CORRECT ACTION TIMING IS NOT ACTION FIDELITY IN WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

World models used for planning must predict how changing an action changes the future. Under non-anticipation and a nonzero response moment, we prove that an external randomized code identifies action timing from its rightmost nonzero code-transition moment. Even with exact timing, a restricted predictor at the population-risk optimum can estimate action effects less accurately than predicting no effect. An exact paired-risk identity gives an architecture-independent sufficient prediction-accuracy condition at the action-effect scale. Controlled visual experiments recover all 18 offset families, while only 7 of 36 aligned models beat zero response after 12 epochs. This distinction persists with longer controlled training and larger models on public Walker/Walk. Response weighting, motivated by the identity, improves eleven of twelve controlled comparisons and both Walker architectures under matched labels.

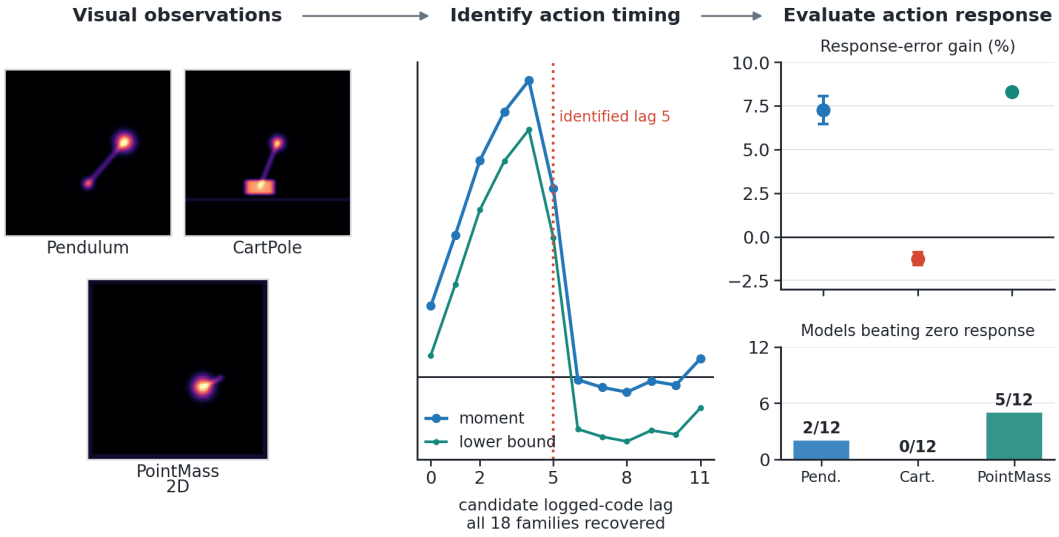

Figure 1: Correct timing and learned response are separate tests. Three visual systems supply transitions for registration. The rightmost positive simultaneous lower bound recovers the lag in all 18 families. At 12 epochs, alignment improves two systems but degrades CartPole; only 7 of 36 aligned models beat zero response. Error bars give 95% confidence intervals.

1 INTRODUCTION

World models are increasingly used to choose actions by predicting their consequences, from compact latent planning to interactive visual simulation (Yang et al., 2023; Bruce et al., 2024; Zhou et al., 2025; Alzayer et al., 2026). This use requires accuracy in the difference between possible futures. A model can predict the dominant motion well while misrepresenting the smaller change caused by an action. That distinction matters whenever predicted futures inform an action choice.

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.

A world model needs both a meaningful action input and an accurate response to it. We call the correct assignment of an action to its transition *temporal correspondence*, and the accuracy of the predicted change under an action intervention *action fidelity*. Recent evaluations test action adherence, action-to-motion correspondence and matched counterfactual outcomes (Arai et al., 2024; Co et al., 2026; Zhang et al., 2026a). An unknown offset in an asynchronous action log makes temporal correspondence an identification problem. Once that problem is solved, what guarantees action fidelity?

A shifted action stream can be identified when the data-collection controller accepts an external randomized code. Earlier actions remain predictive under feedback, complicating input-delay estimation (Gabay & Merhav, 1976; Ljung, 1999; Gilson & Van den Hof, 2005). Active watermarking supplies independent excitation (Satchidanandan & Kumar, 2017). Our right-boundary theorem uses non-anticipation: future codes cannot affect the current transition. A nonzero current-code response identifies its lag as the rightmost nonzero moment, even when an earlier moment is larger.

Exact timing alone does not guarantee an accurate learned response. When one parameter controls both state and action terms, fitting the larger state component can reverse the smaller action response at the exact population squared-risk optimum. The failure needs neither finite data nor early stopping. It sharpens the distinction between prediction and control-relevant accuracy studied in value-aware and representation learning (Farahmand et al., 2017; Gelada et al., 2019; Grimm et al., 2020; Zhang et al., 2021; Lambert et al., 2020).

The sufficient condition concerns prediction error at the scale of the action effect. An exact identity separates paired prediction risk into shared error and one quarter of response error. Under our randomized action law, paired and factual population risks coincide. The sharp bound specifies what prediction accuracy suffices relative to the action effect. It also motivates response weighting, complementing counterfactual constraints and action-specific readouts (Zhang et al., 2026b; Shi et al., 2026; Li et al., 2026).

Experiments test these implications separately (Figure 1). Across three deterministic visual systems and two compact architectures, registration recovers every offset, but only seven of 36 aligned models beat zero response after 12 epochs. At 48 epochs, factual prediction beats frame persistence in eleven of twelve conditions, while response beats zero in only five. Response weighting improves eleven matched-label comparisons. Larger models on the public Walker/Walk task extend this test beyond the controlled systems.

2 Related Work

2.1 Action-conditioned world models

Action-conditioned prediction has progressed from Atari frames to interactive video (Oh et al., 2015; Bruce et al., 2024). Model-based learning connects prediction with planning and policy learning (Ha & Schmidhuber, 2018; Hafner et al., 2020; 2025). Value-aware and control-relevant objectives preserve information for specified rewards and policies (Farahmand et al., 2017; Gelada et al., 2019; Grimm et al., 2020; Zhang et al., 2021). Objective mismatch concerns how prediction losses serve a downstream decision (Lambert et al., 2020). We formalize an earlier requirement: predicting the difference between two action outcomes, before a reward or planner is chosen.

Recent world models also broaden the action interface. UniSim models driving scenes, while Genie learns interactive environments without recorded actions (Yang et al., 2023; Bruce et al., 2024). Video PreTraining recovers native action labels from video (Baker et al., 2022). Olaf-World aligns latent action semantics across contexts, while causally debiased latent action models reduce action-irrelevant visual bias (Jiang et al., 2026; Wei et al., 2026). Masked Visual Actions expresses control through a partially revealed visual trajectory (Alzayer et al., 2026). DINO-WM predicts visual features, while representation studies ask which features retain action information (Zhou et al., 2025; Yeom et al., 2026). These choices can aid learning without certifying a shifted log or its learned response.

2.2 Temporal delay and active identification

Delayed observations and actions change both learning and control. Closed-loop identification addresses recovery under feedback (Gabay & Merhav, 1976; Ljung, 1999; Gilson & Van den Hof, 2005), while dynamic watermarking tests whether a process follows its intended dynamics (Satchidanandan & Kumar, 2017). Reinforcement-learning methods model or compensate delayed feedback and actions (Walsh et al., 2007; Ramstedt et al., 2021; Chen et al., 2021). Our active estimator identifies a fixed action-log offset. We compare it with a raw-level peak and evaluate the finite model after registration.

2.3 Action fidelity and response objectives

Action fidelity is now evaluated separately from visual quality. ACT-Bench measures driving-action adherence, WorldSimProbe aligns rollouts for action-to-motion checks, and matched driving counterfactuals test episode-specific open-loop outcomes (Arai et al., 2024; Co et al., 2026; Zhang et al., 2026a). These evaluations do not estimate the fixed external-log offset considered here. For linear Gaussian latent dynamics with invertible nonlinear observations, identifiability theory relates counterfactual error to action excitation (Zhang et al., 2026b). CoCo uses counterfactual consistency constraints, while Dueling World Models subtracts action-mean predictions at readout (Shi et al., 2026; Li et al., 2026). We distinguish timing recovery from response accuracy, then test direct response supervision with matched labels.

3 From Temporal Identification to Action Fidelity

3.1 Logged action codes

We consider a visual process with dynamic observation X_t and a context C_t available before the current action code is sampled. Let $q_t = q(C_t) \in \{-1, +1\}$ be a bounded context sign, and let $E_t \in \{-1, +1\}^p$ be an independent Rademacher code. A feedback controller applies

$$U_t = \pi(S_t) + \epsilon q_t E_t, \tag{1}$$

where S_t contains the controller state and ϵ keeps actions in range. Transitions depend on past and current codes, never future codes. The codes affect dynamic observations only through the action. The logger has an unknown fixed coordinatewise offset d_j:

$$\bar{E}_{t+d_j,j} = E_{t,j}. \tag{2}$$

Unoccupied logger positions contain independent codes from the same distribution. Thus padding statistics do not reveal the offset.

The response statistic uses an observable temporal increment. For a bounded feature map ϕ, define

$$D_t = \phi(X_{t+1}) - \phi(X_t), \qquad B_{\ell,j} = \mathbb{E}\left[q_t D_t \bar{E}_{t+\ell,j}\right]. \tag{3}$$

This pre-action subtraction preserves the current-code moment and every future-code null because $\phi(X_t)$ is available before E_t is sampled. Unlike a fully residualized statistic, it does not require the observed history to be a sufficient state.

3.2 The rightmost nonzero moment

Identification uses the last nonzero moment. Past codes can correlate with the current state under feedback, so $B_{\ell,j}$ may be nonzero for $\ell < d_j$. Future codes remain independent of the current transition.

Theorem 1 (Right-boundary registration)**.** *Assume that E_t is independent of the complete pre-code history and all transition innovations, q_t is pre-code measurable, and the logger follows Equation 2. If $\Gamma_j = \mathbb{E}[q_t D_t E_{t,j}] \neq 0$, then*

$$B_{d_j,j} = \Gamma_j, \qquad B_{\ell,j} = 0 \quad \textit{for every } \ell > d_j. \tag{4}$$

Therefore d_j is the rightmost lag with a nonzero population moment. Moments before d_j are unrestricted.

The theorem follows from non-anticipation. At $\ell = d_j$, the logged code is $E_{t,j}$, and later entries are independent future codes. The feature map and code amplitude enter only through Γ_j, so they affect finite-sample power but not the population boundary when $\Gamma_j \neq 0$. The result does not depend on a neural architecture, differentiability, or learned residualization. Appendix B gives the full filtration-level proof and vector extension.

3.3 Independent balance for context-dependent effects

Factorized contexts create a sampling issue when the physical perturbation is $W = qE$. A paired design that uses $(q, E) = (1, E)$ and $(-1, -E)$ from the same initial state produces the same physical command twice. It preserves the mean but duplicates one trajectory. We instead assign equal numbers of $q = +1$ and $q = -1$ trajectories and sample every initial state and code sequence independently.

Proposition 1 (Independent balance)**.** *Let E be symmetric Rademacher, let $q \in \{-1, +1\}$ be assigned before an independent trajectory is generated, and let the response depend on (q, E) through $W = qE$, with other randomness identically distributed across strata. Then*

$$qD(qE)E^\top = D(W)W^\top. \tag{5}$$

Both context strata estimate the same population moment. At nominal count $N = 2n$, independent balance has raw-mean variance σ^2/N, while duplicating n paired trajectories has variance $2\sigma^2/N$.

The variance statement applies to the raw sample mean. Direction normalization and bootstrap coverage are handled by the separate inference procedure. Independent balance preserves the target moment while using every trajectory as an independent unit.

3.4 Exact timing does not guarantee response accuracy

Temporal registration specifies the input variable, while the predictor class determines which response can be fitted. This distinction already matters at the population optimum. We compare the predicted difference between two actions with the true difference; predicting no difference defines the zero-response reference.

Proposition 2 (Population-optimal prediction can fail the response test)**.** *Let H and A be independent Rademacher variables, let $Y = H + \varepsilon A$ for $0 < \varepsilon < 1$, and suppose that A is observed at its exact time. In the restricted class $g_w(H, A) = w(H - A)$, ordinary squared-risk minimization gives an action response whose error exceeds that of predicting zero response.*

The unique population-risk minimizer is $w^* = (1 - \varepsilon)/2$. Its squared response error is $(1 + \varepsilon)^2 > 4\varepsilon^2$, the zero-response error. Fitting the state component makes the learned action coefficient negative, although the true coefficient is positive. An unrestricted predictor represents Y exactly. The counterexample proves that timing alone supplies no guarantee, even without optimization or estimation error. Appendix B gives the calculation and the general finite-risk decomposition.

3.5 Prediction accuracy at the action-response scale

Prediction error does control response error once measured at the response scale. For m-dimensional targets Y^+, Y^- at the same history h, define $e^\pm = g(h, a^\pm) - Y^\pm$. Let R_{pair} be their average per-feature squared error, R_{resp} the per-feature error in their difference, and $R_0 = \mathbb{E}\|Y^+ - Y^-\|^2/m$ the zero-response risk.

Proposition 3 (Response-scale prediction guarantee)**.** *For any predictor and matched target pairs with finite second moments,*

$$R_{\mathrm{pair}} = \frac{1}{m}\mathbb{E}\left\|\frac{e^+ + e^-}{2}\right\|^2 + \frac{1}{4}R_{\mathrm{resp}}, \qquad R_{\mathrm{resp}} \leq 4R_{\mathrm{pair}}. \tag{6}$$

The factor four is sharp. If $R_0 > 0$, then $R_{\mathrm{pair}} < R_0/4$ suffices for response risk below zero-response risk.

Table 1: Evaluation design. Counts are per environment, context, and offset family. Each 38-step world-model rollout contributes 37 prediction rows.

Component	Fixed design
Registration	12 direction seeds and 12 held-out evaluation seeds
Registration episodes	96 per seed; candidate lags $0, \ldots, 11$
World-model train	8 seeds $\times$ 32 episodes; 9,472 rows
World-model validation	4 seeds $\times$ 16 episodes; 2,368 rows
World-model test	12 seeds $\times$ 16 episodes; 7,104 rows
Architectures	Convolutional recurrent model and patch Transformer
Primary endpoint	coordinatewise counterfactual response error
Relative baselines	timestamp and validation-selected raw-level peak
Absolute baseline	zero predicted action response

The identity distinguishes shared prediction error from error in the action effect. Under our independent Rademacher action law, uniformly mixing the positive and negative branches gives the factual action marginal, so population paired and factual risks coincide. Improvement over frame persistence alone need not reach the sufficient threshold. Conversely, shared branch errors can cancel in the response. Adding $R_{\mathrm{resp}}/4$ to paired loss doubles its response-error weight without changing the shared-error coefficient. Appendix B.7 proves the identity and marginal equality.

4 Empirical Tests of the Theory

4.1 Visual systems and factorial design

The evaluation separates timing identification from response learning using three deterministic visual control systems: a pendulum, a cart-pole, and a two-dimensional point mass. Each state is rendered as a 24×24 image, and ϕ is an 8×8 average-pooled dynamic feature. The context is either homogeneous, with $q = 1$, or factorized, with $q \in \{-1, +1\}$. Scalar systems use offsets 2, 5, and 8. PointMass2D uses coordinatewise offsets $(2, 5)$, $(5, 8)$, and $(8, 2)$. Crossing system, context, and offset gives 18 independent registration families.

Known transitions make timing and action response separately observable. Table 1 summarizes the alignment study, and Appendix C specifies the systems.

4.2 Registration estimator and inference

Registration separates direction estimation from confirmation. Twelve direction seeds estimate normalized feature-space moments for each lag and coordinate; twelve disjoint evaluation seeds are projected onto these frozen directions. Factorized panels balance 96 independent trajectories across context signs. We select the largest lag with a positive simultaneous lower bound. A one-sided max-statistic bootstrap covers all coordinates and candidate lags, with the familywise level divided across 18 families and two confirmatory response definitions.

The observable estimator uses Equation 3. Subtracting the next feature under unperturbed feedback gives an exact-response positive control. Raw-level and matched temporal-increment peaks are descriptive baselines, separate from the registration decision. Zero-amplitude panels test for spurious boundaries. All confirmatory inference uses independently balanced trajectories.

4.3 Finite-model evaluation

Both finite models receive two observations, each containing dynamic image X and visible context C, plus one logged code per action coordinate. A convolutional neural network with a gated recurrent unit (CNN-GRU) encodes the two-frame history. A two-layer patch Transformer uses frame tokens and an action token. Both predict the next 8×8 feature and train for 12 epochs with the same per-example squared feature loss. Appendix E specifies the models and optimizer.

Four input alignments are compared: timestamp lag zero, raw-level peak, registered right boundary and known oracle offset. The stronger timestamp or raw-peak baseline is selected by validation response error before test evaluation. Identical alignments reuse the same fitted model.

The primary endpoint is coordinatewise counterfactual response error. At a held-out state, changing one current code between $+1$ and -1, with other coordinates fixed, gives $r_t = \phi(X^+_{t+1}) - \phi(X^-_{t+1})$. For the predicted difference $\widehat{r}_t$, response error is $\|\widehat{r}_t - r_t\|_2^2/64$. Relative gain divides the paired reduction from the strongest non-oracle baseline by its error. Querying a misregistered model changes its code slot relative to training, measuring sensitivity to corrected input rather than causal risk under the historical variable. Zero-response error, $\|r_t\|_2^2/64$, is an absolute reference with the same causal meaning for registered and oracle models.

Uncertainty uses 5,000 bootstrap replicates of test seeds and then episodes, with shared draws for methods on identical rows and independent draws across families. Environment and overall summaries weight cell ratios equally; PointMass2D pools coordinate interventions within a cell. The 95% confidence intervals (CIs) condition on each fitted model, excluding retraining variation. Before test evaluation, we fixed a 5% aggregate point target, a positive aggregate CI and no negative environment-level CI.

4.4 Training budget and matched response supervision

The training study separates optimization budget from supervision. Both architectures train for 48 epochs across systems and contexts at offsets 5 or $(5, 8)$. Three initializations and three objectives give 108 fits. Factual loss uses observed targets; paired prediction uses both counterfactual targets; response weighting uses $R_{\text{pair}} + R_{\text{resp}}/4$. Paired objectives share labels, initializations, data order and branch evaluations. A fresh 192-trajectory panel per setting evaluates epochs 12, 24 and 48 and factual-validation-selected models in 32-bit floating point (FP32). Intervals also resample initialization. Appendix G gives optimization and bfloat16 comparison details.

Larger residual models test the same objectives on Walker/Walk, a public locomotion task in DeepMind Control Suite (Tunyasuvunakool et al., 2020). They use 64×64 images, 16×16 targets, 96 training, 24 validation and 32 test episodes. Three initializations per objective give 18 FP32 fits. Same-state interventions change one action coordinate; an independent sign selects the factual branch. Epoch 24 is primary and factual-validation selection is secondary. Appendix H gives the protocol and intervals.

5 Results

5.1 Temporal registration succeeds in every family

The right-boundary estimator recovers all 18 true offset vectors. The exact-response control also recovers 18 of 18, and neither method certifies a lag after the true offset. In contrast, trusting the timestamp and selecting the largest raw-level peak recover 0 of 18 families. A matched descriptive rule that selects the largest held-out temporal-increment mean recovers only 3 of 18, showing that the boundary rule, rather than temporal differencing alone, drives successful registration. Figure 2 shows why: past moments can be larger than the true-lag moment, whereas the simultaneous lower bound becomes negative after the true offset.

Independent balance preserves recovery with unique initial states and code sequences in both context strata. Zero-amplitude controls return no boundary; positive-amplitude controls return offsets 2, 5 and 8. Registered and oracle alignments coincide in all 36 fitted-model cells. Temporal mismatch therefore cannot explain the response failures that follow.

5.2 Correct alignment helps overall at 12 epochs

Correct alignment reduces aggregate response error by 4.78% (95% CI, 4.47% to 5.07%). The estimate lies just below the prespecified 5% point target, while its confidence interval is entirely above zero. Environment-level results differ sharply: Pendulum improves by 7.27% and PointMass2D by 8.31%, while CartPole degrades by 1.25% (95% CI, 0.90% to 1.61% degradation). See Table 2 and Figure 3.

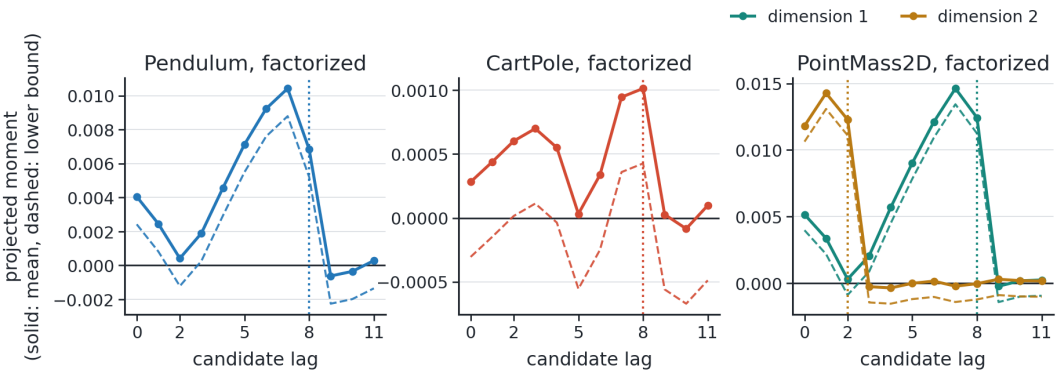

Figure 2: Observable temporal-increment moments in three factorized families. Solid curves show projected means, dashed curves show simultaneous lower bounds, and vertical dotted lines mark true coordinatewise offsets. Past moments are unrestricted. The rightmost positive lower bound recovers each offset, including both PointMass2D coordinates.

Table 2: Timing is recovered in every family, while learned action response remains system dependent. Response gain compares correct alignment with the stronger non-oracle model selected on validation. "Models beating zero" uses the zero-response predictor as an absolute reference, and response energy is measured relative to ordinary transition energy. Parentheses give 95% CIs.

System	Families registered	Response gain (%), 95% CI ↑	Models beating zero	Response energy (%)
Pendulum	6/6	7.27 (6.45, 8.05)	2/12	3.71
CartPole	6/6	-1.25 (-1.61, -0.90)	0/12	0.65
PointMass2D	6/6	8.31 (8.14, 8.48)	5/12	29.85
All	18/18	4.78 (4.47, 5.07)	7/36	–

Correct timing changes the learned response. Relative to timestamp alignment, oracle alignment reduces response error by 5.85% overall (95% CI, 5.53% to 6.15%). Coordinate-level averages give gains of 4.87% for the CNN-GRU and 7.87% for the Transformer, both with positive CIs. Appendix I specifies these aggregation units. The benefit over the stronger non-oracle baseline remains system dependent.

5.3 Relative improvement does not establish action fidelity

The absolute test is stricter. Figure 4 shows that only 7 of 36 correctly aligned cells beat the zero-response reference, including none of the 12 CartPole cells. Aligned-model response error averages 5.07 times the zero-response error overall and 12.20 times in CartPole. Correct alignment can therefore improve one learned model over another while leaving its action effect less accurate than predicting no effect.

CartPole combines the weakest response signal with the largest response error. Its response energy is 0.65% of ordinary dynamics energy, compared with 3.71% for Pendulum and 29.85% for PointMass2D. Context provides a second separation: alignment reduces response error by 11.45% in homogeneous cells but increases it by 1.90% in factorized cells, although the context sign is visible. Ordinary next-feature error increases by only 0.57% on average. These comparisons motivate a direct test of learning at the response scale with fixed systems and architectures.

5.4 Response weighting with matched labels

Longer training improves ordinary prediction more widely than action response. Table 3 reports means over three initializations for each of twelve system-context-architecture conditions. After 48 epochs, factual training beats frame persistence in eleven conditions, but six of those still fail the zero-response test. The same counts hold under factual-validation model selection. Thus the response deficit persists even where ordinary prediction has become useful.

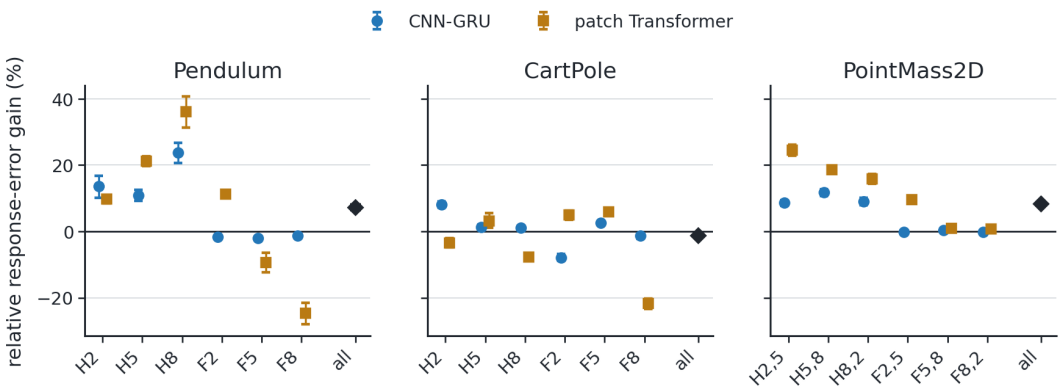

Figure 3: Finite-model response gain for all 36 system, context, offset, and architecture cells. H and F denote homogeneous and factorized contexts. The final diamond in each panel is the equal-stratum environment aggregate. Correct and oracle alignments coincide in every cell, yet gains vary in sign across systems and model classes.

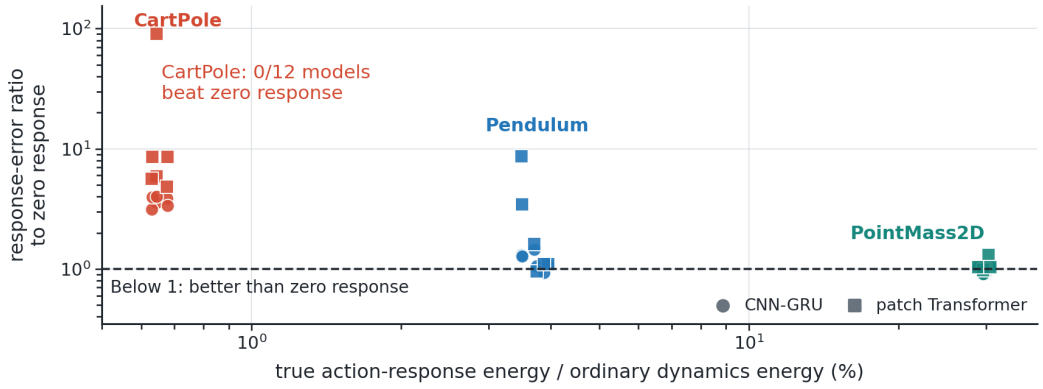

Figure 4: Absolute response quality requires more than correct timing. Each point is one correctly aligned model. The horizontal line is a predictor that always outputs zero action response, and values below one are better. CartPole action response contains about 0.65% of ordinary transition energy, and no aligned CartPole model beats zero.

Direct response weighting improves the matched-label comparison. At 48 epochs it reduces response error in eleven of twelve condition means, with nine pointwise 95% CIs excluding zero. For homogeneous Pendulum with the CNN-GRU, the response-to-zero ratio falls from 0.330 to 0.230; factual prediction risk relative to frame persistence remains 0.019 at the displayed precision. Response weighting brings eight condition means below zero-response risk, compared with seven for

Table 3: Training budget and response supervision, evaluated in FP32 on fresh test trajectories. Counts use three-initialization condition means. Persistence copies the current feature; zero response predicts no action effect. Paired objectives use identical labels and branch evaluations.

Training objective	Epochs	Beats persistence	Beats zero response
Factual	12	8/12	3/12
Factual	24	10/12	4/12
Factual	48	11/12	5/12
Paired	48	11/12	7/12
Paired + response	48	11/12	8/12

paired prediction and five for factual training. Four conditions still fail this test. Appendix G gives full intervals and training curves.

5.5 The separation persists on Walker/Walk

Walker/Walk retains the distinction with larger models. All three final Transformer condition means beat persistence, yet none beats zero response. Response weighting improves on paired prediction by 0.78% for the CNN-GRU and 2.56% for the Transformer, with both response-difference CIs excluding zero. Their response-to-zero ratios remain 1.008 and 1.042. Under factual-validation selection, all six architecture-objective means beat persistence and none beats zero. Appendix H reports every fit and interval.

6 Discussion

6.1 Timing identification and response accuracy

Temporal identification and response accuracy require different guarantees. Our theorem identifies the input; the population-optimal counterexample establishes the remaining learning problem. This connects identification under feedback with value-aware learning (Ljung, 1999; Farahmand et al., 2017; Grimm et al., 2020; Zhang et al., 2021).

The relevant accuracy scale is the action effect itself. Proposition 3 gives a sharp worst-case condition from prediction risk alone. Shared branch errors can cancel, so direct response evaluation is more informative when pairs are available. None of the 36 controlled condition-objective means meets the sufficient condition, although twenty beat zero response. Appendix G measures the shared-error term that explains this difference.

6.2 Implications for world-model design

The paired-risk decomposition identifies a learning target across architectures. Feature prediction and action representations determine what information a model retains (Zhou et al., 2025; Yeom et al., 2026; Alzayer et al., 2026); our result specifies the error component relevant to comparing actions. CoCo constrains rollout consistency, whereas Dueling World Models centers action predictions at readout (Shi et al., 2026; Li et al., 2026). A common action-independent centering term cancels in a paired difference. Response supervision changes that fitted difference, improving both architectures under matched labels.

Exact simulator resets supply the matched labels and isolate response weighting. Active registration also requires control during data collection. Passive logs do not guarantee independent excitation or same-state counterfactuals. Appendix B.9 distinguishes simulator access from conditional-mean estimation in offline data.

Response-aware evaluation complements delay compensation (Arai et al., 2024; Co et al., 2026; Zhang et al., 2026a; Walsh et al., 2007; Ramstedt et al., 2021; Chen et al., 2021). Our results suggest an ordered evaluation: establish temporal correspondence, compare learned response with zero, then measure task-specific planning or control utility.

The evidence covers fixed offsets in three controlled systems and response learning on public Walker/Walk. Precision matters separately: homogeneous CartPole's response-weighted Transformer changes from 0.982 times zero-response risk in FP32 to 1.578 in bfloat16. The population counterexample is independent of this effect. Variable delays, large generative models and downstream planning require separate evaluation.

7 Conclusion

Correct timing leaves a response-learning problem, even at a restricted population optimum. The paired-risk identity gives a sufficient prediction-accuracy condition at the action-effect scale. Controlled systems and public Walker/Walk support evaluating action differences alongside ordinary prediction quality.

AI USE STATEMENT

Generative AI tools supported mathematical formalization, experimental code checking, and manuscript polishing. The authors developed the theoretical framework and experimental design, interpreted the results, and verified the citations, equations, code, figures, insights, claims, and prose. The authors take responsibility for all content.

REPRODUCIBILITY STATEMENT

Appendix B gives complete proofs. The remaining appendix specifies the controlled systems, public Walker/Walk task, data generation, registration and bootstrap units, neural models, seed hierarchy, adaptivity, and complete cell results. The complete experimental code and reported results are provided in the Supplementary Material. The anonymous reproduction program generates the experimental data, computes the statistics, and exports machine-readable results for reconstructing the reported tables and figures.

REFERENCES

Hadi Alzayer, Wenlong Huang, Haonan Chen, Christopher Luey, Lvmin Zhang, Maneesh Agrawala, Gordon Wetzstein, Li Fei-Fei, Yilun Du, Jiajun Wu, and Jia-Bin Huang. Masked visual actions for unified world modeling. *arXiv preprint arXiv:2607.19343*, 2026. URL https://arxiv.org/abs/2607.19343.

Hidehisa Arai, Keishi Ishihara, Tsubasa Takahashi, and Yu Yamaguchi. ACT-Bench: Towards action controllable world models for autonomous driving. *arXiv preprint arXiv:2412.05337*, 2024. URL https://arxiv.org/abs/2412.05337.

Bowen Baker, Ilge Akkaya, Peter Zhokhov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (VPT): Learning to act by watching unlabeled online videos. In *Advances in Neural Information Processing Systems*, pp. 24639–24654, 2022. URL https://arxiv.org/abs/2206.11795.

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 *International Conference on Machine Learning*, pp. 4603–4623, 2024. URL https://proceedings.mlr.press/v235/bruce24a.html.

Baiming Chen, Mengdi Xu, Liang Li, and Ding Zhao. Delay-aware model-based reinforcement learning for continuous control. *Neurocomputing*, 450:119–128, 2021. doi: 10.1016/j.neucom.2021.04.015. URL https://arxiv.org/abs/2005.05440.

Peterson Co, Sicheng Hu, Chunxuan Jiao, Hongyang Cheng, Yulin Luo, Yijie Xu, Sixiang Chen, Zhongxia Zhao, Zihao Wang, DaFeng Chi, Peidong Liu, YuTong Chen, Henghua Liu, Zhihao Yuan, Huizhu Jia, Yuzheng Zhuang, Tianle Zhang, Liang Lin, Huajie Tan, and Shanghang Zhang. WorldSimProbe: Diagnosing simulator faithfulness in action-conditioned world models for embodied manipulation. *arXiv preprint arXiv:2608.09298*, 2026. URL https://arxiv.org/abs/2608.09298.

Amir-Massoud Farahmand, André Barreto, and Daniel Nikovski. Value-aware loss function for model-based reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 1486–1494, 2017. URL https://proceedings.mlr.press/v54/farahmand17a.html.

E. Gabay and S. Merhav. Identification of linear systems with time-delay operating in a closed loop in the presence of noise. *IEEE Transactions on Automatic Control*, 21(5):711–716, 1976. doi: 10.1109/TAC.1976.1101353.

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*, pp. 2170–2179, 2019. URL https://proceedings.mlr.press/v97/gelada19a.html.

Marion Gilson and Paul M. J. Van den Hof. Instrumental variable methods for closed-loop system identification. *Automatica*, 41(2):241–249, 2005. doi: 10.1016/j.automatica.2004.09.016.

Christopher Grimm, André Barreto, Satinder Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2011.03506.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018. URL https://arxiv.org/abs/1803.10122.

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://arxiv.org/abs/1912.01603.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025. doi: 10.1038/s41586-025-08744-2.

Yuxin Jiang, Yuchao Gu, Ivor W. Tsang, and Mike Zheng Shou. Olaf-World: Orienting latent actions for video world modeling. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/66245. arXiv:2602.10104.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. In *Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pp. 761–770. PMLR, 2020. URL https://proceedings.mlr.press/v120/lambert20a.html.

Jiazhuo Li, Yiming Fei, Zhiruo Zhou, and Heikichi Hayashi. Dueling world models: Advantage-style action channels for common-mode distractor rejection. *arXiv preprint arXiv:2608.06706*, 2026. URL https://arxiv.org/abs/2608.06706.

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

Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in Atari games. In *Advances in Neural Information Processing Systems*, 2015. URL https://arxiv.org/abs/1507.08750.

Simon Ramstedt, Yann Bouteiller, Giovanni Beltrame, Christopher Pal, and Jonathan Binas. Reinforcement learning with random delays. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.02966.

Bharadwaj Satchidanandan and P. R. Kumar. Dynamic watermarking: Active defense of networked cyber-physical systems. *Proceedings of the IEEE*, 105(2):219–240, 2017. doi: 10.1109/JPROC.2016.2575064.

Yuhong Shi, Zhenhao Chu, Jie Wei, Jun Hao, Jianyi Liu, and Jingwen Fu. Overcoming statistical bias in action-controllable world models. *arXiv preprint arXiv:2608.04653*, 2026. URL https://arxiv.org/abs/2608.04653.

Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020. doi: 10.1016/j.simpa.2020.100022. URL https://doi.org/10.1016/j.simpa.2020.100022.

Thomas J. Walsh, Ali Nouri, Lihong Li, and Michael L. Littman. Planning and learning in environments with delayed feedback. In *European Conference on Machine Learning*, pp. 442–453, 2007. doi: 10.1007/978-3-540-74958-5_41.

Yufan Wei, Kun Zhou, Lingjun Mao, Zijun Zhang, Ziming Xu, Ziqiao Xi, Shuang Liang, Ruobing Han, Yuchen Yan, Xinyue Wang, Fan Feng, and Biwei Huang. Causally debiased latent action model for embodied action conditioned world models. *arXiv preprint arXiv:2607.09185*, 2026. URL https://arxiv.org/abs/2607.09185.

Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. UniSim: A neural closed-loop sensor simulator. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1389–1399, 2023. doi: 10.1109/CVPR52729.2023.00140.

Jewon Yeom, Hanseul Kim, Jeongjae Park, Sungmok Jung, Jaejin Lee, and Taesup Kim. What makes video world model latents action-relevant: Prediction over reconstruction. *arXiv preprint arXiv:2606.07687*, 2026. URL https://arxiv.org/abs/2606.07687.

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://arxiv.org/abs/2006.10742.

Jiaru Zhang, Can Cui, Yi Xu, Xin Ye, Ruqi Zhang, and Ziran Wang. How can driving world models do counterfactual prediction? *arXiv preprint arXiv:2608.11601*, 2026a. URL https://arxiv.org/abs/2608.11601.

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*, 2026b. 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*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79115–79135. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhou25t.html.

A Appendix Overview

The appendix follows the main argument from identification to response learning. The guide below locates the proofs, complete results and reproduction details. The alignment, controlled training and public-task studies retain separate data summaries and uncertainty calculations.

Section	Contents
B	Identification, finite-learner example and paired-risk proofs
C	Dynamics, controller, renderer and observed context
D	Registration design and complete offset recovery
E	Model architectures, data and alignment comparisons
F	Every alignment-study cell, interval and coordinate
G	Training budgets, paired supervision and all initializations
H	Public Walker/Walk protocol, all fits and matched contrasts
I	Alignment-study sampling and inference
J	Sampling independence and adaptivity
K	Executable artifact and evidence reconstruction

B Formal Results

B.1 Filtration and identification assumptions

The registration claim follows from an external randomized code and non-anticipation. Let $\mathcal{F}_t$ contain the initial state, all transition innovations through time t, the context C_t, and all action codes before E_t. The dynamic observation X_t, controller state S_t, and context sign $q_t = q(C_t)$ are $\mathcal{F}_t$-measurable. The sequence $\{E_t\}$ is independent across time and coordinates, each coordinate is centered Rademacher, and E_t is independent of $\mathcal{F}_t$ and all transition innovations. The next state and observation may depend arbitrarily on $(\mathcal{F}_t, E_t)$ and the next transition innovation, but not on a future code. Thus $D_t = \phi(X_{t+1}) - \phi(X_t)$ may depend on E_t but cannot depend on E_{t+k} for $k > 0$.

The action-response interpretation also uses an exclusion restriction. Conditional on the pre-code state and transition innovation, (q_t, E_t) affect the dynamic observation only through the commanded action in Equation 1. The context can be exposed to a learned model as a separate observation channel, but the controlled dynamic renderer has no direct path from q_t or E_t. Without this restriction, the moment would identify the timing of a randomized code effect but need not identify an action effect.

The logger stores each coordinate with a fixed offset. For coordinate j,

$$\bar{E}_{s,j} = E_{s-d_j,j} \quad \text{whenever the source index is in range.} \tag{7}$$

Every unoccupied position is filled by an independent centered Rademacher padding code. These codes are independent of the physical code sequence, state, context, and transition innovations. This padding law prevents the offset from being identified through missingness or another distributional difference at unoccupied positions. Each fixed d_j belongs to the tested candidate set.

The population response condition is coordinate specific. Define

$$\Gamma_j = \mathbb{E}[q_t D_t E_{t,j}]. \tag{8}$$

Registration requires $\Gamma_j \neq 0$. This is a relevance condition for the external code. It does not require linear dynamics, a Markov visual observation, or a neural model.

B.2 Pre-action subtraction preserves the moment

The temporal increment removes a large pre-action visual component without changing the identifying moment. This property follows from measurability rather than from a learned residual model.

Proposition 4 (Pre-action subtraction)**.** *Under the assumptions above,*

$$\mathbb{E}[q_t \phi(X_t) E_{t,j}] = 0, \tag{9}$$

$$\mathbb{E}[q_t \phi(X_t) \bar{E}_{t+\ell,j}] = 0 \quad \textit{for every } \ell > d_j. \tag{10}$$

Consequently, replacing $\phi(X_{t+1})$ by D_t preserves the moment at the true offset and every future-code null.

Proof. Both q_t and $\phi(X_t)$ are $\mathcal{F}_t$-measurable. Since $E_{t,j}$ is centered and independent of $\mathcal{F}_t$,

$$\mathbb{E}[q_t\phi(X_t)E_{t,j}] = \mathbb{E}[q_t\phi(X_t)\mathbb{E}[E_{t,j} \mid \mathcal{F}_t]] = 0. \tag{11}$$

For $\ell > d_j$, Equation 7 maps $\bar{E}_{t+\ell,j}$ to a future centered code, apart from independent padding entries. The same conditional-expectation argument gives the second equality. Subtracting these zero moments proves the final statement. □

B.3 Proof of right-boundary registration

The right boundary is identifiable even when closed-loop dynamics create nonzero moments at earlier lags. This distinction is why a largest-peak rule is not implied by the assumptions.

Proof of Theorem 1. At $\ell = d_j$, the logger identity gives $\bar{E}_{t+d_j,j} = E_{t,j}$. Therefore

$$B_{d_j,j} = \mathbb{E}[q_t D_t E_{t,j}] = \Gamma_j \neq 0. \tag{12}$$

For $\ell > d_j$, write $k = \ell - d_j > 0$. The logged variable is $E_{t+k,j}$, or an independent padding code at a boundary. In either case, it is centered and independent of (q_t, D_t) by the code and padding assumptions. Hence

$$B_{\ell,j} = \mathbb{E}\big[q_t D_t \mathbb{E}[\bar{E}_{t+\ell,j} \mid q_t, D_t]\big] = 0. \tag{13}$$

The theorem places no restriction on $\ell < d_j$. At such a lag, the logged code is from the past and may already have changed X_t under feedback. Since the moment at the true lag is nonzero and every later moment is zero, d_j is the rightmost lag with a nonzero population moment. □

The coordinatewise result extends directly to vector actions. Each d_j is identified from the jth logger coordinate, while D_t may be vector valued and may depend jointly on all current action coordinates. The proof does not require the offsets to be equal. This extension supports the PointMass2D settings with $(d_1, d_2) \in \{(2,5),(5,8),(8,2)\}$.

B.4 Proof of independent balance

Independent balance preserves the target moment when context reverses the physical code. Let $W = qE$, let ξ collect the initial state and all other trajectory randomness, and write the response as $D(W,\xi)$. The variables E and ξ are sampled independently in each trajectory, and their laws are the same in both context strata. Since $q^2 = 1$,

$$qD(qE,\xi)E^\top = D(W,\xi)(qE)^\top = D(W,\xi)W^\top. \tag{14}$$

Because E is symmetric Rademacher, W has the same distribution in both fixed context strata. Equal numbers of independently generated trajectories from $q = +1$ and $q = -1$ therefore target the same mean.

The variance comparison is most transparent for any scalar matrix element or projection onto a fixed feature direction. Let that raw summand in Equation 14 have variance σ^2. With $N = 2n$ independent trajectories, the raw mean has variance σ^2/N. A duplicated-companion construction instead generates n independent physical trajectories and enters each summand twice. Its nominal N-row mean equals the mean of the n independent values, so its variance is $\sigma^2/n = 2\sigma^2/N$. This statement does not extend through direction normalization, the maximum statistic, or bootstrap coverage.

B.5 Finite risk ordering

The risk decomposition gives an exact condition for when correct alignment improves a chosen finite learner. For a baseline alignment b and correct alignment c, write $R(\widehat{g}_{A,z}) = R^*_z + \mathcal{E}_{A,z}$, where R^*_z is the best risk under information set z, and $\mathcal{E}_{A,z}$ collects approximation, optimization and estimation excess for algorithm A. Then

$$R(\widehat{g}_{A,b}) - R(\widehat{g}_{A,c}) = (R^*_b - R^*_c) + (\mathcal{E}_{A,b} - \mathcal{E}_{A,c}). \tag{15}$$

Therefore correct alignment has lower finite risk if and only if

$$\mathcal{E}_{A,c} - \mathcal{E}_{A,b} < R_b^* - R_c^*. \tag{16}$$

An ordering $R_c^* \leq R_b^*$ follows when the correct information set contains the baseline information set, or under a corresponding conditional-independence argument. The historical and current code slots in the relative comparison are not nested, so we do not assume this ordering. Even when an ideal-risk advantage exists, Equation 16 shows why it need not survive finite approximation, optimization, and estimation.

B.6 Proof of the finite-learner counterexample

Proof of Proposition 2. Independence and centering give

$$\mathbb{E}[(Y - g_w(H, A))^2] = (1 - w)^2 + (\varepsilon + w)^2. \tag{17}$$

The unique minimizer is $w^* = (1-\varepsilon)/2 > 0$. The true coefficient of A is ε, whereas the learned coefficient is $-w^*$. Ignoring the action gives squared coefficient error ε^2. The restricted risk minimizer gives

$$(\varepsilon + w^*)^2 = \left(\frac{1+\varepsilon}{2}\right)^2 > \varepsilon^2, \tag{18}$$

where the inequality follows from $\varepsilon < 1$. Thus exact temporal access to A does not ensure a useful learned action response under ordinary prediction risk and a finite shared representation. □

The unrestricted predictor $g(H, A) = H + \varepsilon A$ has zero error. The restricted class couples its state and action coefficients, so fitting the larger state component reverses the smaller action response. The neural study measures whether a response deficit also occurs in compact architectures without this constructed parameter constraint.

B.7 Paired prediction risk and response risk

The paired-risk identity holds for any predictor with finite second moments. Let h be the observed history, let Y^+ and Y^- be next-feature targets under two interventions from the same physical state, and let m be the feature dimension: 64 in the controlled systems and 256 in Walker/Walk. Define errors $e^\pm = g(h, a^\pm) - Y^\pm$ and risks

$$R_{\text{pair}} = \frac{1}{2m}\mathbb{E}\big[\|e^+\|^2 + \|e^-\|^2\big], \tag{19}$$

$$R_{\text{resp}} = \frac{1}{m}\mathbb{E}\|e^+ - e^-\|^2. \tag{20}$$

For vector actions, the expectation also averages uniformly over the intervened coordinate, holding the other action coordinates fixed.

Proposition 3 follows from the following identity and its nonnegative shared-error term:

$$R_{\text{pair}} = \frac{1}{m}\mathbb{E}\left\|\frac{e^+ + e^-}{2}\right\|^2 + \frac{1}{4}R_{\text{resp}}, \tag{21}$$

$$R_{\text{resp}} \leq 4R_{\text{pair}}. \tag{22}$$

If zero-response risk $R_0 = \mathbb{E}\|Y^+ - Y^-\|^2/m$ is positive, then $R_{\text{pair}} < R_0/4$ is sufficient for $R_{\text{resp}} < R_0$.

Proof of Proposition 3. The parallelogram identity gives

$$\frac{1}{2}(\|e^+\|^2 + \|e^-\|^2) = \left\|\frac{e^+ + e^-}{2}\right\|^2 + \frac{1}{4}\|e^+ - e^-\|^2. \tag{23}$$

Take expectations and divide by m. Nonnegativity of the first term yields the bound; comparison with R_0 gives the sufficient condition. The factor four is attained when $e^+ = -e^-$. No architectural, linearity or differentiability assumption is used. □

A scalar example attains equality under the stated randomized action law. Let $Y(H, A) = H + \varepsilon A$ with independent Rademacher H, A, and let $g(H, A) = H$. Then $e^+ = -\varepsilon$, $e^- = \varepsilon$, $R_{\text{pair}} = \varepsilon^2$ and $R_{\text{resp}} = R_0 = 4\varepsilon^2 > 0$. Equality at $R_{\text{pair}} = R_0/4$ therefore cannot guarantee improvement over zero response.

Paired and factual risks have the same population target under the experiment's action law. Conditional on the complete pre-code state and observed history, the current coordinate E_j is independent and uniformly distributed on $\{-1, +1\}$. A uniformly selected positive or negative intervention branch therefore has the factual distribution of E_j. The other coordinates retain their factual values, and both branches share the same pre-code state and context. Averaging over coordinates and histories gives $R_{\text{pair}} = R_{\text{factual}}$ in population. The empirical losses differ because paired training observes both branch targets at each sampled state. Thus the comparison tests additional supervision and objective weighting under action coverage.

The decomposition specifies the response-weighted objective without a coefficient search. Paired prediction minimizes R_{pair}. Adding $R_{\text{resp}}/4$ doubles the response-error coefficient from $1/4$ to $1/2$ and leaves the mean-error coefficient unchanged. This construction emphasizes a measured prediction component without changing the architecture or branch labels. The held-out comparison determines whether that emphasis improves generalization.

B.8 Response energy and correlated branch errors

Response failure can arise from predicted variation that is weakly aligned with the physical effect. Define $r = Y^+ - Y^-$ and $\widehat{r} = g(h, a^+) - g(h, a^-)$. Expanding their squared difference gives

$$R_{\text{resp}} - R_0 = \frac{1}{m}\mathbb{E}\|\widehat{r}\|^2 - \frac{2}{m}\mathbb{E}\langle \widehat{r}, r\rangle. \tag{24}$$

The first term is predicted response energy and the second measures alignment with the true effect. A model improves on zero when twice its alignment exceeds its predicted energy. A small response is therefore useful only when its direction agrees with the true change.

Correlated branch errors also explain why the bound need not be tight. Direct expansion yields

$$R_{\text{resp}} = \frac{1}{m}\mathbb{E}\|e^+\|^2 + \frac{1}{m}\mathbb{E}\|e^-\|^2 - \frac{2}{m}\mathbb{E}\langle e^+, e^-\rangle. \tag{25}$$

Common prediction errors cancel in the response difference. The evidence retains both branch risks, their cross term, predicted energy and alignment, allowing Equations 21, 24 and 25 to be checked on the same rows.

B.9 Data access and response targets

Data access determines which part of the analysis can be implemented. Right-boundary registration requires a randomized code introduced before the physical transition and recorded by the action logger. This permits calibration through a controllable collection interface, including a simulator or an instrumented controller. Adding random labels to an existing passive log cannot create the required intervention. Passive video and recovered action labels motivate the need for temporal correspondence, but do not by themselves satisfy the active identification assumptions (Baker et al., 2022; Bruce et al., 2024; Ljung, 1999).

Simulator resets provide direct response labels. Starting from the same physical state, the two branches vary one action coordinate while holding the history and remaining coordinates fixed. Both paired objectives use these same outcomes. Their contrast therefore measures the loss weighting, whereas their comparison with factual prediction also changes the number of observed targets. The theorem on timing and the identity for prediction risk address separate data requirements: the identity remains valid when timing is already known and no registration code is used.

Offline data can instead identify a conditional mean response under additional assumptions. Let Y^a be the potential next target under action a and let h be the observed history. If the recorded action is conditionally independent of Y^a given h, the observed target equals Y^a when that action is taken, and each queried action has conditional support, then $\mu_a(h) = \mathbb{E}[Y^a \mid h] = \mathbb{E}[Y \mid h, A = a]$. Conditional regression can estimate the difference $\mu_{a^+}(h) - \mu_{a^-}(h)$. This target averages over

unobserved states compatible with the history; it differs from the same-state response measured in our simulator. Approximate labels also introduce estimation error. For a true response r, an approximate label $\widetilde{r}$, and a predicted response $\widehat{r}$, the triangle inequality in mean square gives

$$\sqrt{\mathbb{E}\|\widehat{r}-r\|^2/m} \le \sqrt{\mathbb{E}\|\widehat{r}-\widetilde{r}\|^2/m} + \sqrt{\mathbb{E}\|\widetilde{r}-r\|^2/m}. \tag{26}$$

Thus fitting approximate paired labels supports a true-response guarantee only when their error is also controlled. Matching visually similar histories alone does not establish that control.

Stochastic transitions preserve the algebraic identity but change the target being evaluated. Let z contain the observed history, complete simulator state and fixed action pair. Write $Y^{\pm} = \mu_{\pm}(z) + \xi^{\pm}$, where $\mathbb{E}[\xi^{\pm} \mid z] = 0$, and let $d(z) = \mu_{+}(z) - \mu_{-}(z)$. For a fixed predictor, conditional centering yields

$$\frac{1}{m}\mathbb{E}\|\widehat{r} - (Y^{+} - Y^{-})\|^2 = \frac{1}{m}\mathbb{E}\|\widehat{r} - d(z)\|^2 + \frac{1}{m}\mathbb{E}\|\xi^{+} - \xi^{-}\|^2. \tag{27}$$

The final term depends on how the two branches share transition noise. Independent and shared-noise rollouts therefore measure different response risks. A stochastic extension must specify that coupling, or evaluate conditional mean responses using repeated transitions. Our deterministic transitions have zero noise conditional on z. Conditioning only on observed history can leave uncertainty about the hidden state, even with deterministic dynamics.

Representation choice determines what an accurate response means. The identity applies to a fixed target map, including a frozen learned representation. A target representation trained jointly with the predictor can change or discard action-relevant variation. Its response risk then requires an additional representation-quality criterion. Our fixed dynamic feature makes that target explicit; it does not evaluate end-to-end representation learning in a generative world model.

C Controlled Visual Systems

C.1 Common protocol

All systems are deterministic so that the true counterfactual response is available. A macro transition contains four numerical substeps. At each macro time, a stabilizing feedback policy produces $\pi(S_t)$ and the executed action is

$$U_t = \mathrm{clip}(\pi(S_t) + 0.35 q_t E_t, -2, 2)\,. \tag{28}$$

The homogeneous context uses $q_t = 1$. In registration panels, the factorized sign is fixed within an episode and exactly balanced across independent episodes. In world-model episodes, q_t is sampled independently at each time step. In both cases, the sign is exposed as a separate constant image channel, and the dynamic renderer never reads it. Table 4 lists the initial-state distributions and feedback policies.

Table 4: Initial-state distributions and feedback policies. Each interval is sampled independently.

System	Initial state	Feedback action
Pendulum	$\theta \sim U[-1.1, 1.1]$, $\omega \sim U[-.45, .45]$	$[-1.05\sin\theta - .24\omega]_{-1.35}^{1.35}$
CartPole	$x \sim U[-.45, .45]$, $\dot{x} \sim U[-.25, .25]$	$[-.32x - .22\dot{x} + 1.25\theta + .34\dot{\theta}]_{-1.35}^{1.35}$
	$\theta \sim U[-.35, .35]$, $\dot{\theta} \sim U[-.25, .25]$	
PointMass2D	$x, y \sim U[-.65, .65]$, $v_x, v_y \sim U[-.25, .25]$	$[-.72(x, y) - .34(v_x, v_y)]_{-1.35}^{1.35}$

C.2 Pendulum dynamics

The pendulum state is (θ, ω) and the scalar action is torque u. One substep with $\Delta = 0.045$ is

$$\omega' = 0.996\omega + \Delta(-0.82\sin\theta + 1.10u), \tag{29}$$

$$\theta' = \mathrm{wrap}_{[-\pi,\pi)}(\theta + \Delta\omega'). \tag{30}$$

After four substeps, angular velocity is clipped to $[-4, 4]$. The renderer places a pivot at $(0, 0.38)$, a bob at distance 0.68, and Gaussian intensity along the connecting rod.

C.3 CartPole dynamics

The CartPole state is $(x, \dot{x}, \theta, \dot{\theta})$ and the applied force is $F = 5u$. We use gravity $g = 9.8$, cart mass $m_c = 1$, pole mass $m_p = 0.1$, half-length $l = 0.5$, and $\Delta = 0.02$. One substep is

$$a = \frac{F + m_p l \dot{\theta}^2 \sin\theta}{m_c + m_p}, \tag{31}$$

$$\ddot{\theta} = \frac{g \sin\theta - \cos\theta\, a}{l\,(4/3 - m_p \cos^2\theta/(m_c + m_p))}, \tag{32}$$

$$\ddot{x} = a - \frac{m_p l \ddot{\theta} \cos\theta}{m_c + m_p}, \tag{33}$$

$$\dot{x}' = 0.998\dot{x} + \Delta\ddot{x}, \qquad \dot{\theta}' = 0.998\dot{\theta} + \Delta\ddot{\theta}, \tag{34}$$

$$x' = x + \Delta\dot{x}', \qquad \theta' = \theta + \Delta\dot{\theta}'. \tag{35}$$

After each substep, $(x, \theta, \dot{x}, \dot{\theta})$ is clipped to $[-2.35, 2.35] \times [-1.35, 1.35] \times [-4, 4] \times [-5, 5]$. The renderer draws the cart, rail, pole, and tip from (x, θ).

C.4 PointMass2D dynamics

The point-mass state is (x, y, v_x, v_y) and the action is (u_x, u_y). With $\Delta = 0.065$, one substep is

$$v_x' = 0.985 v_x + \Delta u_x, \qquad x' = x + \Delta v_x', \tag{36}$$

$$v_y' = 0.985 v_y + \Delta u_y, \qquad y' = y + \Delta v_y'. \tag{37}$$

Positions reflect at ± 0.98 with velocity magnitude multiplied by 0.55, and velocities are clipped to $[-3, 3]$. The renderer shows the point, a short velocity trace, and the square boundary.

C.5 Rendering and response features

Each state is rendered deterministically to a 24×24 grayscale image in $[0, 1]$. Pendulum and PointMass2D use Gaussian blobs for dynamic objects, while CartPole combines a smooth rectangular cart, Gaussian pole samples, and a rail. Images are quantized through a 16-bit floating representation and then returned as 32-bit values. This round trip is applied to all methods and counterfactuals.

The response map ϕ averages each non-overlapping 3×3 image block, producing an 8×8 feature and then a 64-dimensional vector. Registration uses only this dynamic feature. World models receive a two-channel observation: the dynamic image and a constant context channel. Their target is the next 64-dimensional dynamic feature.

D Registration Protocol

D.1 Panel construction

Registration uses independent episodes as sampling units. Each episode draws a fresh initial state, a fresh length-48 Rademacher code sequence, and fresh Rademacher padding codes for unused logger locations. Factorized panels assign exactly half of the episodes to each context sign before generating independent trajectories. No initial state or code sequence is reused across the two signs.

The factorial study crosses three systems, two context regimes, and three offset settings. Pendulum and CartPole use scalar offsets 2, 5, and 8. PointMass2D uses $(2, 5)$, $(5, 8)$, and $(8, 2)$. Each of the 18 families has 12 direction seeds and 12 disjoint evaluation seeds, with 96 independent episodes per seed. These 216 direction-evaluation seed pairs contain 41,472 registration episodes. The preceding registration controls contain 72 seed pairs and 23,040 separate episodes, as reported in Appendix D.4. The complete registration evidence therefore contains 64,512 independent episodes.

D.2 Direction estimation and held-out confirmation

The feature-valued moment must be converted to a scalar test without using evaluation outcomes to choose its direction. For method m, lag ℓ, coordinate j, and direction seed s, let $\widehat{B}^{(m)}_{s,\ell,j}$ be the

equal-context-stratum mean when the context is factorized and the ordinary mean otherwise. We first average the 12 direction-seed means and normalize:

$$v_{\ell,j}^{(m)} = \frac{\sum_s \widehat{B}_{s,\ell,j}^{(m)}}{\left\| \sum_s \widehat{B}_{s,\ell,j}^{(m)} \right\|_2}. \tag{38}$$

Each episode from the 12 disjoint evaluation seeds is then projected onto this frozen direction. No evaluation seed contributes to $v_{\ell,j}^{(m)}$.

The aggregate statistic gives every evaluation seed equal weight. Each family and response definition uses 50,000 outer bootstrap draws. Every draw resamples the 12 evaluation-seed means. A one-sided maximum statistic covers every candidate lag and action coordinate in that family. The global level 0.05 is divided across 18 families and the two confirmatory response definitions, temporal increment and exact response. The raw-level method is descriptive and has no role in this familywise decision. A coordinate is registered at the largest lag whose simultaneous lower bound is positive.

D.3 Response definitions

The temporal increment is the observable response used by the estimator:

$$D_t^{\mathrm{inc}} = \phi(X_{t+1}) - \phi(X_t). \tag{39}$$

The exact response is a positive control available only because the transition kernel is known:

$$D_t^{\mathrm{exact}} = \phi(X_{t+1}(\pi(S_t) + 0.35 q_t E_t)) - \phi(X_{t+1}(\pi(S_t))). \tag{40}$$

The raw level $D_t^{\mathrm{raw}} = \phi(X_{t+1})$ supplies the largest-peak diagnostic. A matched diagnostic applies the same largest-mean rule to the held-out temporal-increment projections. Neither diagnostic changes the right-boundary decision. All three response definitions use the same trajectories, logger candidates, feature map, direction split, and candidate lags 0 through 11.

D.4 Registration controls

The registration controls test signal absence, positive offsets, and evaluation-seed stability before the full factorial study. They use factorized Pendulum panels, amplitudes 0 and 0.35, offsets 2, 5, and 8, 12 direction seeds, 12 held-out evaluation seeds, and 160 independent episodes per seed. Aggregate inference uses the same right-boundary rule. At the seed level, exact one-sided 90% Clopper-Pearson lower bounds summarize recovery stability. Their conjunction is a design criterion rather than simultaneous 95% inference.

Table 5: Registration-control outcomes. A zero-amplitude panel is correct when no boundary is returned. The raw-level diagnostic is excluded.

Response	Amplitude	True lag	Estimated	Correct	Future false
increment	0.00	2	none	yes	0
increment	0.35	2	2	yes	0
increment	0.00	5	none	yes	0
increment	0.35	5	5	yes	0
increment	0.00	8	none	yes	0
increment	0.35	8	8	yes	0
exact control	0.00	2	none	yes	0
exact control	0.35	2	2	yes	0
exact control	0.00	5	none	yes	0
exact control	0.35	5	5	yes	0
exact control	0.00	8	none	yes	0
exact control	0.35	8	8	yes	0

Table 5 shows that the temporal-increment estimator recovers offsets 2 and 8 in all 12 held-out evaluation seeds and offset 5 in 11 of 12. The exact-response control recovers every offset in all 12 seeds. The corresponding one-sided 90% exact lower bounds are 0.825, 0.713, and 0.825 for the temporal increment, and 0.825 for each exact-response offset. Aggregate zero-amplitude panels return no boundary for either confirmatory response.

D.5 COMPLETE REGISTRATION OUTCOMES

Table 6 reports all 18 families. The timestamp and raw-level peak are wrong in every family. A matched temporal-increment peak rule uses the same held-out projections but selects the largest mean. It recovers only three families, all from CartPole. The observable right boundary and exact-response control equal the oracle in every family, including all vector offsets.

Table 6: Complete registration results. Boundary denotes the observable temporal-increment estimator; Exact denotes the exact-response control. Parentheses show coordinatewise lags.

System	Context	True lag	Timestamp	Raw peak	Boundary	Exact
Pendulum	hom.	(2)	(0)	(0)	(2)	(2)
Pendulum	hom.	(5)	(0)	(0)	(5)	(5)
Pendulum	hom.	(8)	(0)	(3)	(8)	(8)
Pendulum	fact.	(2)	(0)	(0)	(2)	(2)
Pendulum	fact.	(5)	(0)	(0)	(5)	(5)
Pendulum	fact.	(8)	(0)	(3)	(8)	(8)
CartPole	hom.	(2)	(0)	(1)	(2)	(2)
CartPole	hom.	(5)	(0)	(3)	(5)	(5)
CartPole	hom.	(8)	(0)	(5)	(8)	(8)
CartPole	fact.	(2)	(0)	(0)	(2)	(2)
CartPole	fact.	(5)	(0)	(2)	(5)	(5)
CartPole	fact.	(8)	(0)	(0)	(8)	(8)
PointMass2D	hom.	(2,5)	(0,0)	(0,0)	(2,5)	(2,5)
PointMass2D	hom.	(5,8)	(0,0)	(0,3)	(5,8)	(5,8)
PointMass2D	hom.	(8,2)	(0,0)	(3,0)	(8,2)	(8,2)
PointMass2D	fact.	(2,5)	(0,0)	(0,0)	(2,5)	(2,5)
PointMass2D	fact.	(5,8)	(0,0)	(0,3)	(5,8)	(5,8)
PointMass2D	fact.	(8,2)	(0,0)	(3,0)	(8,2)	(8,2)

E FINITE WORLD-MODEL EVALUATION

E.1 WORLD-MODEL ROWS AND ACTION ALIGNMENTS

Each world-model episode contains 38 executed transitions and contributes 37 prediction rows. A row contains observations at times $t-1$ and t, a candidate logged code for each lag, and target feature $\phi(X_{t+1})$. Training, validation, and test splits use disjoint top-level seeds. Per family, training contains $8 \times 32 \times 37 = 9{,}472$ rows, validation contains $4 \times 16 \times 37 = 2{,}368$ rows, and test contains $12 \times 16 \times 37 = 7{,}104$ rows. Across 18 families, the corresponding totals are 170,496, 42,624, and 127,872 unique rows. Architectures and alignments share these rows and do not create additional samples.

Four alignment labels map the logger to model input. Timestamp uses lag zero. Raw peak uses the coordinatewise largest raw-level moment estimated from the registration panels. Right boundary uses the observable registered lag. Oracle uses the known lag. Identical lag vectors share the same trained model and stored predictions. Across all families and architectures, 92 distinct models are trained. The four labels index 144 evaluations after exact reuse.

The baseline for each family and architecture is selected using validation counterfactual response error only. We compare timestamp and raw peak, retain the lower-error model, and use the test split once for the frozen comparison with the right boundary. Timestamp is selected in 26 of 36 cells and raw peak in 10. This selection uses simulator counterfactual ground truth and is an evaluation control, not a deployable selector for natural logs. Since right-boundary and oracle lags coincide in all cells, their model parameters and predictions are identical.

E.2 Architectures and optimization

Table 7 describes two compact architectures with different temporal inductive biases. The registration theorem is architecture independent, while the finite-model evaluation asks how each fitted function class represents the recovered action response.

Table 7: World-model architectures. Parameter counts differ slightly between scalar and two-dimensional action inputs.

Model	Visual and temporal encoder	Action and output head
CNN-GRU	$2 \to 12 \to 20 \to 28$ convolutions, 64-unit gated recurrent unit	action concatenation, 96-unit multilayer perceptron, 61,172 or 61,268 parameters
Patch Transformer	4×4 patches, width 64, 2 layers, 4 heads	action token, linear 64-dimensional head, 78,272 or 78,336 parameters

The CNN-GRU uses three 3×3 convolutions with strides $(1, 2, 2)$ and padding one, each followed by a sigmoid linear unit. Adaptive average pooling produces a 3×3 grid, which is flattened and linearly projected to 64 units. A single-layer 64-unit GRU processes the previous and current frame embeddings. The final hidden state is concatenated with the action code, then passed through a 96-unit hidden layer and a linear 64-dimensional output. The hidden layer uses the same activation.

The Transformer embeds each frame with a 4×4 stride-four convolution. The action code is linearly mapped to one token and prepended to the previous-frame and current-frame patch tokens. Learned positional embeddings and separate learned time embeddings encode token location and frame order. Two pre-normalized layers use four attention heads, width 64, feedforward width 128, Gaussian error linear units and zero dropout. A final layer normalization and linear head read the action token. Positional and time embeddings are initialized from a normal distribution with standard deviation 0.01.

Both models predict the next 64-dimensional dynamic feature. The training loss sums the 64 squared feature errors and averages over examples. We train for 12 epochs with AdamW, learning rate 2×10^{-3}, weight decay 10^{-5}, gradient-norm clipping at 5, and effective batch size 512. This single configuration is shared across architectures and objectives without architecture-specific tuning. The comparisons test learning under a common optimization protocol, not each architecture's best attainable performance. Training and inference use bfloat16 autocasting, while stored predictions and metrics use 32-bit floating point. Data are deterministically shuffled at each epoch. Microbatches accumulate gradients before one optimizer update; they preserve the effective batch and update count. No hyperparameter is selected from the held-out test split.

E.3 Counterfactual action response

The primary endpoint measures whether the model predicts the current action effect accurately. For test row i and action coordinate j, we replace the intended current code coordinate by $+1$ and -1 while holding the state, context, and other coordinates fixed. The true response is

$$r_{i,j} = \phi(X_{i,t+1}^{E_j=+1}) - \phi(X_{i,t+1}^{E_j=-1}), \tag{41}$$

and the learned response is the corresponding prediction difference $\widehat{r}_{i,j}$. Coordinatewise response error is

$$e_{i,j} = \frac{1}{64}\|\widehat{r}_{i,j} - r_{i,j}\|_2^2. \tag{42}$$

For baseline b and right-boundary model c, relative response gain is

$$G = \frac{\mathbb{E}[e_b - e_c]}{\mathbb{E}[e_b]}. \tag{43}$$

Positive values favor correct alignment. Errors are computed for each intervention coordinate. For PointMass2D cell-level gain, numerator and denominator errors are pooled over both coordinate interventions within each episode and model cell before their ratio is formed. One ratio per model cell

enters the 36-stratum environment and overall summaries. Coordinate-specific ratios are reported separately.

Two secondary quantities separate relative sensitivity from absolute quality. The zero-response error is $e^0_{i,j} = \|r_{i,j}\|_2^2/64$. A right-boundary model beats zero when its mean response error is below this reference. The response-to-dynamics energy ratio is

$$\rho = \frac{\mathbb{E}[\|r_{i,j}\|_2^2/64]}{\mathbb{E}[\|\phi(X_{t+1}) - \phi(X_t)\|_2^2/64]}. \tag{44}$$

Ordinary next-feature error is reported separately and is not used as an action-response endpoint.

E.4 Interpretation of misregistered comparisons

Timestamp and raw-peak models are trained with a historical logged-code slot. The test instead changes the intended current-action code. Their response errors therefore measure how a deployed model transports an intervention through its learned action interface. They do not identify the causal risk of the historical code variable under a common information set. The absolute zero-response comparison avoids this ambiguity for the correctly registered model, and the oracle identity confirms that no residual registration error remains.

F Complete Results

F.1 All model cells

Table 8 reports every system, context, offset, and architecture cell. The environment-level pattern in the main paper is accompanied by a context split. Averaged over the fixed cells, homogeneous contexts gain 11.45%, whereas factorized contexts change by -1.90%. This descriptive split shows that response energy is not the only source of difficulty: the visible context determines the sign of the physical perturbation, and ordinary next-feature training need not represent this interaction.

Table 8: Complete cell diagnostics. Gain compares the right-boundary model with the validation-selected non-oracle model. Energy is the true response-to-dynamics ratio. Aligned/zero below one means the correctly aligned model beats a zero-response predictor. Context entries are homogeneous (hom.) or factorized (fact.).

System	Ctx.	Model	Lag	Gain (%)	Energy (%)	Aligned/zero
Pendulum	hom.	CNN-GRU	2	13.57	3.75	1.06
Pendulum	hom.	Transformer	2	9.71	3.75	0.95
Pendulum	hom.	CNN-GRU	5	10.84	3.95	1.07
Pendulum	hom.	Transformer	5	21.22	3.95	1.10
Pendulum	hom.	CNN-GRU	8	23.77	3.88	0.93
Pendulum	hom.	Transformer	8	36.04	3.88	1.10
Pendulum	fact.	CNN-GRU	2	-1.65	3.49	1.31
Pendulum	fact.	Transformer	2	11.25	3.49	8.66
Pendulum	fact.	CNN-GRU	5	-2.04	3.70	1.45
Pendulum	fact.	Transformer	5	-9.43	3.70	1.61
Pendulum	fact.	CNN-GRU	8	-1.34	3.50	1.28
Pendulum	fact.	Transformer	8	-24.63	3.50	3.46
CartPole	hom.	CNN-GRU	2	8.10	0.65	3.56
CartPole	hom.	Transformer	2	-3.44	0.65	5.91
CartPole	hom.	CNN-GRU	5	1.20	0.63	3.15
CartPole	hom.	Transformer	5	3.22	0.63	5.61
CartPole	hom.	CNN-GRU	8	1.18	0.68	3.88
CartPole	hom.	Transformer	8	-7.76	0.68	4.85
CartPole	fact.	CNN-GRU	2	-7.88	0.63	3.98
CartPole	fact.	Transformer	2	4.95	0.63	8.56
CartPole	fact.	CNN-GRU	5	2.61	0.68	3.36
CartPole	fact.	Transformer	5	5.92	0.68	8.55
CartPole	fact.	CNN-GRU	8	-1.35	0.64	4.00

System	Ctx.	Model	Lag	Gain (%)	Energy (%)	Aligned/zero
CartPole	fact.	Transformer	8	-21.82	0.64	90.98
PointMass2D	hom.	CNN-GRU	2,5	8.58	29.91	0.92
PointMass2D	hom.	Transformer	2,5	24.53	29.91	0.98
PointMass2D	hom.	CNN-GRU	5,8	11.77	29.72	0.91
PointMass2D	hom.	Transformer	5,8	18.62	29.72	1.00
PointMass2D	hom.	CNN-GRU	8,2	9.07	29.59	0.91
PointMass2D	hom.	Transformer	8,2	15.89	29.59	0.98
PointMass2D	fact.	CNN-GRU	2,5	-0.15	30.37	1.06
PointMass2D	fact.	Transformer	2,5	9.52	30.37	1.31
PointMass2D	fact.	CNN-GRU	5,8	0.41	28.92	1.06
PointMass2D	fact.	Transformer	5,8	0.91	28.92	1.04
PointMass2D	fact.	CNN-GRU	8,2	-0.15	30.57	1.06
PointMass2D	fact.	Transformer	8,2	0.72	30.57	1.03

The absolute diagnostic is demanding and informative. Only two Pendulum models and five PointMass2D models beat zero response. None of the 12 CartPole models do so. The largest ratio is the factorized CartPole patch Transformer at offset 8, whose response error is 90.98 times the zero-response error. The small true CartPole response makes this ratio sensitive to inaccurate predicted changes even when ordinary next-feature prediction remains well defined.

F.2 Cell intervals and ordinary prediction

Table 9 gives cellwise hierarchical CIs for the primary gain and point estimates for oracle-versus-timestamp sensitivity and ordinary next-feature gain. Correct timing changes learned responses for both architectures in aggregate, while ordinary prediction error does not move consistently with response quality. The equal-cell mean ordinary-error gain is -0.57%, compared with the 4.78% response gain. This difference supports separate reporting of prediction and action-response metrics.

Table 9: Cellwise uncertainty and secondary outcomes. Each gain CI resamples top-level test seeds and episodes conditional on the fitted model. Oracle sensitivity compares oracle with timestamp timing. Ordinary gain uses next-feature error. Context entries are homogeneous (hom.) or factorized (fact.).

System	Context	Model	Offset	Gain (95% CI), %	Oracle sensitivity (%)	Ordinary gain (%)
Pendulum	hom.	CNN-GRU	2	13.57 (10.22, 16.83)	13.57	2.65
Pendulum	hom.	Transformer	2	9.71 (8.42, 10.95)	9.71	-5.10
Pendulum	hom.	CNN-GRU	5	10.84 (9.29, 12.44)	10.84	-5.90
Pendulum	hom.	Transformer	5	21.22 (19.69, 22.74)	21.22	3.92
Pendulum	hom.	CNN-GRU	8	23.77 (20.70, 26.72)	23.77	-4.87
Pendulum	hom.	Transformer	8	36.04 (31.27, 40.65)	36.04	-4.19
Pendulum	fact.	CNN-GRU	2	-1.65 (-2.03, -1.29)	-1.65	0.06
Pendulum	fact.	Transformer	2	11.25 (10.27, 12.14)	11.25	1.76
Pendulum	fact.	CNN-GRU	5	-2.04 (-2.43, -1.68)	-2.04	0.88
Pendulum	fact.	Transformer	5	-9.43 (-12.37, -6.47)	-9.43	0.80
Pendulum	fact.	CNN-GRU	8	-1.34 (-1.82, -0.89)	-1.34	-3.10
Pendulum	fact.	Transformer	8	-24.63 (-28.06, -21.45)	-15.94	-0.38
CartPole	hom.	CNN-GRU	2	8.10 (7.20, 8.96)	11.23	5.71
CartPole	hom.	Transformer	2	-3.44 (-4.82, -2.02)	-3.44	-2.91
CartPole	hom.	CNN-GRU	5	1.20 (0.31, 2.05)	-0.33	-2.43
CartPole	hom.	Transformer	5	3.22 (1.16, 5.55)	9.43	-4.58
CartPole	hom.	CNN-GRU	8	1.18 (0.75, 1.62)	8.28	1.65
CartPole	hom.	Transformer	8	-7.76 (-8.91, -6.65)	4.07	-4.72
CartPole	fact.	CNN-GRU	2	-7.88 (-8.88, -6.87)	-7.88	-1.46
CartPole	fact.	Transformer	2	4.95 (3.59, 6.28)	4.95	6.58
CartPole	fact.	CNN-GRU	5	2.61 (2.18, 3.04)	2.61	-3.16
CartPole	fact.	Transformer	5	5.92 (5.25, 6.57)	7.98	-0.53
CartPole	fact.	CNN-GRU	8	-1.35 (-2.21, -0.50)	-1.35	-1.77
CartPole	fact.	Transformer	8	-21.82 (-23.36, -20.31)	-21.82	-1.21
PointMass2D	hom.	CNN-GRU	$(2, 5)$	8.58 (7.80, 9.37)	8.58	1.83
PointMass2D	hom.	Transformer	$(2, 5)$	24.53 (22.86, 26.08)	24.53	-2.05
PointMass2D	hom.	CNN-GRU	$(5, 8)$	11.77 (10.77, 12.76)	12.36	3.02
PointMass2D	hom.	Transformer	$(5, 8)$	18.62 (17.93, 19.29)	18.62	0.04
PointMass2D	hom.	CNN-GRU	$(8, 2)$	9.07 (7.85, 10.21)	9.63	-1.70
PointMass2D	hom.	Transformer	$(8, 2)$	15.89 (14.34, 17.37)	15.89	-0.81
PointMass2D	fact.	CNN-GRU	$(2, 5)$	-0.15 (-0.28, -0.03)	-0.15	-0.26
PointMass2D	fact.	Transformer	$(2, 5)$	9.52 (9.22, 9.83)	9.52	0.77
PointMass2D	fact.	CNN-GRU	$(5, 8)$	0.41 (0.28, 0.53)	0.41	0.73
PointMass2D	fact.	Transformer	$(5, 8)$	0.91 (0.71, 1.09)	0.91	0.96

System	Context	Model	Offset	Gain (95% CI), %	Oracle sensitivity (%)	Ordinary gain (%)
PointMass2D	fact.	CNN-GRU	$(8, 2)$	-0.15 (-0.27, -0.03)	-0.15	-0.49
PointMass2D	fact.	Transformer	$(8, 2)$	0.72 (0.49, 0.96)	0.72	-0.14

F.3 COORDINATEWISE POINTMASS2D OUTCOMES

The two-dimensional system tests whether one scalar lag is adequate when coordinates have different offsets. Every response intervention and error is computed separately for the two coordinates. Table 10 shows that the right-boundary model receives the correct coordinatewise lags, including the asymmetric $(8, 2)$ and $(2, 5)$ cases.

Table 10: Coordinatewise PointMass2D results. Gain uses the validation-selected non-oracle model. Oracle sensitivity compares oracle with timestamp timing. Context entries are homogeneous (hom.) or factorized (fact.).

System	Ctx.	Model	Coord.	Lag	Gain (%)	Oracle sensitivity (%)
PointMass2D	hom.	CNN-GRU	1	2	7.50	7.50
PointMass2D	hom.	CNN-GRU	2	5	9.64	9.64
PointMass2D	hom.	Transformer	1	2	14.63	14.63
PointMass2D	hom.	Transformer	2	5	32.46	32.46
PointMass2D	hom.	CNN-GRU	1	5	10.85	10.96
PointMass2D	hom.	CNN-GRU	2	8	12.65	13.69
PointMass2D	hom.	Transformer	1	5	19.68	19.68
PointMass2D	hom.	Transformer	2	8	17.57	17.57
PointMass2D	hom.	CNN-GRU	1	8	9.85	10.97
PointMass2D	hom.	CNN-GRU	2	2	8.27	8.22
PointMass2D	hom.	Transformer	1	8	16.82	16.82
PointMass2D	hom.	Transformer	2	2	14.94	14.94
PointMass2D	fact.	CNN-GRU	1	2	-0.02	-0.02
PointMass2D	fact.	CNN-GRU	2	5	-0.28	-0.28
PointMass2D	fact.	Transformer	1	2	14.11	14.11
PointMass2D	fact.	Transformer	2	5	1.47	1.47
PointMass2D	fact.	CNN-GRU	1	5	0.54	0.54
PointMass2D	fact.	CNN-GRU	2	8	0.27	0.27
PointMass2D	fact.	Transformer	1	5	1.16	1.16
PointMass2D	fact.	Transformer	2	8	0.65	0.65
PointMass2D	fact.	CNN-GRU	1	8	-0.00	0.04
PointMass2D	fact.	CNN-GRU	2	2	-0.30	-0.33
PointMass2D	fact.	Transformer	1	8	1.18	1.18
PointMass2D	fact.	Transformer	2	2	0.28	0.28

F.4 AGGREGATE CHECKS

The aggregate comparisons separate registration accuracy from learned response. Oracle and right-boundary models are identical in all 36 cells. Oracle timing improves response relative to timestamp timing by 5.85% overall (95% CI, 5.53% to 6.15%). The coordinate-level gain is 4.87% for the CNN-GRU (95% CI, 4.59% to 5.15%) and 7.87% for the patch Transformer (95% CI, 7.49% to 8.25%). Against the validation-selected non-oracle baseline, CartPole instead has a negative CI, as reported in Table 2.

G TRAINING BUDGET AND RESPONSE SUPERVISION

G.1 QUESTIONS AND MATCHED COMPARISONS

The training study tests two explanations for inaccurate action response after correct registration. A finite model may need more optimization, or its loss may assign too little accuracy to the response component. The first comparison extends the original training budget. The second compares two objectives with the same counterfactual labels, isolating the effect of response weighting from the effect of additional supervision.

Three objectives use the same two neural architectures. Factual prediction minimizes the observed next-feature error. Paired prediction minimizes R_{pair}, using both simulator outcomes for each coordinate intervention. Paired prediction plus response minimizes $R_{\mathrm{pair}} + R_{\mathrm{resp}}/4$, following Equation 21. The response coefficient was fixed before evaluating the new test panel. The paired objectives share every branch target, initialization, training permutation and optimizer update count. Factual prediction receives only the observed branch and therefore uses fewer labels and branch evaluations.

The study holds temporal correspondence correct throughout. It uses lag 5 for scalar actions and $(5, 8)$ for PointMass2D, the middle entries of the original offset lists, across all systems and contexts. These fixed offsets define six generated settings. Crossing them with the two architectures gives twelve conditions. Three initializations and three objectives yield 108 fits. All conditions enter the reported summaries.

G.2 Data, optimization and checkpoint selection

The original data-generation law, architectures and optimizer remain unchanged. Each setting contains 9,472 training rows, 2,368 validation rows and 7,104 rows from the original test panel. A new test panel contains another 192 independently generated trajectories, or 7,104 rows. All architectures and objectives within a setting share the same trajectories. The new panel is the primary test; the original panel connects the training comparison with the earlier alignment study. New initializations and the evaluation precision are recorded separately, so the new 12-epoch fit is a within-study budget comparison.

Training runs for 48 epochs with AdamW, learning rate 0.002, weight decay 10^{-5}, gradient-norm clipping at 5 and effective batch size 512. The loss sums the 64 feature errors and averages over the logical batch, matching the alignment study's optimization scale. Stored risks average over features and, for response risk, intervention coordinates. Both paired objectives evaluate two branches per action coordinate. Shared image encoding reuses the same history features across branches without changing the model function.

Every epoch records the online training loss, a fixed 512-row training diagnostic subset and the full validation panel. Fixed checkpoints at epochs 12, 24 and 48 are evaluated independently of their test performance. A separate selected model minimizes factual validation risk over epochs 0 through 48, retaining the earliest minimizer. Response validation risk does not select this model. Figure 5 reports the complete validation histories, including epoch zero.

G.3 Risk definitions and sampling units

The ordinary-prediction reference copies the current dynamic feature. Its risk is $R_{\mathrm{persist}} = \mathbb{E}\|\phi(X_{t+1}) - \phi(X_t)\|^2/64$. We report $R_{\mathrm{factual}}/R_{\mathrm{persist}}$ and R_{resp}/R_0, where values below one favor the learned model. Neither ratio is an accuracy percentage. A condition mean averages the three initialization risks before dividing by the shared reference risk. Counts of conditions below one are descriptive counts of these means, not counts of individual trained models or simultaneous significance tests.

Uncertainty includes both initialization and held-out trajectory variation. Each crossed bootstrap replicate independently samples the initialization axis and a test panel: twelve top-level test seeds, followed by sixteen trajectories inside each selected seed. The same sampled panel is applied to every selected initialization and matched objective. Step-level errors are first averaged within each trajectory. The 2,000-replicate percentile intervals are pointwise 95% intervals, conditional on the shared training and validation data and selected checkpoints. Individual initialization results are reported alongside them.

G.4 Complete final risks

The final risks show a broad improvement from response supervision within the controlled panel. Table 11 reports all twelve condition means. Factual training beats persistence in eleven conditions and zero response in five. Paired prediction brings seven conditions below zero-response risk, and response weighting brings eight below it. Four response-weighted conditions remain above the

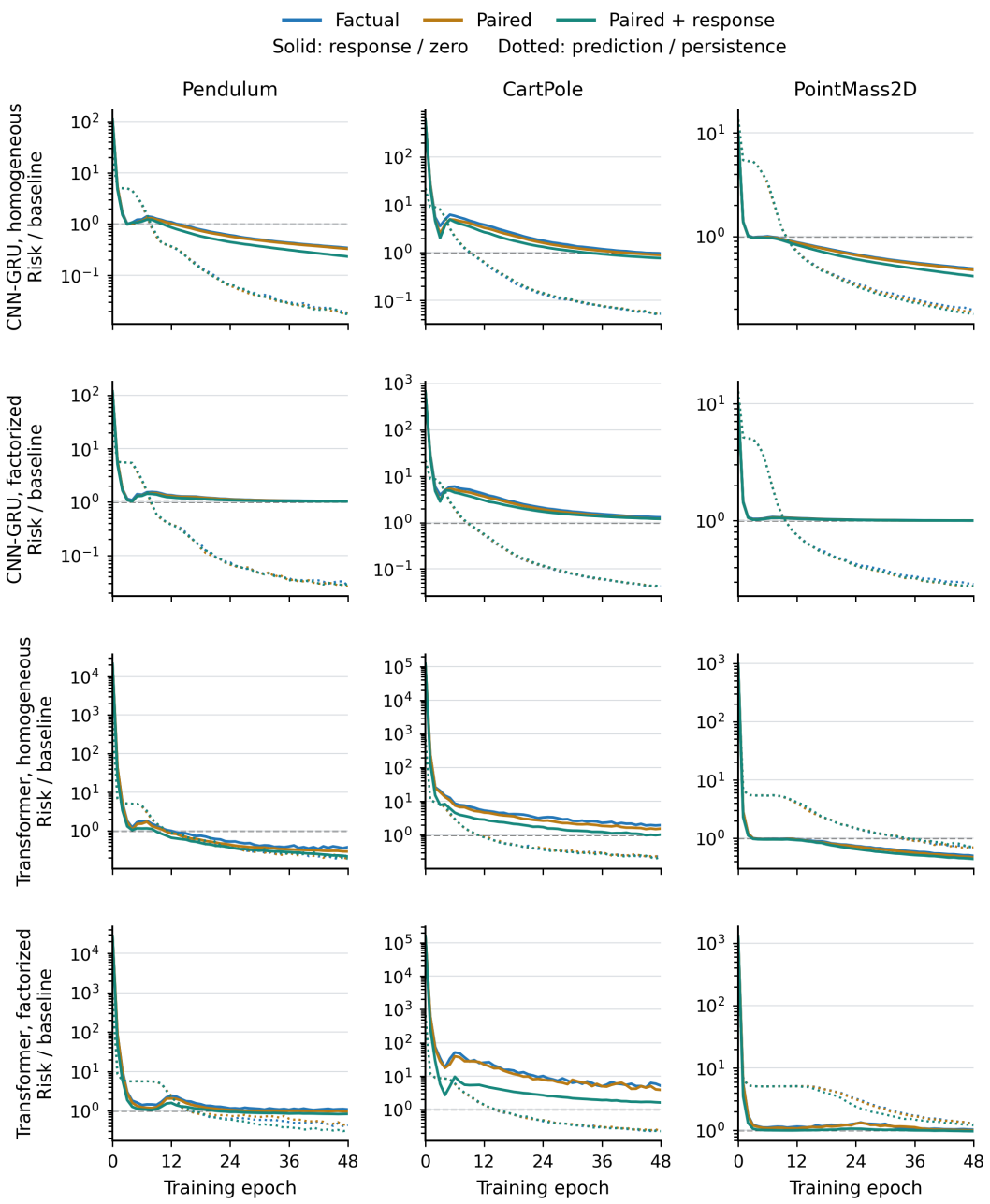

Figure 5: Prediction and response improve at different rates. Each panel is one system-context-architecture condition, with means over three initializations. Solid curves show validation response risk relative to zero response; dotted curves show validation prediction risk relative to frame persistence. Color identifies the training objective. The horizontal line marks equality to each reference. All 49 recorded epochs are shown, including initialization.

zero reference, including all three factorized CNN-GRU conditions. The factorized PointMass2D Transformer also remains worse than persistence on ordinary prediction.

The matched-label comparison identifies an effect of the learning objective. Response weighting improves eleven of twelve final condition means, and nine pointwise intervals exclude zero improvement. Its largest response-ratio reduction occurs for the factorized CartPole Transformer, although the resulting response still exceeds the zero reference. In homogeneous Pendulum, both architec-

Table 11: Final risk ratios on the fresh test panel, averaged over three initializations. The first three numeric columns give response risk relative to zero; the last three give factual prediction risk relative to persistence. H and F denote homogeneous and factorized contexts.

		Response / zero			Factual / persistence		
Setting	Model	Factual	Paired	+ response	Factual	Paired	+ response
Pendulum H	CNN-GRU	0.342	0.330	0.230	0.020	0.019	0.019
Pendulum H	Transformer	0.373	0.281	0.214	0.226	0.198	0.197
Pendulum F	CNN-GRU	1.036	1.032	1.024	0.025	0.024	0.026
Pendulum F	Transformer	1.094	0.967	0.837	0.414	0.415	0.310
CartPole H	CNN-GRU	0.959	0.889	0.762	0.047	0.047	0.047
CartPole H	Transformer	1.888	1.479	0.982	0.212	0.216	0.211
CartPole F	CNN-GRU	1.307	1.212	1.210	0.042	0.042	0.042
CartPole F	Transformer	5.309	3.983	1.631	0.232	0.228	0.226
PointMass2D H	CNN-GRU	0.526	0.512	0.449	0.220	0.210	0.201
PointMass2D H	Transformer	0.533	0.505	0.473	0.791	0.785	0.808
PointMass2D F	CNN-GRU	1.005	1.002	1.002	0.296	0.285	0.286
PointMass2D F	Transformer	1.034	0.999	0.972	1.362	1.311	1.242

tures already beat zero with paired prediction and improve further under response weighting. Thus the benefit includes more accurate responses in successful conditions as well as reductions in still-inaccurate responses.

Table 12: Matched-label response-supervision contrasts. Positive differences favor paired prediction plus response loss over paired prediction. Each numeric cell gives the estimate above its pointwise 95% confidence interval.

Setting	Model	Epoch	Response-ratio reduction	Factual-ratio reduction
Pendulum H	CNN-GRU	48	0.101 (0.082 to 0.120)	0.001 (-0.002 to 0.003)
Pendulum H	CNN-GRU	Val.	0.107 (0.095 to 0.120)	0.001 (-0.001 to 0.003)
Pendulum H	Transformer	48	0.067 (0.043 to 0.090)	0.001 (-0.025 to 0.043)
Pendulum H	Transformer	Val.	0.067 (0.051 to 0.085)	0.003 (-0.025 to 0.022)
Pendulum F	CNN-GRU	48	0.009 (0.005 to 0.012)	-0.002 (-0.006 to 0.000)
Pendulum F	CNN-GRU	Val.	0.008 (0.006 to 0.011)	-0.000210 (-0.000471 to 0.000118)
Pendulum F	Transformer	48	0.130 (0.081 to 0.191)	0.106 (0.007 to 0.239)
Pendulum F	Transformer	Val.	0.130 (0.082 to 0.188)	0.120 (0.006 to 0.278)
CartPole H	CNN-GRU	48	0.128 (0.109 to 0.146)	0.000051 (-0.000830 to 0.000997)
CartPole H	CNN-GRU	Val.	0.129 (0.107 to 0.154)	0.000313 (-0.001395 to 0.002059)
CartPole H	Transformer	48	0.497 (0.238 to 0.828)	0.005 (-0.028 to 0.027)
CartPole H	Transformer	Val.	0.535 (0.274 to 0.844)	0.024 (-0.024 to 0.064)
CartPole F	CNN-GRU	48	0.002 (-0.011 to 0.012)	0.000051 (-0.001016 to 0.001446)
CartPole F	CNN-GRU	Val.	0.007 (-0.005 to 0.016)	-0.000177 (-0.000931 to 0.000512)
CartPole F	Transformer	48	2.352 (0.594 to 5.733)	0.002 (-0.004 to 0.007)
CartPole F	Transformer	Val.	2.583 (0.746 to 5.896)	0.007 (-0.001 to 0.016)
PointMass2D H	CNN-GRU	48	0.063 (0.058 to 0.067)	0.009 (0.005 to 0.013)
PointMass2D H	CNN-GRU	Val.	0.067 (0.059 to 0.073)	0.009 (0.004 to 0.013)

Setting	Model	Epoch	Response-ratio reduction	Factual-ratio reduction
PointMass2D H	Transformer	48	0.032 (0.006 to 0.054)	-0.023 (-0.076 to 0.018)
PointMass2D H	Transformer	Val.	0.033 (0.009 to 0.063)	-0.038 (-0.107 to 0.027)
PointMass2D F	CNN-GRU	48	-0.000064 (-0.000316 to 0.000180)	-0.001 (-0.003 to 0.000)
PointMass2D F	CNN-GRU	Val.	0.000221 (-0.000157 to 0.000692)	-0.003 (-0.008 to -0.000)
PointMass2D F	Transformer	48	0.027 (-0.001 to 0.047)	0.070 (-0.082 to 0.318)
PointMass2D F	Transformer	Val.	0.028 (-0.003 to 0.047)	0.080 (-0.059 to 0.331)

G.5 Full intervals and validation-selected models

Validation selection retains the main pattern without using test response to select a checkpoint. Eleven factual condition means beat persistence and five beat zero, while eight response-weighted condition means beat zero. The matched-label response difference is positive in all twelve selected-model condition means, with nine pointwise intervals excluding zero. Table 13 includes both final and validation-selected models for every objective.

Table 13: Response and factual risk ratios on the fresh test panel. H and F denote homogeneous and factorized contexts; Transf. denotes the patch Transformer; + response adds response loss to paired prediction. Val. selects the lowest factual validation risk. Parentheses give pointwise 95% bootstrap confidence intervals.

Setting	Model	Epoch	Loss	Response / zero	Factual / persistence
Pendulum H	CNN-GRU	48	Factual	0.342 (0.302 to 0.387)	0.020 (0.017 to 0.023)
Pendulum H	CNN-GRU	48	Paired	0.330 (0.291 to 0.373)	0.019 (0.017 to 0.022)
Pendulum H	CNN-GRU	48	+ response	0.230 (0.204 to 0.256)	0.019 (0.015 to 0.023)
Pendulum H	CNN-GRU	Val.	Factual	0.345 (0.306 to 0.384)	0.020 (0.017 to 0.023)
Pendulum H	CNN-GRU	Val.	Paired	0.340 (0.303 to 0.379)	0.019 (0.016 to 0.022)
Pendulum H	CNN-GRU	Val.	+ response	0.234 (0.207 to 0.262)	0.018 (0.015 to 0.022)
Pendulum H	Transf.	48	Factual	0.373 (0.313 to 0.427)	0.226 (0.202 to 0.254)
Pendulum H	Transf.	48	Paired	0.281 (0.234 to 0.338)	0.198 (0.171 to 0.223)
Pendulum H	Transf.	48	+ response	0.214 (0.168 to 0.268)	0.197 (0.168 to 0.225)
Pendulum H	Transf.	Val.	Factual	0.380 (0.332 to 0.419)	0.186 (0.169 to 0.207)
Pendulum H	Transf.	Val.	Paired	0.284 (0.240 to 0.337)	0.190 (0.168 to 0.214)
Pendulum H	Transf.	Val.	+ response	0.217 (0.174 to 0.268)	0.187 (0.168 to 0.209)
Pendulum F	CNN-GRU	48	Factual	1.036 (1.028 to 1.044)	0.025 (0.022 to 0.029)
Pendulum F	CNN-GRU	48	Paired	1.032 (1.024 to 1.040)	0.024 (0.021 to 0.027)
Pendulum F	CNN-GRU	48	+ response	1.024 (1.018 to 1.029)	0.026 (0.022 to 0.032)
Pendulum F	CNN-GRU	Val.	Factual	1.036 (1.028 to 1.044)	0.025 (0.022 to 0.029)
Pendulum F	CNN-GRU	Val.	Paired	1.033 (1.026 to 1.041)	0.024 (0.021 to 0.027)
Pendulum F	CNN-GRU	Val.	+ response	1.025 (1.019 to 1.030)	0.024 (0.021 to 0.027)
Pendulum F	Transf.	48	Factual	1.094 (0.982 to 1.193)	0.414 (0.303 to 0.519)
Pendulum F	Transf.	48	Paired	0.967 (0.885 to 1.028)	0.415 (0.293 to 0.508)
Pendulum F	Transf.	48	+ response	0.837 (0.762 to 0.927)	0.310 (0.256 to 0.378)
Pendulum F	Transf.	Val.	Factual	1.107 (0.985 to 1.224)	0.374 (0.298 to 0.426)
Pendulum F	Transf.	Val.	Paired	0.967 (0.886 to 1.033)	0.411 (0.288 to 0.517)
Pendulum F	Transf.	Val.	+ response	0.838 (0.762 to 0.936)	0.290 (0.244 to 0.343)
CartPole H	CNN-GRU	48	Factual	0.959 (0.909 to 1.008)	0.047 (0.041 to 0.053)
CartPole H	CNN-GRU	48	Paired	0.889 (0.842 to 0.939)	0.047 (0.040 to 0.054)
CartPole H	CNN-GRU	48	+ response	0.762 (0.724 to 0.800)	0.047 (0.041 to 0.054)
CartPole H	CNN-GRU	Val.	Factual	0.967 (0.910 to 1.032)	0.046 (0.040 to 0.052)
CartPole H	CNN-GRU	Val.	Paired	0.896 (0.847 to 0.958)	0.046 (0.039 to 0.054)
CartPole H	CNN-GRU	Val.	+ response	0.767 (0.729 to 0.810)	0.046 (0.040 to 0.052)
CartPole H	Transf.	48	Factual	1.888 (1.710 to 2.056)	0.212 (0.186 to 0.235)
CartPole H	Transf.	48	Paired	1.479 (1.239 to 1.768)	0.216 (0.195 to 0.234)
CartPole H	Transf.	48	+ response	0.982 (0.902 to 1.056)	0.211 (0.178 to 0.255)
CartPole H	Transf.	Val.	Factual	1.886 (1.678 to 2.097)	0.184 (0.170 to 0.199)

Setting	Model	Epoch	Loss	Response / zero	Factual / persistence
CartPole H	Transf.	Val.	Paired	1.483 (1.256 to 1.772)	0.197 (0.164 to 0.229)
CartPole H	Transf.	Val.	+ response	0.947 (0.873 to 1.026)	0.173 (0.156 to 0.192)
CartPole F	CNN-GRU	48	Factual	1.307 (1.252 to 1.366)	0.042 (0.036 to 0.049)
CartPole F	CNN-GRU	48	Paired	1.212 (1.178 to 1.249)	0.042 (0.036 to 0.049)
CartPole F	CNN-GRU	48	+ response	1.210 (1.170 to 1.246)	0.042 (0.036 to 0.049)
CartPole F	CNN-GRU	Val.	Factual	1.307 (1.258 to 1.359)	0.041 (0.034 to 0.048)
CartPole F	CNN-GRU	Val.	Paired	1.225 (1.194 to 1.263)	0.041 (0.035 to 0.049)
CartPole F	CNN-GRU	Val.	+ response	1.218 (1.184 to 1.253)	0.042 (0.035 to 0.049)
CartPole F	Transf.	48	Factual	5.309 (2.646 to 10.254)	0.232 (0.212 to 0.254)
CartPole F	Transf.	48	Paired	3.983 (2.187 to 7.314)	0.228 (0.206 to 0.250)
CartPole F	Transf.	48	+ response	1.631 (1.547 to 1.742)	0.226 (0.204 to 0.248)
CartPole F	Transf.	Val.	Factual	6.371 (2.657 to 13.223)	0.222 (0.203 to 0.243)
CartPole F	Transf.	Val.	Paired	4.254 (2.433 to 7.594)	0.227 (0.205 to 0.249)
CartPole F	Transf.	Val.	+ response	1.671 (1.546 to 1.774)	0.220 (0.201 to 0.240)
PointMass2D H	CNN-GRU	48	Factual	0.526 (0.501 to 0.555)	0.220 (0.203 to 0.241)
PointMass2D H	CNN-GRU	48	Paired	0.512 (0.485 to 0.542)	0.210 (0.195 to 0.230)
PointMass2D H	CNN-GRU	48	+ response	0.449 (0.422 to 0.478)	0.201 (0.185 to 0.222)
PointMass2D H	CNN-GRU	Val.	Factual	0.526 (0.499 to 0.556)	0.220 (0.203 to 0.238)
PointMass2D H	CNN-GRU	Val.	Paired	0.516 (0.488 to 0.546)	0.210 (0.194 to 0.228)
PointMass2D H	CNN-GRU	Val.	+ response	0.449 (0.421 to 0.480)	0.201 (0.185 to 0.219)
PointMass2D H	Transf.	48	Factual	0.533 (0.489 to 0.587)	0.791 (0.722 to 0.856)
PointMass2D H	Transf.	48	Paired	0.505 (0.456 to 0.564)	0.785 (0.707 to 0.863)
PointMass2D H	Transf.	48	+ response	0.473 (0.432 to 0.518)	0.808 (0.699 to 0.897)
PointMass2D H	Transf.	Val.	Factual	0.541 (0.496 to 0.590)	0.769 (0.688 to 0.844)
PointMass2D H	Transf.	Val.	Paired	0.510 (0.457 to 0.575)	0.759 (0.684 to 0.839)
PointMass2D H	Transf.	Val.	+ response	0.478 (0.440 to 0.519)	0.796 (0.690 to 0.891)
PointMass2D F	CNN-GRU	48	Factual	1.005 (1.004 to 1.006)	0.296 (0.276 to 0.321)
PointMass2D F	CNN-GRU	48	Paired	1.002 (1.002 to 1.003)	0.285 (0.263 to 0.310)
PointMass2D F	CNN-GRU	48	+ response	1.002 (1.002 to 1.003)	0.286 (0.264 to 0.311)
PointMass2D F	CNN-GRU	Val.	Factual	1.005 (1.004 to 1.006)	0.295 (0.275 to 0.319)
PointMass2D F	CNN-GRU	Val.	Paired	1.003 (1.002 to 1.003)	0.282 (0.261 to 0.307)
PointMass2D F	CNN-GRU	Val.	+ response	1.003 (1.002 to 1.003)	0.285 (0.264 to 0.312)
PointMass2D F	Transf.	48	Factual	1.034 (1.002 to 1.060)	1.362 (1.223 to 1.475)
PointMass2D F	Transf.	48	Paired	0.999 (0.966 to 1.026)	1.311 (1.188 to 1.417)
PointMass2D F	Transf.	48	+ response	0.972 (0.961 to 0.983)	1.242 (1.051 to 1.436)
PointMass2D F	Transf.	Val.	Factual	1.037 (1.008 to 1.060)	1.361 (1.228 to 1.476)
PointMass2D F	Transf.	Val.	Paired	0.999 (0.964 to 1.026)	1.311 (1.192 to 1.416)
PointMass2D F	Transf.	Val.	+ response	0.971 (0.960 to 0.982)	1.231 (1.029 to 1.429)

G.6 Independent model initializations

The repeated initializations reveal variation that the original alignment-study intervals did not measure. Table 14 gives every final response ratio separately. The initialization seeds are 202609091, 202609193 and 202609297, in the listed order. Each seed determines initialization and training order, with both matched across objectives. All nine final conditions whose paired-difference intervals exclude zero improve at every initialization. The remaining three conditions have mixed signs. All fits share each setting's training and validation trajectories, so these repetitions measure initialization and order variation rather than variation across independently sampled training sets.

Table 14: Final response-risk ratios for every model initialization on the fresh test panel. Initialization indices identify the three seeds in their listed order.

Setting	Model	Init.	Factual	Paired	+ response
Pendulum H	CNN-GRU	1	0.349	0.332	0.229
Pendulum H	CNN-GRU	2	0.315	0.307	0.225
Pendulum H	CNN-GRU	3	0.363	0.352	0.235
Pendulum H	Transformer	1	0.383	0.247	0.204
Pendulum H	Transformer	2	0.316	0.262	0.171
Pendulum H	Transformer	3	0.420	0.335	0.266
Pendulum F	CNN-GRU	1	1.040	1.038	1.026

Setting	Model	Init.	Factual	Paired	+ response
Pendulum F	CNN-GRU	2	1.039	1.035	1.026
Pendulum F	CNN-GRU	3	1.028	1.024	1.019
Pendulum F	Transformer	1	0.983	0.884	0.776
Pendulum F	Transformer	2	1.109	1.013	0.932
Pendulum F	Transformer	3	1.191	1.003	0.803
CartPole H	CNN-GRU	1	0.993	0.937	0.790
CartPole H	CNN-GRU	2	0.970	0.878	0.751
CartPole H	CNN-GRU	3	0.914	0.853	0.743
CartPole H	Transformer	1	1.938	1.432	1.012
CartPole H	Transformer	2	2.003	1.774	0.931
CartPole H	Transformer	3	1.724	1.233	1.005
CartPole F	CNN-GRU	1	1.367	1.250	1.242
CartPole F	CNN-GRU	2	1.306	1.210	1.222
CartPole F	CNN-GRU	3	1.248	1.178	1.166
CartPole F	Transformer	1	2.886	2.367	1.741
CartPole F	Transformer	2	10.379	7.385	1.588
CartPole F	Transformer	3	2.663	2.196	1.563
PointMass2D H	CNN-GRU	1	0.534	0.524	0.466
PointMass2D H	CNN-GRU	2	0.517	0.499	0.436
PointMass2D H	CNN-GRU	3	0.528	0.512	0.446
PointMass2D H	Transformer	1	0.504	0.477	0.440
PointMass2D H	Transformer	2	0.508	0.470	0.465
PointMass2D H	Transformer	3	0.587	0.567	0.512
PointMass2D F	CNN-GRU	1	1.005	1.002	1.002
PointMass2D F	CNN-GRU	2	1.005	1.003	1.003
PointMass2D F	CNN-GRU	3	1.005	1.002	1.002
PointMass2D F	Transformer	1	1.052	1.011	0.973
PointMass2D F	Transformer	2	0.995	0.961	0.965
PointMass2D F	Transformer	3	1.054	1.023	0.977

G.7 Budget and test-panel comparisons

Longer factual training changes the absolute outcome in several conditions. On the fresh test panel, the number beating persistence rises from eight at 12 epochs to ten at 24 and eleven at 48. The corresponding zero-response counts are three, four and five. These results attribute part of the early deficit to the finite training budget while preserving the separation between useful ordinary prediction and accurate response. Table 15 reports all objectives and budgets, together with the original test panel at 48 epochs.

The learning curves also show continued fitting near the final budget. From epoch 40 to 48, factual-training validation prediction risk falls by 2.06% to 29.65% across the twelve condition means. Many factual-validation selections occur at or near epoch 48. The comparison therefore establishes a response deficit at specified budgets and a matched-objective improvement within those budgets. Agreement between final and selected models checks sensitivity to checkpoint choice; it does not establish optimization convergence.

Table 15: Response-risk ratios across all fixed training budgets and both test panels. The first three numeric columns use fresh test trajectories; the last uses the original test trajectories at 48 epochs.

Setting	Model	Objective	12	24	48	Original test
Pendulum H	CNN-GRU	Factual	1.052	0.597	0.342	0.337
Pendulum H	CNN-GRU	Paired	0.991	0.570	0.330	0.325
Pendulum H	CNN-GRU	Paired + response	0.837	0.444	0.230	0.224
Pendulum H	Transformer	Factual	0.972	0.507	0.373	0.365
Pendulum H	Transformer	Paired	0.813	0.421	0.281	0.276
Pendulum H	Transformer	Paired + response	0.649	0.357	0.214	0.210
Pendulum F	CNN-GRU	Factual	1.338	1.131	1.036	1.041
Pendulum F	CNN-GRU	Paired	1.292	1.115	1.032	1.037
Pendulum F	CNN-GRU	Paired + response	1.218	1.080	1.024	1.028

Setting	Model	Objective	12	24	48	Original test
Pendulum F	Transformer	Factual	2.380	1.163	1.094	1.104
Pendulum F	Transformer	Paired	2.064	1.049	0.967	0.976
Pendulum F	Transformer	Paired + response	1.584	0.941	0.837	0.845
CartPole H	CNN-GRU	Factual	3.822	1.768	0.959	1.000
CartPole H	CNN-GRU	Paired	3.288	1.583	0.889	0.928
CartPole H	CNN-GRU	Paired + response	2.672	1.310	0.762	0.792
CartPole H	Transformer	Factual	5.181	3.238	1.888	2.028
CartPole H	Transformer	Paired	4.509	2.591	1.479	1.588
CartPole H	Transformer	Paired + response	2.792	1.659	0.982	1.048
CartPole F	CNN-GRU	Factual	4.265	2.118	1.307	1.280
CartPole F	CNN-GRU	Paired	3.772	1.954	1.212	1.195
CartPole F	CNN-GRU	Paired + response	3.093	1.751	1.210	1.193
CartPole F	Transformer	Factual	25.389	10.263	5.309	4.863
CartPole F	Transformer	Paired	21.716	8.080	3.983	3.680
CartPole F	Transformer	Paired + response	5.138	2.719	1.631	1.581
PointMass2D H	CNN-GRU	Factual	0.894	0.701	0.526	0.517
PointMass2D H	CNN-GRU	Paired	0.883	0.690	0.512	0.502
PointMass2D H	CNN-GRU	Paired + response	0.847	0.637	0.449	0.440
PointMass2D H	Transformer	Factual	0.981	0.763	0.533	0.526
PointMass2D H	Transformer	Paired	0.963	0.746	0.505	0.497
PointMass2D H	Transformer	Paired + response	0.947	0.682	0.473	0.466
PointMass2D F	CNN-GRU	Factual	1.055	1.020	1.005	1.005
PointMass2D F	CNN-GRU	Paired	1.048	1.016	1.002	1.002
PointMass2D F	CNN-GRU	Paired + response	1.040	1.014	1.002	1.002
PointMass2D F	Transformer	Factual	1.153	1.277	1.034	1.028
PointMass2D F	Transformer	Paired	1.072	1.289	0.999	0.995
PointMass2D F	Transformer	Paired + response	1.006	1.064	0.972	0.966

G.8 Numerical precision and retained statistics

The main training-study evaluation uses 32-bit floating-point inference. Training uses bfloat16 autocasting, and each final model is also evaluated under that autocasting mode. Table 16 reports both results on identical fresh test trajectories. This comparison measures whether rounding the two predictions changes their small difference; it does not retrain either model. The alignment study retains its original inference precision and is reported separately.

Absolute response quality is precision-sensitive for the CartPole Transformer. In the homogeneous condition, response-weighted prediction has response-to-zero ratio 0.982 in FP32 and 1.578 with bfloat16 autocasting. The number of response-weighted conditions below zero therefore falls from eight to seven. The direction of the paired-objective contrast remains positive in eleven of twelve condition means under either precision. Only aggregate final-model bfloat16 metrics were retained, so its table reports point estimates without trajectory-level intervals.

Table 16: Inference-precision comparison at 48 epochs on the fresh test panel. Training uses bfloat16 in every condition; the main evaluation uses 32-bit floating-point inference.

Setting	Model	Objective	32-bit response / zero	bfloat16 response / zero
Pendulum H	CNN-GRU	Factual	0.342	0.346
Pendulum H	CNN-GRU	Paired	0.330	0.334
Pendulum H	CNN-GRU	Paired + response	0.230	0.234
Pendulum H	Transformer	Factual	0.373	0.486
Pendulum H	Transformer	Paired	0.281	0.398
Pendulum H	Transformer	Paired + response	0.214	0.333
Pendulum F	CNN-GRU	Factual	1.036	1.040
Pendulum F	CNN-GRU	Paired	1.032	1.036
Pendulum F	CNN-GRU	Paired + response	1.024	1.027
Pendulum F	Transformer	Factual	1.094	1.208
Pendulum F	Transformer	Paired	0.967	1.084
Pendulum F	Transformer	Paired + response	0.837	0.945

Setting	Model	Objective	32-bit response / zero	bfloat16 response / zero
CartPole H	CNN-GRU	Factual	0.959	0.982
CartPole H	CNN-GRU	Paired	0.889	0.905
CartPole H	CNN-GRU	Paired + response	0.762	0.779
CartPole H	Transformer	Factual	1.888	2.528
CartPole H	Transformer	Paired	1.479	2.119
CartPole H	Transformer	Paired + response	0.982	1.578
CartPole F	CNN-GRU	Factual	1.307	1.329
CartPole F	CNN-GRU	Paired	1.212	1.243
CartPole F	CNN-GRU	Paired + response	1.210	1.234
CartPole F	Transformer	Factual	5.309	6.018
CartPole F	Transformer	Paired	3.983	4.879
CartPole F	Transformer	Paired + response	1.631	2.326
PointMass2D H	CNN-GRU	Factual	0.526	0.526
PointMass2D H	CNN-GRU	Paired	0.512	0.512
PointMass2D H	CNN-GRU	Paired + response	0.449	0.450
PointMass2D H	Transformer	Factual	0.533	0.544
PointMass2D H	Transformer	Paired	0.505	0.516
PointMass2D H	Transformer	Paired + response	0.473	0.484
PointMass2D F	CNN-GRU	Factual	1.005	1.005
PointMass2D F	CNN-GRU	Paired	1.002	1.003
PointMass2D F	CNN-GRU	Paired + response	1.002	1.003
PointMass2D F	Transformer	Factual	1.034	1.045
PointMass2D F	Transformer	Paired	0.999	1.009
PointMass2D F	Transformer	Paired + response	0.972	0.982

The evidence supports additional CPU analysis without retraining. Every reported checkpoint retains trajectory-level factual risk, persistence risk, response risk, zero-response energy, predicted energy, alignment, paired risk, the two branch risks and their error cross term. Coordinate-specific response and energy terms are also retained. These quantities reconstruct the ratios and intervals and check the exact identities in Appendix B.7. Full training histories distinguish ongoing fitting from changes in the relative response metric.

G.9 Empirical risk decomposition

Shared prediction error explains why the sufficient response bound can be conservative. Define $M = R_{\mathrm{pair}} - R_{\mathrm{resp}}/4$, the mean squared within-pair average error, and $B = 4R_{\mathrm{pair}}/R_0$. Equation 21 gives $B = 4M/R_0 + R_{\mathrm{resp}}/R_0$. None of the 36 final condition-objective means meets $B < 1$, although twenty beat zero response directly. The sufficient condition therefore certifies none of these empirical successes. Table 17 retains every mean and both decisions.

The homogeneous Pendulum Transformer gives a concrete example of error cancellation. With response weighting, its paired risk is $5.171R_0$, while response risk is only $0.214R_0$. The normalized sum of branch risks is 10.341, and twice their error cross term is 10.128. Subtraction before rounding gives the response ratio 0.214. Thus a large component shared by the two branch errors cancels, allowing an accurate action difference despite a loose bound on total prediction error. Direct response evaluation captures this accuracy without requiring the sufficient condition to hold.

The bound and the identity serve different purposes. The bound gives the best universal guarantee obtainable from paired risk alone, since opposite branch errors attain its factor four. It specifies the response scale a prediction-only guarantee must reach. When paired outcomes are available, the identity additionally measures the shared component M, giving the exact response ratio $B - 4M/R_0$. Improving the certificate then requires information about this component, rather than a smaller universal constant. The matched-label experiment uses the same decomposition to change the response coefficient while retaining the shared-error coefficient.

Table 17: Paired-risk bounds at 48 epochs on the shared fresh-test panel with FP32 inference. Risks are averaged over three initializations before normalization. Here $B = 4R_{\mathrm{pair}}/R_0$ and $M = R_{\mathrm{pair}} - R_{\mathrm{resp}}/4$. F, P and PR denote factual, paired, and paired plus response losses; H and F in setting names denote homogeneous and factorized contexts. Yes/no decisions use unrounded values.

Setting	Model	Loss	B	M/R_0	R_{resp}/R_0	$B < 1$	$R_{\mathrm{resp}} < R_0$
Pendulum H	CNN-GRU	F	2.0846	0.4356	0.3422	No	Yes
Pendulum H	CNN-GRU	P	2.0547	0.4311	0.3304	No	Yes
Pendulum H	CNN-GRU	PR	1.9810	0.4379	0.2295	No	Yes
Pendulum H	Transformer	F	23.7794	5.8516	0.3730	No	Yes
Pendulum H	Transformer	P	20.7990	5.1294	0.2813	No	Yes
Pendulum H	Transformer	PR	20.6828	5.1173	0.2138	No	Yes
Pendulum F	CNN-GRU	F	2.5814	0.3863	1.0360	No	No
Pendulum F	CNN-GRU	P	2.4553	0.3558	1.0322	No	No
Pendulum F	CNN-GRU	PR	2.6751	0.4129	1.0236	No	No
Pendulum F	Transformer	F	41.5209	10.1067	1.0940	No	No
Pendulum F	Transformer	P	41.7476	10.1952	0.9666	No	Yes
Pendulum F	Transformer	PR	31.1355	7.5746	0.8370	No	Yes
CartPole H	CNN-GRU	F	27.0087	6.5124	0.9589	No	Yes
CartPole H	CNN-GRU	P	27.0858	6.5492	0.8891	No	Yes
CartPole H	CNN-GRU	PR	27.0436	6.5705	0.7616	No	Yes
CartPole H	Transformer	F	122.0734	30.0463	1.8884	No	No
CartPole H	Transformer	P	124.4408	30.7404	1.4793	No	No
CartPole H	Transformer	PR	121.5332	30.1377	0.9824	No	Yes
CartPole F	CNN-GRU	F	27.0836	6.4442	1.3067	No	No
CartPole F	CNN-GRU	P	27.2140	6.5004	1.2124	No	No
CartPole F	CNN-GRU	PR	27.1802	6.4926	1.2100	No	No
CartPole F	Transformer	F	149.1814	35.9681	5.3091	No	No
CartPole F	Transformer	P	146.9996	35.7542	3.9828	No	No
CartPole F	Transformer	PR	145.5560	35.9813	1.6308	No	No
PointMass2D H	CNN-GRU	F	3.0554	0.6323	0.5260	No	Yes
PointMass2D H	CNN-GRU	P	2.9189	0.6017	0.5119	No	Yes
PointMass2D H	CNN-GRU	PR	2.7979	0.5872	0.4491	No	Yes
PointMass2D H	Transformer	F	11.0313	2.6245	0.5331	No	Yes
PointMass2D H	Transformer	P	10.9431	2.6096	0.5046	No	Yes
PointMass2D H	Transformer	PR	11.2722	2.6999	0.4727	No	Yes
PointMass2D F	CNN-GRU	F	4.1653	0.7901	1.0050	No	No
PointMass2D F	CNN-GRU	P	3.9964	0.7485	1.0024	No	No
PointMass2D F	CNN-GRU	PR	4.0120	0.7524	1.0024	No	No
PointMass2D F	Transformer	F	19.1945	4.5402	1.0336	No	No
PointMass2D F	Transformer	P	18.4756	4.3693	0.9986	No	Yes
PointMass2D F	Transformer	PR	17.4952	4.1309	0.9717	No	Yes

H Public Walker/Walk Response Evaluation

H.1 Purpose and data collection

The public task tests response learning with official simulator dynamics and larger visual models. We use Walker/Walk from DeepMind Control Suite, implemented in `dm_control` and MuJoCo (Tunyasuvunakool et al., 2020). The task supplies a six-dimensional action and rendered observations. We generate trajectories for supervised next-feature prediction. The measured outcomes are prediction and response risks, rather than policy returns. Timing is correct throughout this study, so the comparison isolates response learning after the input has been identified.

The data contain 96 training episodes, 24 validation episodes and 32 test episodes. Each episode has four warmup steps followed by 48 prediction rows, giving 4,608, 1,152 and 1,536 rows, respectively. The model observes the previous and current 64×64 grayscale images. The target is the next image averaged over nonoverlapping 4×4 blocks, yielding 256 features. Camera zero renders the images.

Grayscale conversion uses red, green and blue weights 0.299, 0.587 and 0.114, followed by rounding to eight-bit integers, normalization to $[0, 1]$ and average pooling. The target representation is fixed.

Each row supplies two matched actions at the same simulator state. Starting from the previous factual action, the background command follows a first-order process with coefficient 0.72, independent Gaussian noise of standard deviation 0.55, and clipping to $[-0.95, 0.95]$. The intervened coordinate is $(i + t) \bmod 6$ for episode index i and step t, both starting at zero. That coordinate is set to $+0.8$ or -0.8, while all other coordinates retain their common background values. An independent equiprobable sign selects one branch as the factual transition. Thus the factual conditional action law is exactly the uniform mixture of the two paired actions, as required for the population-risk equality in Appendix B.7.

The two branches share the complete pre-action integration state. We copy MuJoCo's integration state, including its solver-related state, together with the task random state and episode counters. Position and velocity calculations are then refreshed without rerunning the constraint solver. The branch selected by the factual sign is evaluated twice, once in the factual environment and once in its cloned environment. The stored checks show identical integration states and rendered pixels at every row in all three splits. The pair therefore changes the commanded action while preserving the pre-action simulator state.

The data seeds for training, validation and test are 20320101, 20330101 and 20340101. Episode i adds $97i$ to its split seed. An independent generator with seed offset 1900001 chooses the factual branch signs. Data are shared across objectives and model initializations. The implementation uses `dm_control` 1.0.44 and MuJoCo 3.11.0.

H.2 Models and optimization

Both architectures predict a residual added to the current pooled feature. The output layer starts at zero, so the initial model exactly copies the current feature and predicts zero response. The residual is not clipped. This common initialization makes the persistence and zero-response references explicit properties of the initial prediction function.

The CNN-GRU has 2,850,816 parameters. A shared frame encoder uses three 3×3 stride-two convolutions with padding one and widths 32, 64 and 128. Gaussian error linear unit activations follow each convolution. The flattened $8 \times 8 \times 128$ output is projected to 256 features and passed through the same activation. A 256-unit GRU processes the two frame embeddings. Its final output is concatenated with the six-dimensional action and mapped through a 512-unit hidden layer to the 256-dimensional residual.

The Transformer has 1,867,840 parameters. An 8×8 stride-eight convolution maps each frame to 64 patch tokens of width 192. The two frames and a projected action token form 129 tokens, with learned positional embeddings. Four pre-normalized Transformer layers use six attention heads and feedforward width 768. Each feedforward block uses a Gaussian error linear unit. A final layer normalization and linear readout of the action token produce the residual. Dropout is zero.

The objectives match the controlled training study. Factual prediction uses the observed branch, paired prediction minimizes R_{pair}, and paired plus response minimizes $R_{\mathrm{pair}} + R_{\mathrm{resp}}/4$. Here every risk is averaged over the 256 target features. The two paired objectives share all labels, branch evaluations, initializations and data permutations. The coefficient follows the paired-risk identity and was fixed before evaluating test outcomes.

Training uses AdamW for 24 epochs with learning rate 3×10^{-4}, weight decay 10^{-4}, coefficients $(0.9, 0.999)$, numerical epsilon 10^{-8} and gradient-norm clipping at 5. The effective batch size is 256. Training and primary evaluation use FP32; TensorFloat-32 acceleration is disabled. The initialization seeds are 20310921, 20311019 and 20311123. For each seed, the epoch permutation is generated from that seed, the epoch index and the constant 701. Two architectures, three objectives and three initializations give 18 fits.

The final epoch is the primary endpoint. A secondary checkpoint minimizes factual validation risk over epochs zero through 24, with ties resolved in favor of the earlier epoch. Response validation performance does not select a checkpoint. Final CNN-GRU prediction risk exceeds persistence,

whereas its selected checkpoint improves on persistence. The two endpoints separate final-budget behavior from prediction-selected performance.

H.3 Prediction and response on the public task

The two risk ratios test different properties of the same fitted model. Prediction risk is divided by the risk of copying the current feature. Response risk is divided by the risk of predicting zero action response. Both references use the same test rows as the learned models. A ratio below one favors the model; a condition mean averages risks over the three initializations before division by the common reference. The factual persistence risk is 1.477294×10^{-3} and zero-response risk is 9.420284×10^{-5}.

Table 18 shows that useful next-feature prediction can coexist with inaccurate action response on the public task. At the final epoch, all three Transformer condition means improve on persistence, but none improves on zero response. Under factual-validation selection, all six condition means improve on persistence and none improves on zero response. This extends the controlled result to official simulator dynamics, higher-resolution observations and larger residual models under a fixed FP32 protocol.

Table 18: All Walker/Walk condition means. Each row averages three initializations. Final denotes epoch 24; selected denotes factual-validation selection. Values below one favor the learned model.

Model	Objective	Checkpoint	Prediction / persistence	Response / zero
CNN-GRU	Factual	Final	1.0452	1.0259
CNN-GRU	Factual	Selected	0.9417	1.0284
CNN-GRU	Paired	Final	1.0363	1.0160
CNN-GRU	Paired	Selected	0.9413	1.0203
CNN-GRU	Paired + response	Final	1.0308	1.0081
CNN-GRU	Paired + response	Selected	0.9408	1.0115
Transformer	Factual	Final	0.9829	1.1747
Transformer	Factual	Selected	0.9799	1.1583
Transformer	Paired	Final	0.9823	1.0692
Transformer	Paired	Selected	0.9797	1.0567
Transformer	Paired + response	Final	0.9817	1.0418
Transformer	Paired + response	Selected	0.9794	1.0332

Response weighting consistently improves the matched-label response comparison. At the final epoch, the relative reductions are 0.78% for the CNN-GRU and 2.56% for the Transformer. Both paired-difference intervals exclude zero, and the direction also holds under factual-validation selection. The prediction-risk differences are less certain: their intervals include zero (Table 19).

Uncertainty includes initialization and held-out episode variation. Each of 2,000 bootstrap replicates independently resamples three initializations and 32 test episodes with replacement. Every selected initialization uses the same sampled episode panel, and paired objectives share all draws. Risks are first averaged over the 48 rows within an episode. The percentile intervals are pointwise 95% intervals, conditional on the training and validation data and checkpoint rule. The bootstrap seed is 20350919.

The individual initializations retain the complete comparison behind the condition means. Table 20 reports all 18 fits at both endpoints. Seed indices follow the order given above. The selected epoch is determined independently for each fit, while every final result uses epoch 24.

H.4 Risk decomposition and numerical diagnostics

The public-task errors also satisfy the paired-risk decomposition. Let $M = R_{\mathrm{pair}} - R_{\mathrm{resp}}/4$ denote the shared-error term. Table 21 reports each component normalized by R_0, together with the sufficient-bound ratio $4R_{\mathrm{pair}}/R_0$. None of these public-task condition means satisfies the sufficient condition or beats zero response directly. Response weighting reduces the error without crossing the zero-response threshold in these conditions.

Table 19: Response-weighted minus paired risk on Walker/Walk, in units of 10^{-6}. Negative values favor response weighting. Each cell gives the mean difference and a pointwise 95% CI from the crossed bootstrap.

Model	Checkpoint	Response difference	Prediction difference
CNN-GRU	Final	-0.743 (-0.792 to -0.694)	-8.209 (-20.870 to 3.175)
CNN-GRU	Selected	-0.836 (-0.950 to -0.757)	-0.616 (-3.741 to 4.308)
Transformer	Final	-2.574 (-2.905 to -2.230)	-0.918 (-7.639 to 7.082)
Transformer	Selected	-2.212 (-2.649 to -1.783)	-0.479 (-5.567 to 2.871)

Table 20: Every Walker/Walk initialization. Prediction columns divide factual risk by persistence; response columns divide response risk by zero-response risk. Epoch gives the validation-selected checkpoint. Seed indices identify the three listed initialization seeds.

Model	Objective	Seed	Epoch	Final pred.	Final resp.	Selected pred.	Selected resp.
CNN-GRU	Factual	1	16	1.0340	1.0260	0.9417	1.0268
CNN-GRU	Factual	2	16	1.0486	1.0264	0.9380	1.0280
CNN-GRU	Factual	3	17	1.0529	1.0251	0.9455	1.0304
CNN-GRU	Paired	1	17	1.0220	1.0161	0.9422	1.0204
CNN-GRU	Paired	2	16	1.0415	1.0172	0.9389	1.0201
CNN-GRU	Paired	3	17	1.0455	1.0146	0.9427	1.0205
CNN-GRU	Paired + response	1	17	1.0177	1.0080	0.9404	1.0120
CNN-GRU	Paired + response	2	16	1.0440	1.0092	0.9390	1.0119
CNN-GRU	Paired + response	3	18	1.0306	1.0069	0.9432	1.0105
Transformer	Factual	1	23	0.9841	1.2018	0.9753	1.1524
Transformer	Factual	2	24	0.9773	1.1670	0.9773	1.1670
Transformer	Factual	3	24	0.9872	1.1554	0.9872	1.1554
Transformer	Paired	1	23	0.9739	1.0808	0.9718	1.0576
Transformer	Paired	2	23	0.9849	1.0651	0.9793	1.0507
Transformer	Paired	3	24	0.9881	1.0616	0.9881	1.0616
Transformer	Paired + response	1	23	0.9793	1.0505	0.9727	1.0336
Transformer	Paired + response	2	23	0.9810	1.0412	0.9807	1.0322
Transformer	Paired + response	3	24	0.9849	1.0338	0.9849	1.0338

The action effects remain measurable after image quantization. No row has exactly zero response energy, either before or after grayscale rounding. Each action coordinate contributes 256 test rows. On the test panel, quantization changes the response by a mean squared error of 2.137×10^{-8}, about 0.023% of the quantized response energy. Table 22 reports the corresponding quantities for every split. These checks measure target construction, separately from neural inference precision.

All reported Walker comparisons use FP32 training and inference. Native bfloat16 evaluation was unavailable in this run and has no reported result. The precision comparison in Appendix G belongs to the separate controlled study. The public-task archive retains every fit's final and selected metrics, per-episode statistics, training history, data diagnostics and bootstrap draws.

I Statistical Protocol

I.1 Registration inference

Registration directions and evaluation data are disjoint. Direction-seed moments are first reduced to equal-weight seed means. The frozen direction is then applied to evaluation episodes. In factorized panels, episode means are calculated separately within $q = +1$ and $q = -1$ and averaged with equal weight. This prevents random context imbalance from changing the target estimand.

Table 21: Walker/Walk paired-risk components and sufficient-bound ratios. The two middle components sum to R_{pair}/R_0, up to rounding. A sufficient-bound ratio below one certifies response risk below zero-response risk.

Model	Objective	Checkpoint	R_{pair}/R_0	M/R_0	$R_{\mathrm{resp}}/(4R_0)$	$4R_{\mathrm{pair}}/R_0$
CNN-GRU	Factual	Final	16.405	16.149	0.256	65.62
CNN-GRU	Factual	Selected	14.775	14.517	0.257	59.10
CNN-GRU	Paired	Final	16.265	16.011	0.254	65.06
CNN-GRU	Paired	Selected	14.765	14.510	0.255	59.06
CNN-GRU	Paired + response	Final	16.172	15.920	0.252	64.69
CNN-GRU	Paired + response	Selected	14.759	14.507	0.253	59.04
Transformer	Factual	Final	15.427	15.133	0.294	61.71
Transformer	Factual	Selected	15.377	15.088	0.290	61.51
Transformer	Paired	Final	15.414	15.146	0.267	61.66
Transformer	Paired	Selected	15.369	15.105	0.264	61.47
Transformer	Paired + response	Final	15.404	15.144	0.260	61.62
Transformer	Paired + response	Selected	15.362	15.104	0.258	61.45

Table 22: Walker/Walk data diagnostics. Energy and quantization-error columns are per-feature squared quantities in units of 10^{-6}. Raw energy uses grayscale values before eight-bit rounding. Quantization error measures the change in the paired response caused by rounding. Every duplicated factual branch matches its original state and pixels exactly.

Split	Rows	Rows / coordinate	Quantized energy	Raw energy	Quantization error
Train	4608	768	98.924	98.847	0.02065
Validation	1152	192	95.632	95.571	0.02110
Test	1536	256	94.203	94.147	0.02137

For registration inference, the top-level bootstrap resamples the 12 evaluation-seed means and does not resample episodes again inside that aggregate bootstrap. Within a family, the maximum deviation over all candidate lags and action coordinates gives one simultaneous critical value. Bonferroni division across the 18 families and two confirmatory response definitions yields family-method level $0.05/36$. A lag is certified when its one-sided simultaneous lower bound is positive. Recovery requires every coordinate to return its true right boundary and no coordinate to certify a future lag.

I.2 Downstream inference

Downstream uncertainty follows the data-generation hierarchy. Each cell first averages its 37 step-level errors within an episode. A bootstrap replicate samples the 12 top-level test seeds with replacement and then samples 16 episodes with replacement inside each selected seed. Numerator and denominator in Equation 43 use the same sampled rows. The intervals condition on one fitted model instance per cell. They quantify held-out trajectory variation and do not include variation from retraining with a different initialization.

Environment and overall summaries treat system, context, offset, and architecture cells as fixed strata. PointMass2D pools its two coordinate errors within each cell before computing the ratio. The overall point averages 36 cell ratios equally. Architecture-specific oracle sensitivity instead averages 24 coordinate-specific ratios per architecture: six each from Pendulum and CartPole and twelve from PointMass2D. These coordinate-level summaries use a different weighting from the overall cell mean. The two architectures and coordinates within a family share seed and episode draws; independently generated families use independent draws. Every downstream interval uses 5,000 bootstrap replicates.

I.3 Decision criteria and reported evidence

Before the held-out test outcomes were examined, the downstream protocol fixed three criteria. The aggregate point estimate had to reach 5%, its two-sided interval had to exclude zero, and no environment-level interval could lie wholly below zero. The point target and confidence interval

have different roles: a positive interval does not establish a five-percent population lower bound. The observed aggregate point, 4.78%, missed the fixed target, and the CartPole interval failed the environment criterion. Successful registration therefore did not meet the specified downstream improvement criteria.

The zero-response count, response-energy analysis, homogeneous-versus-factorized split, and full cell tables are descriptive analyses of the same held-out test predictions. They explain the failed implication but do not replace the fixed decision criteria and receive no confirmatory p-values. Raw-level peak is also descriptive and never enters the registration decision.

J Sampling Independence and Adaptivity

During method development, a paired context construction reused an initial state and code sequence across transformed pairs. Because (q, E) and $(-q, -E)$ generate the same physical perturbation $W = qE$, this construction duplicated one physical trajectory and understated the raw-mean variance.

After identifying this dependence, we specified the independently balanced design used throughout the paper. Context signs are assigned in equal numbers, while every trajectory receives an independent initial state, action-code sequence, and logger-padding sequence. All 18 registration families and all downstream models were regenerated with mutually disjoint random seeds. This design was specified after the paired result was observed, and no trajectory, fitted direction, trained model, prediction, or bootstrap draw is shared between the two designs.

The correction changes the sampling unit while preserving the scientific question, external code law, candidate lags, offsets, action amplitude, feature map, multiplicity correction, systems, contexts, response definitions, architectures, training objective, and endpoints. Every reported estimate uses only the independent data.

K Reproducibility and Scope

K.1 Artifact contents

The single-cell reproduction program generates trajectories, trains the models and computes the reported results for all three studies. It implements the controlled systems and downloads the official public-task software. No previously measured outcome is needed as an input. The returned archive contains alignment summaries and per-example errors, controlled-training histories and trajectory-level risk decompositions, and public-task episodes, predictions and matched comparisons. Temporary optimizer checkpoints support interrupted runs. Separate compact evidence retains the summaries needed for report generation without model training.

The experiment sources support resuming from committed episode and epoch boundaries. Temporary checkpoints are deleted after the completed evidence archive is stored persistently. Execution batch sizes adapt to available memory while preserving scientific sample counts, data order, effective batch size, model initialization and optimizer update counts. Public-task tables summarize all fit metrics and matched bootstrap intervals.

K.2 Independent reconstruction checks

An independent reconstruction recomputed the 18 observable and 18 exact registration outcomes, the absence of future-lag certificates, equality of right-boundary and oracle alignments in 36 cells, aggregate and environment-level response gains, architecture-level oracle sensitivity, and zero-response counts. The LaTeX tables and figures in this submission are generated directly from the machine-readable archive.

K.3 Scope of the results

The registration theorem covers fixed coordinatewise delays, observed context, an external randomized code and a candidate set containing the true lag. The paired-risk identity holds for any predictor

with finite second moments. The empirical evaluation covers two architecture families at two model scales, three controlled systems, public Walker/Walk, factual and paired supervision, and fixed training budgets. Variable delays, natural logs, large video generators and downstream planning require separate evidence.

The combined conclusion concerns both measurement and learning. Correct registration identifies the external action channel; the paired-risk identity specifies the prediction accuracy needed at the response scale. The longer-training and matched-label studies then measure how optimization budget and response weighting change the remaining error. Every condition remains in the reported evidence, including those whose learned response fails the zero baseline.