ADAPTING WORLD MODELS THROUGH RESPONSE-PRESERVING REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

A world model can predict the next observation almost perfectly and still be the wrong model to plan with, because deployment physics rarely match the simulator it was trained in. A controller must read the change from a short prefix of frames and actions, then apply the same correction to every future it imagines. We show that ordinary forward training quietly undermines the first step: a history-dependent predictor explains part of a shifted successor from the history itself, absorbing the change and leaving too little residual for any calibration procedure to read. Forward and controlled-pair objectives cannot tell this allocation from an informative one, since models that divide the change differently make identical predictions. Response-preserving representations (RPR) fix the allocation rather than the symptom: the encoder keeps a grid of spatial motion tokens, controlled source pairs anchor an explicit response channel, and encoder, predictor, and response are trained through the very weighted ridge solve that runs at deployment. With the networks frozen, one solve on the prefix returns the change coefficient and a calibrated covariance; the coefficient is reused at every imagined step and refreshed as observations arrive. A linear-Gaussian analysis ties retained response to prefix estimation risk. On three simulated visual-control tasks, RPR recovers the change from eight transitions in under nine milliseconds, matches twenty-step gradient adaptation on the all-profile mean, and surpasses it wherever the change lies in the source family, while true-coefficient controls and absorption diagnostics attribute the gain to a better estimate and a better forward model alike.

1 INTRODUCTION

A world model earns its place in a control loop by predicting what an action will do in a learned representation (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2020; 2025). Train one in a simulator and deploy it where the physics differ – a stiffer actuator, a heavier object, more drag – and it has two jobs at once. From a short prefix of frames and recorded actions it must infer how the dynamics changed, and while it imagines candidate futures it must apply that correction the same way at every step. Neither job is guaranteed by next-observation accuracy, whether encoder and predictor are learned jointly, as in LeWorldModel (LeWM; Maes et al., 2026), or built on pretrained visual features, as in DINO-WM and V-JEPA 2-AC (Zhou et al., 2025; Assran et al., 2025). Our subject is the deployment-time change that such predictors must expose to a planner.

The first job fails in a way that is easy to state and hard to see. A predictor that looks at a few frames can infer a great deal about the current physics – how far an object moved for a command, how quickly it came to rest – and the forward objective rewards it for folding that inference into its prediction. Whatever the predictor absorbs from the history is missing from the residual that a calibration procedure would later read, so the model can fit shifted trajectories closely while leaving the change it was meant to expose without any reliable representation. We call this *predictor absorption* (Figure 1). It has a mirror image: an explicit response channel can attenuate on shifted histories, reporting the change at a fraction of its true size. Forward and controlled-pair objectives see neither allocation, because models that divide a shifted successor differently between history and response predict identical successors.

The allocation matters because calibration is used inside planning. A context vector inferred from the prefix drifts as soon as real frames are replaced by imagined ones, and gradient-based test-time training

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.

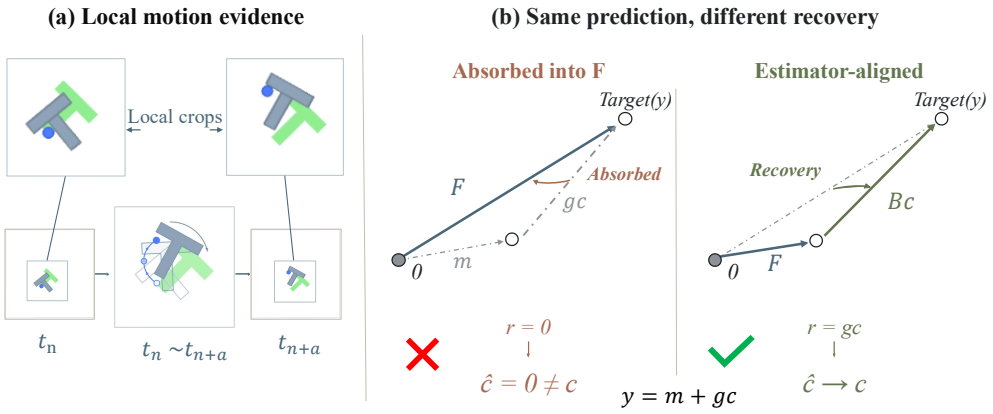

Figure 1: **Same prediction, different recovery. Left:** public Push-T observations carry the evidence of a physical change as small local motion (Chi et al., 2023), which is why the encoder keeps a spatial token grid. **Right:** in a noiseless example, two allocations of the same shifted successor between F and Bc predict identically, yet the ridge solve (limit $\lambda \downarrow 0$) recovers c from only one of them.

rewrites the weights but leaves no explicit quantity to carry forward. Recovery and persistence are therefore separate requirements. What a planner needs is a coefficient in a source-defined coordinate system: estimated from real observations, held fixed while candidates are imagined, and applied to every candidate under the same calibrated dynamics.

Response-preserving representations (RPR) train for exactly this. The encoder keeps a grid of spatial motion tokens, so that small local displacements remain visible. Controlled source pairs, formed by replaying one action from the same state under two physical settings, anchor how a successor should respond to the change. Ordinary source prefixes with known coefficients then pass through a differentiable weighted ridge solve, and supervising its output sends gradients into the encoder, predictor, and response channel, so the residual used for recovery and the response used for prediction are shaped together. At deployment the networks are frozen: one solve returns the coefficient and a calibrated covariance, the coefficient conditions every imagined transition, and new observations refresh it between replans.

The outcome is a calibration step that costs one small linear solve, under nine milliseconds on eight transitions, yet matches twenty-step gradient adaptation across six stress profiles and surpasses it wherever the change lies in the affine source family. Our contributions are threefold.

- We identify predictor absorption and response attenuation as two allocations of the same physical change, show why forward and pair losses cannot separate them, and relate the retained response to estimation risk.
- We introduce RPR, which combines spatial-token residuals, controlled-pair anchoring, and a differentiable weighted-ridge objective, and reuses one prefix estimate throughout planning.
- We evaluate recovery, persistent forecasting, and control under matched budgets on three visual-control tasks; true-coefficient controls, absorption and retention diagnostics, stress profiles, and candidate-action transfer tests separate calibration gains from representation error.

2 Related Work

Latent prediction and visual resolution. TD-MPC learns latent dynamics and values for planning (Hansen et al., 2022; 2024), and JEPAs predict learned targets without reconstructing pixels (Assran et al., 2023; Bardes et al., 2024). LeWM combines next-embedding prediction with LeJEPA's SIGReg (Balestriero & LeCun, 2025; Maes et al., 2026), while DINO-WM and V-JEPA 2-AC predict on pretrained features (Zhou et al., 2025; Assran et al., 2025). Identifiability results relate such representations to state and controlled dynamics under excitation conditions (Klindt et al., 2026; Zhang et al., 2026), and Tan et al. (2026) document a mismatch between prediction and parameter

identification that direct parameter supervision only partly closes. We estimate source-defined changes from spatial residuals instead, so the same coefficients condition future rollouts.

Context and test-time adaptation. History-based identification and meta-learned dynamics support rapid adaptation (Yu et al., 2017; Kumar et al., 2021; Nagabandi et al., 2019). CaDM separates a context from transition prediction (Lee et al., 2020), and HiP-RSSM infers latent factors shared by related systems (Shaj et al., 2022); both persist a context through planning, as an explicit coefficient does. AdaJEPA adapts the model before each replan (Wang et al., 2026), prefix test-time training updates weights once from the prefix (Sun et al., 2020), and ReDRAW fits a latent residual from offline target data (Lanier et al., 2026). We compare against all three regimes under matched observation budgets.

Learning representations through estimation. R2-D2 differentiates through a closed-form regression (Bertinetto et al., 2019), ALPaCA learns features and priors for Bayesian regression (Harrison et al., 2018), and task-based and control-oriented learning optimize models for the decisions made with them (Donti et al., 2017; Richards et al., 2021). RPR carries this idea into world-model calibration: controlled pairs anchor the response directions, and coefficient recovery through the deployed solver trains E, F, and B jointly, so the residual used to estimate a change is aligned with the response used to predict its effect. Gaussian conditioning links prefix information to action comparisons (Bishop, 2006); Appendix E traces the connections to system identification.

3 Method

3.1 Problem setup

We train in a simulator whose physical parameters can be set, and at deployment we must calibrate their unknown values from observations alone. Source-defined coordinates $c \in \mathbb{R}^p$ specify the physical parameters through a fixed map $\vartheta = h(c)$, with $c = 0$ the nominal environment. The gain–drag experiments use affine coordinates; the native-contact extension uses log-ratios of positive physical parameters. Within an episode the coordinates are constant.

An encoder E_θ maps each observation x_t to a spatial token grid $Z_t \in \mathbb{R}^{N\times q}$, and we write $z_t = \mathrm{vec}(Z_t) \in \mathbb{R}^d$ with $d = Nq$. For ViT-Tiny/14 we project the 16×16 patch tokens to $q = 32$ channels, giving $N = 256$ and $d = 8192$; no global pooling is applied, so local displacements stay separately visible. A length-L history is $H_t = (z_{t-L+1:t}, a_{t-L+1:t-1})$. The model augments a nominal predictor F_ϕ with an explicit response channel:

$$G(H_t, a_t; c) = F_\phi(H_t, a_t) + B_\psi(H_t, a_t)c, \qquad B_\psi(H_t, a_t) \in \mathbb{R}^{d\times p}. \tag{1}$$

Thus $B_\psi = \partial G/\partial c$ exactly, while the response itself may vary with state, action, and contact; the affine form is a local latent approximation to the simulator's response, not a restriction on the encoder.

Simulator rollouts supply the source coefficient labels. To anchor the response, we restore a source state and replay one action under two settings with shared randomness, which yields controlled successor pairs. At deployment the model receives T transitions with recorded actions and the $L-1$ preceding frames, but no physical state, nominal counterpart, or future frame. Fixed-prefix calibration solves once on this prefix; continual calibration refits the coefficient after each new real transition. Both keep the networks and source-calibrated weights frozen during every imagined rollout: one coefficient conditions every candidate future, while $B(H, a)$ follows the imagined state and action. Appearance is sampled independently of c.

3.2 A predictor with an explicit response channel

Source prediction uses the known episode coefficients:

$$\mathcal{L}_{\mathrm{pred}} = \mathbb{E}\big[d^{-1}\|E_\theta(x_{t+1}) - G(H_t, a_t; c)\|_2^2\big]. \tag{2}$$

Nominal episodes train F_ϕ at $c = 0$. A controlled pair shares a nominal history H^0, an action a, and a random stream, but changes the physical setting by δ. The difference between the encoded successors anchors the response channel:

$$\mathcal{L}_{\mathrm{ctrl}} = \mathbb{E}\left[\frac{\|E_\theta(x_+^\delta) - E_\theta(x_+^0) - B_\psi(H^0, a)\delta\|_2^2}{d\|\delta\|_2^2}\right]. \tag{3}$$

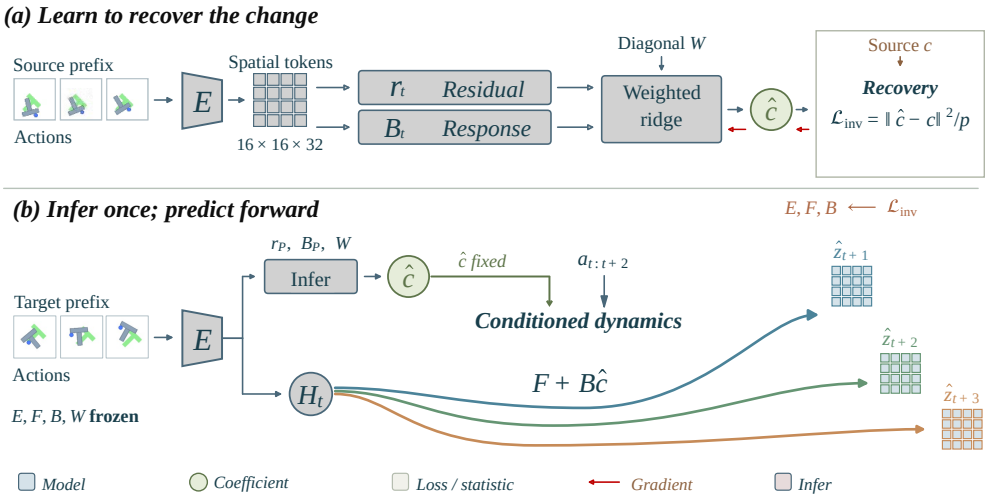

Figure 2: RPR **learns to recover the change, then reuses the estimate. Top:** spatial residuals and response matrices from an ordinary source prefix enter a differentiable weighted ridge solve, and the recovery loss trains E, F, and B through it. **Bottom:** at deployment the networks are frozen; one solve on the target prefix yields $\widehat{c}$, which conditions every imagined step. Public Push-T frames are from Chi et al. (2023); token tiles depict spatial features.

These objectives pin down the sum $F + Bc$ on ordinary data and the response at the histories used to form pairs, but they leave open how a shifted successor is divided between the two terms. A history-dependent F can absorb a shifted successor while B compensates, shrinking the estimator's residual without changing the forward prediction. This is predictor absorption, and the next objective is designed to rule it out.

3.3 Training through the deployed estimator

For an ordinary source prefix P of T transitions, form spatial residuals and response matrices:

$$r_t = E_\theta(x_{t+1}) - F_\phi(H_t, a_t), \qquad r_P = \mathrm{col}_{t\in P}(r_t), \qquad B_P = \mathrm{col}_{t\in P}\, B_\psi(H_t, a_t). \tag{4}$$

An independent source split estimates a diagonal residual variance for every token and channel, $\widehat{\Sigma} = \mathrm{diag}(s_1^2, \ldots, s_d^2)$, and the prefix weight is $W_T = I_T \otimes \widehat{\Sigma}^{-1}$. The same weighted ridge solve used at deployment estimates the change:

$$\widehat{c}_P = \arg\min_b \|r_P - B_P b\|_{W_T}^2 + \lambda\|b\|_2^2 = (B_P^\top W_T B_P + \lambda I_p)^{-1} B_P^\top W_T r_P. \tag{5}$$

We solve the positive-definite $p \times p$ system by Cholesky factorization and supervise its output on ordinary prefixes with known coefficients:

$$\mathcal{L}_{\mathrm{inv}} = \mathbb{E}_{c,P}\left[p^{-1}\|\widehat{c}_P - c\|_2^2\right], \tag{6}$$

$$\mathcal{L} = \mathcal{L}_{\mathrm{pred}} + \lambda_{\mathrm{SIG}}\mathcal{L}_{\mathrm{SIG}} + \lambda_{\mathrm{ctrl}}\mathcal{L}_{\mathrm{ctrl}} + \lambda_{\mathrm{inv}}\mathcal{L}_{\mathrm{inv}}. \tag{7}$$

The label c is the target of the solve, never an input to it. Gradients through the solve update E, F, and B to reduce coefficient error; setting $\lambda_{\mathrm{inv}} = 0$ gives the forward-pair baseline with the same data, labels, and capacity. A separate history-to-parameter head would bypass this coupling entirely: it never has to make r_P and B_P compatible with the solver that runs at deployment.

What the solve teaches the model. Write $M = B_P^\top W_T B_P + \lambda I$ and $e_P = r_P - B_P\widehat{c}_P$. Differentiating the normal equations gives

$$M\,\mathrm{d}\widehat{c}_P = (\mathrm{d}B_P)^\top W_T e_P + B_P^\top W_T(\mathrm{d}r_P - \mathrm{d}B_P\widehat{c}_P) + B_P^\top(\mathrm{d}W_T)e_P. \tag{8}$$

Coefficient error therefore reaches the predictor through $\mathrm{d}r_P$, the response channel through $\mathrm{d}B_P$, and the encoder through both successors and histories. The inverse objective is not a post-hoc probe: it selects, among all allocations that predict equally well, the one that makes the deployed estimator informative.

Spatial weighting and cost. The diagonal covariance is estimated from ordinary source prediction errors and frozen after source calibration. Whitening costs $O(Td(p+1))$ and the $p \times p$ solve $O(Tdp^2 + p^3)$, with $p = 2$ for gain–drag and $p \leq 4$ for native contact. Coefficient scales, variances, and λ stay fixed across prefixes, so additional observations enter under the same prior; weakly excited directions have small information eigenvalues, ridge stabilizes the solve, and the covariance below reports what remains.

3.4 Persistent coefficients and posterior-aware planning

Fixed-prefix RPR estimates $\widehat{c}_P$ once per episode; continual RPR applies Equation (5) to the latest window of real transitions before each replan. In both, the estimate stays fixed throughout every imagined rollout while F and B are reevaluated on the imagined history. With P the initial prefix or current real window, each candidate action advances an imagined history from the observed H_t:

$$\widehat{z}_{k+1} = G(\widehat{H}_k, a_k; \widehat{c}_P), \qquad \widehat{H}_{k+1} = \mathrm{shift}(\widehat{H}_k, a_k, \widehat{z}_{k+1}). \tag{9}$$

Each update replaces the oldest state and action with an imagined successor, so after L steps the history is entirely model-generated while $\widehat{c}_P$ still comes from observations. A re-inferred context would read this imagined history instead; the fixed-context baseline therefore separates persistence from estimator alignment.

Planning with the inferred change. CEM ranks candidate action sequences by the terminal distance to the encoded goal. All candidates share the same prefix estimate and model parameters, while F and B depend on each candidate's imagined history. After an action is executed, its observation joins the history; this updates the state in both protocols and updates the coefficient only in the continual variant (Appendix F).

Under the Gaussian prior $c \sim \mathcal{N}(0, \lambda^{-1} I)$, define

$$\mathcal{I}_P = B_P^\top W_T B_P, \qquad V_P = (\lambda I_p + \mathcal{I}_P)^{-1}, \qquad \widetilde{V}_P = \gamma_T V_P. \tag{10}$$

A source-held-out coverage rule sets γ_T to account for correlated tokens and model error (Appendix B). The posterior-aware variant, RPR-P, draws four coefficient samples from $\mathcal{N}(\widehat{c}_P, \widetilde{V}_P)$ and minimizes the mean terminal cost plus 0.5 times the sample standard deviation. Each sample is fixed through its rollout, and common samples are shared across action candidates; the point-estimate comparison receives the same fourfold rollout budget. Algorithm 1 in Appendix B summarizes training and deployment.

3.5 From prefix recovery to action comparisons

Prediction and calibration constrain different quantities. Forward losses constrain predictions and controlled response differences; calibration constrains recovery through $r = z_+ - F$. Two allocation failures live in the gap between them. With a fixed response channel, the history-dependent predictor can absorb a shifted successor and leave a small residual. With a learned response channel, B can instead attenuate on shifted histories, reducing the information available to the estimator while the forward fit is unchanged. Appendix A gives compact constructions for both effects; the first is quantified next.

Proposition 1 (Predictor absorption increases estimation risk)**.** *Fix $b \neq 0$, $\Sigma \succ 0$, and $r = (1-\beta)bc + \epsilon$, where $\mathbb{E}[c] = 0$, $\mathrm{Var}(c) = \tau^2$, $\epsilon \sim \mathcal{N}(0, \Sigma)$ independently, and $\beta \geq 0$. For $s = b^\top \Sigma^{-1} b$ and ridge penalty $\lambda \geq 0$, the coefficient mean-squared error satisfies*

$$\mathcal{R}(\beta) - \mathcal{R}(0) = \frac{\tau^2(2\lambda\beta s + \beta^2 s^2)}{(s+\lambda)^2} \geq 0. \tag{11}$$

Here βbc is the part of the response absorbed into F, with b and Σ held fixed. If instead B is attenuated to αb on shifted histories, its information becomes $\alpha^2 s$, and coefficient risk grows even though the forward predictor is untouched. The inverse objective must therefore shape both the residual allocation and the response direction; reducing predictor absorption alone is not enough, and the experiments test the two separately.

Table 1: Fixed-prefix success (%, ↑) at $T = 8$. The all-six column is the equally weighted mean over nominal, interpolation, extrapolation, weak excitation, appearance, and nonlinear-drag profiles. Entries show mean ± 95% bootstrap half-width.

	Planar drag		Push-T		Cube	
Method	Interp.	All six	Interp.	All six	Interp.	All six
Forward-pair	70.8±12.2	64.7±9.8	40.4±13.8	38.5±9.9	33.8±12.0	33.9±8.8
Param. + ridge	79.0±7.8	69.0±9.0	51.4±12.2	46.1±11.4	46.8±12.6	42.3±9.5
Param. + head	82.2±7.2	71.5±9.1	55.2±13.0	49.2±11.0	50.6±10.0	45.0±9.4
Fixed context	84.6±8.0	74.0±8.2	61.0±14.8	55.7±12.3	55.0±8.8	49.3±8.8
Recurrent ctx.	80.4±9.4	71.4±8.8	49.2±12.0	48.1±11.8	43.8±12.0	42.1±9.6
Long history	78.4±7.2	70.3±8.9	52.2±15.0	49.3±12.3	45.6±13.8	43.3±10.1
TTT-F	90.4±5.6	81.3±7.1	68.2±14.4	63.7±12.3	69.4±9.4	61.8±9.3
TTT-LoRA	88.8±5.4	81.5±7.0	69.6±12.4	65.1±12.4	65.0±9.6	61.9±8.9
RPR	92.2±5.4	79.6±7.0	78.8±12.0	65.8±11.6	70.8±9.0	59.2±9.2

Information sets the required prefix length. For a correctly specified, fixed-design linear-Gaussian residual model with $c \sim \mathcal{N}(0, \tau^2)$, the posterior-mean coefficient MSE is $\mathcal{R}_c = (\tau^{-2} + \mathcal{I}_P)^{-1}$ (Proposition 2). Independent transitions with equal information ι give $\mathcal{I}_P = T\iota$: less information per transition demands a longer prefix. Contact-rich prefixes carry direction-dependent information – repeated free-space motion need not reveal grasp friction – and an attenuated B shrinks the corresponding columns of $\mathcal{I}_P$ even when F predicts the histories well.

Planning depends on action contrasts. For affine candidate costs, let $\mathcal{D}$ stack the differences of coefficient gradients between candidates. Estimation error with mean μ_e and covariance V_e perturbs the cost gaps by $\|\mathcal{D}\mu_e\|^2 + \mathrm{tr}(\mathcal{D}V_e\mathcal{D}^\top)$, and a noiseless prefix resolves every contrast exactly when $\ker B_P \subseteq \ker \mathcal{D}$ (Proposition 3). The planner needs the coefficient only along directions that change the ranking of candidates, which is what the candidate-transfer tests of Appendix C.4 measure.

4 Experiments

The experiments ask three questions. Does training through the deployed solve recover the change better from the same prefix, and does the gain survive long-horizon imagination and closed-loop control? Where does the gain come from – a better estimate, or a better forward model? And what does one solve cost, and where does an affine coefficient stop being enough?

4.1 Experimental setup

Tasks and shifts. The suite covers planar motion with drag, object pushing in Push-T, and cube manipulation in OGBench's `visual-cube-single-v0` (Chi et al., 2023; Park et al., 2025). Each task varies gain and drag, fixed within an episode, and provides 4,500 source episodes split 3,600/450/450 plus 24,000 controlled successor pairs. Targets follow six profiles – nominal, interpolation, extrapolation, weak excitation, appearance, and nonlinear drag – with reset, action, and rendering seeds disjoint from the source (Appendix C).

Matched baselines. All fixed-prefix comparisons share the spatial ViT-Tiny/14 backbone, the source data, and 70,000 training updates. Forward-pair training drops $\mathcal{L}_{\mathrm{inv}}$ and uses the same ridge solve at deployment; parameter-supervised ridge and a direct parameter head (Param. + ridge, Param. + head) add coefficient supervision without the solve; CaDM-style fixed context and HiP-RSSM-style recurrent context infer a latent context (Lee et al., 2020; Shaj et al., 2022); a long-history predictor attends to the full prefix. Prefix test-time training updates F (TTT-F) or rank-8 LoRA adapters (TTT-LoRA) for 20 steps on the prefix and then freezes (Sun et al., 2020; Hu et al., 2022). True-c conditioning keeps each model's networks and planner but replaces its estimate by the true coefficient, which isolates estimation error from representation error. Source validation selects every hyperparameter, including $(\lambda_{\mathrm{ctrl}}, \lambda_{\mathrm{inv}}) = (0.25, 0.05)$, fixed for all tasks (Table 15).

Budgets and metrics. Nested prefixes $T \in \{2, 4, 8, 16, 32\}$ end at the same physical state; the default is $T = 8$ with history length $L = 3$. Forecasts replay common future actions at horizons $h \in \{1, 5, 10, 20\}$, and control starts from a separate copy of the same endpoint. CEM uses 512 candidates, five iterations, 64 elites, and a ten-step horizon (Rubinstein & Kroese, 2004). We report

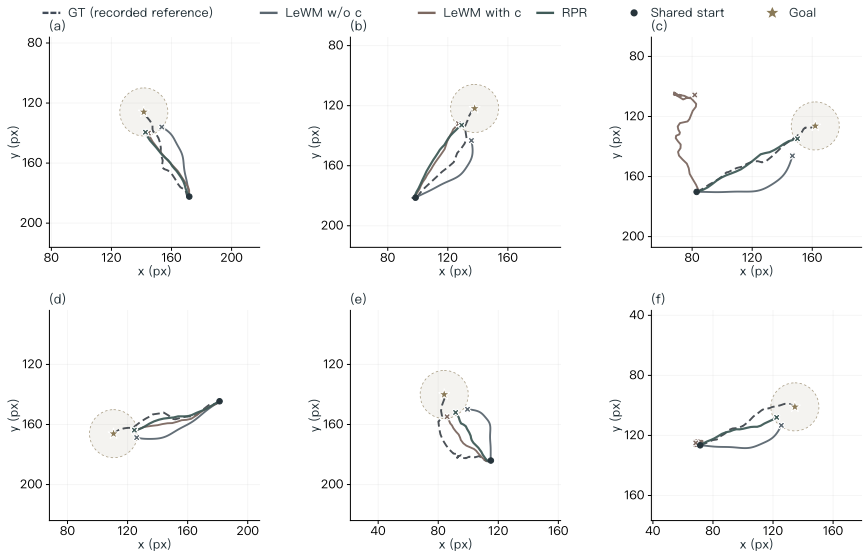

Figure 3: **Planar-drag trajectory cases.** Six cases share a start (dot) and a target region (star); the recorded reference is dashed. The nominal LeWM predictor (w/o c) and the same predictor conditioned on the forward-pair ridge estimate (with c) overshoot or stall under the shifted drag, while RPR follows the reference into the target. Aggregates are in Tables 1 and 10.

coefficient RMSE, source-normalized physical forecast error (NRMSE) read out from imagined and observed frames by one source-trained readout, success within 200 control steps (five balanced goals for Cube), and latency, over five seeds and 100 paired scenarios per target group with 95% hierarchical bootstrap intervals. Table 1 reports interpolation next to the equal-weight mean over all six profiles (Appendix D).

4.2 Adaptation from a shared observation budget

Table 1 shows the intended operating point. On affine interpolation, RPR is the best method on all three tasks: it improves over parameter-supervised ridge by 13 to 27 points and over the strongest gradient-based adapter by 1 to 9 points, with the widest margin on Push-T, where contact makes the change hardest to read from a few frames. Supervising the coefficient is not what closes the gap – Param. + ridge and Param. + head see the same labels – nor is a richer context, since both context models and the long-history predictor trail. What the solve changes is what the residual contains. On the all-profile mean, RPR and the two 20-step adapters lie within one interval of each other, because extrapolation, weak excitation, appearance, and nonlinear drag enter with equal weight; TTT-LoRA wins the nonlinear-drag column on every task (Tables 10–12), which marks where a two-coordinate affine estimate stops describing the change. Figures 3 and 4 show individual episodes in which the nominal and forward-pair predictors overshoot or lose contact while the corrected model reaches the goal.

Persistence during imagination. Methods with similar one-step errors diverge over longer rollouts. On Push-T at $h = 20$, RPR reaches an NRMSE of 0.199, against 0.437 for forward-pair training and 0.361 for parameter-supervised ridge, while the readout floor on observed frames stays near 0.03 for all three (Table 9). Context drift is visible in Table 1 as well: the recurrent context, which re-infers from the imagined history, trails the fixed context it shares a backbone with by 4 to 12 points on every task. A coefficient estimated once and reused keeps the correction intact as imagined states replace real ones.

4.3 What makes the change recoverable?

Absorption and retention. At a shifted endpoint, controlled replays give the reference response $\Delta^* = E(x_+^c) - E(x_+^0)$; let $u = F(H^{\text{shift}}, a) - E(x_+^0)$ be the part of it that the nominal predictor already

Table 2: Push-T controls at $T = 8$. A_F: predictor absorption; K_B: response retention; lower recovery and rollout RMSE are better. Each true-c column uses that row's own checkpoint. Entries include 95% intervals.

Training	A_F	K_B	Recovery	h=20, $\hat{c}$	h=20, c
Forward-pair	0.589 ± 0.027	0.290 ± 0.013	0.473 ± 0.011	0.437 ± 0.011	0.278 ± 0.006
Param. + ridge	0.420 ± 0.022	0.460 ± 0.020	0.366 ± 0.011	0.361 ± 0.008	0.241 ± 0.006
RPR	0.118 ± 0.005	0.910 ± 0.034	0.218 ± 0.005	0.199 ± 0.005	0.158 ± 0.003
RPR / no pairs	0.254 ± 0.010	0.620 ± 0.026	0.309 ± 0.008	0.285 ± 0.007	0.224 ± 0.005
RPR / stop E	0.206 ± 0.009	0.660 ± 0.028	0.281 ± 0.006	0.255 ± 0.006	0.204 ± 0.006
RPR / stop F	0.434 ± 0.018	0.490 ± 0.023	0.352 ± 0.008	0.332 ± 0.008	0.236 ± 0.007
Nominal-only F	0.255 ± 0.012	0.670 ± 0.032	0.317 ± 0.007	0.298 ± 0.007	0.249 ± 0.006
Anchored F	0.073 ± 0.003	0.710 ± 0.036	0.250 ± 0.006	0.236 ± 0.005	0.205 ± 0.004

produces from the shifted history. Signed absorption is $A_F = \langle u, \Delta^* \rangle_W / (\|\Delta^*\|_W^2 + \eta)$, which is independent of B and therefore measurable even when the learned response vanishes. Response retention K_B compares the learned response on shifted and nominal histories (Appendix C.4).

Training ablations. Table 2 separates the two failure modes. Forward-pair training absorbs 59% of the reference response into F and retains 29% of its response channel; RPR absorbs 12% and retains 91%, and its recovery error falls by more than half. Removing controlled pairs or stopping gradients into E weakens both diagnostics together, whereas stopping gradients into F returns absorption to the level of parameter-supervised ridge: the solve has to be allowed to reshape the predictor. The anchored-F control is the decisive case. It has the smallest absorption of all, yet retains less response and recovers worse than RPR, so removing the predictor's share of the change is not enough when the learned response direction is weak or poorly conditioned. Recovery needs both allocations fixed at once, exactly as Section 3.5 predicts.

Calibration or representation? The true-c columns keep each row's checkpoint and future actions fixed and remove only the coefficient-estimation error, so the estimated-to-true gap is the calibration cost and the remaining true-c error is the forward-model cost. RPR reduces both: its true-c rollout error is 0.158 against 0.278 for forward-pair training, so inverse training changes the forward representation and not merely the estimate, and its calibration gap shrinks to 0.04, a quarter of the forward-pair gap. The intervention repeats on all three tasks with the same ordering (Table 7).

Transfer to candidate actions. Planning only benefits if the estimate improves comparisons between actions that were never observed. Appendix C.4 fixes 512 candidate sequences before adaptation and scores them by offline simulator replays. On Push-T, RPR ranks candidate pairs in agreement with the simulator 78.5% of the time, against 65.9% for forward-pair training and 82.8% with the true coefficient, and its selected-candidate regret is halved (Table 8). Erasing the coefficient after the first L imagined steps, once the history is model-generated, drops agreement to 68.8%, and swapping in another episode's estimate drops it to 64.9%: the gain comes from the right coefficient being reused, not from conditioning as such.

4.4 Efficiency, robustness, and the edge of the affine family

Adaptation cost. Table 3 reports the cost of test-time updates. RPR fits in a median of 8.7 ms, including encoding, weighting, and the solve, against 167.2 ms for TTT-LoRA and 285.2 ms for updating F; planning cost is the same for every affine model. The gap matters under a budget: with a 100-ms cap, LoRA can afford at most ten steps and reaches 62.2% capped success, while 20 steps time out on every episode and fall back to the source model at 40.4% (Table 13). RPR sits at 78.8% inside that budget with an order of magnitude to spare.

Calibrated uncertainty. The covariance of Equation (10) is calibrated on held-out source episodes and transfers: the 90% ellipsoid covers 90.2% of interpolation targets, 86.0% under weak excitation, and 75.4% under extrapolation, with areas that grow as the prefix becomes less informative (Table 14). Spending the same rollout budget on four coefficient draws instead of four times as many candidates, RPR-P raises weak-excitation success from 59.2% to 62.8% and does not help extrapolation: the posterior helps where the prefix is ambiguous, not outside the source support.

Table 3: Fixed-prefix accuracy and cost on Push-T at $T = 8$. Fitting and planning latency are median/p95 in ms; peak device memory is in MB, and control success is a percentage.

Method	Fit (ms)	Plan (ms)	Memory	Success
Forward-pair	8.7/10.0	72.6/83.4	1755	40.4
Param. + ridge	8.9/10.4	72.5/85.2	1777	51.4
Param. + head	5.2/6.0	72.6/84.6	1748	55.2
Fixed context	5.9/6.8	81.7/95.2	1944	61.0
Recurrent ctx.	6.3/7.4	96.4/111.6	2065	49.2
Long history	7.6/8.7	105.6/120.5	2482	52.2
TTT-F	285.2/348.1	71.8/88.0	3381	68.2
TTT-LoRA	167.2/204.6	69.3/82.8	2695	69.6
RPR	8.7/9.9	71.8/82.7	1774	78.8



(a) Push-T case A **(b)** Push-T case B

Figure 4: **Push-T trajectory cases.** Each panel shows a shared start, three push checkpoints, and the terminal state for the recorded reference, the nominal LeWM predictor (w/o c), the same predictor conditioned on the forward-pair estimate (with c), and RPR. Circles mark contact with object motion; RPR reaches the goal pose in both cases.

Reading the stress profiles. Tables 10–12 expose the boundary of prefix calibration. On nominal, interpolation, and appearance shifts, RPR matches or leads every method, since appearance changes leave the source-defined coordinates intact. On extrapolation the two routes trade wins. Under weak excitation the prefix carries little information along the excited direction, and under nonlinear drag a two-coordinate estimate misses the extra velocity term; there, repeated gradient updates lead by two to six and ten to twelve points. Continual RPR closes most of that gap at a fraction of the cost: on Push-T it lifts interpolation success from 78.8% to 85.4% and nonlinear-drag success from 51.4% to 70.2% in 18 adaptation seconds per episode, where continual LoRA reaches 87.2% and 78.2% in 102 seconds (Table 17).

Native contact parameters. Appendix C.2 replaces the injected gain and drag by cube mass, table and finger friction, and arm stiffness in the native MuJoCo model, with log-ratio coordinates and $p \in \{1, 2, 4\}$. RPR leads with one and two factors, ties TTT-LoRA with four, and recovers all four coordinates with an aggregate RMSE of 0.398 against 0.491 for parameter-supervised ridge (Tables 4–5). Composition across withheld sign orthants favors gradient adaptation.

5 Conclusion

A world model that predicts well is not thereby a model that can be calibrated: the forward objective lets the predictor absorb the very change a planner needs, and controlled pairs alone cannot prevent it. RPR trains the encoder, predictor, and response channel through the weighted ridge solve that runs at deployment, so the residual read from a short prefix carries the change and the recovered coefficient persists through every imagined future. One solve costs under nine milliseconds and matches or beats twenty-step gradient adaptation wherever the change lies in the affine source family; true-coefficient controls and absorption diagnostics show that the gain is a better forward representation as much as a better estimate, and continual refitting carries part of the gain beyond that family.

AI USE STATEMENT

We used AI-based tools to assist with language editing, LaTeX typesetting, and figure scripting. All research ideas, methods, experiments, analyses, and conclusions are the authors' own, and every AI-assisted passage was checked by the authors.

REFERENCES

Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J. Zico Kolter. Differentiable convex optimization layers. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://papers.neurips.cc/paper_files/paper/2019/hash/9ce3c52fc54362e22053399d3181c638-Abstract.html.

Brandon Amos and J. Zico Kolter. OptNet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, volume 70, pp. 136–145, 2017. URL https://proceedings.mlr.press/v70/amos17a.html.

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*, 2023. URL https://arxiv.org/abs/2301.08243.

Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. URL https://arxiv.org/abs/2506.09985.

Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025. URL https://arxiv.org/abs/2511.08544.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=xm6YD62D1Ub.

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. URL https://openreview.net/forum?id=QaCCuDfBk2.

Luca Bertinetto, João F. Henriques, Philip H. S. Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyxnZh0ct7.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. URL https://link.springer.com/book/9780387310732.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023. URL https://roboticsproceedings.org/rss19/p026.html.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/3de568f8597b94bda53149c7d7f5958c-Abstract.html.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 2 edition, 2006. doi: 10.1002/047174882X. URL https://onlinelibrary.wiley.com/doi/book/10.1002/047174882X.

Priya Donti, Brandon Amos, and J. Zico Kolter. Task-based end-to-end model learning in stochastic optimization. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1703.04529.

Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G. Bellemare. DeepMDP: Learning continuous latent space models for representation learning. In *International Conference on Machine Learning*, volume 97, pp. 2170–2179, 2019. URL https://proceedings.mlr.press/v97/gelada19a.html.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31, pp. 2451–2463, 2018. URL https://worldmodels.github.io/.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, volume 97, pp. 2555–2565, 2019. URL https://proceedings.mlr.press/v97/hafner19a.html.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025. doi: 10.1038/s41586-025-08744-2. URL https://www.nature.com/articles/s41586-025-08744-2.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Oxh5CstDJU.

Nicklas A Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In *International Conference on Machine Learning*, volume 162, pp. 8387–8406, 2022. URL https://proceedings.mlr.press/v162/hansen22a.html.

James Harrison, Apoorva Sharma, and Marco Pavone. Meta-learning priors for efficient online bayesian regression. *arXiv preprint arXiv:1807.08912*, 2018. URL https://arxiv.org/abs/1807.08912.

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. URL https://arxiv.org/abs/2106.09685.

Taylor W. Killian, Samuel Daulton, George Konidaris, and Finale Doshi-Velez. Robust and efficient transfer learning with hidden parameter markov decision processes. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/2227d753dc18505031869d44673728e2-Abstract.html.

David Klindt, Yann LeCun, and Randall Balestriero. When does LeJEPA learn a world model? *arXiv preprint arXiv:2605.26379*, 2026. URL https://arxiv.org/abs/2605.26379.

Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: Rapid motor adaptation for legged robots. In *Proceedings of Robotics: Science and Systems*, 2021. doi: 10.15607/RSS.2021.XVII.011. URL https://roboticsproceedings.org/rss17/p011.html.

JB Lanier, Kyungmin Kim, Armin Karamzade, Yifei Liu, Ankita Sinha, Kat He, Davide Corsi, and Roy Fox. Adapting world models with latent-state dynamics residuals. In *Proceedings of the 8th Annual Learning for Dynamics and Control Conference*, volume 331 of *Proceedings of Machine Learning Research*, pp. 117–144, 2026. URL https://proceedings.mlr.press/v331/lanier26a.html.

Kimin Lee, Younggyo Seo, Seunghyun Lee, Honglak Lee, and Jinwoo Shin. Context-aware dynamics model for generalization in model-based reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 5757–5766. PMLR, 2020. URL https://proceedings.mlr.press/v119/lee20g.html.

Lennart Ljung. *System Identification: Theory for the User*. Prentice Hall, 2 edition, 1999.

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

Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S. Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyztsoC5Y7.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=M992mjgKzI.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3803–3810, 2018. doi: 10.1109/ICRA.2018.8460528. URL https://arxiv.org/abs/1710.06537.

Kate Rakelly, Aurick Zhou, Deirdre Quillen, Chelsea Finn, and Sergey Levine. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International Conference on Machine Learning*, volume 97, pp. 5331–5340, 2019. URL https://proceedings.mlr.press/v97/rakelly19a.html.

Spencer M. Richards, Navid Azizan, Jean-Jacques Slotine, and Marco Pavone. Adaptive-control-oriented meta-learning for nonlinear systems. In *Proceedings of Robotics: Science and Systems*, 2021. doi: 10.15607/RSS.2021.XVII.056. URL https://www.roboticsproceedings.org/rss17/p056.html.

Reuven Y. Rubinstein and Dirk P. Kroese. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning*. Springer, 2004.

Vaisakh Shaj, Dieter Büchler, Rohit Sonker, Philipp Becker, and Gerhard Neumann. Hidden parameter recurrent state space models for changing dynamics scenarios. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=ds8yZOUsea.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, volume 119, pp. 9229–9248, 2020. URL https://proceedings.mlr.press/v119/sun20b.html.

Kaizhen Tan, Xin Xu, Siru Tao, Yixiao Li, Hanzhe Hong, Yang Feng, and Heqing Du. What can latent world models know? physical parameter identifiability in multimodal predictive representations. *arXiv preprint arXiv:2607.27017*, 2026. URL https://arxiv.org/abs/2607.27017.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uXl3bZLkr3c.

Ying Wang, Oumayma Bounou, Yann LeCun, and Mengye Ren. AdaJEPA: An adaptive latent world model. *arXiv preprint arXiv:2606.32026*, 2026. URL https://arxiv.org/abs/2606.32026.

Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/a1afc58c6ca9540d057299ec3016d726-Abstract.html.

Wenhao Yu, Jie Tan, C. Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. In *Proceedings of Robotics: Science and Systems*, 2017. doi: 10.15607/RSS.2017.XIII.048. URL https://roboticsproceedings.org/rss13/p48.html.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, volume 139, pp. 12310–12320, 2021. URL https://proceedings.mlr.press/v139/zbontar21a.html.

Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=-2FCwDKRREu.

Marvin Zhang, Sharad Vikram, Laura Smith, Pieter Abbeel, Matthew Johnson, and Sergey Levine. SOLAR: Deep structured representations for model-based reinforcement learning. In *International Conference on Machine Learning*, volume 97, pp. 7444–7453, 2019. URL https://proceedings.mlr.press/v97/zhang19m.html.

Xiangteng Zhang, Yang Guan, Bo Zhang, Hongyang Li, Ya-Qin Zhang, and Shengbo Eben Li. On the identifiability of controlled world models. *arXiv preprint arXiv:2607.22430*, 2026. URL https://arxiv.org/abs/2607.22430.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2411.04983.

A Theoretical Details and Proofs

Two channel-allocation constructions. Let a noiseless ordinary source history contain (m, q) with $q = c$, where $c \in \{0, \pm 1\}$, and let the successor be $y = m + gc$ for fixed $g \neq 0$. Controlled pairs use the same history $q = 0$ and outcomes $m, m + g\delta$.

Fixed response channel (predictor absorption). Set $F(m, q) = m + gq$ and $B(m, q) = g$ for every history. Ordinary predictions and controlled pairs are exact, but on every shifted ordinary prefix the residual is $r = y - F = 0$ although the response channel remains nonzero. A ridge fit from these ordinary residuals therefore returns zero. This is the fixed-b situation analyzed by Proposition 1: the predictor has absorbed the change, while B itself has not attenuated.

Learned response channel (channel attenuation). For $0 \leq \alpha \leq 1$, set $B(m, 0) = g$ for the controlled pairs, $B(m, q) = \alpha g$ on shifted ordinary histories, and $F(m, q) = m + (1 - \alpha)gq$. Ordinary predictions and pair fits remain exact, but the ordinary-prefix residual is $r = \alpha gc$, so the response information is reduced by α^2. At $\alpha = 0$, the predictor absorbs the change and the response channel disappears; for intermediate α, the residual and B are both nonzero but attenuated. This construction isolates the learned B allocation while showing how it can approach predictor absorption. Together the constructions show why the inverse loss must be evaluated through the deployed residual and response, rather than by prediction or pair fit alone.

Proof of Proposition 1. The estimate is $\widehat{c} = (1 - \beta)sc/(s + \lambda) + b^\top \Sigma^{-1}\epsilon/(s + \lambda)$. Independence removes the cross term and the numerator noise variance is s, so

$$\mathcal{R}(\beta) = \frac{\tau^2(\lambda + \beta s)^2 + s}{(s + \lambda)^2}.$$

Subtracting its value at $\beta = 0$ gives Equation (11).

Proposition 2 (Information controls coefficient risk)**.** *For fixed B_P and the Gaussian experiment $r_P = B_P c + \epsilon_P$, $c \sim \mathcal{N}(0, C)$, $\epsilon_P \sim \mathcal{N}(0, \Sigma_P)$ independently, with $C, \Sigma_P \succ 0$, define $\mathcal{I}_P = B_P^\top \Sigma_P^{-1} B_P$. The posterior mean and covariance are $m_P = V_P B_P^\top \Sigma_P^{-1} r_P$ and $V_P = (C^{-1} + \mathcal{I}_P)^{-1}$. The posterior mean's coefficient risk is $\mathrm{tr}(V_P)$, and its fixed-query risk is $\mathrm{tr}(B_Q V_P B_Q^\top)$. Both are nonincreasing when information increases in the positive-semidefinite order.*

Proof. Completing the Gaussian posterior square gives (m_P, V_P) (Bishop, 2006; Cover & Thomas, 2006). Conditional squared-error decomposition gives $\mathbb{E}[\|f - c\|^2 \mid r_P] = \|f - m_P\|^2 + \mathrm{tr}(V_P)$; replacing the norm by $\|B_Q(\cdot)\|$ gives the risk at a fixed query, bounded by $\|B_Q\|_{\mathrm{op}}^2 \mathrm{tr}(V_P)$. For added information $\Delta \succeq 0$, the covariance becomes $V_P^{1/2}(I + V_P^{1/2}\Delta V_P^{1/2})^{-1} V_P^{1/2} \preceq V_P$. Setting $C^{-1} = \lambda I$ recovers weighted ridge with the same prior. These calculations assume a design independent of c and independent noise. For learned designs, the likelihood and calibration must account for their dependence on history.

Proposition 3 (Prefix information and action contrasts)**.** *Fix c, B_P, and a finite set of affine candidate costs $\ell_u(v) = \alpha_u + j_u^\top v$, independently of estimation noise. Let $\mathcal{D}$ stack rows $(j_u - j_v)^\top$ for all candidate pairs, and let $e = \widehat{c} - c$ have mean μ_e and covariance V_e. Then*

$$\mathbb{E}\|\mathcal{D}e\|^2 = \|\mathcal{D}\mu_e\|^2 + \mathrm{tr}(\mathcal{D}V_e\mathcal{D}^\top) \leq \|\mathcal{D}\|_{\mathrm{op}}^2 \mathbb{E}\|e\|^2.$$

In the noiseless model $r_P = B_P c$, all cost differences are determined for every c exactly when $\ker B_P \subseteq \ker \mathcal{D}$. For unique minimizer $u_\star$, let $\Delta_u = \ell_u(c) - \ell_{u_\star}(c) > 0$ and $d_u = j_u - j_{u_\star}$. The probability of choosing another candidate is at most

$$\min\left\{1, \sum_{u \neq u_\star} \frac{(d_u^\top \mu_e)^2 + d_u^\top V_e d_u}{\Delta_u^2}\right\}.$$

Proof. Cost-gap errors equal $\mathcal{D}e$; expand their second moments and apply the operator-norm bound. Two coefficients agreeing on the noiseless prefix differ by an element of $\ker B_P$; all cost gaps agree precisely when that element lies in $\ker \mathcal{D}$. A competing candidate can displace $u_\star$ only if $|d_u^\top e| \geq \Delta_u$. The second-moment inequality and a union bound give the probability bound.

For learned rollout costs, Appendix C.4 uses local coefficient gradients through the imagined history and reports simulator- scored ranking and regret.

Lemma 1 (Decision-relevant prefix ambiguity)**.** *Let P_i be the law of all information available to a selector under setting $i \in \{0,1\}$. For two fixed candidate actions, assume $J_0(u_1) - J_0(u_0) \geq \Delta$ and $J_1(u_0) - J_1(u_1) \geq \Delta > 0$. Any randomized prefix-based selector $\widehat{u} \in \{u_0, u_1\}$ satisfies*

$$\max_i \mathbb{E}_i[J_i(\widehat{u}) - J_i(u_i)] \geq \tfrac{\Delta}{2}\big(1 - \mathrm{TV}(P_0, P_1)\big).$$

Proof. For $q(x) = \Pr(\widehat{u} = u_1 \mid x)$, the sum of error probabilities is $1+\mathbb{E}_0 q - \mathbb{E}_1 q \geq 1 - \mathrm{TV}(P_0, P_1)$. Each error costs at least Δ; the maximum risk is at least the average risk. This is the two-point testing bound applied to action choice. For every prefix length at which the complete laws coincide, the bound remains $\Delta/2$. The native-contact protocol therefore measures both prefix responses and changes in candidate ordering.

B Implementation Protocol

Algorithm 1 Fixed-prefix RPR: train recovery, then reuse the estimate during planning

1: Warm up E, F on nominal episodes; initialize B with controlled pairs.
2: **for** each source update **do**
3: Sample ordinary prefixes, controlled pairs, and independent data for residual variances.
4: Form spatial r_P, B_P and diagonal $\widehat{\Sigma}$; solve for $\widehat{c}_P$.
5: Update E, F, B through Equation (7), including the solve.
6: **end for**
7: Freeze networks; calibrate variances and covariance scales on source episodes.
8: Encode the target prefix and infer $(\widehat{c}_P, \widetilde{V}_P)$.
9: **for** each CEM candidate and imagined step **do**
10: Reevaluate F, B and apply the same $\widehat{c}_P$ (fixed draw for RPR-P).
11: **end for**

B.1 Spatial architecture and optimization

Images are resized to 224^2, with pixel values scaled to $[0,1]$; we apply no crops or flips. ViT-Tiny/14 has 12 blocks, three heads, and width 192 (Maes et al., 2026). A shared linear projection and channelwise batch normalization map its 16×16 patch grid to 32 channels per location; the class token is discarded. The predictor embeds these tokens to width 384 and applies six factorized blocks, each with within-frame spatial attention and past-only temporal attention at each location (16 heads, feed-forward width 1536). Spatial and temporal embeddings and action-conditioned adaptive layer normalization encode location, time, and commands. History length is $L = 3$; a shared output projection produces $F_n \in \mathbb{R}^{32}$. A two-layer GELU MLP of width 256 maps each final contextual token to $B_n \in \mathbb{R}^{32\times p}$; its last weights have standard deviation 10^{-3}, zero bias, and no column normalization. Source c enters G only through the explicit response multiplier.

We set $(\lambda_{\mathrm{SIG}}, \lambda_{\mathrm{ctrl}}, \lambda_{\mathrm{inv}}) = (0.1, 0.25, 0.05)$ and $\lambda = 2$. For the gain–drag tests, $C = 0.5I$ matches the coefficient covariance of the balanced 25-setting grid before nominal oversampling. We test $\lambda \in \{0.01, 0.1, 1, 2, 10\}$; any selected alternative uses source validation and is reported separately. SIGReg uses 1,024 random channel projections and 33 trapezoidal nodes on $[-4, 4]$, averaged over spatial and relative temporal positions. We warm up E, F on nominal data for 20,000 updates, train B on controlled pairs with the backbone fixed for 10,000, then train jointly for 40,000, giving a total of 70,000 updates; peak learning rates are $3 \times 10^{-4}, 10^{-3}, 10^{-4}$. Each stage resets AdamW: betas $(0.9, 0.95)$, matrix weight decay 0.05, gradient clipping 1, 5% linear warm-up, and cosine decay to 10^{-6}. Joint batches contain 64 clips of eight transitions (half nominal), 256 pairs, 32 ordinary prefixes with $T \sim \mathrm{Uniform}\{2, 4, 8, 16, 32\}$, and 512 independent weighting transitions. Each clip or prefix includes $L - 1$ preceding context frames, excluded from the losses. When varying L, we pad histories to 16 positions and mask unused positions, keeping tensor shapes and updates fixed. Dropout is disabled; normalization statistics freeze after warm-up, while affine parameters remain trainable. We use the final checkpoint.

B.2 Source data, weighting, and calibration

Each gain–drag task supplies 2,000 nominal and 2,500 controlled 64-transition episodes (100 per cell of $c_j \in \{-1, -0.5, 0, 0.5, 1\}$). Dynamics, initialization, exploration, and rendering are independently sampled. Stratifying by episode and scene seed gives 3,600/450/450 episodes for training/validation/calibration; replays retain their split. Another 24,000 controlled pairs, 3,000 per $\delta \in \{\pm 0.5, \pm 1\}e_j$, $j = 1, 2$, are split 19,200/2,400/2,400. Each pair restores state and repeats an action with shared randomness and the original nominal history. Ordinary prefixes for inverse training have no paired input. Matched baselines share these data; weakly excited episodes are retained. Appendix C.2 defines the native-contact source distribution.

For an independent batch with equal numbers of nominal and shifted transitions, define $e_j = E(x_{j+1}) - G(H_j, a_j; c_j)$, $u_k = n^{-1} \sum_j e_{j,k}^2$, $\bar{u} = d^{-1} \sum_k u_k$, and $v = (nd)^{-1} \sum_j \|z_j - \bar{z}\|^2$. Use

$$s_k^2 = 0.9u_k + 0.1\bar{u} + 10^{-3}v, \qquad \widehat{\Sigma} = \mathrm{diag}(s_1^2, \ldots, s_d^2). \tag{12}$$

Training differentiates through the moments and the solve (Bertinetto et al., 2019). We use 225 episodes for variances and 225 for coverage, then freeze calibration. For each T, use one coverage prefix to evaluate $q_i = (c_i - \widehat{c}_i)^\top V_i^{-1}(c_i - \widehat{c}_i)$. With $m = 225$, we set $\gamma_T = \max\{1, q_{(\lceil 0.9(m+1) \rceil)}/\chi^2_{p,0.9}\}$, where $q_{(j)}$ is the jth order statistic. This calibrates the scale of the 90% coefficient ellipsoid on source episodes. Another 225 source episodes, generated from disjoint seed families after model selection, assess source coverage independently of this calibration; they are excluded from training. Table 14 compares held-out source coverage with target interpolation, weak excitation, and extrapolation. Table 15 records the selected response and inverse weights.

B.3 Deployment and posterior planning

We divide each row of B, r by s_k, then accumulate and solve the symmetric $p \times p$ system in float64; neural operations use float32. Nonfinite inputs, zero latent variance, or failed factorizations stop training; deployment failures use $c = 0$ and remain in all metrics. With $L = 3$, the T-transition prefix contains $T + L$ frames and $T + L - 1$ commands, including its history. The point planner fixes $\widehat{c}$ and recomputes F, B at every imagined step.

CEM minimizes $\|\widehat{z}_{t+10} - E(x_g)\|^2/d$: 512 candidates, five iterations, 64 elites, initial action mean zero/standard deviation one, clipping to environment action bounds, 0.1 old/0.9 elite updates, and standard-deviation floor 0.05. We execute the first action of the final mean sequence, then shift it and append a neutral action. RPR-P uses four shared coefficient samples per planning call, fixed within each rollout, and mean cost plus 0.5 sample standard deviation. Neither fitted coefficients nor Gaussian samples are clipped. A point-planner baseline uses 2,048 candidates and 256 elites. Table 3 compares end-to-end fitting and planning costs; Table 14 compares candidate and rollout budgets.

RPR-P uses exactly the same fitted mean and covariance as point RPR; only candidate scoring changes. Sharing coefficient samples reduces comparison noise across candidates; separate planner seeds quantify variation between planning calls. The 2,048-candidate point baseline uses the same number of model transitions as RPR-P, both over five CEM iterations.

C Evaluation protocol

C.1 Tasks, target conditions, and paired comparisons

The three gain–drag comparisons use $g = 1 + 0.2c_g$ and $b = 0.5 + 0.25c_b\,\mathrm{s}^{-1}$, with native physics fixed. Added drag is applied at each substep, separately from contact friction. Source datasets are specified in Appendix B.2; target reset, action, and rendering seeds are disjoint from the source seeds.

Task wrappers. Planar drag integrates $\dot{q} = v$, $\dot{v} = ga - bv$ by semi-implicit Euler at 0.01 s with five substeps per observation, unit mass, $a \in [-1, 1]^2$, and reflecting walls at $[-2, 2]^2$. Resets sample $q \in [-1, 1]^2$, $v \in [-0.1, 0.1]^2$; commands change every four observations. Success requires position error and speed below 0.05 for five observations. Push-T maps commanded pusher position u to $p_{\text{eff}} + g(u - p_{\text{eff}})$ before clipping, and applies $f = -bmv$, $\tau = -bI\omega$ to the pushed body; success is goal IoU ≥ 0.95. Cube uses `visual-cube-single-v0` (Park et al., 2025), five balanced goals, 64×64 renders upsampled to the shared 224^2 encoder input, gain on the first four actions,

and unchanged gripper commands. Drag uses world-frame inertia; physics/observation intervals are $0.002/0.05$ s. Success is cube–goal distance ≤ 0.04 m. All control evaluations allow 200 steps after the prefix.

Target conditions. Nominal episodes use $c = 0$. The interpolation grid contains all combinations of $c_j \in \{-0.75, -0.25, 0.25, 0.75\}$. Extrapolation includes single-coordinate offsets of ± 1.5 and all four corners $(\pm 1.5, \pm 1.5)$. Weak excitation scales prefix action deviations by 0.1 while retaining the interpolation settings. Appearance shifts combine interpolation with brightness $0.7/1.3$, image-noise SD $0.02/0.05$, and disjoint textures. Nonlinear drag adds $f_{\text{extra}} = -\eta m \|v\| v / v_{\text{ref}}$, $\eta \in \{0.25, 0.5\}\,\text{s}^{-1}$, where v_{ref} is the median nonzero nominal-source speed. Settings are balanced within each test group. The nonlinear-drag comparison reports control success because the two coefficients cannot fully describe the changed dynamics.

Shared scenarios. Each scenario stores a 47-transition lead-in, an endpoint snapshot, a goal, future commands, and simulator states. For each T, we reveal the last T transitions plus the required history; shorter prefixes hide earlier observations. Every method starts from the same endpoint. When varying L, we use a shared $T + 15$ buffer and the same fitted residuals, masking unused history positions. Forecasts replay common actions; control uses shared snapshots and planner seeds.

C.2 Native contact dynamics

Four physical factors. Native-Cube uses OGBench Cube with its original action interface and 25 MuJoCo substeps per 0.05-second control step. The injected gain and drag are disabled. Set $c_j = \log_2(\vartheta_j / \vartheta_{0,j})$ for cube mass, table friction, finger friction, and arm servo stiffness. Mass and principal inertia scale together. An explicit floor–cube pair sets both tangential friction coefficients, preserving the original contact dimension, compliance, margin, and other friction components. The four finger-pad sliding coefficients share one multiplier and retain their contact priority. Arm stiffness scales both position gain and position bias for six actuators, preserving velocity feedback and force limits. Their nominal gains are $[4500, 4500, 4500, 2000, 2000, 500]$ N m rad^{-1} in shoulder-pan, shoulder-lift, elbow, and wrist-1/2/3 order. These changes use native simulator fields; $F + Bc$ remains a learned local approximation across contact modes.

Source and target settings. For each nested family—mass only ($p = 1$), mass plus table friction ($p = 2$), and all factors ($p = 4$)—use 2,000 nominal and 2,500 shifted episodes with active $c_j \sim \text{Uniform}[-1, 1]$. Keep the 3,600/450/450 split, 24,000 axis-aligned controlled pairs, 70,000 updates, and source-only hyperparameter selection. Balance pairs over $\delta \in \{\pm 0.5, \pm 1\} e_j$; inactive factors stay nominal. Single-factor and joint interpolation use fresh settings inside this support. Single-factor extrapolation uses $c_j = \pm 2$, giving quarter or fourfold native values. A separate $p = 4$ composition experiment trains, validates, and calibrates shifted episodes only where $\prod_j \text{sign}(c_j) = 1$; targets use the opposite sign parity. Axis-only pairs remain shared, and all eight withheld joint orthants stay outside the ordinary source pools.

Which observations support which actions? Compare RPR, no-inverse training, direct parameter conditioning, fixed context, and true-c conditioning; the continual track adds AdaJEPA and LoRA with matched real-feedback access. Report coordinatewise and aggregate coefficient RMSE, candidate ranking, selected-candidate regret, forecast error, success, and measured latency. In each phase, true-c comparisons assess estimation sensitivity while retaining representation and rollout errors. Table 4 reports success for the nested families and the composition split, and Table 5 reports the $p = 4$ coefficient RMSE per coordinate: RPR recovers every coordinate better than the parameter-supervised solve, with mass and stiffness the best-resolved factors and the two friction coefficients, which only contact-rich prefixes excite, the least.

Joint-response nonadditivity. From the same integration state and command sequence, record the cube and pinch-site positions $s_k(c)$. Define

$$\nu_h(c) = \frac{\|s_{1:h}(c) - s_{1:h}(0) - \sum_j [s_{1:h}(c_j e_j) - s_{1:h}(0)]\|_{\text{RMS}}}{\|s_{1:h}(c) - s_{1:h}(0)\|_{\text{RMS}} + 10^{-4}\,\text{m}}.$$

The RMS averages over h times and six Cartesian coordinates. Stratify joint-shift errors by ν_h using source-fixed bins; true-c prediction tests response-model accuracy within each group. Simulator state and contact metadata are offline diagnostics; deployed models receive images and recorded actions.

Table 4: Native-Cube learned-model comparison: success (%, ↑) with 95% intervals. Nested families vary mass ($p = 1$), mass/friction ($p = 2$), then four native factors ($p = 4$). Composition withholds eight joint orthants.

Method	$p = 1$	$p = 2$	$p = 4$	Composition
Forward-pair	49.8±20.6	41.2±19.4	30.6±16.6	23.4±14.2
Param. + ridge	57.4±18.4	48.8±19.8	37.6±17.4	29.8±16.2
Fixed context	60.0±19.4	53.2±22.0	43.4±20.8	35.8±16.8
TTT-LoRA	63.8±20.2	57.4±19.6	49.6±22.4	44.8±18.4
RPR	65.2±19.8	58.6±19.4	49.0±20.2	39.8±17.4

Table 5: Native-Cube $p = 4$ coefficient RMSE. Coordinates use the source log-scale. Aggregate RMSE is the square root of the mean coordinate MSE.

Method	Mass	Table friction	Finger friction	Stiffness	Aggregate
Forward-pair	0.480±0.025	0.660±0.024	0.700±0.024	0.480±0.018	0.589±0.008
Param. + ridge	0.380±0.012	0.570±0.026	0.580±0.017	0.400±0.025	0.491±0.005
RPR	0.290±0.013	0.460±0.021	0.470±0.022	0.340±0.019	0.398±0.007

Table 6: Native-parameter implementation audit. Position-response RMS in mm (mean ± sample SD, three seeds) after halving one factor. Replays share the initial integration state and 12 actions over 0.6 s; responses cover cube and pinch-site positions. Phases denote the initial snapshots.

Factor	Nominal value	Free-space start	Table slide	Grasp start
Cube mass	79.36 g	< 0.001	< 0.001	0.704 ± 0.487
Table friction	1.0	< 0.001	$2.719 \pm <0.001$	0.484 ± 0.236
Finger friction	0.7/0.6	< 0.001	< 0.001	0.612 ± 0.389
Arm stiffness	4500/2000/500	28.527 ± 2.748	0.868 ± 0.115	31.632 ± 1.193

Count numerical warnings, saturation, task failures, and invalid resets separately, retaining every preregistered setting in the report.

Implementation checks. The implementation audit uses MuJoCo 3.3.3 and OGBench commit `1d4140997f60`. It compares explicit-pair and original contacts at nominal settings and checks restoration after parameter changes. Mass updates recompute constants on temporary simulator data; replays restore the complete integration state. Recorded contact coefficients verify that friction changes reach the intended contacts. Across 15 matched nominal replays, explicit-pair and original-contact trajectories agree exactly; restoring nominal parameters also reproduces the trajectories. The 150 stored nominal/half/double rollouts and 135 additional quarter/fourfold rollouts remain finite with no new numerical warnings. In the half-scale checks of Table 6, mass barely affects unforced sliding, while stiffness dominates the position metric. The audit establishes factor effectiveness in these sampled interactions. The all-double grasp response has mean $\nu_1 = 0.415$ and $\nu_{12} = 0.038$; native variation therefore need not produce large multistep interactions. The learning protocol tests whether visual calibration exploits these responses.

C.3 Baselines and estimator controls

Shared training information. Matched models share labeled episodes, pairs, and updates. Forward-pair training omits $\mathcal{L}_{\text{inv}}$; ridge and direct-head variants condition $F + B\hat{c}$ throughout imagination. Source validation selects the controlled-response and inverse weights from the two-range sweep summarized in Table 15. Models without a deployed response channel fit an auxiliary B for 10,000 updates before jointly training prediction and pairs. The auxiliary head is used only for source training; deployment uses the response channel.

Context and long-history models. Fixed and recurrent context use two Transformer layers, width 192, three heads, and context width selected from $\{2, 16, 32\}$, with or without coefficient supervision. A linear projection and adaptive normalization condition the shared predictor. Fixed context summarizes the prefix once; recurrent context updates as history advances. Its frozen-context control holds

Table 7: Checkpoint-matched coefficient interventions. Physical NRMSE (↓), mean ± 95% intervals. Estimated and true coefficients use identical networks, future actions, and readouts.

Task / training	h=10, $\hat{c}$	h=10, c	h=20, $\hat{c}$	h=20, c
Planar drag / Forward-pair	0.197 ± 0.005	0.119 ± 0.003	0.333 ± 0.006	0.215 ± 0.005
Planar drag / Param. + ridge	0.154 ± 0.004	0.105 ± 0.003	0.248 ± 0.005	0.186 ± 0.004
Planar drag / RPR	0.087 ± 0.002	0.071 ± 0.002	0.132 ± 0.003	0.105 ± 0.002
Push-T / Forward-pair	0.263 ± 0.006	0.165 ± 0.004	0.437 ± 0.011	0.278 ± 0.006
Push-T / Param. + ridge	0.219 ± 0.006	0.146 ± 0.004	0.361 ± 0.008	0.241 ± 0.006
Push-T / RPR	0.126 ± 0.003	0.101 ± 0.002	0.199 ± 0.005	0.158 ± 0.003
Cube / Forward-pair	0.277 ± 0.006	0.199 ± 0.005	0.455 ± 0.012	0.337 ± 0.006
Cube / Param. + ridge	0.231 ± 0.006	0.176 ± 0.003	0.374 ± 0.008	0.306 ± 0.008
Cube / RPR	0.163 ± 0.004	0.136 ± 0.004	0.266 ± 0.006	0.222 ± 0.005

the same checkpoint's initial estimate. The long-history model attends to the full prefix, replacing the oldest tokens during imagination. All share the training budget.

Test-time gradient adaptation. TTT freezes E and updates the mixed-dynamics predictor F, or rank-8 LoRA in its attention and feed-forward maps. The loss combines prefix next-latent MSE and squared weight deviation. LoRA initializes one factor randomly and the other at zero, penalizing the induced weight increment. Source validation selects rates $\{1, 3, 10, 30\} \times 10^{-5}$ and anchor weights $\{10^{-4}, 10^{-3}, 10^{-2}\}$. Use 20 AdamW steps by default; sweep 1, 5, 10, 20, and 50, resetting each episode. A 100-ms variant returns the source model on timeout and reports timeout and success rates.

True-coefficient reference. True-c conditioning retains each row's learned networks, checkpoint, and CEM budget and replaces the estimate by the true coefficient, so the estimated-to-true gap measures sensitivity to coefficient-estimation error while representation and rollout errors remain. Table 7 reports the intervention at two horizons on all three tasks: the ordering of the three training schemes is the same in every column, and RPR has both the smallest true-c error and the smallest gap.

C.4 Mechanism and transfer tests

From calibration to candidate actions. At each shared endpoint, fix 512 action sequences independently of model estimates, using clipped unit-Gaussian commands held for two steps. Use nested horizons $h \in \{1, 5, 10, 20\}$ and prefixes $T \in \{2, 4, 8, 16, 32\}$. Compare RPR, forward-pair ridge, direct parameter conditioning, and fixed context on the same candidates. Rank them by each model's terminal latent goal cost. Offline simulator replays score squared position error for planar drag and Cube, and $1 - \mathrm{IoU}$ for Push-T; these scores remain evaluation targets. Report pairwise ranking agreement (ties receive half credit) and selected candidate regret $J^{\mathrm{sim}}_{\widehat{u}} - \min_u J^{\mathrm{sim}}_u$.

For the persistence intervention, retain the fitted coefficient for the first L imagined steps, then set it to zero for the remaining rollout. A second intervention substitutes another episode's estimate, permuted within task and estimate-norm quartiles. Keep the recipient history, goal, checkpoint, candidate bank, and computation fixed. Prefix RMSE checks that calibration is unchanged; ranking and regret test its use after the observation window has been replaced. Contrasting full reuse with delayed erasure, while comparing matched forward-pair and parameter-supervised models, separates persistence from estimator-aligned training. Table 8 reports the Push-T results at $T = 8$ and $h = 20$: full reuse of the RPR estimate raises pairwise ranking agreement from 65.9% (forward-pair) to 78.5% and halves regret, erasing the coefficient once the history is model-generated removes most of the gain, and swapping in another episode's estimate removes all of it; the true coefficient bounds what a better estimate could still add.

Table 8: Push-T branch transfer at $T = 8, h = 20$. Pairwise ranking agreement (%, ↑) and selected-candidate regret (↓) use a shared 512-candidate bank.

Intervention	Ranking	Regret
Forward-pair	65.9 ± 0.2	0.138 ± 0.013
Param. + ridge	69.2 ± 0.2	0.105 ± 0.012
RPR / full reuse	78.5 ± 0.1	0.067 ± 0.006
RPR / erase after L	68.8 ± 0.3	0.113 ± 0.011
RPR / swap coefficient	64.9 ± 0.2	0.133 ± 0.016
RPR / true c	82.8 ± 0.2	0.051 ± 0.004

Absorption. Replay each shifted-history endpoint and action under nominal and changed dynamics with shared randomness. For u_t, Δ_t^* defined in the main text and $W = \widehat{\Sigma}^{-1}$, signed absorption is

$$A = \frac{\langle u_t, \Delta_t^* \rangle_W}{\|\Delta_t^*\|_W^2 + \eta}.$$

Set $\eta = \max(10^{-12}, 10^{-6} m_\Delta)$, with m_Δ the median positive source reference energy. This diagnostic compares the nominal prediction with a reference response independent of the learned B. We report it as predictor absorption A_F. We separately report response retention

$$K_B = \mathrm{clip}_{[0,1]} \left(\mathrm{median}_t \frac{\|B(H_t^{\mathrm{shift}}, a_t)c\|_W}{\|B(H_t^{\mathrm{nom}}, a_t)c\|_W + \eta} \right),$$

which uses the learned channel rather than the simulator reference. A small A_F can coexist with a small K_B; the anchored-F control is included precisely to make that distinction testable. The fixed-B proposition applies to A_F, whereas channel attenuation changes $\mathcal{I}_P$ through K_B. Nominal-only F freezes after 20,000 warm-up updates; the same total schedule trains the remaining components. Counterfactual anchoring adds the nominal-successor target on shifted histories, with its weight selected on the source validation split under the same 70,000-update budget. The anchored control and RPR use identical additional source replays. The gradient-stopping ablations retain prediction and controlled-response gradients.

C.5 Metrics, uncertainty, and reporting

Coefficient RMSE averages squared errors across the active source-scaled coordinates before taking the square root; per-coordinate RMSE values are also reported. Physical-state readouts use existing source splits (10,000/2,000/2,000 training/validation/check transitions), with ridge penalties selected from $10^{-5}, \ldots, 10^{2}$. Readouts use history for velocity, sine/cosine for angles, and rotation matrices for orientation. All physical coordinates are standardized by the same source-state SD, floored at 10^{-6}. NRMSE compares readouts of imagined histories with true states. Applying the same frozen readout to encoded true future frames measures its readout error at each horizon (Table 9).

For each target test group, five seeds and 100 scenarios per seed support 200 hierarchical bootstrap replicates: resample seed groups, then scenarios within each group. Resampling is performed separately for each reported metric; the 2.5/97.5 percentiles give marginal intervals. Nonfinite forecasts incur an error of ten standardized units. Failed or timed-out planners receive zero success. Timing comparisons summarize per-case latency distributions by their median and 95th percentile. Fitting includes data transfer, encoding, weighting, and solving or all gradient updates; planning includes all candidate rollouts. Table 3 also reports peak memory. Posterior coverage is the fraction inside the source-calibrated 90% ellipsoid, paired with mean area and coefficient RMSE; Table 14 compares calibration and planning.

Qualitative comparisons. Figures 4 and 3 show two Push-T rollouts and six planar-drag trajectory cases. In both figures, "LeWM w/o c" is the nominal predictor F with no coefficient and "LeWM with c" is the forward-pair model conditioned on its own ridge estimate; the Push-T panels retain their native checkpoint times and terminal steps, and the drag panels retain pixel coordinates and the displayed target region.

Figure notes. Figures 1–2 use schematic elements to keep the mechanism readable at paper scale. The trajectory panels in Figures 4 and 3 are vector exports that retain their native labels, endpoint

Table 9: Physical readout and forecast errors (NRMSE, ↓). Observed errors use encoded ground-truth frames; imagined errors use model rollouts. Both use the same source-trained physical-state readout.

Task / method	Observed $h = 1$	Observed $h = 20$	Imagined $h = 20$
Planar drag			
Forward-pair	0.021	0.022	0.333
Param. + ridge	0.020	0.021	0.248
RPR	0.020	0.022	0.132
Push-T			
Forward-pair	0.031	0.033	0.437
Param. + ridge	0.031	0.033	0.361
RPR	0.030	0.032	0.199
Cube			
Forward-pair	0.039	0.041	0.455
Param. + ridge	0.039	0.042	0.374
RPR	0.040	0.042	0.266

Table 10: Planar drag: success (%) for every target profile, with 95% bootstrap half-widths.

Method	Nominal	Interp.	Extrap.	Weak	Appearance	Nonlinear
Forward-pair	97.4±2.4	70.8±12.2	52.0±12.4	64.0±11.0	53.6±10.4	50.4±13.8
Param. + ridge	96.0±2.6	79.0±7.8	55.8±12.0	69.4±11.2	61.2±11.2	52.8±11.8
Param. + head	96.2±3.0	82.2±7.2	59.4±12.4	71.2±10.0	64.6±11.2	55.4±13.0
Fixed context	96.8±3.0	84.6±8.0	61.0±11.2	74.4±8.8	69.2±9.4	58.2±13.0
Recurrent ctx.	95.4±2.6	80.4±9.4	58.2±10.8	71.4±10.0	66.0±10.8	57.0±12.6
Long history	96.4±2.8	78.4±7.2	57.0±11.2	70.0±11.2	62.8±12.2	57.4±11.4
TTT-F	96.2±2.6	90.4±5.6	72.0±8.6	78.4±9.0	75.4±10.4	75.2±9.4
TTT-LoRA	95.6±2.6	88.8±5.4	73.8±10.8	76.8±10.0	76.2±8.4	77.6±8.0
RPR	97.0±2.4	92.2±5.4	73.0±9.6	72.4±8.8	77.0±9.2	65.8±10.2

Table 11: Push-T: success (%) for every target profile, with 95% bootstrap half-widths.

Method	Nominal	Interp.	Extrap.	Weak	Appearance	Nonlinear
Forward-pair	89.0±6.8	40.4±13.8	24.2±7.8	31.4±11.6	24.8±12.0	21.4±11.0
Param. + ridge	88.4±7.8	51.4±12.2	32.0±12.6	41.0±12.6	36.4±15.4	27.2±10.6
Param. + head	88.0±7.0	55.2±13.0	36.4±11.6	44.2±13.4	40.0±13.6	31.6±11.2
Fixed context	89.8±5.8	61.0±14.8	42.4±14.0	51.4±13.0	50.6±15.2	38.8±13.8
Recurrent ctx.	88.6±7.0	49.2±12.0	35.8±15.8	42.4±11.8	40.2±12.4	32.6±13.2
Long history	88.2±8.2	52.2±15.0	35.6±13.2	43.8±15.0	41.8±14.6	34.4±11.6
TTT-F	86.4±7.4	68.2±14.4	55.2±13.8	54.8±13.2	58.2±13.4	59.4±13.8
TTT-LoRA	85.8±8.0	69.6±12.4	56.0±13.8	56.0±13.8	61.6±15.0	61.8±13.2
RPR	87.8±6.6	78.8±12.0	57.8±13.4	54.0±13.2	64.8±13.6	51.4±12.8

conventions, and pixel coordinates; they serve as qualitative case studies, while the quantitative tables and intervals follow the aggregate protocol above.

D Additional experimental comparisons

The following fixed-prefix comparisons use the splits, paired scenarios, and uncertainty estimates of Appendix C. Unless varied, $T = 8$, $L = 3$, and the point-estimate planner uses 512 candidates.

Tables 10–12 give the complete six-profile success matrices with intervals for every fixed-prefix method; the all-six column of Table 1 is their row-wise mean. Table 13 varies the number of gradient steps for the two test-time-training baselines and reports success under a 100-ms fitting budget, which is the regime of Section 4.

Table 12: Cube: success (%) for every target profile, with 95% bootstrap half-widths.

Method	Nominal	Interp.	Extrap.	Weak	Appearance	Nonlinear
Forward-pair	79.2±8.6	33.8±12.0	23.6±9.6	28.4±9.6	23.2±10.2	15.0±7.8
Param. + ridge	79.0±8.4	46.8±12.6	32.6±8.8	38.6±10.4	34.8±10.8	22.2±9.6
Param. + head	79.4±8.0	50.6±10.0	35.4±10.2	40.2±11.6	38.8±10.8	25.6±10.2
Fixed context	80.8±6.4	55.0±8.8	38.0±9.2	45.4±10.8	45.8±11.4	30.8±10.4
Recurrent ctx.	78.2±7.8	43.8±12.0	31.6±8.8	38.0±9.8	35.4±11.4	25.8±11.2
Long history	78.8±6.8	45.6±13.8	32.4±11.6	39.0±11.2	36.6±12.0	27.4±9.0
TTT-F	78.8±10.0	69.4±9.4	57.6±9.6	51.8±10.4	58.4±10.6	54.6±10.2
TTT-LoRA	78.4±8.4	65.0±9.6	58.8±9.2	53.0±10.6	59.6±9.2	56.8±10.8
RPR	79.6±6.6	70.8±9.0	53.2±9.8	47.8±12.0	59.0±9.2	45.0±12.4

Table 13: Push-T gradient-update sweep. Timeout is the percentage of fits exceeding 100 ms; capped success returns the frozen source model on timeout.

Method / steps	Success	Median ms	Timeout %	Capped success
TTT-F, 1	54.4±21.2	22.5	0.0±0.0	54.4±21.2
TTT-F, 5	62.2±22.6	77.5	4.2±2.0	61.0±22.4
TTT-F, 10	66.0±20.8	148.1	99.2±1.0	41.2±13.8
TTT-F, 20	68.2±21.0	284.7	100.0±0.0	40.4±13.8
TTT-F, 50	68.8±20.2	704.4	100.0±0.0	40.4±13.8
TTT-LoRA, 1	53.6±19.6	16.9	0.0±0.0	53.6±19.6
TTT-LoRA, 5	60.4±21.8	49.1	0.0±0.0	60.4±21.8
TTT-LoRA, 10	66.8±19.0	89.5	24.2±5.2	62.2±17.0
TTT-LoRA, 20	69.6±20.0	170.2	100.0±0.0	40.4±13.8
TTT-LoRA, 50	70.4±20.8	404.4	100.0±0.0	40.4±13.8

Table 14: Push-T posterior calibration and planning. Coverage and mean ellipsoid area assess the coefficient posterior. Planning comparisons fix either candidate count or total rollout count.

Evaluation data	90% coverage (%)	Ellipsoid area	Coeff. RMSE
Held-out source	90.4	0.880	0.206
Target interpolation	90.2	0.963	0.218
Target weak excitation	86.0	3.890	0.469
Target extrapolation	75.4	1.667	0.387

Planner	Candidates	Draws	Weak success	Extrap. success	Time (ms)
RPR	512	1	54.0	57.8	71.8
RPR	2,048	1	59.2	61.8	246.3
RPR-P	512	4	62.8	60.8	247.1

Table 15: Push-T source-only response-weight search. The selected pair is shared by the two shift ranges reported here.

Shift range	λ_{ctrl}	λ_{inv}	Best success (%)
±1	0.25	0.05	87
±1.5	0.25	0.05	85

E Additional related work

Control representations. Structured dynamics (Watter et al., 2015; Zhang et al., 2019), transition uncertainty (Chua et al., 2018), feature geometry (Bardes et al., 2022; Zbontar et al., 2021), and behavioral abstractions (Gelada et al., 2019; Zhang et al., 2021) complement RPR's objective of estimating dynamics changes from observations.

Identification and adaptation. System identification requires informative observations (Ljung, 1999); randomization and latent context model environment variation (Peng et al., 2018; Killian et al., 2017; Rakelly et al., 2019). TTT and Tent update network parameters (Sun et al., 2020; Wang et al.,

Table 16: Adaptation protocols. A replan follows one executed model-step action in the matched-backbone setting. Only real observations enter updates.

Method	Target observations	Updated quantities / schedule
Prefix TTT	Initial prefix	F or LoRA / 20 steps once
Fixed-prefix RPR	Initial prefix	c / one solve
Continual TTT-F	Prefix + feedback	F / one step per replan
Continual LoRA	Prefix + feedback	Rank-8 maps / one step per replan
AdaJEPA protocol	Prefix + feedback	E head, final F block / per replan
Continual RPR	Prefix + feedback	c / one solve per replan
ReDRAW-style residual	Prefix or offline set	Residual head / gradient updates
ReDRAW (original)	Offline target set	Residual + actor–critic / offline

Table 17: Continual Push-T: success (%) under affine interpolation and nonlinear drag, and cumulative adaptation seconds over 200 actions. Ports share the backbone and planner.

Method	Interp.	Nonlinear	Adapt. seconds
Fixed-prefix RPR	78.8 ± 12.0	51.4 ± 12.8	14.0 ± 1.5
Continual RPR	85.4 ± 11.8	70.2 ± 18.8	18.0 ± 0.9
Continual TTT-F	86.8 ± 11.4	77.4 ± 14.8	112.0 ± 6.6
Continual LoRA	87.2 ± 11.6	78.2 ± 17.6	102.0 ± 6.9
AdaJEPA-style port	86.2 ± 11.6	77.8 ± 14.8	94.0 ± 7.7
ReDRAW-style online	85.0 ± 11.6	77.6 ± 17.0	104.0 ± 7.5

2021); RPR instead fits source-defined coefficients. AdaJEPA's authors suggest that history-based adaptation contributes to strong frozen-model results (Wang et al., 2026), motivating absorption tests.

Differentiable estimation. RPR uses differentiable optimization (Amos & Kolter, 2017; Agrawal et al., 2019) to align source-trained representations with the deployed coefficient estimator.

F Continual and offline adaptation comparisons

Continual control. Start from $T = 8$ real transitions, retain the latest $K = 8$, and update after each executed action. Match CEM, goals, seeds, and the 200-action budget; report success, cumulative adaptation time, and replan latency.

AdaJEPA and gradient controls. The port updates the final predictor block and encoder projection with one Adam step on stop-gradient targets, using source-selected rates and the same window, cadence, backbone, and CEM budget as continual TTT-F and LoRA (Wang et al., 2026). Remaining weights and normalization statistics stay frozen; episode weights reset between trials.

Continual coefficient fitting. Solve Equation (5) on each real window with the source prior; freeze E, F, B, W during rollouts and compare fixed and continual fits on stress profiles. Table 17 reports Push-T success under interpolation and nonlinear drag together with the cumulative adaptation time over the 200-action episode. Refitting the coefficient after every executed action improves both profiles for four additional seconds per episode, whereas the gradient-based ports spend 94–112 seconds; their advantage under nonlinear drag is the misspecification margin discussed in Section 4.

Residual adaptation and offline transfer. The ReDRAW-style control adds a zero-initialized width-128 residual to F; E, F remain frozen and the residual receives the same 20-step budget. Offline comparisons use 1,000, 10,000, or 40,000 disjoint target transitions, while the original ReDRAW protocol is reported separately with its actor–critic access pattern (Lanier et al., 2026). Pooled RPR solves Equation (5) once on the whole offline set with the same frozen networks. Table 18 shows that the two routes coincide at 1,000 transitions; the residual head keeps improving with more target data, which a two-coordinate estimate cannot use, whereas RPR reaches its plateau from a prefix-sized sample.

Table 18: Push-T offline adaptation success (%). Both methods receive the same condition-specific, disjoint target transitions. Collection cost is additional to fitting.

Method	1,000 transitions	10,000	40,000
Pooled RPR	78.6 ± 16.4	80.4 ± 13.4	80.8 ± 15.2
ReDRAW-style offline	78.2 ± 16.0	83.4 ± 14.0	86.2 ± 11.8