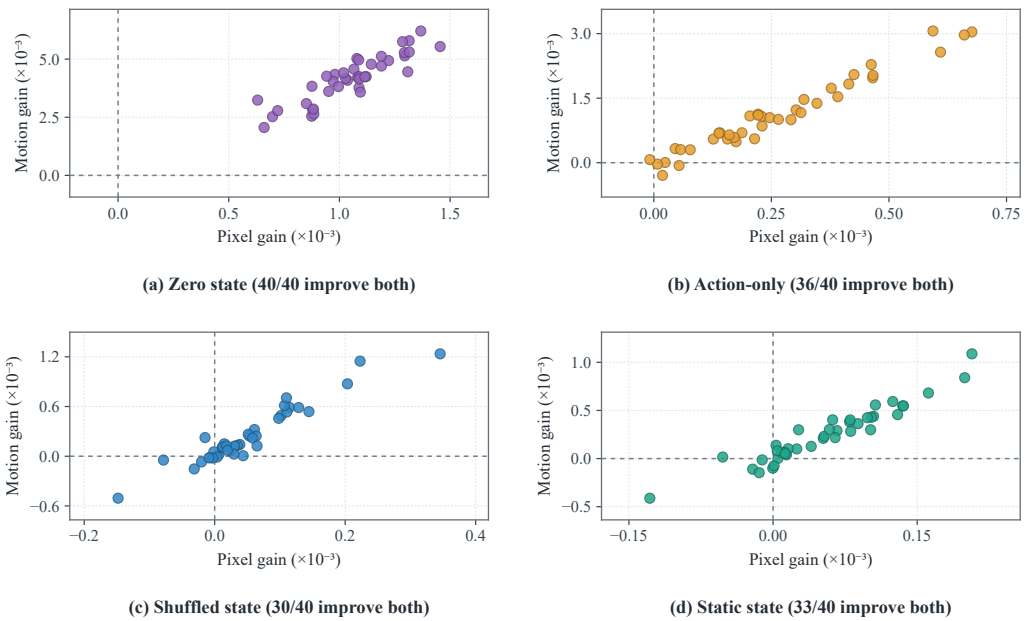



Figure 10: Joint pixel and motion gains within the same validation episode. Each panel includes forty points; positive coordinates favor correct state. Bottom annotations give the number of episodes with strictly positive gains on both metrics. Quadrants show the joint distribution of the two fidelity gains for each state intervention.

non-positive pairs for shuffled state, and two single-metric gains and five non-positive pairs for static state. These counts describe the distribution of state benefits across episodes.

Both coordinates derive from the same predicted images, with motion error computed on their target-defined changing regions. The joint distribution therefore describes how two related fidelity measures respond to each state intervention.

### F.6 Cumulative and endpoint prediction quality

The prefix averages used in the main comparison emphasize the whole forecast interval. Figure 11 additionally shows recorded endpoint errors at steps 1, 5, 10, and 20 for the correct and zero states. At step one the two aggregation conventions coincide, providing a direct consistency check. At longer horizons the endpoint curves resolve final-step prediction quality alongside the accumulated mean.

For the correct-state validation predictor, H20 pixel MSE is $1.685 \times 10^{-3}$ while endpoint pixel MSE is $2.574 \times 10^{-3}$. Motion MSE changes from the H20 mean $7.654 \times 10^{-3}$ to the endpoint value $11.455 \times 10^{-3}$. The correct state improves over zero state under both aggregation conventions, including at the final prediction step.

An additional reference characterizes image reconstruction through the frozen encoder and decoder. Applying the fixed decoder to encoded future target frames gives H20 pixel and motion reconstruction errors of $0.688 \times 10^{-3}$ and $2.791 \times 10^{-3}$ on these same validation queries. These are target-conditioned reconstruction scores, evaluated at encoded target tokens, whereas forecasts use the predictor's output tokens. Copying the last observed image instead gives H20 pixel and motion errors of $9.424 \times 10^{-3}$ and $55.953 \times 10^{-3}$. Together, the two references characterize encoder–decoder reconstruction and static persistence alongside the action-conditioned forecast.

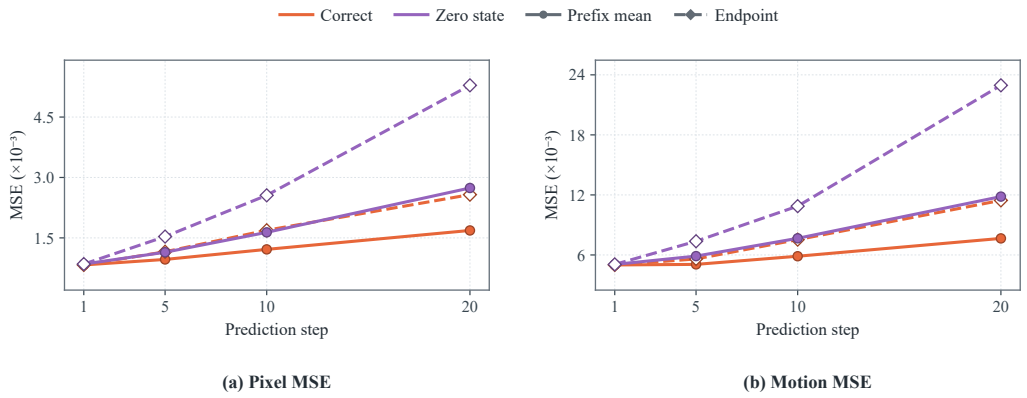



Figure 11: Prefix-mean versus endpoint errors for the same validation queries. Solid lines average predictions $1, \ldots, h$; dashed lines score only prediction $h$. Colors distinguish correct and zero state. Markers identify the evaluated horizons, connected by visual guides.

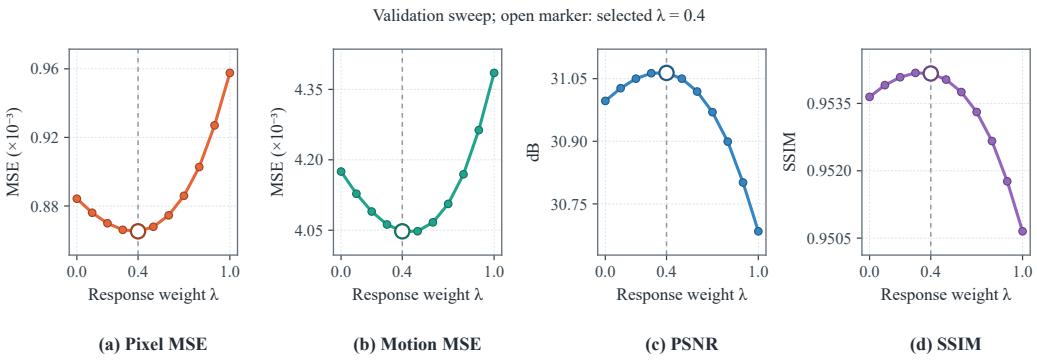



Figure 12: Complete RoboMimic validation mixture sweep with fixed fitted branches and shared-history inference. Every marker is one recorded weight in $\{0, 0.1, \ldots, 1\}$; the open marker identifies the selected $\lambda = 0.4$. Pixel and motion MSE favor smaller values, whereas PSNR and SSIM favor larger values. Selection uses RoboMimic validation episodes, with the chosen weight subsequently fixed across sources.

### F.7 Mixture-weight selection and sensitivity

The validation mixture sweep evaluates $\lambda = 0, 0.1, \ldots, 1$ with frozen branch weights and shared-history inference. The selected $\lambda = 0.4$ minimizes both H20 pixel and motion MSE on RoboMimic; the SSIM maximum occurs at $\lambda = 0.3$. The selected coefficient therefore follows the two MSE criteria and transfers unchanged to LIBERO and LanguageTable. Equation (24) characterizes same-context interpolation, while this sweep evaluates the resulting recursive decoded rollouts.

Figure 12 shows all eleven evaluated weights. Relative to the companion-only prediction at $\lambda = 0$, the selected mixture reduces validation pixel MSE by 2.15% and motion MSE by 3.05%. The corresponding selected values are $0.86523 \times 10^{-3}$ and $4.04768 \times 10^{-3}$. The latter is close to the $\lambda = 0.5$ value, $4.04788 \times 10^{-3}$, indicating a nearly flat motion-error curve between these adjacent weights.

The error curves rise toward the response-only endpoint, while an interior interval remains competitive across the four metrics. This supports combining the fitted branches at an interior weight. The mixture sweep characterizes how branch interpolation affects forecasting accuracy, while the state ablations isolate the contribution of memory content. Together, these experiments examine branch combination and state conditioning as complementary components of the predictor.

Table 12: H20 errors for the qualitative comparisons (MSE $\times 10^{-3}$). Bold identifies the lowest error for each query and metric.

| | Pixel MSE | | | Motion MSE | | |
|---|---|---|---|---|---|---|
| Case | PARM | JEPA | TAU | PARM | JEPA | TAU |
| RoboMimic Lift (R1) | **1.288** | 3.325 | 3.741 | **7.025** | 18.628 | 19.191 |
| RoboMimic Lift (R2) | **0.974** | 3.313 | 1.645 | **4.570** | 14.733 | 6.769 |
| LIBERO New (L1) | **3.389** | 4.512 | 4.702 | **15.481** | 21.293 | 19.938 |
| LIBERO New (L2) | **2.526** | 2.850 | 3.922 | **13.016** | 14.581 | 16.673 |
| LanguageTable (T1) | **2.849** | 4.695 | 4.523 | **7.796** | 16.856 | 17.191 |
| LanguageTable (T2) | **4.820** | 7.107 | 6.700 | **13.223** | 20.964 | 22.139 |

# G Qualitative comparisons

## G.1 Comparison protocol

We compare twenty-step predictions of PARM-WM, JEPA-WM, and action-conditioned TAU from their evaluated checkpoints. Each comparison fixes the seven-frame observed prefix and the future action sequence. PARM-WM and JEPA-WM use the same source-specific latent-to-RGB decoder; TAU predicts RGB directly. Following the aligned rollout-grid presentation used in visual world modeling (Zhou et al., 2025; Terver et al., 2026), rows identify methods and columns identify the same prediction times, $t+1, t+5, t+10, t+15, t+20$. All panels evaluate open-loop visual prediction under the recorded action sequence.

**Query protocol.** For RoboMimic, LIBERO, and LanguageTable, we rank held-out queries by the recorded JEPA-WM minus PARM-WM H20 motion-MSE difference, requiring PARM-WM to improve both pixel and motion MSE over both displayed baselines. We take two distinct episodes per source and prefer distinct tasks when available. The six queries are organized as follows: Figures 13–15 show the first query per source, and Figure 16 shows the second. Table 12 reports their twenty-step average errors.

**Display and error-map convention.** Observed and target RGB values are mapped from 8-bit intensities to $[0, 1]$. Predictions are clipped to that interval only for display; all reported errors and error maps use the unmodified floating-point predictions. The final column of each rollout grid shows the endpoint dynamic mask in the ground-truth row and endpoint RGB root-mean-square error in the model rows, with a shared color scale across methods. These maps visualize the endpoint, whereas the H20 scores average all twenty predicted frames. The detail panel uses the same $24 \times 24$ crop for every method, chosen by maximizing the target-versus-last-observation absolute RGB change summed over the crop. Crop placement therefore depends exclusively on target and observed images. All methods use nearest-neighbor display and the same intensity mapping.

## G.2 Qualitative results across three sources

**RoboMimic Lift.** Figure 13 shows a visibly changing arm and gripper near the end of the horizon. PARM-WM reduces the concentrated endpoint errors around these moving boundaries relative to the two displayed baselines. The second episode in Figure 16 shows a similar reduction in the aligned arm-detail crop. Together, these examples illustrate more accurate visual continuation around the moving arm and gripper.

**LIBERO.** The LIBERO sequence in Figure 14 concentrates prediction differences around the arm and gripper. PARM-WM has lower twenty-step pixel and motion errors. The aligned crops in this example and the second LIBERO episode in Figure 16 place the corresponding gripper-region differences in a common spatial frame.

**LanguageTable.** In Figure 15, the target arm changes its visible configuration over the prediction horizon. PARM-WM follows more of this change, whereas the displayed JEPA-WM and TAU rollouts retain a straighter arm appearance. The common-scale maps localize much of the difference

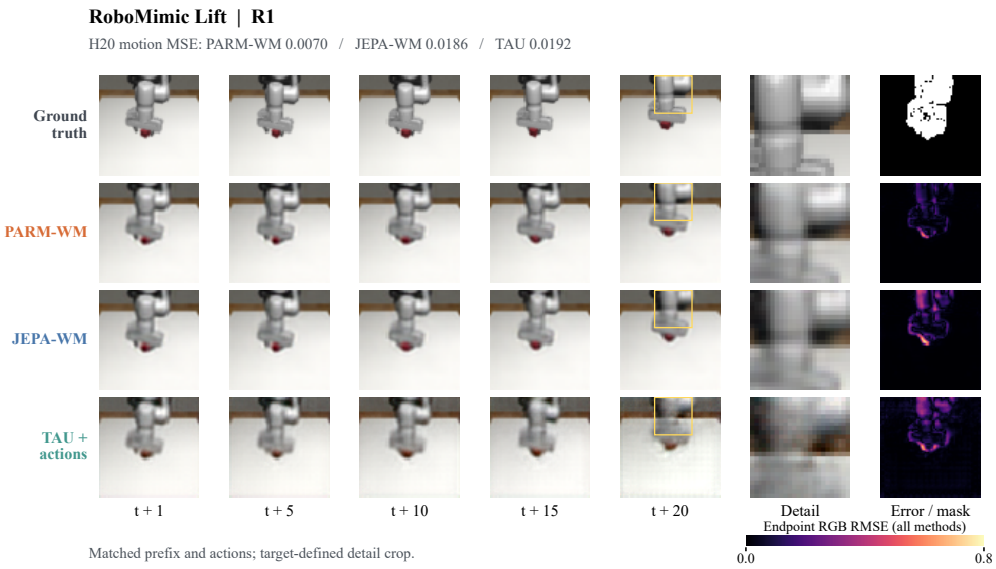



Figure 13: RoboMimic Lift query R1. The aligned rollout, endpoint detail, and common-scale error maps show reduced error around moving arm boundaries. Yellow boxes mark identical crops. H20 scores average twenty frames; the error maps and binary dynamic mask refer only to $t + 20$.

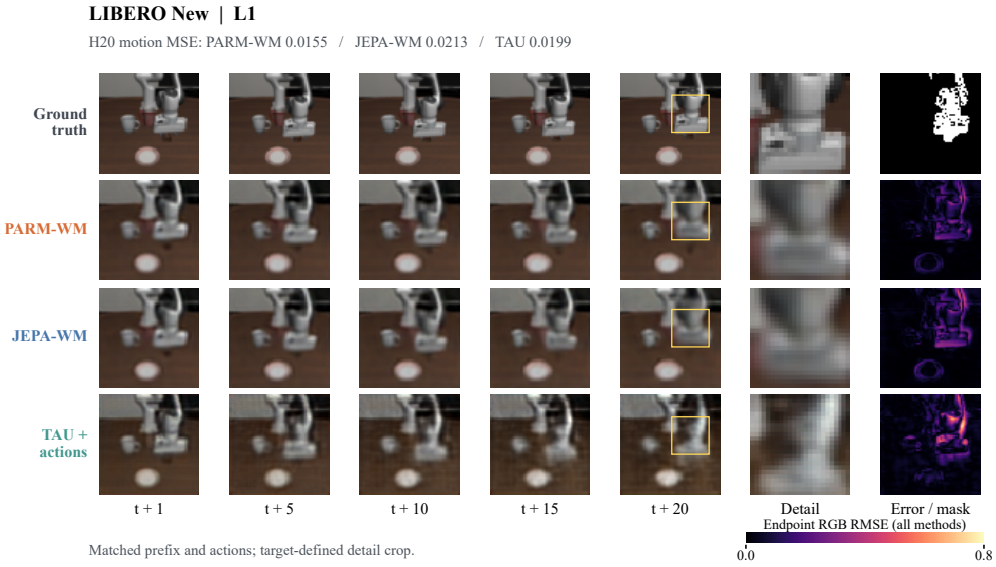



Figure 14: LIBERO query L1 under identical conditioning. PARM-WM improves the H20 errors of both displayed comparators. The matched gripper crop resolves local prediction differences. Layout, endpoint error maps, and target-defined crop follow Figure 13.

to the arm rather than to the static tabletop. The second query in Figure 16 likewise shows reduced local arm-boundary error under the image-and-action conditioning used in this experiment.

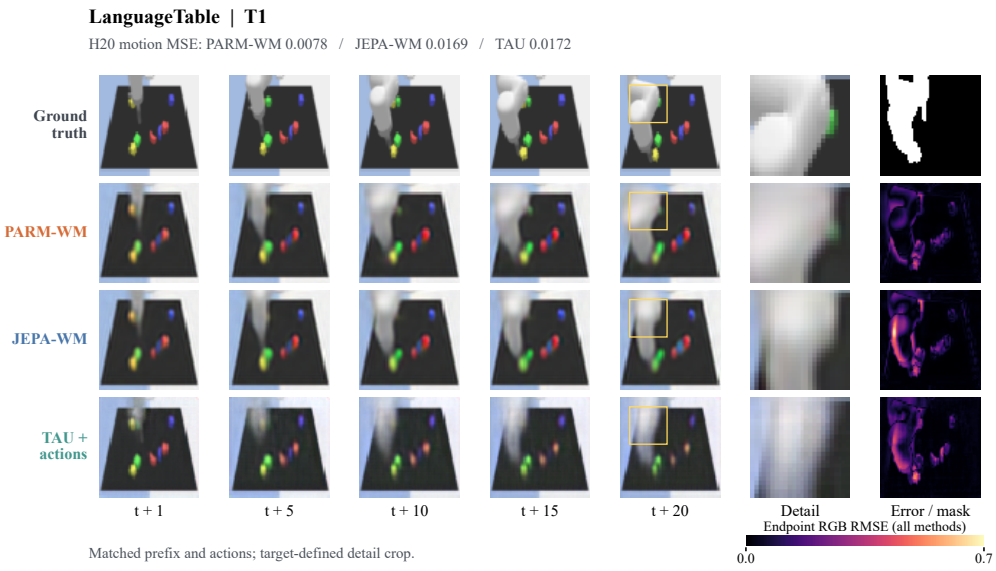



Figure 15: LanguageTable query T1. PARM-WM captures more of the changing arm configuration. All rows use the same observed prefix, numerical actions, and endpoint error scale.

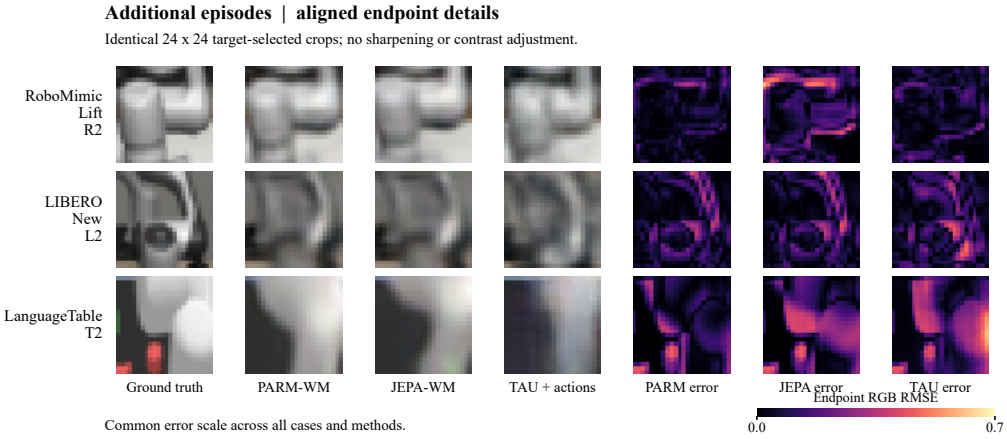



Figure 16: Second-episode endpoint details: R2, L2, and T2. Each row uses an identical target-defined crop across ground truth and predictions. All nine error maps share one scale. These localized endpoint views complement the full-image, twenty-step errors in Table 12.

### G.3 Second-episode rollout trajectories

The endpoint atlas provides a common local comparison scale. Figures 17–19 extend it to the full second-query sequences using the same five sampled times, target-only crop rule, and method ordering as the first set. These are the same R2, L2 and T2 queries reported in Table 12.

For RoboMimic R2, the approaching gripper changes its position relative to the small red object. The aligned rows expose a larger arm-boundary discrepancy for JEPA-WM; PARM-WM and TAU are closer to each other, with lower aggregate errors for PARM-WM. The H20 motion errors are $4.570$, $14.733$, and $6.769$ in $10^{-3}$ units for PARM-WM, JEPA-WM, and TAU, respectively, consistent with the observed differences around the moving arm.

In LIBERO L2, all three predictions reproduce the broad scene structure with similar smoothing around the moving gripper. PARM-WM's H20 motion error $13.016 \times 10^{-3}$ is below JEPA-WM's

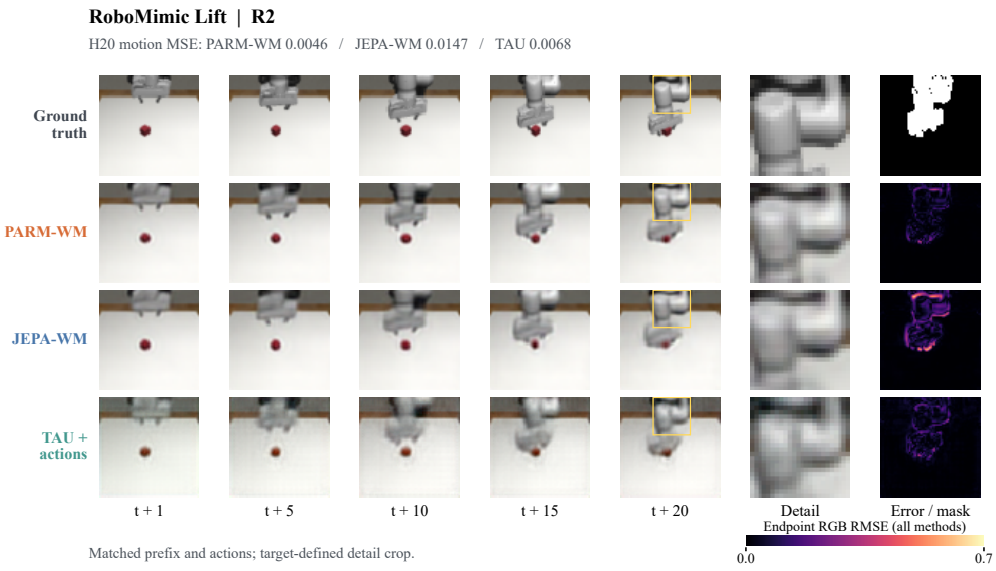



Figure 17: Second RoboMimic sequence R2. Identical observed conditioning and actions produce different gripper continuations. The right-hand crop uses one target-defined location for all methods, and the error maps use a common scale within the figure.

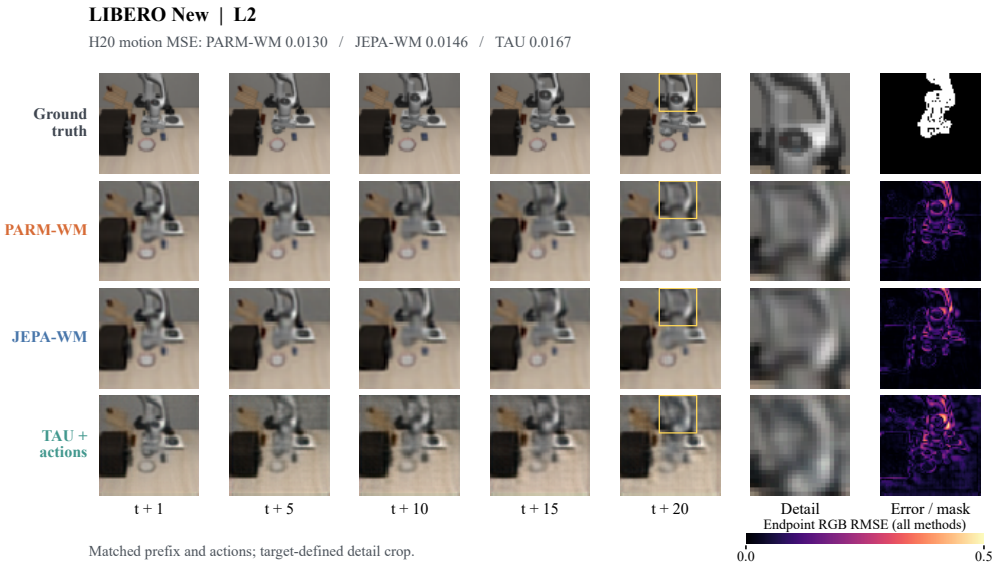



Figure 18: Second LIBERO sequence L2. PARM-WM reduces changing-region error under matched conditioning. The same images appear as localized endpoint details in Figure 16; this full sequence additionally exposes their temporal development and surrounding scene context.

$14.581 \times 10^{-3}$ and TAU's $16.673 \times 10^{-3}$. The matched crop and common-scale error maps resolve the local differences at the native $64 \times 64$ resolution.

In LanguageTable T2, the visible arm occupies a different part of the tabletop as the sequence advances. The late prediction and matched crop show differences in the arm boundary and its surrounding image region; the common-scale error map localizes those discrepancies. Its motion errors are 13.223, 20.964, and 22.139 in $10^{-3}$ units for PARM-WM, JEPA-WM, and TAU, respectively.

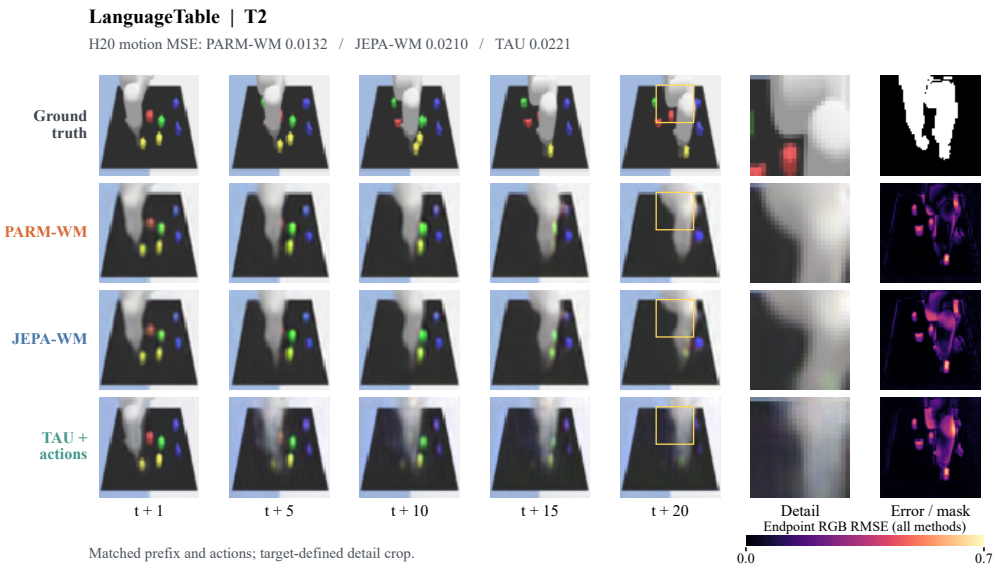



Figure 19: Second LanguageTable sequence T2. Full frames and a matched arm-region crop relate the favorable aggregate errors to visible local differences. Every method receives the same image prefix and numerical actions.

### G.4 Spatial decomposition of prediction error

The image plates show location, whereas Figure 20 quantifies how changing and complementary regions contribute to each query's full-image MSE. Each stack is the exact sum of time-averaged per-frame contributions in Equation (41). Its total therefore reproduces that query's pixel MSE. The darker segment is the changing-region error weighted by its per-frame area fraction. Across the six queries, the mean changing-region fraction ranges from 12.0% to 23.9%.

For R1, changing pixels occupy 14.5% of the image on average but account for 83.9% of PARM-WM's remaining pixel error. Relative to JEPA-WM, both the changing-region contribution (1.081 versus 2.899) and its complement (0.208 versus 0.426), in $10^{-3}$ units, decrease. Both regions contribute to the full-image improvement.

For T1, PARM-WM reduces the changing-region contribution to $1.475 \times 10^{-3}$, compared with $3.406 \times 10^{-3}$ for JEPA-WM and $3.408 \times 10^{-3}$ for TAU. The corresponding complement contributions are $1.374 \times 10^{-3}$, $1.289 \times 10^{-3}$, and $1.116 \times 10^{-3}$, respectively. The changing-region reduction yields lower full-image error.

### G.5 Temporal profiles of the displayed queries

Figure 21 retains every one of the twenty predicted frames for each displayed query. It connects the five-frame image samples to their H20 summaries through the complete error trajectory. Each panel uses a labeled, case-specific vertical range to resolve variation at its error scale. Every trace corresponds to one predicted query.

The RoboMimic trajectories show sustained motion-error reductions relative to JEPA-WM. LIBERO L1 and L2 exhibit different magnitudes of H20 gain, while LanguageTable shows clear separation late in the forecast.

H20 and endpoint scores summarize cumulative and final-step accuracy, respectively. For the displayed R2 example, PARM-WM's H20 and endpoint motion errors are 4.570 and 5.631 in $10^{-3}$ units, compared with TAU's 6.769 and 6.817. PARM-WM improves both summaries, with the time-resolved profiles showing how the separation develops over the forecast.

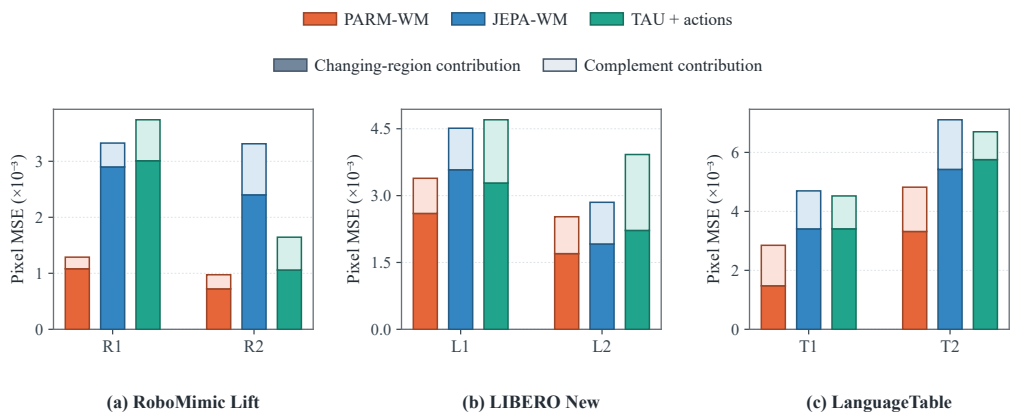



Figure 20: Exact regional decomposition of pixel MSE for all six queries. Color identifies the method; darker and lighter segments are the changing-region and complement contributions, respectively. Products with the target-defined area fractions are computed per frame before averaging. The two cases within each panel are distinct episodes. Query identifiers follow Table 12.

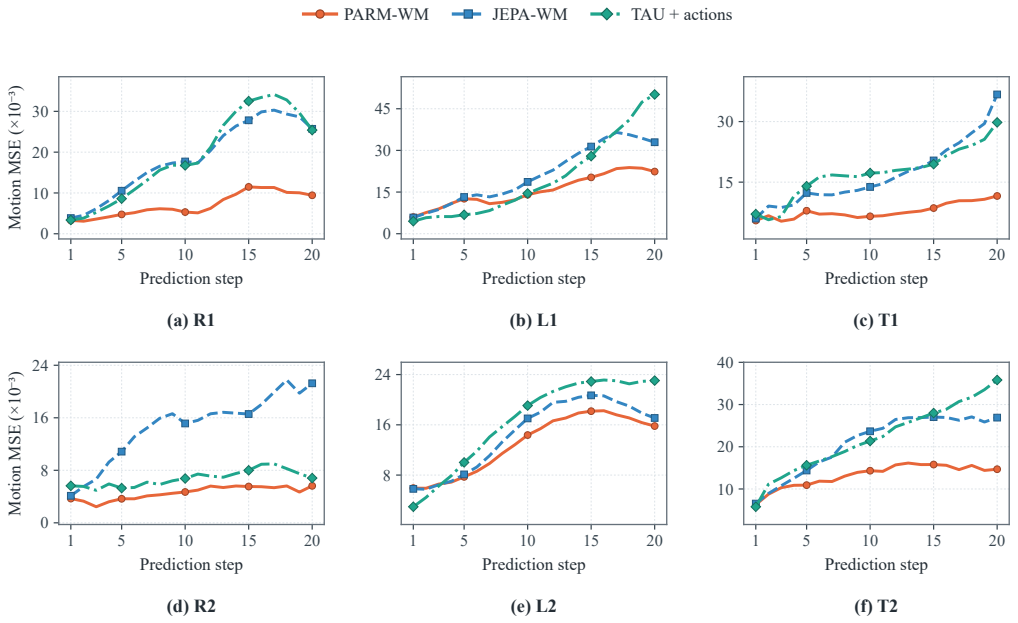



Figure 21: Frame-level motion-error trajectories for all six queries. Columns correspond to the three sources; rows correspond to their first and second cases. Curves use raw, unclipped predictions and the same target-defined masks as the aggregate metrics. Straight segments connect the recorded frame-level scores of the queries shown in the image plates.

Together, the images, spatial decompositions, and temporal profiles connect measured prediction differences to their locations and development within each rollout. They complement the complete held-out comparisons with query-level views of action-conditioned visual prediction.