LEARNING PHYSICAL DYNAMICS THROUGH RELATIONAL PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Predicting physical trajectories in unfamiliar configurations requires reusing local interactions. Trajectory supervision evaluates their combined effect on motion, so fitting training trajectories need not capture how individual relations evolve. We introduce **HORME**, which trains the same pair features for motion computation and prediction of relational evolution. To supervise this evolution during recursive prediction, we propose Reference-Anchored Joint-Embedding Predictive Architecture (RA-JEPA). It matches predicted and observed feature changes pair by pair in a shared space, retaining initial features as a fixed reference. The forecasts condition physical pair features, allowing trajectory errors to train the predictor. Computing displacement further requires resolving when acceleration acts within each step. To capture timing effects missed by sparse acceleration samples, we propose Temporal Response Decomposition (TRD), separating displacement contributions from mean acceleration and its timing. A readout shared across integration stages corrects the timing-dependent component while preserving the current step's velocity update. HORME achieves the lowest trajectory root mean square error (RMSE) among compared methods on I-PHYRE and gravitational N-body, including unseen games, larger systems, and extended horizons.

1 INTRODUCTION

Predicting physical trajectories supports world modeling, planning, and control in unfamiliar environments (Battaglia et al., 2013; Assran et al., 2025). We predict future object states, including positions and velocities, from their initial states, physical attributes, and any prescribed interventions. Removing a support in I-PHYRE (Li et al., 2024) or adding bodies to a gravitational system changes how local interactions combine. Generalization therefore requires interaction rules that remain useful across these combinations.

Relational simulators support such reuse by representing objects as graph nodes and interactions as edges (Battaglia et al., 2016; 2018). Neural Force Field (NFF) (Li et al., 2026), for example, decodes pairwise forces from edge features, sums them per object, and integrates the resulting acceleration. Trajectory loss, however, supervises these contributions only through their combined effect on motion. Consider a ball resting on two supports (Figure 1). Different model contributions can produce the same net force while both supports are present, yet give different predictions when one is removed. A good trajectory fit can therefore hide errors in individual interactions that become apparent after a configuration change. During rollout, the resulting state errors also alter the relations evaluated at subsequent steps.

The same trajectories also reveal how each pair's relative position and motion change. This temporal information can train pair features to retain what is needed to predict their evolution, beyond what suffices to fit aggregate motion. We introduce **HORME** to impose this additional requirement on a shared relational state: the graph of pair features used for physical computation. Its relational predictor, Reference-Anchored JEPA (RA-JEPA), uses the initial graph and prescribed interventions to recursively forecast pair-feature changes. The initial pair features remain fixed as a reference within each window, preserving the starting context as predictions become inputs to subsequent steps. Forecasts and observed targets share the physical encoder's feature space. Each forecast is matched to changes between encodings of observed states, with errors scored pair by pair even when the combined motion is correct. During rollout, the forecasts condition current-state pair features

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.

before motion computation, allowing trajectory errors to train the predictor through the physical consequences of its outputs. The same trajectories thus supervise both relational evolution and its use in generating motion without force labels.

The resulting features must be converted into motion over a finite time step. Velocity change depends on mean acceleration, but displacement also depends on when the acceleration acts. With few acceleration samples per step, this timing can be estimated inaccurately, producing position errors that alter later relations. To correct these errors, we propose Temporal Response Decomposition (TRD), separating displacement contributions from mean acceleration and its timing. A readout of the same pair features corrects the timing-dependent component, with parameters shared across integration stages and outputs combined using Simpson weights. Observed endpoints supervise the correction, which respects physical pair constraints and leaves the current step's velocity update unchanged.

Our contributions are summarized below.

- We introduce **HORME**, which trains a shared graph of pair features for motion computation and prediction of relational evolution using observed trajectories.
- RA-JEPA predicts changes from a fixed initial reference, with forecast conditioning connecting the predictor to trajectory supervision.
- TRD reads the same pair features across integration stages to correct displacement while preserving physical pair constraints and the current step's velocity update.
- Experiments on I-PHYRE and N-body show the lowest trajectory RMSE among the compared methods, including unseen configurations and extended horizons. Ablations support joint training and a displacement readout shared across stages.

2 Related Work

2.1 Physical Dynamics Modeling

Relational simulators share local computations across configurations (Battaglia et al., 2016; 2018). Visual Interaction Networks (Watters et al., 2017), RPIN (Qi et al., 2021), and SlotFormer (Wu et al., 2023) predict object states or object-centric representations. STICA (Nishimoto & Matsubara, 2026) couples object-level dynamics with causality-aware policy learning. NRI (Kipf et al., 2018) infers interaction graphs, while DPI-Net (Li et al., 2019) and GNS (Sanchez-Gonzalez et al., 2020) simulate particle systems. Graph neural ordinary differential equations (ODEs) (Poli et al., 2019), SEGNO (Liu et al., 2024), and NFF (Li et al., 2026) integrate predicted graph dynamics, equivariant accelerations, and pairwise forces, respectively. IGNS (Hoang et al., 2026) uses port-Hamiltonian dynamics to preserve information during graph propagation, while EqCollide (Chen et al., 2026) models deformable-body collisions through equivariant graph neural ODEs. In pairwise models, motion losses evaluate aggregated interaction effects. Hypersolvers (Poli et al., 2020) learn residual numerical corrections and consider joint optimization of model and solver. HORME predicts pair-feature evolution and corrects displacement through a shared stage readout, preserving physical constraints and the current step's velocity update.

2.2 Joint-Embedding Predictive Architectures

I-JEPA (Assran et al., 2023) predicts image-region features, while V-JEPA (Bardes et al., 2024) extends feature prediction to video. The target feature space determines which distinctions predictive supervision preserves. Graph-JEPA (Skenderi et al., 2023) and Causal-JEPA (Nam et al., 2026) use subgraph and object-level masking. JEPA-x (Wen et al., 2026) supervises visual transitions with physical trajectories, and Semigroup-JEPA (Liu et al., 2026) studies autoregressive prediction in a learned visual space. LeWorldModel (Maes et al., 2026) jointly trains visual encoding and latent dynamics using prediction loss and Gaussian regularization. Fast-LeWM (Gao & Xu, 2026) predicts future latents from action prefixes, while Flow-JEPA (Huo et al., 2026) generates latent trajectories with conditional flow matching. V-JEPA 2 (Assran et al., 2025) uses predicted features for planning. SPR (Schwarzer et al., 2021) shares its encoder between Q-learning and latent prediction. HORME forecasts individual pair features and conditions physical computation on these forecasts, so trajectory errors also train the predictor.

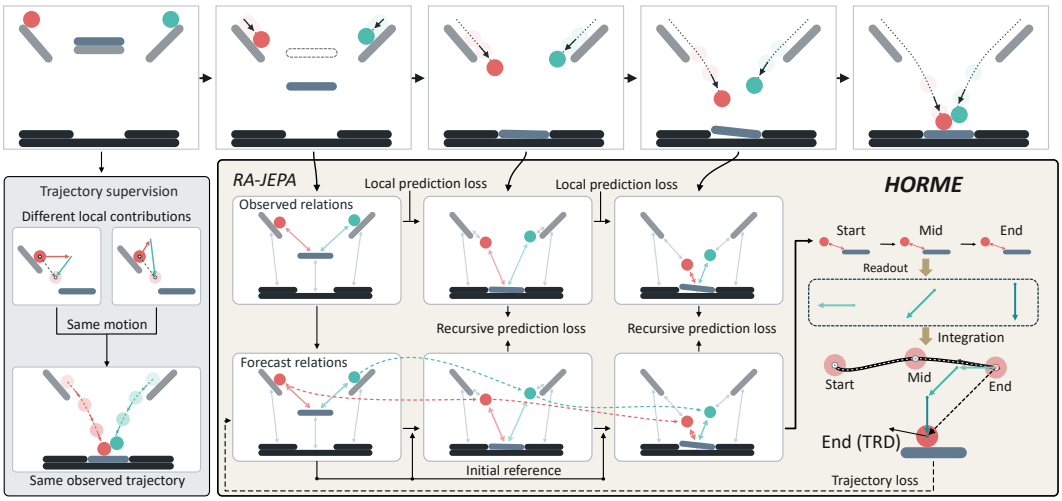


Figure 1: HORME pipeline. **Top:** schematic of an observed I-PHYRE trajectory. Inputs are the initial object states (first frame) and prescribed removal times. The goal is to predict subsequent object states, shown here as ground-truth targets for training and evaluation. **Left:** different pairwise contributions can yield the same motion. **Center:** RA-JEPA forecasts pair-feature changes from a fixed initial reference. Observed features supervise local one-step and recursive predictions. **Right:** TRD combines shared readouts of forecast-conditioned features at the step's start, midpoint, and end. It shifts the endpoint from "End" to "End (TRD)" while preserving the step's velocity update. Trajectory loss trains the predictor through integration. Edges denote object relations and colored arrows illustrate displacement contributions.

3 METHODOLOGY

3.1 OVERVIEW

Given an initial state S_0 of N objects, HORME (Figure 1) predicts L subsequent states at intervals of duration h. The state S_t contains object positions x_t, velocities v_t, and physical attributes such as mass, geometry, or mobility. Prescribed interventions u are supplied when applicable. Ordered-pair features form the shared relational state. RA-JEPA forecasts its evolution over K intervals. At each integration stage, forecasts condition current-state features. TRD reads these features to advance positions and velocities. Algorithm 1 indexes window starts by n and their intervals by k. Dividing `Encode`'s output by σ gives the reference `U0`, to which `Forecast` appends K recursive predictions. `StageResponses` reads acceleration and displacement coefficients (a, c) at three stages, which `TRDUpdate` combines into the next state. Each new window rebuilds its reference from the predicted endpoint.

Algorithm 1 HORME rollout for L steps in K-step windows, given initial state S_0, interventions u, interval h, and fixed feature scale σ.

```
S = S0
trajectory = [S]
for n in range(0, L, K):
  U0 = Encode(S, h) / sigma
  UJ = Forecast(U0, u, n*h, K, h)
  for k in range(min(K, L - n)):
    t = (n + k) * h
    a, c = StageResponses(
        S, U0, UJ[k:k+2], u, t, h)
    S = TRDUpdate(S, a, c, h)
    trajectory.append(S)
return trajectory
```

3.2 REFERENCE-ANCHORED JEPA

3.2.1 A SHARED RELATIONAL STATE

To supervise relations before aggregation, RA-JEPA predicts features for each ordered pair (i, j), with source i acting on receiver j. Let E_η denote the physical encoder with parameters η. At query time s, it produces the physical feature graph $H_s^D = E_\eta(S_s, h)$ before forecast conditioning, with superscript D denoting physical-state encoding. Two encoder branches each supply one d-dimensional token (feature vector) per pair, giving $H_s^D \in \mathbb{R}^{N\times N\times 2\times d}$. Fixed root mean square (RMS) scales $\sigma > 0$, calibrated per encoder branch on training data, define $U_s^D = H_s^D/\sigma$ for

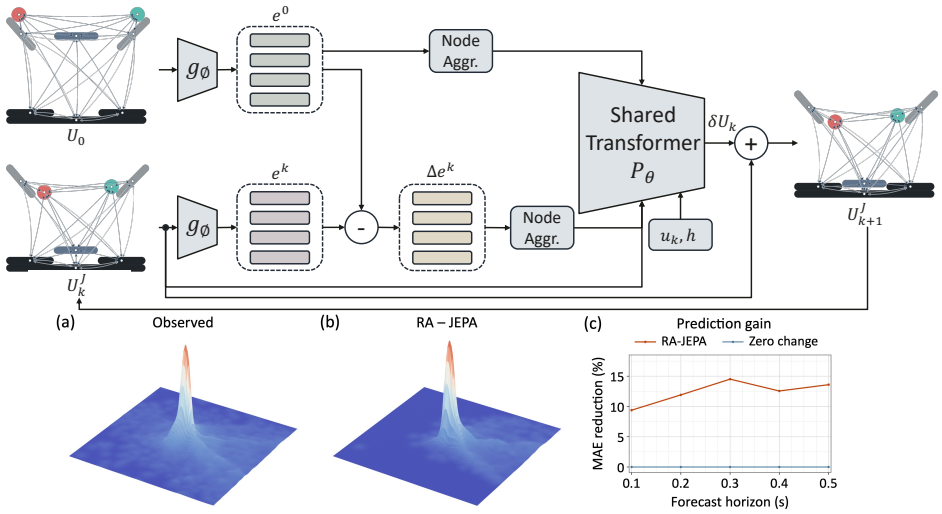


Figure 2: Reference-anchored prediction. Top, shared projection g_ϕ gives initial pair features e^0 and current features e^k. Node aggregation summarizes e^0 and changes $\Delta e^k = e^k - e^0$ for Transformer P_θ, whose increments δU_k recursively update U_k^J. Bottom, densities of observed (a) and forecast (b) feature changes share a principal-component (PC) basis and logarithmic height scale. Panel (c) reports mean absolute error (MAE) reduction over all feature coordinates relative to a zero-change forecast ($U_k^J = U_0$) from a frozen I-PHYRE model.

encodings, forecasts, and targets. Self-pairs ($i = j$) and absent objects are masked. The I-PHYRE encoder follows NFF (Li et al., 2026) in using source-geometry and relative-state inputs with event-related and persistent feature branches. N-body uses pair features of relative position, motion, and mass. Prediction losses compare each pair's feature change with targets derived from observed states, then average the absolute errors.

3.2.2 Predicting Changes from an Initial Reference

We reset time at each window and write $U_0 = E_\eta(S_0, h)/\sigma$, with S_0 now denoting that window's initial state. To retain the starting context, the initial reference $R_0 = \mathcal{R}_\phi(\mathrm{sg}(U_0))$ stores projected pair features, query projections, and a validity mask. The operator sg stops input gradients, while the projection parameters ϕ remain trainable. The shared projection g_ϕ gives initial pair features $e^0 = g_\phi(\mathrm{sg}(U_0))$, current features $e^k = g_\phi(U_k^J)$, and changes $\Delta e^k = e^k - e^0$. Starting from $\Delta U_0 = 0$, the increment predictor P_θ shares parameters across recursive steps:

$$U_k^J = U_0 + \Delta U_k, \qquad \Delta U_{k+1} = \Delta U_k + P_\theta(U_k^J, R_0, u_k, h), \tag{1}$$

Here U_k^J is the forecast feature graph, with J denoting JEPA, ΔU_k its cumulative change, and u_k the next-interval interventions. The two node-aggregation branches in Figure 2 produce the initial context bank from e^0 and the change context bank from Δe^k. Each bank is a collection of per-object summaries combining incoming and outgoing pair means to distinguish source and receiver roles. The Transformer adds the stored initial query to each current-pair query and attends to both banks, letting each pair use changes involving other objects. Gradients pass through recursive forecasts. Initial pair features and queries stay fixed within the window. I-PHYRE removals update both banks' masks and summaries. Only pairs involving a dynamic object receive increments.

3.2.3 Conditioning the Physical Response

To use relational forecasts in motion computation, we combine the current-state encoding U_s^D with U_s^J, forecast from the window start for the same query time s. A conditioning reference U_s^{ref} defines the feature difference read by attention $\mathcal{C}_\psi$, with trainable parameters ψ:

$$Z_s = U_s^J - U_s^{\mathrm{ref}}, \qquad H_s^\star = H_s^D + \sigma\,\mathcal{C}_\psi(U_s^D, U_s^J, Z_s). \tag{2}$$

Here Z_s is the normalized forecast difference and $H_s^\star$ the conditioned feature graph. Queries, keys, and values are U_s^D, U_s^J, and Z_s, respectively. Response heads consume $H_s^\star$, so trajectory supervision trains the forecast's contribution to motion. I-PHYRE uses $U_s^{\mathrm{ref}} = U_0$ to read predicted

changes from the window start. N-body uses $U_s^{\text{ref}} = U_s^D$ to read forecast disagreement with the current encoding. This conditioning reference is distinct from the predictor's initial reference R_0. At inference, both U_0 and U_s^D come from rollout states, so conditioning requires no future observations. Forecast alignment respects prescribed removal times, as detailed in Appendix B.3.

3.3 Temporal Response Decomposition

3.3.1 From Relational Features to Finite-Time Responses

The conditioned pair features must now produce velocity and displacement responses over a finite interval. Let $a(\tau)$ denote an integrable acceleration history at normalized time $\tau \in [0, 1]$, corresponding to physical time $t + h\tau$. Second-order dynamics give

$$\begin{aligned} v_{t+h} - v_t &= h \int_0^1 a(\tau)\, d\tau, \\ x_{t+h} - x_t - hv_t &= h^2 \int_0^1 (1-\tau) a(\tau)\, d\tau. \end{aligned} \tag{3}$$

Writing $1-\tau = 1/2 + (1/2-\tau)$ separates the mean-acceleration contribution from the centered temporal moment $B = \int_0^1 (1/2 - \tau) a(\tau)\, d\tau$, which captures acceleration timing. An acceleration perturbation with zero integral can change this centered temporal moment while preserving endpoint velocity. TRD parameterizes this displacement freedom with a shared pairwise readout supervised by observed endpoints. Response heads map $H_s^\star$ to pair contributions aggregated into acceleration a_s and displacement coefficient c_s, both with acceleration units. In $c_s = \mathcal{A}_s(g_c(H_s^\star))$, the readout g_c shares parameters across pairs and stages. The assembly operator $\mathcal{A}_s$ enforces physical constraints before aggregation. Assembly uses known geometry, masses, and mobility. Prescribed I-PHYRE gravity contributes only to a_s. N-body uses invariant pair inputs with mass and gravitational-time scaling, followed by antisymmetric vector readouts. These readouts preserve total linear momentum and center-of-mass motion in the closed N-body system.

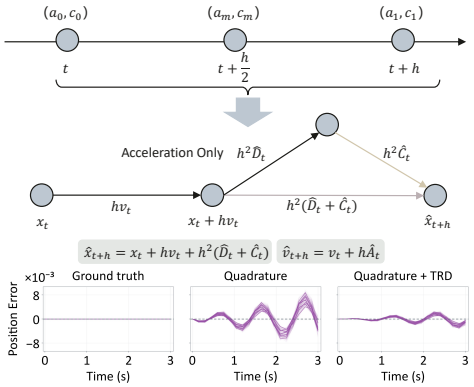


Figure 3: TRD and a spring experiment. Stages read acceleration a and displacement coefficient c. TRD adds $h^2\widehat{C}_t$ to acceleration-based displacement $h^2\widehat{D}_t$, with velocity change $h\widehat{A}_t$. Arrows mark additive contributions. Purple curves show signed vertical-position errors against zero-error dashed lines. Quadrature uses acceleration alone. Quadrature + TRD adds the learned correction at the same analytic field and force-query budget.

3.3.2 Shared Readouts and Simpson Quadrature

We apply Simpson's three-node rule to the response integrals in Eq. (3) at t, $t + h/2$, and $t + h$. Aligned forecasts condition the stage features, whose response heads produce (a_0, c_0), (a_m, c_m), and (a_1, c_1). Midpoint and endpoint states are previewed from the same S_t using constant acceleration a_0 for $h/2$ and a_m for h, respectively. For $a(\tau)$, Simpson's rule gives mean acceleration $\widehat{A}_t = (a_0 + 4a_m + a_1)/6$. For $(1-\tau)a(\tau)$, it gives $\widehat{D}_t = a_0/6 + a_m/3$. At the endpoint, no time remains for acceleration to affect position.

The averaged displacement coefficient $\widehat{C}_t = (c_0 + 4c_m + c_1)/6$ gives the displacement correction $h^2\widehat{C}_t$. The learned function is shared across stages, while Simpson weights determine how its outputs contribute to the interval response. TRD produces the predicted position $\widehat{x}_{t+h}$ and velocity $\widehat{v}_{t+h}$ through

$$\widehat{v}_{t+h} = v_t + h\widehat{A}_t, \qquad \widehat{x}_{t+h} = x_t + hv_t + h^2(\widehat{D}_t + \widehat{C}_t). \tag{4}$$

The estimated centered temporal moment is $\widehat{B}_t = \widehat{D}_t + \widehat{C}_t - \widehat{A}_t/2 = (a_0 - a_1)/12 + \widehat{C}_t$. Thus $\widehat{C}_t$ corrects the centered temporal moment at fixed $\widehat{A}_t$, preserving the current step's velocity update and acceleration-based previews (Figure 3). The corrected position becomes input to subsequent steps, connecting the learned displacement response to later interactions. Observed endpoints supervise

the stage coefficients within this fixed update. Angular states use the same construction and object masks.

3.4 Training and Inference

Let $\bar{U}_k$ be the normalized encoding of the observed state at kh from a target encoder whose parameters are an exponential moving average (EMA) of E_η. The online and EMA target encoders share fixed normalization before forecast conditioning. With fixed R_0, the local loss $\mathcal{L}_{\text{local}}$ supervises one-step transitions, matching increments from online encodings of observed states to $\text{sg}(\bar{U}_{k+1} - \bar{U}_k)$. The recursive loss $\mathcal{L}_{\text{recursive}}$ supervises accumulated changes when the predictor consumes its own outputs, matching ΔU_k to $\text{sg}(\bar{U}_k - \bar{U}_0)$. The total objective $\mathcal{L}$ combines the trajectory loss $\mathcal{L}_{\text{traj}}$ with these prediction losses,

$$\mathcal{L} = \mathcal{L}_{\text{traj}} + \frac{\lambda}{2}\big(\mathcal{L}_{\text{local}} + \mathcal{L}_{\text{recursive}}\big). \tag{5}$$

The prediction losses average absolute errors over valid pairs and horizons, weighted by $\lambda \geq 0$. Trajectory supervision uses normalized state errors for I-PHYRE, whereas N-body supervises position, velocity, and endpoint-derived response moments. Prediction losses update E_η and P_θ through online encodings and forecasts, with stopped target gradients. Trajectory loss reaches E_η through physical queries and reaches P_θ through forecast conditioning, alongside the response heads and $\mathcal{C}_\psi$. At inference, forecasts from each window's initial state and prescribed interventions condition integration. Observed targets and local one-step predictions are training-only.

4 Experiments

We first evaluate trajectory prediction under configuration and horizon shifts, then examine the predicted feature changes and displacement responses. Table 4 tests how predictive supervision, forecast conditioning, and trajectory gradients contribute to the joint model. The lower block of Table 5 tests whether the organization of stage readouts matters under matched head budgets and physical constraints.

4.1 Experimental Setup

Benchmarks. We follow the experimental setup of NFF (Li et al., 2026) for structured-state prediction on I-PHYRE and gravitational N-body dynamics. I-PHYRE provides 100 training trajectories from 10 basic games, 200 within-scenario (Within) test trajectories, and 600 cross-scenario (Cross) trajectories from 30 unseen games. The Cross games cover noisy, compositional, and multi-ball settings. Rollouts cover 150 frames given the initial state and prescribed removals. N-body uses 200 training trajectories from two- or three-body systems with planetary and cometary initializations at $T = 50$ frames. Within varies initial conditions, whereas Cross uses eight or ten bodies with broader mass and initial-radius distributions. N-body evaluation covers horizons T and $3T$.

Baselines. We compare four baselines using published results from Li et al. (2026). Interaction Networks (IN) (Battaglia et al., 2016) aggregate pairwise effects for an object network to predict next states. SlotFormer (Wu et al., 2023) uses spatiotemporal attention for autoregressive slot prediction, with state-derived embeddings in the cited benchmark. SEGNO (Liu et al., 2024) integrates equivariant accelerations in a second-order neural ODE to predict continuous trajectories. NFF (Li et al., 2026) numerically integrates accelerations obtained from learned pairwise force fields.

Evaluation. Models predict complete trajectories through open-loop rollout from the initial system state. Root mean square error (RMSE) is computed per trajectory over the scored channels before averaging: normalized $x, y, \theta/10$ for I-PHYRE, and mass, position, and velocity for N-body. Final position error (FPE) and position change error (PCE) follow the benchmark: componentwise absolute errors over the scored channels at the last valid frame and over adjacent valid changes, respectively. Correlation (R) follows the benchmark's Pearson implementation. Standard errors of the mean (SEM) are computed across test trajectories.

4.2 PREDICTION ON UNSEEN CONFIGURATIONS

Results on I-PHYRE. Table 1 shows that HORME achieves the lowest trajectory RMSE and highest correlation on both splits. Relative to NFF, RMSE falls from 0.048 to 0.039 on Within and from 0.131 to 0.099 on Cross, reductions of 18.8% and 24.4%, respectively. The larger relative gain on Cross extends to unseen games with changed contact configurations. Cross PCE also decreases from 0.003 to 0.002, while Within PCE matches the best value of 0.001. RA-JEPA supervises pair-feature evolution across horizons, while TRD refines displacement at each step. Together with trajectory supervision over training windows, these objectives target agreement across successive states. FPE instead scores only the last valid frame, where NFF retains lower error on both splits. Improved agreement over the full trajectory therefore need not imply the smallest terminal error.

Table 1: I-PHYRE prediction over 150 frames with prescribed removals. Within uses familiar games, Cross unseen games. Means ± SEM across trajectories. Bold and underlined entries mark best and second-best values. Metrics are defined in Section 4.1.

Setting	Method	RMSE ↓	FPE ↓	PCE ↓	R ↑
Within	IN (2016)	0.124 ± 0.007	0.129 ± 0.010	0.004 ± 0.000	0.993 ± 0.001
	SlotFormer (2023)	0.067 ± 0.006	0.068 ± 0.007	<u>0.002</u> ± 0.000	<u>0.997</u> ± 0.001
	SEGNO (2024)	0.203 ± 0.070	0.212 ± 0.082	0.008 ± 0.008	0.977 ± 0.026
	NFF (2026)	<u>0.048</u> ± 0.004	**0.042** ± 0.004	**0.001** ± 0.000	<u>0.997</u> ± 0.001
	HORME	**0.039** ± 0.003	<u>0.062</u> ± 0.005	**0.001** ± 0.000	**0.998** ± 0.000
Cross	IN (2016)	0.194 ± 0.004	0.178 ± 0.005	0.005 ± 0.000	0.974 ± 0.001
	SlotFormer (2023)	0.206 ± 0.005	0.198 ± 0.005	0.004 ± 0.000	0.967 ± 0.001
	SEGNO (2024)	0.314 ± 0.106	0.289 ± 0.109	0.017 ± 0.018	0.928 ± 0.064
	NFF (2026)	<u>0.131</u> ± 0.004	**0.111** ± 0.004	<u>0.003</u> ± 0.000	<u>0.980</u> ± 0.002
	HORME	**0.099** ± 0.004	<u>0.118</u> ± 0.003	**0.002** ± 0.000	**0.984** ± 0.001

Results on N-body. Table 2 shows that HORME leads all four metrics in every reported setting of gravitational dynamics. At Within T, RMSE decreases from SEGNO's 0.079 to 0.030, a 62.4% reduction at the training horizon. Extending prediction to $3T$ yields RMSE of 0.170, compared with NFF's 0.525, a reduction of 67.5%. This setting keeps body counts within the training range while extending rollout duration, testing temporal extrapolation. Cross $3T$ additionally introduces eight or ten bodies and broader mass and radius distributions. Here, RMSE decreases from NFF's 1.226 to 0.339, a 72.4% reduction under the combined shifts. At $3T$, HORME also reduces FPE from 0.592 to 0.152 on Within and from 1.418 to 0.335 on Cross relative to NFF. Cross correlation rises from 0.803 to 0.962. Lower PCE and FPE accompany the RMSE reduction, showing improvements in successive changes and final states. The gains therefore persist when more interactions are combined over longer rollouts.

Table 2: N-body prediction at $T = 50$ and $3T = 150$ frames. Within uses familiar body counts, Cross eight or ten bodies with broader physical parameters. Means ± SEM across trajectories. Bold and underlined entries mark best and second-best values.

Setting	Method	RMSE ↓	FPE ↓	PCE ↓	R ↑
Within T	IN (2016)	0.200 ± 0.015	0.198 ± 0.014	0.012 ± 0.001	0.984 ± 0.002
	SlotFormer (2023)	0.214 ± 0.018	0.215 ± 0.017	0.013 ± 0.002	0.977 ± 0.003
	SEGNO (2024)	<u>0.079</u> ± 0.009	<u>0.074</u> ± 0.009	<u>0.005</u> ± 0.001	<u>0.996</u> ± 0.001
	NFF (2026)	0.097 ± 0.009	0.104 ± 0.009	0.007 ± 0.001	<u>0.996</u> ± 0.001
	HORME	**0.030** ± 0.004	**0.032** ± 0.005	**0.002** ± 0.000	**1.000** ± 0.000
Within $3T$	IN (2016)	0.752 ± 0.026	0.785 ± 0.027	0.029 ± 0.002	0.895 ± 0.009
	SlotFormer (2023)	1.092 ± 0.043	1.311 ± 0.064	0.035 ± 0.002	0.854 ± 0.010
	SEGNO (2024)	0.690 ± 0.144	1.535 ± 0.056	0.022 ± 0.004	0.947 ± 0.010
	NFF (2026)	<u>0.525</u> ± 0.026	<u>0.592</u> ± 0.029	<u>0.021</u> ± 0.002	<u>0.949</u> ± 0.006
	HORME	**0.170** ± 0.016	**0.152** ± 0.013	**0.011** ± 0.002	**0.988** ± 0.002
Cross $3T$	IN (2016)	6.942 ± 0.235	11.568 ± 0.492	0.101 ± 0.003	0.287 ± 0.015
	SlotFormer (2023)	2.533 ± 0.068	2.751 ± 0.060	0.053 ± 0.002	0.438 ± 0.019
	SEGNO (2024)	2.759 ± 0.076	2.302 ± 0.062	0.063 ± 0.001	0.291 ± 0.013
	NFF (2026)	<u>1.226</u> ± 0.023	<u>1.418</u> ± 0.029	<u>0.035</u> ± 0.002	<u>0.803</u> ± 0.010
	HORME	**0.339** ± 0.018	**0.335** ± 0.017	**0.019** ± 0.002	**0.962** ± 0.004

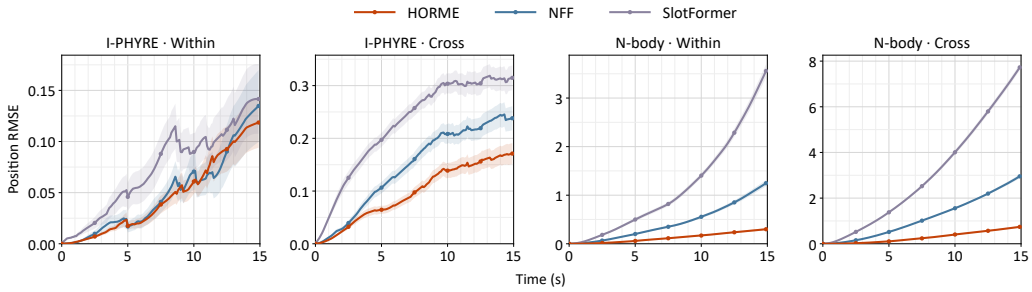


Figure 4: Open-loop position RMSE for HORME (orange), NFF (blue), and SlotFormer (purple). Panels separate familiar (Within) from unseen (Cross) configurations. Curves show trajectory means with ± SEM shading. I-PHYRE uses normalized x, y positions from four fixed test trajectories per game (40 Within, 120 Cross), shared across methods (Appendix D.3). N-body uses x, y, z from all 200 trajectories per split. Lower is better.

Error accumulation. Figure 4 traces position error through open-loop prediction. HORME's advantage becomes clearest late in I-PHYRE Cross rollouts and widens over time in both N-body splits. These position-only curves complement the multichannel scores in Tables 1 and 2 by showing how errors evolve along the sampled trajectories.

Qualitative prediction. Figure 5 compares selected Cross scenes under matched initial states and viewing bounds. In these scenes, HORME follows the ground-truth paths more closely through contact changes and multi-body motion, while baseline rollouts drift farther or leave the viewport.

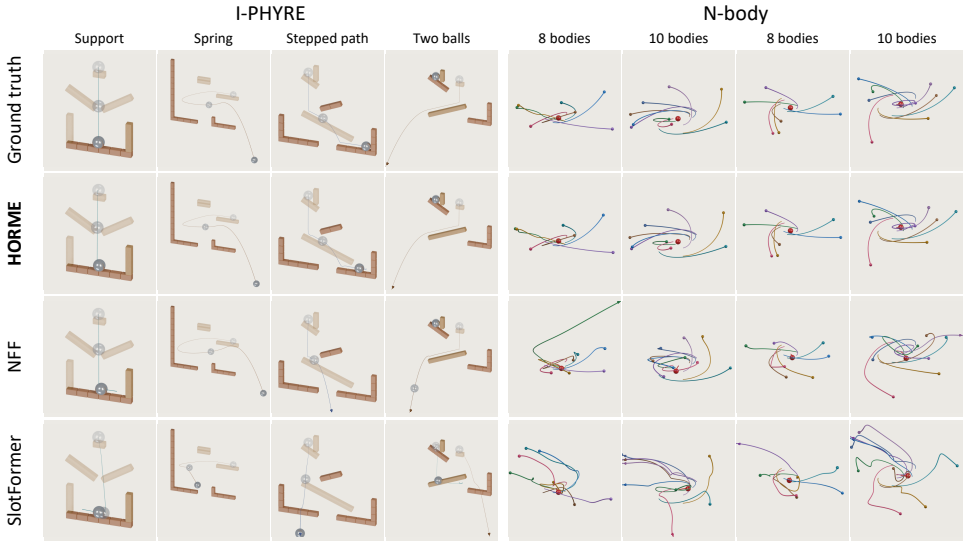


Figure 5: Selected Cross trajectories. Columns are independent scenes and rows compare methods. I-PHYRE states progress from faint to opaque. N-body spheres mark trajectory endpoints. Each column shares initial states, timestamps, and viewing bounds across methods. Arrows mark viewport exits.

4.3 Feature and Response Diagnostics

Predicted feature changes. Figure 2(a,b) pools 800 I-PHYRE test trajectories in a common PC basis. The predicted density follows the dominant observed mode. Panel (c) tests whether individual forecasts improve on retaining the starting features, comparing paired predictions and observations over all 512 coordinates. Across horizons of 0.1–0.5 s, RA-JEPA reduces mean feature MAE by 9.4–14.5% relative to retaining the initial features.

Conceptual response experiment. To test displacement modeling independently of force learning, Figure 3 fixes the analytic spring field and uses three force queries per step. Across 128 held-out

rollouts, vertical-position RMSE falls from 3.15×10^{-3} to 1.07×10^{-3}. Corrected positions change later acceleration queries, also reducing velocity error over the rollout.

4.4 Joint Learning and Response Parameterization

Table 3: Component ablations of HORME. Lower RMSE indicates better prediction.

TRD	RA-JEPA	I-PHYRE		N-body $3T$	
		Within	Cross	Within	Cross
✗	✗	0.089	0.234	0.794	2.831
✓	✗	0.044	0.119	0.391	0.685
✗	✓	<u>0.043</u>	<u>0.113</u>	<u>0.336</u>	<u>0.639</u>
✓	✓	**0.039**	**0.099**	**0.170**	**0.339**

Table 4: Joint-learning controls with the TRD architecture fixed (RMSE ↓).

Variant	I-PHYRE		N-body $3T$	
	Within	Cross	Within	Cross
Auxiliary-only JEPA	0.051	0.126	0.484	0.943
No predictive loss	0.043	0.115	0.343	0.675
No initial reference	0.043	<u>0.107</u>	0.296	0.599
Stop trajectory gradient	<u>0.042</u>	0.119	<u>0.261</u>	<u>0.512</u>
HORME	**0.039**	**0.099**	**0.170**	**0.339**

Component contributions. Either component lowers RMSE across both benchmarks and splits relative to disabling both (Table 3). Together, they reduce Cross RMSE by 57.7% on I-PHYRE and 88.0% on N-body. The full model also outperforms either component alone, supporting complementary roles for relational prediction and displacement modeling.

Joint predictive learning. Table 4 tests how feature supervision and physical optimization interact. Auxiliary-only JEPA retains feature losses without conditioning physical responses. Stop trajectory gradient restores conditioning but blocks trajectory gradients to the predictor. Restoring conditioning lowers RMSE relative to Auxiliary-only JEPA. Allowing trajectory gradients to train the predictor further lowers error in the full model. No predictive loss retains conditioning with $\lambda = 0$ but has higher error than HORME, supporting explicit feature supervision. No initial reference removes the initial context bank, stored query, and e^0 subtraction. Its higher errors support retaining the initial relational context during recursion.

Response parameterization. Table 5 tests the temporal organization of physical readouts. TRD improves over a generic endpoint position correction on both benchmarks. The lower block matches head budgets while fixing the backbone, training protocol, and physical constraints. Joint endpoint combines all three stage features into one correction. Shared, uniform averages a common stage head's outputs. Replacing uniform with Simpson weights lowers Cross RMSE from 0.103 to 0.099 on I-PHYRE and from 0.418 to 0.339 on N-body. These comparisons support a common readout at each stage followed by Simpson weighting when converting features into displacement.

Table 5: Displacement-response controls. Cross RMSE ↓, with N-body evaluated at $3T$. The lower block compares heads with matched parameter budgets.

Displacement response	I-PHYRE	N-body
No correction	0.113	0.639
Generic endpoint residual	0.104	0.549
Heads with matched budgets		
Joint endpoint	0.104	0.494
Shared, uniform	<u>0.103</u>	<u>0.418</u>
TRD	**0.099**	**0.339**

5 Conclusion

HORME connects prediction of relational evolution to motion computation through a shared graph of pair features. RA-JEPA retains the initial context during recursive prediction and conditions physical readouts on its forecasts, allowing trajectory errors to train the predictor. TRD reads the same features to correct the timing-dependent displacement response while preserving physical constraints and the current step's velocity update. Experiments on I-PHYRE and N-body show the lowest trajectory RMSE among the compared methods across familiar and unseen configurations, including extended gravitational rollouts. Feature diagnostics and ablations support training relational forecasts together with their use in physical computation. Observed trajectories thus supervise how pair features evolve and generate finite-time motion, without force labels.

AI USE STATEMENT

We used generative AI for language polishing and selected data visualizations. The authors reviewed all outputs and take responsibility for the final content.

ETHICS STATEMENT

This study uses simulated physical systems and does not involve human participants or personal data. We have identified no specific ethical concerns.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix B describe the model formulation, training objectives, and rollout procedure. Dataset configurations and splits are specified in Appendix C. Appendix D reports architecture and optimization settings, model selection, and evaluation metrics, while Appendix E details the additional experimental protocols.

REFERENCES

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

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. volume 29, 2016.

Peter W Battaglia, Jessica B Hamrick, and Joshua B Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the national academy of sciences*, 110(45):18327–18332, 2013.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2(3):5, 2018.

Qianyi Chen, Tianrun Gao, Chenbo Jiang, and Tailin Wu. Eqcollide: Equivariant and collision-aware deformable objects neural simulator. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD)*, pp. 10623–10634. Association for Computing Machinery, 2026.

Yuntian Gao and Xiangyu Xu. Fast leworldmodel. *arXiv preprint arXiv:2606.26217*, 2026.

Tai Hoang, Alessandro Trenta, Alessio Gravina, Niklas Freymuth, Philipp Becker, Davide Bacciu, and Gerhard Neumann. Improving long-range interactions in graph neural simulators via hamiltonian dynamics. In *International Conference on Learning Representations (ICLR)*, volume 2026, pp. 55811–55847, 2026.

Yanchen Huo, Ziying Song, and Yadan Luo. Flow-jepa: Flow matching for robust latent dynamics in jepa world models. *arXiv preprint arXiv:2608.29029*, 2026.

Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *International conference on machine learning (ICML)*, pp. 2688–2697. Pmlr, 2018.

Shiqian Li, Kewen Wu, Chi Zhang, and Yixin Zhu. I-phyre: Interactive physical reasoning. In *International Conference on Learning Representations (ICLR)*, volume 2024, pp. 28195–28215, 2024.

Shiqian Li, Ruihong Shen, Yaoyu Tao, Chi Zhang, and Yixin Zhu. Neural force field: Few-shot learning of generalized physical reasoning. In *International Conference on Learning Representations (ICLR)*, volume 2026, pp. 146629–146655, 2026.

Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B. Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In *International Conference on Learning Representations (ICLR)*, 2019.

Andy Zeyi Liu, Haoran Sun, Lucas Baker, Randall Balestriero, and John Sous. Semigroup-jepa: Latent dynamics consistency for zero-shot physics generalization. *arXiv preprint arXiv:2609.10464*, 2026.

Yang Liu, Jiashun Cheng, Haihong Zhao, Tingyang Xu, Peilin Zhao, Fugee Tsung, Jia Li, and Yu Rong. Segno: Generalizing equivariant graph neural networks with physical inductive biases. In *International Conference on Learning Representations (ICLR)*, volume 2024, pp. 22764–22787, 2024.

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

Heejeong Nam, Quentin Le Lidec, Lucas Maes, Yann LeCun, and Randall Balestriero. Causal-jepa: Learning world models through object-level latent masking. 2026.

Yosuke Nishimoto and Takashi Matsubara. Object-centric world models for causality-aware reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, pp. 24585–24593, 2026.

Michael Poli, Stefano Massaroli, Junyoung Park, Atsushi Yamashita, Hajime Asama, and Jinkyoo Park. Graph neural ordinary differential equations. *arXiv preprint arXiv:1911.07532*, 2019.

Michael Poli, Stefano Massaroli, Atsushi Yamashita, Hajime Asama, and Jinkyoo Park. Hypersolvers: Toward fast continuous-depth models. volume 33, pp. 21105–21117, 2020.

Haozhi Qi, Xiaolong Wang, Deepak Pathak, Yi Ma, and Jitendra Malik. Learning long-term visual dynamics with region proposal interaction networks. In *International Conference on Learning Representations (ICLR)*, 2021.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International conference on machine learning (ICML)*, pp. 8459–8468. Pmlr, 2020.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations (ICLR)*, 2021.

Geri Skenderi, Hang Li, Jiliang Tang, and Marco Cristani. Graph-level representation learning with joint-embedding predictive architectures. *arXiv preprint arXiv:2309.16014*, 2023.

Nicholas Watters, Daniel Zoran, Theophane Weber, Peter Battaglia, Razvan Pascanu, and Andrea Tacchetti. Visual interaction networks: Learning a physics simulator from video. *Advances in neural information processing systems (NeurIPS)*, 30, 2017.

Kehan Wen, Ziming Li, Siyuan Luo, and Fan Shi. Jepa-x: Cross-predictive physics grounding for forecastable latent dynamics. *arXiv preprint arXiv:2608.24044*, 2026.

Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. Slotformer: Unsupervised visual dynamics simulation with object-centric models. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

A SIGNIFICANCE OF THIS RESEARCH

Predicting the consequences of an action is a core requirement for model-based planning and control. A useful physical simulator must therefore handle changes to the configuration that generates motion. Removing a support can initiate a sequence of contacts, while adding bodies changes the interactions throughout a gravitational system. These settings motivate learning representations of how local interactions evolve and accumulate into system behavior. HORME forecasts changes in the pair features used to compute motion, treating their graph as a shared relational state.

Observed trajectories supply targets for both object motion and changes in pair features. Encoding these states in the physical model's shared feature space gives the targets for relational prediction. Scoring each pair against its own target directly supervises feature evolution before errors are averaged. RA-JEPA retains the initial pair features as a reference while recursively forecasting their changes. These forecasts condition physical readouts, so trajectory errors train the predictor through the consequences of its outputs. The shared representation is thus trained to support both relational prediction and motion computation from state sequences without force labels.

The same relational state must describe how interactions accumulate over a finite interval. Two acceleration histories can produce the same velocity change while yielding different displacements, making interaction timing relevant to future configurations. TRD separates the displacement contributions from mean acceleration and its timing, then learns a correction to the latter. A pairwise readout shared across integration stages acts on the same conditioned features, with geometry, mass, and mobility constraints applied during assembly. In the closed N-body system, antisymmetric readouts preserve total linear momentum and center-of-mass motion. This connects a flexible learned correction to physical structure while retaining the three-stage integration budget.

I-PHYRE and gravitational N-body test this formulation under contact events and continuous interactions, respectively. The benchmark results show improved trajectory accuracy across unfamiliar object arrangements, with N-body rollouts extending to three times the training horizon (Tables 1 and 2). The joint-learning controls support connecting predictive supervision to physical computation (Table 4). The response controls support a displacement readout shared across stages (Table 5). The broader significance is a way to connect predictive representation learning to the internal computations of a simulator, so the features trained to anticipate change are also trained to produce its physical consequences.

B ADDITIONAL METHOD DETAILS

B.1 PAIR REPRESENTATION AND REFERENCE CONTEXT

An ordered pair (i, j) denotes source i acting on receiver j. The physical encoder produces two width-d tokens per pair, forming $H_t^D \in \mathbb{R}^{N\times N\times 2\times d}$. Self-pairs ($i = j$) and absent objects are masked. Fixed RMS scales for each encoder branch, calibrated on training states, define $U_t^D = H_t^D/\sigma$.

At the start of each forecast window, U_0 initializes the recursive forecast. The reference branch uses $e_{ij}^0 = g_\phi(\mathrm{sg}(U_{0,ij}))$, while current projected pair features are $e_{ij}^k = g_\phi(U_{k,ij}^J)$. The projection remains trainable in both branches. The initial context bank summarizes e^0, and the change context bank summarizes $e^k - e^0$. Each bank contains one feature summary per object, produced by the corresponding node-aggregation branch in Figure 2. For validity mask m_{ij}, outgoing and incoming summaries are

$$o_i(e) = \frac{\sum_j m_{ij} e_{ij}}{\max(1, \sum_j m_{ij})}, \qquad \iota_i(e) = \frac{\sum_j m_{ji} e_{ji}}{\max(1, \sum_j m_{ji})}. \tag{A1}$$

Concatenating the incoming and outgoing means with removal information, then projecting them, gives each per-object context summary. Pair queries include summaries for the source and receiver, the stored initial query, and interval h. Each query attends to both banks through the shared Transformer, allowing its prediction to use changes throughout the graph. I-PHYRE masks removed objects in both banks and predicts increments only for pairs involving a dynamic object.

B.2 Prediction Targets and Gradient Flow

At observed time kh, let U_k^D and $\bar{U}_k$ denote online and target encodings. The local and recursive errors are

$$\begin{aligned} \epsilon_k^{\text{local}} &= P_\theta(U_k^D, R_0, u_k, h) - \text{sg}(\bar{U}_{k+1} - \bar{U}_k), \\ \epsilon_k^{\text{recursive}} &= \Delta U_k - \text{sg}(\bar{U}_k - \bar{U}_0). \end{aligned} \tag{A2}$$

Local predictions use observed states for $k = 0, \ldots, K-1$. Recursive predictions accumulate increments for $k = 1, \ldots, K$, with $U_k^J = U_0 + \Delta U_k$. Both use the same reference R_0. For valid ordered pairs $\mathcal{M}$, each error is reduced as

$$\ell(\epsilon, \mathcal{M}) = \frac{\sum_{(i,j)\in\mathcal{M}} \sum_{b=1}^{2} \sum_{r=1}^{d} |\epsilon_{ijbr}|}{2d \max(1, |\mathcal{M}|)}. \tag{A3}$$

b indexes branches and r feature coordinates. Losses average over horizons and windows, with equal weights in Eq. (5). I-PHYRE scores pairs present at both compared times and involving a dynamic object. Empty pair sets contribute zero.

Prediction losses train the shared encoder and predictor on feature changes. Trajectory loss trains their use through conditioning and response heads. The reference input is detached, while its projection and recursive forecasts remain trainable.

B.3 Finite-Time Response and Integration Stages

For acceleration history $a(\tau)$ at physical time $t+h\tau$, with $\tau \in [0,1]$, define its mean $A = \int_0^1 a(\tau)d\tau$ and centered temporal moment $B = \int_0^1 (1/2 - \tau)a(\tau)d\tau$. The exact endpoint response is

$$v_{t+h} = v_t + hA, \qquad x_{t+h} = x_t + hv_t + h^2(\tfrac{1}{2}A + B). \tag{A4}$$

Both A and B have acceleration units. For a vector b, the perturbation $\delta a(\tau) = 6b(1-2\tau)$ has zero integral but satisfies $\int_0^1 (1-\tau)\delta a(\tau)d\tau = b$. It therefore changes displacement by $h^2 b$ while preserving endpoint velocity. This is the displacement freedom parameterized by TRD.

Shared heads produce (a_0, c_0), (a_m, c_m), and (a_1, c_1) at the initial state, midpoint preview, and endpoint preview, respectively. Both previews start from S_t:

$$\begin{aligned} \widetilde{x}_m &= x_t + \tfrac{h}{2}v_t + \tfrac{h^2}{8}a_0, \quad \widetilde{v}_m = v_t + \tfrac{h}{2}a_0, \\ \widetilde{x}_1 &= x_t + hv_t + \tfrac{h^2}{2}a_m, \quad \widetilde{v}_1 = v_t + ha_m. \end{aligned} \tag{A5}$$

Each stage queries the current physical encoding with aligned forecast context. I-PHYRE uses $U_s^J - U_0$, and N-body uses $U_s^J - U_s^D$, as in Eq. (2).

The conditioning attention uses bias-free value and output projections, giving zero correction when $Z_s = 0$. Additional I-PHYRE pair-geometry evaluations share the same-time forecast but supply their own queries. For pairs removed before the right endpoint of a forecast interval, the left forecast is held until removal.

With $\widehat{A} = (a_0 + 4a_m + a_1)/6$, $\widehat{D} = a_0/6 + a_m/3$, and $\widehat{C} = (c_0 + 4c_m + c_1)/6$, the update estimates $\widehat{B} = \widehat{D} + \widehat{C} - \widehat{A}/2 = (a_0 - a_1)/12 + \widehat{C}$. Thus $\widehat{C}$ corrects the temporal moment approximated by the acceleration stages. Adding $h^2\widehat{C}$ to position preserves acceleration-based previews and the current step's velocity update. The corrected position enters subsequent responses. Angular states follow the same construction.

B.4 Momentum and Center-of-Mass Motion

For fixed positive masses and no external forces, the two heads produce antisymmetric pair vectors $f_{ij} = -f_{ji}$ and $g_{ij} = -g_{ji}$. Aggregation gives $m_j a_j = \sum_i f_{ij}$ and $m_j c_j = \sum_i g_{ij}$, so both mass-weighted sums vanish at every stage. With total mass M, momentum $P_t = \sum_j m_j v_{t,j}$, and center of mass $X_t = M^{-1} \sum_j m_j x_{t,j}$, the discrete update satisfies

$$P_{t+h} = P_t, \qquad X_{t+h} = X_t + hP_t/M. \tag{A6}$$

Antisymmetrization follows forecast conditioning, so these identities hold in exact arithmetic for any predicted features.

C Dataset Configurations

We use the structured-state datasets and splits of NFF (Li et al., 2026). Table A1 lists their sizes and horizons. All rollouts start from the initial state, with prescribed removal schedules supplied for I-PHYRE.

Table A1: Dataset splits and rollout lengths. A frame interval is 0.1 seconds in both benchmarks. The N-body training horizon contains $T = 50$ frames, while $3T$ contains 150 frames.

Benchmark	Split	Trajectories	Frames	Configuration
I-PHYRE	Train	100	150	10 basic games
	Within	200	150	Same 10 games
	Cross	600	150	30 unseen games
N-body	Train	200	50	Two or three bodies
	Validation	200	50	Two or three bodies
	Within	200	50 / 150	Two or three bodies
	Cross	200	50 / 150	Eight or ten bodies

C.1 I-PHYRE

Following NFF (Li et al., 2026), training uses ten basic games from I-PHYRE (Li et al., 2024): support, hinder, direction, hole, fill, seesaw, angle, impulse, pendulum, and spring. Each contributes five successful and five unsuccessful action sequences. Within evaluation uses ten of each per game. Cross evaluation has the same balance across 30 unseen games: ten perturbed configurations, ten compositions, and ten multi-ball configurations.

States contain geometry, orientation, velocities, mobility, removability, and joint or spring connections. Positions and object dimensions are normalized by scene size, and angles are represented as $\theta/10$. Prescribed removals update object validity and reference masks. Scoring uses valid dynamic objects up to scene termination.

C.2 Gravitational N-body Dynamics

We use the NFF benchmark (Li et al., 2026), whose planetary and cometary trajectories are generated with REBOUND. Each split contains 50 trajectories per combination of initialization type and body count. Within uses new two- and three-body initial conditions. Cross uses eight or ten bodies with broader radius and mass ranges (Table A2). Parameters are sampled by Latin hypercube sampling. All bodies, including the central body, have evolving three-dimensional positions and velocities.

Table A2: N-body initialization ranges used by the benchmark. Cross evaluation expands both system size and physical parameter ranges. Central and orbiting masses are sampled independently within the prescribed ranges.

Quantity	Train / Within	Cross
Total bodies	2 or 3	8 or 10
Initial orbital radius	[1, 3]	[1, 5]
Orbiting-body mass	[0.05, 0.10]	[0.05, 0.15]
Central-body mass	[3, 5] or [7, 9]	[3, 9]

D Training and Evaluation Details

D.1 Architecture and Optimization

Table A3 summarizes architecture and optimization. Both benchmarks share pair features between RA-JEPA and the response heads (Section 3.2).

Objectives. I-PHYRE uses 10^4 times masked mean squared error (MSE) on $(x, y, \ell, \theta/10)$, with normalized segment length ℓ. N-body equally weights position, velocity, and response-moment

Table A3: HORME training settings. Both models start from random initialization. Each window spans K intervals of duration h. The local-loss fraction α weights local versus recursive prediction in Eq. (A10).

Setting	I-PHYRE	N-body
Hidden width per physical branch	256	256
Predictor width / blocks / heads	256 / 4 / 4	256 / 4 / 4
Forecast interval h	0.1 s	0.1 s
Transitions per window K	5	4
Global batch, trajectories	4	200
Optimizer	Adam	Adam
Training budget	200 epochs	5,000 updates
Initial learning rate	5×10^{-4}	5×10^{-4}
Final learning rate	10^{-5}	10^{-7}
Schedule	Cosine	Cosine
Weight decay	10^{-5}	10^{-5}
Predictive weight λ	0.1	0.1
Local-loss fraction α	0.5	0.5

MSE. The moments $M_0 = m(v_{t+h} - v_t)$ and $M_1 = m(x_{t+h} - x_t - hv_t)$ are velocity change and displacement beyond constant-velocity motion, both weighted by receiver mass. Predictions use model endpoints and targets use observed endpoints.

Windows and model selection. Training windows contain $K + 1$ observed frames: six for I-PHYRE and five for N-body. At inference, forecasts condition every integration stage, and each window starts from the preceding predicted endpoint. I-PHYRE uses the final online model. N-body selects the online checkpoint minimizing state MSE over its 200 validation trajectories at horizon T.

D.2 Evaluation Metrics

Let z_{tjc}, $\widehat{z}_{tjc}$, and m_{tjc} denote ground truth, prediction, and validity at time t, object j, and channel c. Scored channels are normalized $x, y, \theta/10$ for I-PHYRE and mass, position, and velocity for N-body. Per-trajectory RMSE is

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{t,j,c} m_{tjc}(\widehat{z}_{tjc} - z_{tjc})^2}{\sum_{t,j,c} m_{tjc}}}. \tag{A7}$$

For the final frame t_f with valid ground-truth coordinates, define adjacent changes $\Delta z_{tjc} = z_{tjc} - z_{t-1,j,c}$ and mask $m^{\Delta}_{tjc} = m_{tjc} m_{t-1,j,c}$. Then

$$\begin{aligned} \mathrm{FPE} &= \frac{\sum_{j,c} m_{t_f jc} |\widehat{z}_{t_f jc} - z_{t_f jc}|}{\sum_{j,c} m_{t_f jc}}, \\ \mathrm{PCE} &= \frac{\sum_{t \geq 1,j,c} m^{\Delta}_{tjc} |\Delta \widehat{z}_{tjc} - \Delta z_{tjc}|}{\sum_{t \geq 1,j,c} m^{\Delta}_{tjc}}. \end{aligned} \tag{A8}$$

Correlation R follows the released benchmark implementation. I-PHYRE zeros invalid entries before centering and retains centered padding in the covariance and variance sums. Its error metrics use benchmark clipping and masks. N-body timestamps run from 0 to 4.9 at T and 14.9 at $3T$, spaced by 0.1. Metrics are averaged across n test trajectories, with $\mathrm{SEM} = s/\sqrt{n}$ for sample standard deviation s.

D.3 Trajectory Figure Protocol

Trajectory figures score position alone. For valid dynamic objects $\mathcal{V}_r(t)$ in trajectory r and position dimension d_x,

$$e_r(t) = \sqrt{\frac{\sum_{j \in \mathcal{V}_r(t)} \|\widehat{x}_{r,t,j} - x_{r,t,j}\|_2^2}{d_x |\mathcal{V}_r(t)|}}. \tag{A9}$$

I-PHYRE curves use fixed sample IDs $1, 26, 51, 76$ from each game, equivalent to taking every fifth entry of the ordered test list $1, 6, \ldots, 96$. The same subset is used for all methods, giving 40

Within and 120 Cross trajectories. Table 1 uses all 20 test trajectories per game (200 Within and 600 Cross). I-PHYRE scoring clips two-dimensional positions to $[0, 1]$, with terminated scenes excluded at later times. N-body uses unclipped three-dimensional positions from all 200 trajectories per split. Figure 4 shows the trajectory mean and one SEM at each time.

Method rows share initial states, intervention schedules, timestamps, cameras, colors, and bounds. Paths connect saved states. I-PHYRE overlays up to three states from faint to opaque, with removed supports shown translucently. N-body spheres mark endpoints and scale with relative mass. Arrows mark viewport exits.

D.4 Spring Diagnostic Setup

Figure 3 uses the analytic dynamics $x'' = 0$ and $y'' = -36y - 20y^3$ over 20 steps of $h = 0.15$ seconds. The shared cubic readout $c(y, v)$ has ten coefficients in $(y/0.35, v/2.5)$, fitted to 512 independent one-step position targets by ridge least squares with penalty 10^{-10}. Training states sample $y \sim \mathcal{U}[-0.35, 0.35]$ and $v \sim \mathcal{U}[-2.5, 2.5]$. The 128 held-out trajectories use $y \sim \mathcal{U}[-0.055, 0.055]$ and $v \sim \mathcal{U}[0.65, 1.35]$. Fourth-order Runge–Kutta (RK4) integration with step 0.0015 supplies reference trajectories. Quadrature and Quadrature + TRD share Eq. (A5), analytic acceleration, and three queries per step.

E Additional Experimental Results

E.1 Sensitivity to Predictive Loss Weights

To examine the balance between feature prediction and trajectory fitting, Table A4 varies the overall predictive weight λ and the local-loss fraction α. We write the objective as

$$\mathcal{L} = \mathcal{L}_{\text{traj}} + \lambda\big[\alpha\mathcal{L}_{\text{local}} + (1-\alpha)\mathcal{L}_{\text{recursive}}\big]. \tag{A10}$$

The default $\alpha = 0.5$ recovers Eq. (5), with $\lambda = 0.1$. Each sweep varies one weight while fixing the other at its default. Architecture, data, optimization, and evaluation settings remain the same.

Table A4: Predictive loss weights (RMSE ↓). Stars mark defaults, whose results match Table 4. Bold values mark the lowest RMSE in each sweep.

Parameter	Value	I-PHYRE		N-body $3T$	
		Within	Cross	Within	Cross
Predictive weight λ	0.01	0.043	0.112	0.218	0.432
	0.025	0.045	0.116	0.195	0.486
	0.05	0.041	0.106	0.181	0.376
	0.1*	**0.039**	**0.099**	**0.170**	**0.339**
	0.2	0.043	0.106	0.188	0.363
	0.5	0.042	0.108	0.207	0.401
	1.0	0.045	0.119	0.246	0.478
Local-loss fraction α	0.1	0.043	0.112	0.209	0.415
	0.25	0.044	0.115	0.197	0.432
	0.4	0.042	0.111	0.194	0.386
	0.5*	**0.039**	**0.099**	**0.170**	**0.339**
	0.6	0.042	0.108	0.179	0.354
	0.75	0.044	0.114	0.184	0.366
	0.9	0.043	0.111	0.205	0.409

The defaults attain the lowest RMSE in all four evaluation columns of both sweeps. Increasing λ from 0.01 to 0.1 reduces Cross RMSE from 0.112 to 0.099 on I-PHYRE and from 0.432 to 0.339 on N-body. At $\lambda = 1.0$, the corresponding errors rise to 0.119 and 0.478. Smaller weights weaken the explicit constraint on pair-feature evolution, while larger weights emphasize feature matching relative to trajectory fitting. The results favor a moderate predictive weight that couples feature supervision to the physical rollout objective.

The α sweep separates two forms of temporal supervision at a fixed overall predictive weight. The local loss constrains one-step increments from encodings of observed states. The recursive loss constrains cumulative changes when the predictor consumes its own forecasts. Equal weighting gives the lowest reported errors, whereas emphasizing either term through $\alpha = 0.1$ or 0.9 increases Cross RMSE on both benchmarks. This pattern is consistent with the complementary roles of learning local transitions and supervising the errors accumulated during recursive prediction.

For nearby settings $\lambda \in \{0.05, 0.1, 0.2\}$ or $\alpha \in \{0.4, 0.5, 0.6\}$, with the other weight fixed at its default, Cross RMSE ranges from 0.099 to 0.111 on I-PHYRE and from 0.339 to 0.386 on N-body. These comparisons support the shared default $(\lambda, \alpha) = (0.1, 0.5)$ across the two benchmarks.

E.2 Computational Cost

Figure A1 reports Cross accuracy and measured rollout time for the evaluated I-PHYRE inference implementations. Each point uses the same checkpoint for accuracy and timing, with RMSE averaged over all 600 Cross trajectories. Timing uses a fixed support scene (sample 1) with 12 padded object slots, three occupied slots, and 150 output frames. All methods receive the same initial state and removal schedule on one RTX 4090, using float32, batch size one, and four CPU threads, with TensorFloat-32 disabled. Latency is the median of five synchronized complete rollouts after two warmups, excluding model loading, data preparation, scoring, and one-time initialization.

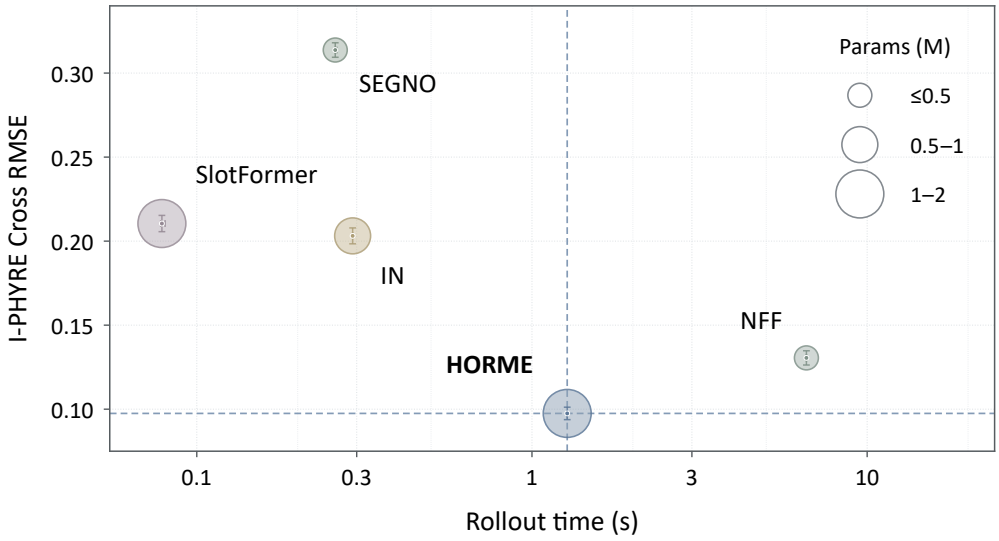


Figure A1: I-PHYRE accuracy and inference cost. Circle sizes indicate parameter intervals. Vertical bars show trajectory SEM ($n = 600$), and horizontal bars span five rollout timings. HORME uses the 1.48M-parameter inference configuration.

This inference configuration achieves a Cross RMSE of 0.0975 with a 1.274-second rollout. Relative to NFF in this workload, RMSE is 25.3% lower and rollout is $5.17\times$ faster. Supported matrix and attention operations total 18.30 billion floating-point operations (GFLOPs) for this workload, counting a multiply-add as two operations.

E.3 Feature Prediction across Horizons

Figures 2 and A2 use feature changes from a frozen I-PHYRE model across 800 test trajectories. Windows start every 20 frames and cover horizons from 0.1 to 0.5 seconds. One valid ordered pair per window and horizon gives 26,562 paired samples. Here U_k^D encodes the observed state at kh. The observed change is $U_k^D - U_0$, and the forecast change is $U_k^J - U_0$.

Principal components are fitted to observed changes and applied to both sets, preserving zero change at the origin. The first two axes explain 54.48% and 11.36% of observed variation. Density estimates use Gaussian smoothing with bandwidth 0.12 and display range $[-4, 4]^2$. Figure 2(a,b) pools horizons and maps density to surface height through a shared logarithmic transform. Figure A2 separates horizons using a common logarithmic color scale. The lower-left panel reports mean Euclidean change norms in this two-dimensional projection. The lower-right panel measures paired MAE across all 512 feature coordinates, using the same sampled pairs for RA-JEPA and a zero-change forecast that retains U_0. All statistics include sampled points outside the displayed density range.

Observed and predicted change norms increase with horizon, with smaller magnitudes for predictions in the displayed PC projection. Paired MAE also increases with horizon but remains below the zero-change error at every step. These paired comparisons show that the forecasts capture individual feature changes beyond retaining the initial features. Figure 2(c) reports the percentage MAE reduction relative to zero change.

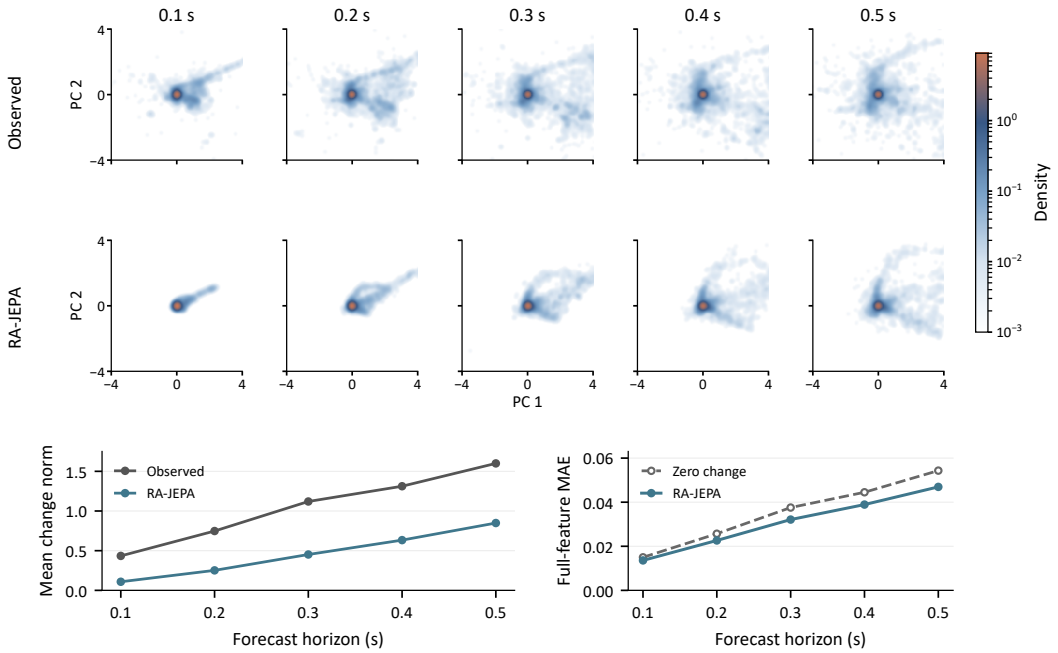


Figure A2: Observed and forecast feature changes across five horizons from 800 I-PHYRE trajectories. The density panels share principal components and a logarithmic color scale. Bottom left, mean change norms in the two-dimensional projection. Bottom right, paired MAE across all 512 feature coordinates, compared with a zero-change forecast that holds U_0 fixed. Both methods use the same 26,562 pair-horizon samples.

F Additional Trajectory Visualizations

The cases in Figures A3 to A7 are selected from ground-truth motion independently of model error. I-PHYRE uses the lower median path length within each game. N-body uses cases near the 35th and 65th path-length percentiles within each body count.

All panels follow the scoring and rendering protocol in Section D.3. Rows compare ground truth, HORME, NFF, and SlotFormer, with per-case position RMSE below. Dotted lines in the N-body curves mark the training horizon at 4.9 seconds.

F.1 I-PHYRE Within Configurations

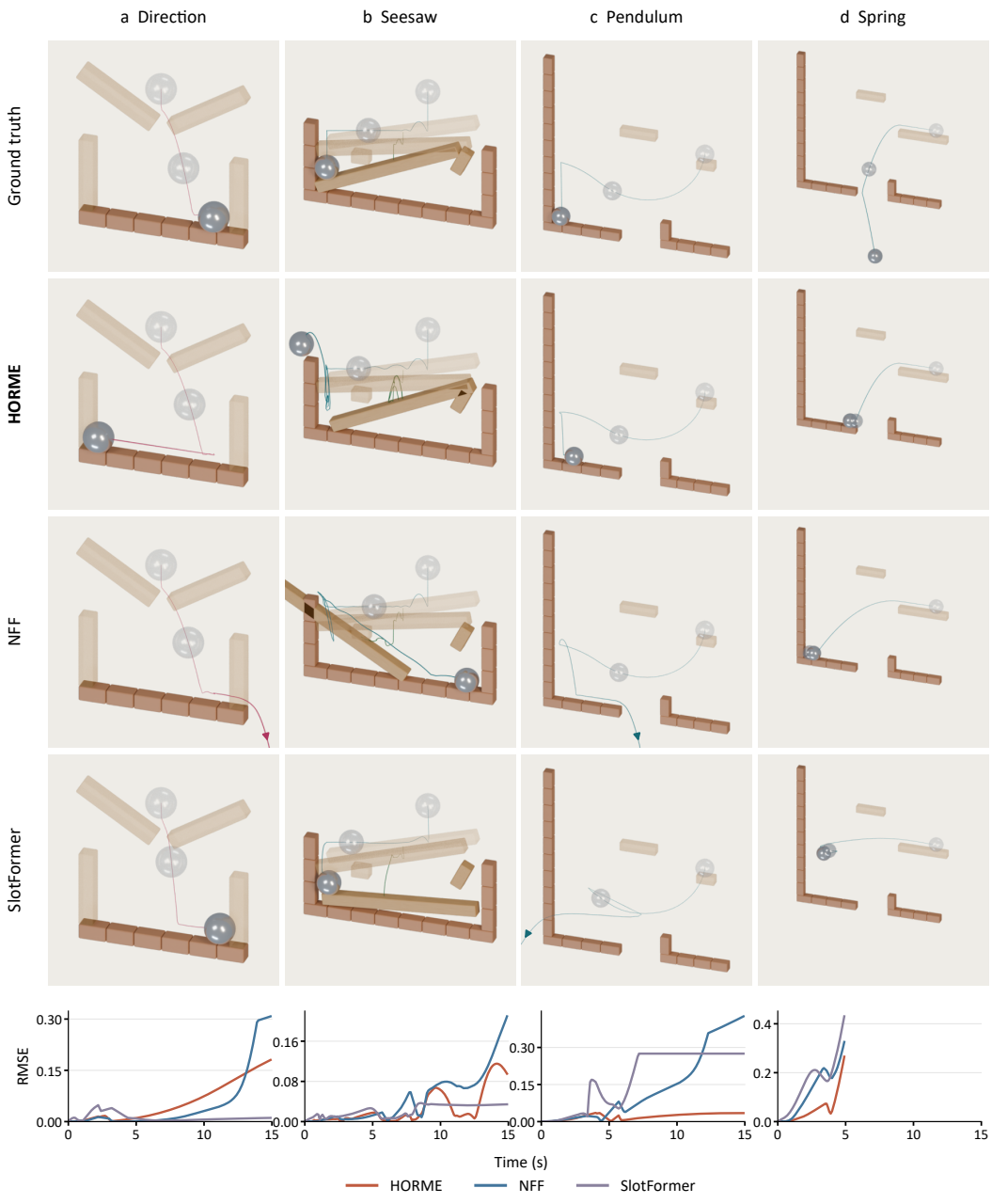


Figure A3: Familiar I-PHYRE games under new action sequences: direction, seesaw, pendulum, and spring. Each column is an independent scene.

F.2 I-PHYRE Perturbed and Compositional Configurations

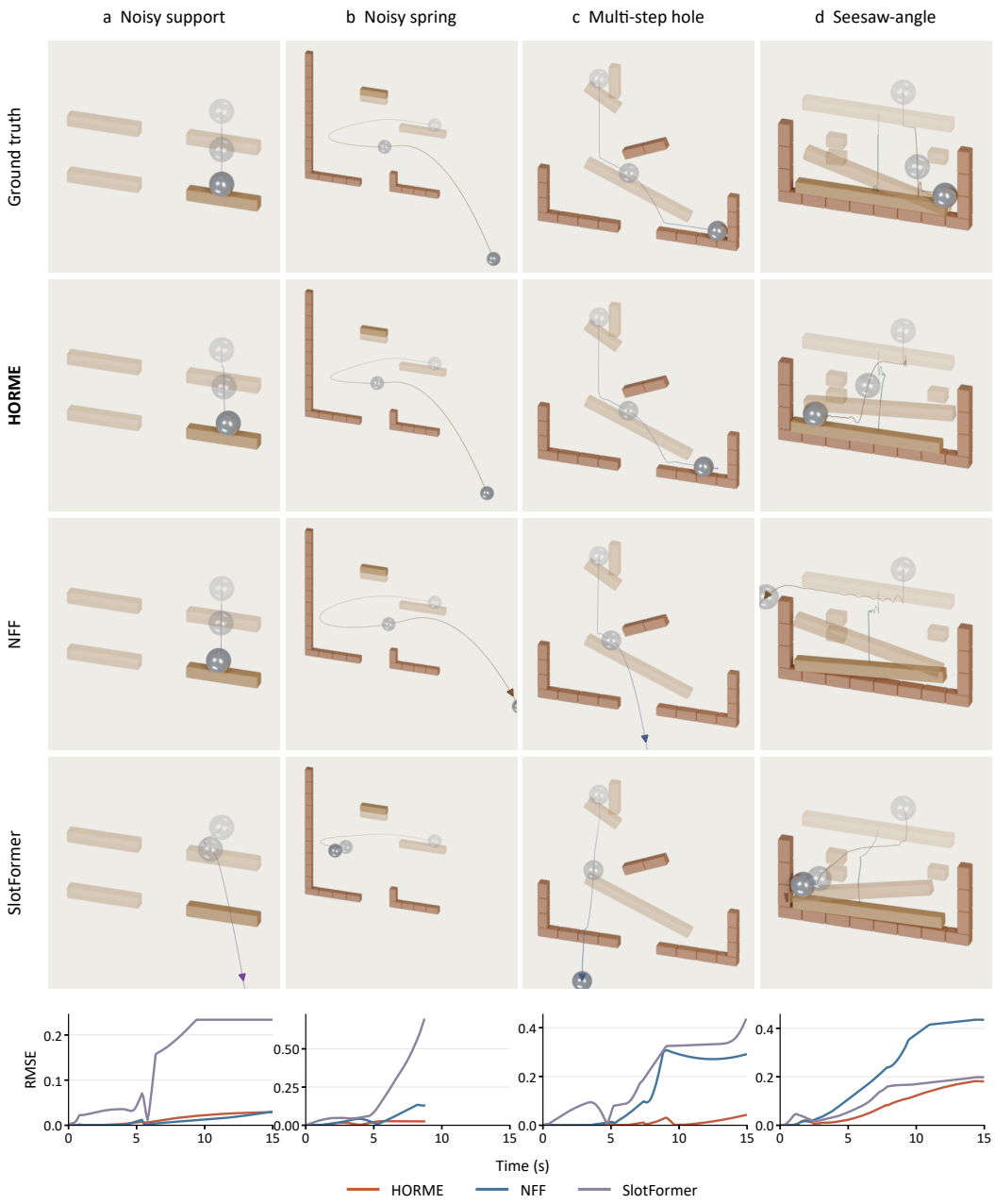


Figure A4: I-PHYRE Cross cases with perturbed or compositional configurations: noisy support, noisy spring, multi-step hole, and seesaw-angle.

F.3 I-PHYRE MULTI-BALL CONFIGURATIONS

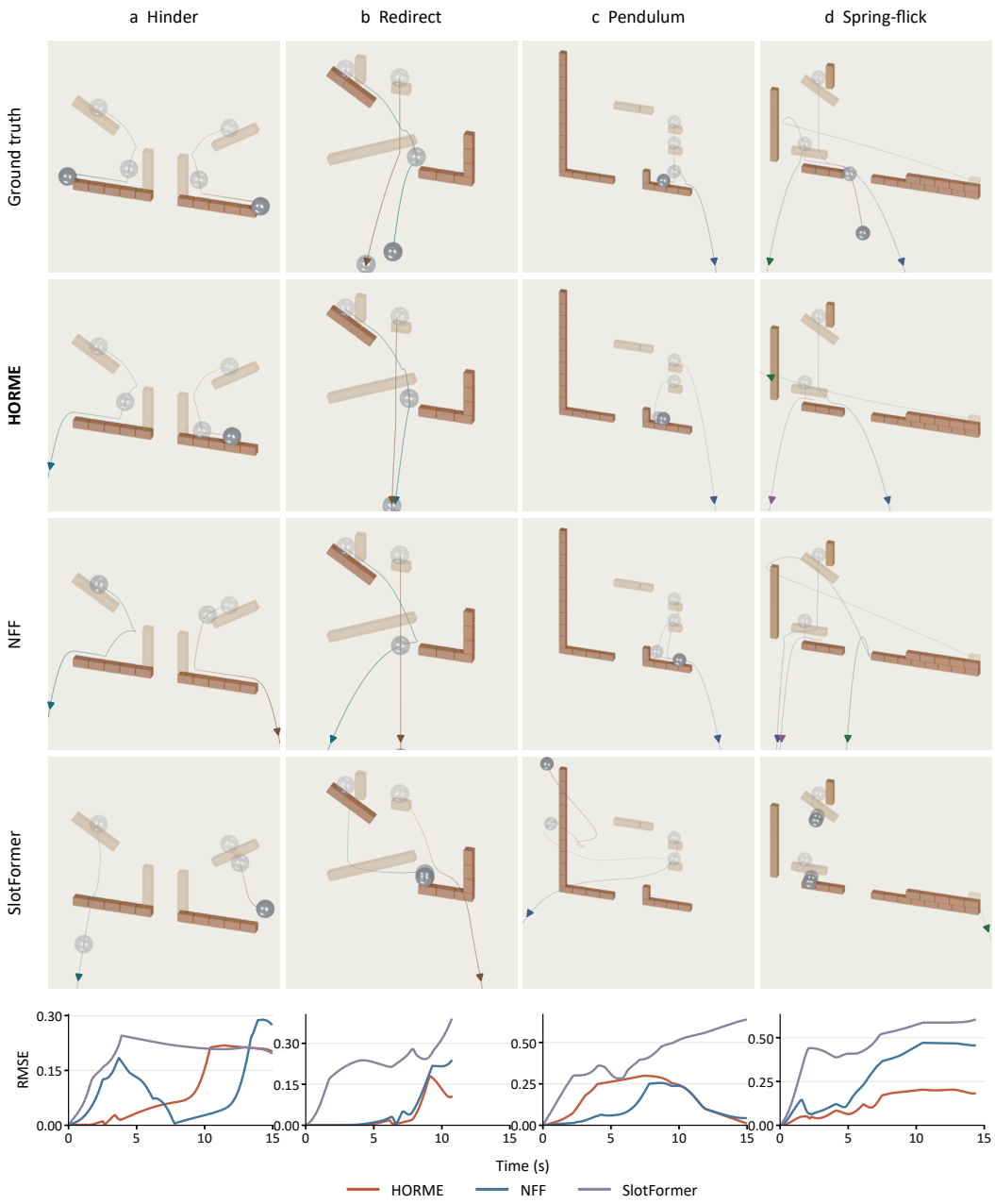


Figure A5: Unseen I-PHYRE games with multiple balls: obstructed paths, redirected motion, pendulum interactions, and a spring mechanism.

F.4 N-BODY WITHIN CONFIGURATIONS

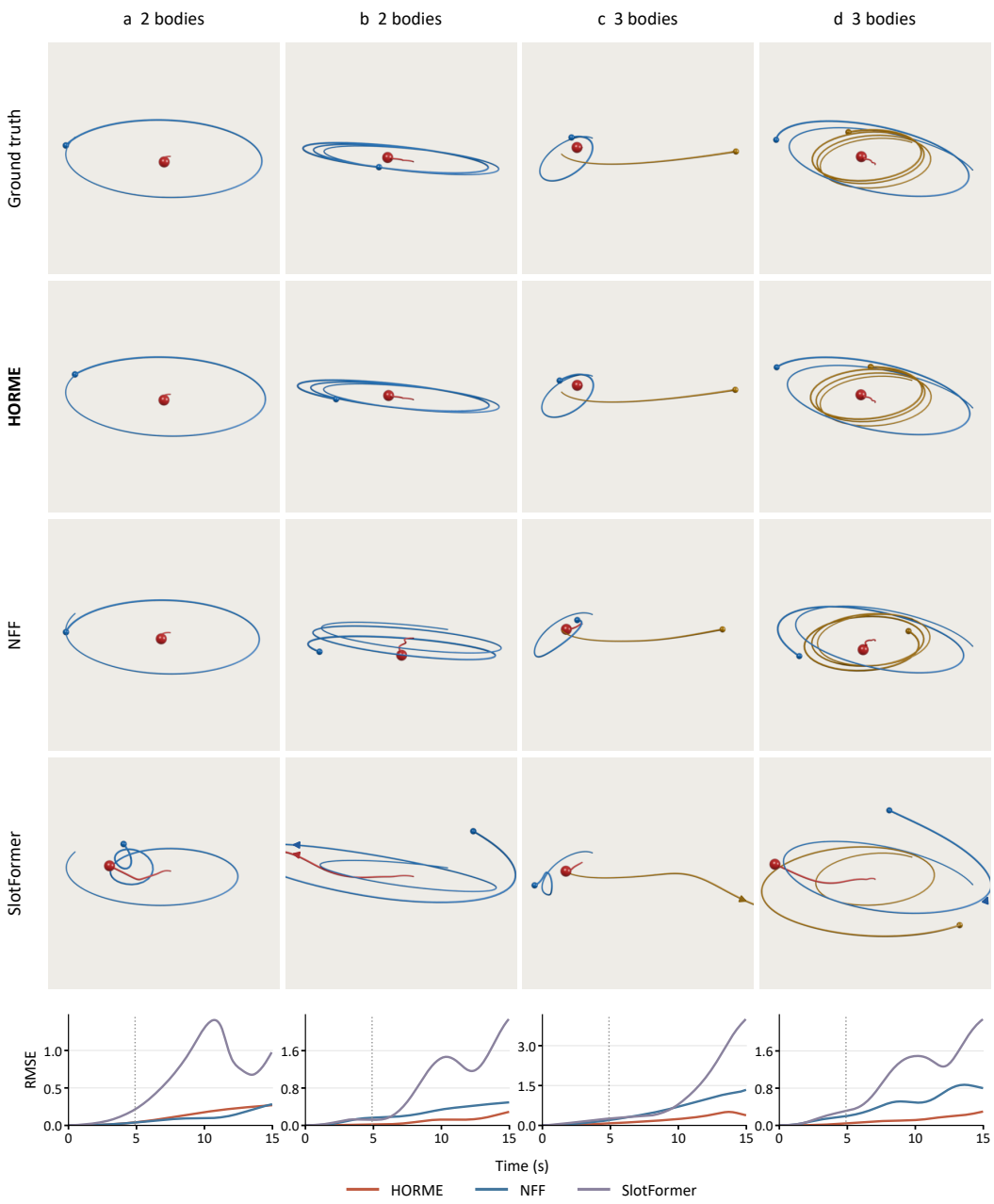


Figure A6: Two- and three-body predictions over 150 frames ($3T$). Paths show complete trajectories and spheres mark endpoints.

F.5 N-BODY CROSS CONFIGURATIONS

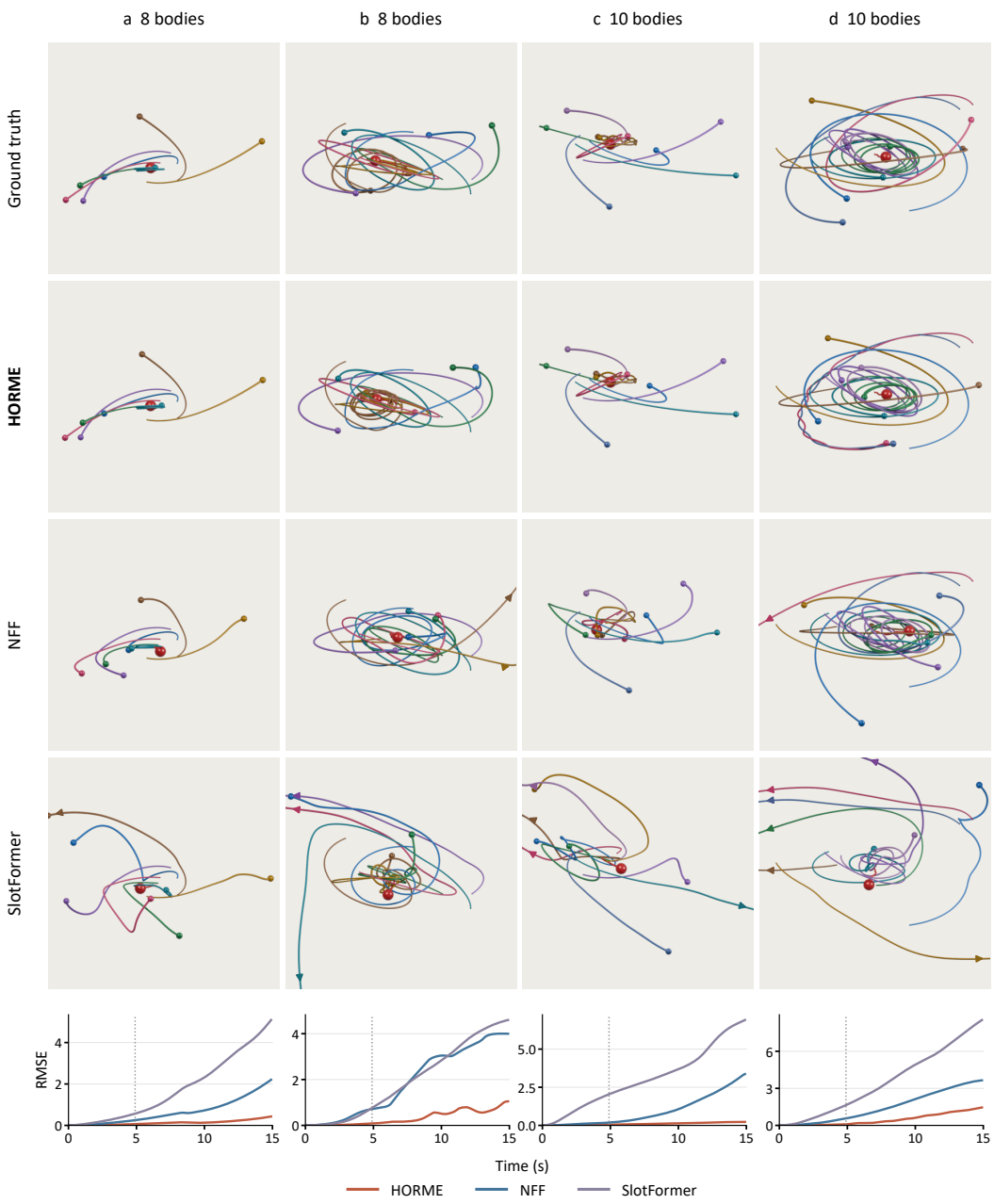


Figure A7: Eight- and ten-body predictions with broader physical parameters over 150 frames ($3T$). Arrows mark viewport exits.

G LIMITATIONS

Our evaluation focuses on simulated contact and gravitational dynamics with structured object states. The applicability of HORME to prediction from raw visual observations remains to be studied. Future work could integrate the framework with learned perception and evaluate its robustness to state-estimation errors and its transfer to real-world physical sequences.