NEURONS AS WORLDS: LEARNING ADAPTIVE MESOSCOPIC DYNAMICS FROM PARTICLES

Anonymous authors
Paper under double-blind review

ABSTRACT

Coarse particle group representations can reduce the communication cost of particle simulation, but changing particle groups disconnects their current contents from the accumulated predictive context. The challenge is to reorganize the representation without restarting its history. We introduce *Neuronal World Dynamics* (NWD), which treats regrouping as a state transition of recurrent particle group representations called *neuronal worlds*. As the particle groups splits and merges, physical moments are recomputed from the unchanged particles, while compact memories are copied or combined locally. Shared recurrent dynamics then refine the inherited context using each successor's current observations, without replaying trajectories. The theoretical analysis identifies when a larger decoder cannot resolve coarse-state ambiguity and quantifies the trade-off between suppressing reorganization and retaining learned dynamics. Across four particle benchmarks, NWD achieves the lowest mean 100-step position NRMSE in all five settings, with 51.8–52.6% reductions over the recurrent-token baseline on N-body. On N-body, NWD reduces this error by 10.7–14.5% relative to fixed-cardinality variants and 51.8–52.6% relative to a recurrent-token baseline; memory ablations provide additional evidence for retained context and its association with particle support.

1 INTRODUCTION

Learning particle dynamics requires resolving local interactions while communicating across the physical system. Graph simulators capture local interactions through message passing, but distant dependencies require multiple hops or an additional hierarchy (Sanchez-Gonzalez et al., 2020; Pfaff et al., 2021; Martinkus et al., 2021). Latent simulators reduce this communication burden by representing many particles with fewer computational units (Alkin et al., 2024; Wu et al., 2024; Wang et al., 2026). These units operate on a *mesoscopic* scale between individual particles and the entire system. The resulting compression procedure introduces a representational challenge: a group that is useful now may later deform, exchange particles, or separate into components with different futures. A fixed allocation can then underresolve the changing dynamics, while rebuilding the groups can disrupt the association between their current contents and accumulated predictive context. Moreover, instantaneous coarse summaries do not need to contain enough information to determine their own future (Zwanzig, 1961; Mori, 1965; Yu et al., 2025).

Spatial adaptation and temporal memory address different parts of this problem. UPT propagates a compressed latent state, and GraMO couples graph interactions with recurrent dynamics (Alkin et al., 2024; Tiwari et al., 2026). Transolver, EvoMesh, and MeshTok adapt spatial aggregation or resolution, while WorldParticle constructs super-tokens from the current particle state (Wu et al., 2024; Deng et al., 2025; Zhao et al., 2026; Wang et al., 2026). In this work, we focus on the interface between these operations: *what recurrent context should initialize the successors of a particle group when it splits or merges?* This state-transfer problem is distinct from propagating a fixed set of latent units or rebuilding a spatial representation from current observations.

Our starting point is to treat the regrouping as a learnable transition of the predictive state, rather than a fresh encoding of the current particles. Consider a particle group separating into two components. The successors have different physical contents, but both descend from the same interaction history. Their masses and momenta should therefore be recomputed from their own particles, while a compact summary of the parent's history can initialize both successors. The shared recurrent dynamics then

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

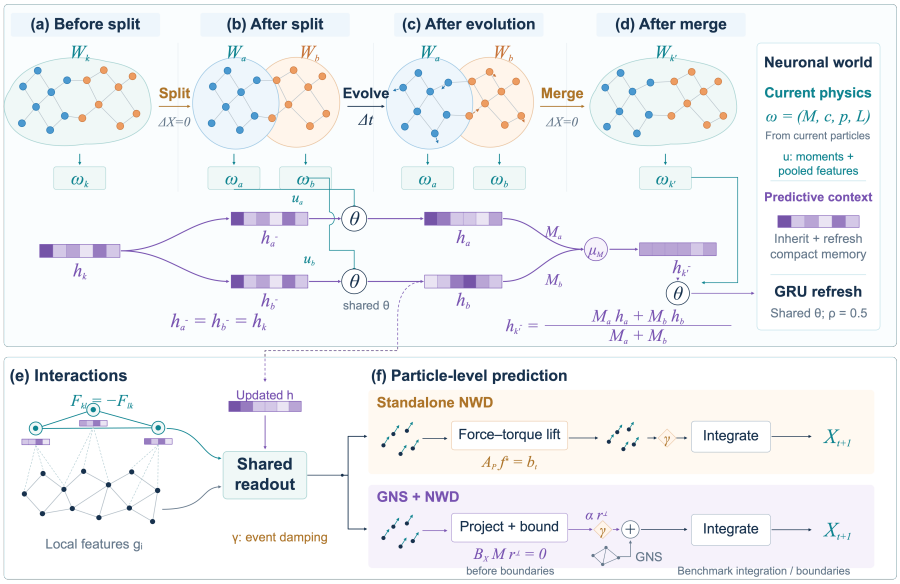


Figure 1: **Neuronal World Dynamics (NWD): reorganizing predictive units without restarting their context. (a–d)** Splits and merges leave physical particles unchanged. Successor moments are recomputed from current support; memories are copied at splits, mass-averaged at merges, and refined through shared recurrent dynamics. **(e)** Local particle features, sparse neuronal-world interactions, and updated memory feed a shared readout. **(f)** Predictions use a force–torque lift for standalone simulation or a projected correction with a bounded mixing coefficient for GNS+NWD, followed by event attenuation and integration. The displayed constraint equalities describe the ideal interfaces before boundary operations.

refines the inherited context using each successor's current observations. The same principle applies in reverse when groups merge. This separation between *remeasured physical state* and *inherited predictive context* makes adaptation possible without freezing the grouping or restarting its memory. The units do not need to correspond to uniquely identifiable objects; they are computational entities whose support and context evolve together.

We realize this principle with *Neuronal World Dynamics*, *i.e.*, NWD (Figure 1). Each *neuronal world* is a recurrent neural unit with sparsely routed particle support, physical moments, and a fixed-width learned memory. Routing changes its receptive field, and split–merge events change the active population while explicitly transferring recurrent state to the successors. Shared parameters let new units immediately reuse the learned dynamics; no new network is trained when a unit is created. Local particle updates retain fine detail, while sparse neuronal-world interactions communicate between groups. The resulting neuronal representation supports either a standalone prediction or a gated correction to a jointly trained graph backbone, denoted GNS+NWD.

The theoretical analysis validates three requirements that are easy to conflate. First, identical coarse observations can hide different futures, imposing an error floor that a larger decoder alone cannot remove. Second, preserving aggregate force and torque still permits many particle-level corrections and does not guarantee accurate trajectories. Third, regrouping can change a predicted update even when the particles are unchanged; attenuating that change also omits part of the learned dynamics. These results motivate state inheritance and explicit output interfaces without claiming that finite memory, adaptive grouping, or event damping guarantees accurate closure.

We evaluate four benchmarks spanning gravity, bonded dynamics, fluids, and granular materials. Across the two three-seed DyWorlds-NBody protocols, NWD reduces 100-step position NRMSE by 51.8–52.6% relative to STT-R and 10.7–14.5% relative to fixed-cardinality variants. Its advantage over the tested fixed-cardinality baseline persists at 50 and 100 steps across 256–2,048 particles. On Dam2D and Sand2D, the jointly trained GNS+NWD hybrid reduces 100-step position MSE by 37.0% and 11.3% relative to enlarged GNS-A under the same 2,000-update budget and within 0.7% of its parameter count; these public-benchmark comparisons use one training seed. N-body ablations show

the largest penalties when persistent memory is removed or its association with particle groups is disrupted. In general, our contributions are:

- **Adaptive predictive units with inherited context.** We introduce NWD, which evolves particle support and recurrent state together as learnable neuronal worlds representation. Its split–merge interface remeasures physical moments while transferring compact context, allowing the representation to change without restarting its predictive history.
- **Analysis of coarse-state ambiguity and reorganization.** We establish when instantaneous coarse observations impose unavoidable prediction error, distinguish aggregate force–torque constraints from trajectory accuracy, and quantify the trade-off between suppressing reorganization shocks and retaining learned updates.
- **Long-rollout evaluation and mechanism analysis.** Across four particle benchmarks, we evaluate NWD as a standalone simulator and a graph correction. The comparisons show long-rollout gains over fixed-cardinality and recurrent-token baselines on N-body and improved graph-hybrid prediction on fluids and granular materials, with N-body ablations probing the roles of memory and its association with particle support.

2 Related Work

Particle and latent simulators. Interaction Networks, GNS, and MeshGraphNets model physical elements with learned graph interactions (Battaglia et al., 2016; Sanchez-Gonzalez et al., 2020; Pfaff et al., 2021). Equivariant and conservation-aware models constrain the interaction law (Brandstetter et al., 2022; Sharma & Fink, 2026). The set Transformer and Perceiver introduced inducing-point and latent bottlenecks for large inputs; UPT applies compressed latent evolution to physics (Lee et al., 2019; Jaegle et al., 2021; Alkin et al., 2024). Transolver, EvoMesh, MeshTok, and WorldParticle adapt physical slices, graph hierarchies, token resolution, and super-token aggregation, respectively (Wu et al., 2024; Deng et al., 2025; Zhao et al., 2026; Wang et al., 2026). NWD retains an explicit correspondence between changing the particle support and the recurrent state, with inheritance rules when that support divides or combines.

Memory and changing representations. GraMO couples graph propagation with recurrent state-space dynamics (Tiwari et al., 2026); coarse-graining theory and graph closure models explain why unresolved dynamics can require memory (Mori, 1965; Zwanzig, 1961; Yu et al., 2025). NWD uses a gated recurrent unit (Cho et al., 2014) as the memory primitive and makes its state transferable across changes in the number and support of predictive units. DoughNet predicts object topology, slot-merging methods adapt latent decompositions, and Topological Neural Operators propagate fields over cell complexes (Bauer et al., 2024; Chatzisavvas et al., 2026; Bastian et al., 2026). Neural Particle Automata learns self-organizing microscopic agents, while numerical merging–splitting changes physical particle resolution (Pajouheshgar et al., 2026; Truong et al., 2021). NWD instead preserves the physical particles and reorganizes their recurrent computational units; classical tree and multipole solvers remain distinct numerical approaches to long-range forces (Barnes & Hut, 1986; Greengard & Rokhlin, 1987). Table 4 summarizes the corresponding state-transition distinctions.

3 Neuronal World Dynamics

3.1 Problem formulation and method overview

Given a particle state $X_t = \{z_{i,t}\}_{i=1}^{N}$, each $z_{i,t} = (x_{i,t}, v_{i,t}, m_i, \xi_{i,t})$ contains position and velocity $x_{i,t}, v_{i,t} \in \mathbb{R}^d$, mass $m_i > 0$, and attributes $\xi_{i,t} \in \mathbb{R}^{d_a}$, with $d \in \{2, 3\}$. Using known external forcing, boundary conditions, and timestep $\Delta t > 0$, we predict $\widehat{X}_{t+1:t+H}$ over H autonomous steps. Causal history is available only up to the prediction start; subsequent inputs use predicted states.

NWD augments the unchanged physical discretization with $K_t \ll N$ neuronal worlds and memberships $P_t \in [0,1]^{N \times K_t}$, satisfying $P_t \mathbf{1} = \mathbf{1}$ with at most r nonzeros per row. Its predictive state and transition are

$$\mathcal{S}_t = (X_t, \mathcal{W}_t, P_t), \qquad \mathcal{S}_{t+1} = \Phi_\theta^{\gamma_t}(\Psi_{e_t}(\mathcal{R}_\theta(\mathcal{S}_t))) \,. \tag{1}$$

The router $\mathcal{R}_\theta$ updates support and physical moments. The event map Ψ_{e_t} transfers support and memory through a split or merge and recomputes successor moments without changing X_t; $\Psi_\varnothing = \mathrm{Id}$.

The shared update $\Phi_\theta^{\gamma_t}$ exchanges messages, refreshes memory, and integrates particle motion, with $\gamma_t \in [0, 1]$ attenuating the learned update on event steps. Causal-history buffers and lifecycle bookkeeping are carried between steps but omitted from $\mathcal{S}_t$ for readability.

3.2 Neuronal-world state and sparse communication

For unit k, $\omega_{k,t}$ contains physical moments, $h_{k,t} \in \mathbb{R}^{d_h}$ is a fixed-width learned memory, id_k links that memory to its current particle support, and ℓ_k records its parent identifiers:

$$\mathcal{W}_t = \{W_{k,t}\}_{k=1}^{K_t}, \qquad W_{k,t} = (\omega_{k,t}, h_{k,t}, \mathrm{id}_k, \ell_k). \tag{2}$$

The column $P_{:k,t}$ is its weighted receptive field; shared routing, recurrence, and decoding define its neural computation. Particle and neuronal-world feature widths are 128 for standalone NWD and 64 for the adapter; hidden-state dimension d_h is fixed throughout a rollout.

Learn the particle receptive fields. A radius graph E_t^P first encodes local interactions into particle features $g_{i,t} \in \mathbb{R}^{d_p}$. For each particle, the router scores q candidate units around their advected centers, including its previous primary unit. Let $s_{ik,t} \in \mathbb{R}$ be the learned particle–unit compatibility score and $\mathcal{C}_i^{(r)}$ the indices of its r highest-scoring candidates. With temperature $\tau > 0$,

$$P_{ik,t} = \begin{cases} \dfrac{\exp(s_{ik,t}/\tau)}{\sum_{l \in \mathcal{C}_i^{(r)}} \exp(s_{il,t}/\tau)}, & k \in \mathcal{C}_i^{(r)}, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The selected configuration uses $q = 8$ and $r = 2$. Support weights induce mass $M_k \in \mathbb{R}_+$, center $c_k \in \mathbb{R}^d$, momentum $p_k \in \mathbb{R}^d$, and angular momentum L_k:

$$\begin{aligned} M_k &= \sum_i P_{ik} m_i, & c_k &= M_k^{-1} \sum_i P_{ik} m_i x_i, \\ p_k &= \sum_i P_{ik} m_i v_i, & L_k &= \sum_i P_{ik} (x_i - c_k) \times m_i v_i. \end{aligned} \tag{4}$$

Time subscripts are suppressed when unambiguous; the cross product is scalar in 2D and vector-valued in 3D. Active neuronal worlds have positive support mass. Write $\omega_k = (M_k, c_k, p_k, L_k) \in \mathbb{R}^{d_\omega}$, where $d_\omega = 1 + 2d + d(d-1)/2$. These moments are computed in FP32 from the particles, rather than predicted ones, and are supplemented by pooled learned features.

Apply shared neural interactions and recurrence. A sparse center graph E_t^W connects neuronal worlds and carries interactions between their particle groups. For one orientation of each neuronal world edge, the model predicts

$$F_{kl} = \alpha_{kl} r_{kl} + \beta_{kl} v_{kl}, \qquad F_{lk} = -F_{kl}, \tag{5}$$

where $F_{kl}, r_{kl}, v_{kl} \in \mathbb{R}^d$, $r_{kl} = c_l - c_k$, and $v_{kl} = p_l/M_l - p_k/M_k$. The scalar coefficients $\alpha_{kl}, \beta_{kl} \in \mathbb{R}$ are outputs of a shared neural interaction rule; any scalar normalization of the relative vectors is absorbed into them. Antisymmetry gives zero total internal neuronal world force, but the velocity-directed term need not give zero torque. Let $h_{k,t}^- \in \mathbb{R}^{d_h}$ denote the incoming memory, after inheritance if an event occurred. A learned embedding $u_{k,t} \in \mathbb{R}^{d_h}$ summarizes current moment descriptors and pooled particle features for the shared gated recurrent unit (GRU) (Cho et al., 2014):

$$h_{k,t+1} = (1 - \rho)\, \mathrm{GRU}_\theta(u_{k,t}, h_{k,t}^-) + \rho\, \mathrm{GRU}_\theta(u_{k,t}, 0), \qquad \rho = 0.5. \tag{6}$$

The first term updates the inherited context; the second evaluates the same current observations without prior memory. Their mixture balances carried context with current evidence when particle membership changes. It does not reconstruct a separate history for every particle or guarantee an exact coarse description. The shared decoder combines local features, neuronal-world interactions, and the refreshed memory to predict particle motion. At fixed feature width, message computation costs $O(|E_t^P| + Nr + |E_t^W|)$ once neighborhoods are available. The implemented candidate search additionally requires $O(NK_t + K_t^2)$ distance comparisons, so sparse messages alone do not establish linear overall complexity.

3.3 REORGANIZING NEURONAL WORLDS WITH COMPACT INHERITANCE

At configured event steps, a discrete scheduler selects legal splits or merges using support residuals, geometry, age, cooldown, and cardinality (Section B). Splits divide support along its leading covariance direction; merges combine the nearest legal pair. Incoming memories follow

$$\text{split: } h^-_{a,t} = h^-_{b,t} = h_{k,t}, \qquad \text{merge: } h^-_{k,t} = \frac{M_a h_{a,t} + M_b h_{b,t}}{M_a + M_b}. \tag{7}$$

Successor moments are remeasured from unchanged particles, and Equation (6) refines inherited context using each successor's observations. New identifiers record parent–successor relations; shared neural parameters make transferred memories immediately usable. Inheritance initializes compact context rather than reconstructing exact support-specific histories. Transferring affected memories costs $O(d_h)$ per event with $O(K_t d_h)$ active storage; moment recomputation separately costs $O(Nr)$ (Section B.5). Discrete event choices receive no rollout-loss gradients; training optimizes the continuous prediction modules.

3.4 PARTICLE-LEVEL OUTPUT INTERFACES

Regrouping leaves the particles unchanged but can change the forces decoded from their representation. We handle two distinct issues: aggregate force–torque constraints restrict the physical output, whereas event attenuation limits the immediate change in the learned update. Let $s_d = d + d(d-1)/2$ be the number of aggregate force–torque constraints per physical system. The standalone and hybrid interfaces differ in their constraint targets, as described below.

Mapping neuronal worlds to particles. For n_e active support edges (i, k), stack raw decoded forces into $\widetilde{f} \in \mathbb{R}^{dn_e}$. The matrix $A_{P_t} \in \mathbb{R}^{s_d \times dn_e}$ sums support-weighted forces and their torques about a reference center. The ideal graph lift is

$$f^\star = \arg\min_f \sum_{(i,k)} P_{ik} \|f_{ik} - \widetilde{f}_{ik}\|^2 \quad \text{subject to} \quad A_{P_t} f = b_t, \tag{8}$$

where $b_t \in \mathbb{R}^{s_d}$ contains the learned neuronal-world field's total force and orbital torque. Particle forces are $f_i = \sum_k P_{ik} f^\star_{ik}$ and accelerations are f_i/m_i. This graph-scoped interface avoids separate constraints within temporary neuronal worlds, but its learned target b_t can change when neuronal worlds change. The implementation uses a damped small linear solve, which introduces a regularization defect even in exact arithmetic; Section A characterizes this defect separately from numerical precision.

Constraint-preserving correction of a graph backbone. In the direct-velocity configuration, the neuronal-world branch decodes mass-normalized particle accelerations. For n moving particles, let $a_A, a_W, a_{\text{ext}} \in \mathbb{R}^{dn}$ denote the backbone, neuronal-world branch, and known external accelerations. The branch projects its learned residual in physical units before adding external forcing and boundary constraints. Specifically, with $M = \text{diag}(m_1 I_d, \ldots, m_n I_d)$ and $B_{X_t} \in \mathbb{R}^{s_d \times dn}$ aggregating particle force and torque, it uses

$$r^\perp = M^{-1}\left[I - B_{X_t}^\mathsf{T}(B_{X_t} B_{X_t}^\mathsf{T})^\dagger B_{X_t}\right] Mr, \qquad B_{X_t} M r^\perp = 0. \tag{9}$$

Here $r \in \mathbb{R}^{dn}$ is the unprojected acceleration residual and $\dagger$ denotes the Moore–Penrose inverse. The zero target depends on particles, not their partition into neuronal worlds. The hybrid then combines the branch output with the backbone:

$$a_{A+W} = \mathcal{B}_{X_t}\left[a_A + \alpha\,(a_W - a_{\text{ext}})\right], \qquad \alpha = \alpha_{\max}\sigma(\ell), \tag{10}$$

where $\mathcal{B}_{X_t}$ applies the shared boundary/acceleration constraints, σ is the logistic sigmoid, and $\ell \in \mathbb{R}$ is learned. The bound $0 < \alpha < \alpha_{\max}$ controls the mixing coefficient, not the norm of an otherwise unrestricted residual. Here a_W denotes the branch output after its configured projection and branch-side constraints; subtracting a_{ext} avoids counting known forcing twice. The backbone and branch are jointly trained; the branch adds 137,354 parameters plus the scalar gate, or 8.6% over GNS-A. The aggregate-preservation statement concerns the projected residual before boundary operators, which can change the aggregate response.

At lifecycle events, γ_t scales the learned update before known forcing and boundary operations; it does not reset recurrent memory. It is zero in the selected standalone N-body configuration, 0.25

in the adapters, and one on steps without events. Thus $\gamma_t = 0$ suppresses the learned event-step correction rather than freezing the physical particles. In the hybrid, this attenuation is already included in the branch output a_W. Reducing the immediate effect of regrouping also omits part of the learned dynamics, giving the trade-off analyzed in Section 4.

3.5 Training

We train on autonomous windows with a multi-step curriculum. The standalone objective is

$$\mathcal{L} = \mathcal{L}_x + 0.1\mathcal{L}_v + 0.1\mathcal{L}_a - 10^{-3}\mathcal{H}(P) + 10^{-4}\,\mathbb{E}_t[K_t], \tag{11}$$

where $\mathcal{L}_x, \mathcal{L}_v, \mathcal{L}_a$ supervise position, velocity, and acceleration over the window, and $\mathcal{H}(P)$ is mean per-particle routing entropy. The cardinality term records active-state cost, but it does not provide a gradient for event selection because K_t is determined by discrete decisions. It should therefore be distinguished from a learned cardinality regularizer. The standalone model has 526,858 parameters. Adapter comparisons share 2,000 updates and task-specific losses across methods: velocity weights are 1 for Dam2D and 2 for Sand2D, with Sand2D contact weight 3. Section B and Table 16 specify state initialization, lifecycle rules, projection tolerances, gate bounds, and optimization.

4 Theory: What an Adaptive Coarse State Must Preserve

Adaptive simulation must distinguish three questions: what information a coarse state retains, which physical aggregates its output preserves, and how regrouping changes its predictions. The following results formalize these questions separately rather than provide a blanket guarantee of improved simulation. Full assumptions and proofs appear in Section A.

Capacity cannot repair information discarded by the coarse state. Let $F : \mathcal{X} \to \mathcal{X}$ be the microscopic transition and $C : \mathcal{X} \to \mathcal{Y}$ a coarse map. An exact memoryless transition G exists precisely when $C(F(X))$ is constant among states sharing $C(X)$ (Theorem A.1). More quantitatively, if $C(X) = C(X')$ but their coarse futures differ by δ, then

$$\max_{Z \in \{X, X'\}} \|G(C(Z)) - C(F(Z))\| \geq \delta/2. \tag{12}$$

The obstruction is informational: a larger decoder acting on the same observation retains the same lower bound. History or refinement can distinguish states that the original coarse observation aliases, although neither is sufficient in general. For NWD, this result motivates augmenting the neuronal-world representation; it is not an error-floor theorem for the full simulator, which also receives particle-level information.

Preserving aggregate physics still leaves room to correct trajectories. For the particle aggregation operator in Equation (9), any constraint-null residual satisfies

$$B_X M(a_A + \alpha r^{\perp}) = B_X M a_A. \tag{13}$$

Yet admissible residuals have dimension $dn - \operatorname{rank}(B_X) \geq dn - s_d$ (Corollary A.12). Thus preserving total force and torque removes at most three degrees of freedom in 2D or six in 3D, while retaining many ways to redistribute particle motion. The same zero target applies after a neuronal world split or merge, unlike a target defined separately inside each neuronal world. This is an interface guarantee before boundary operations, not an energy or trajectory-accuracy guarantee.

A representation event introduces a distinct source of rollout error. At the same particle state, let u_t^- and u_t^+ be learned increments before and after reorganization. Let ζ_t bound $\|u_t^+ - u_t^-\|$, χ_t bound $\|u_t^-\|$, and η_t bound ordinary one-step approximation error. For a Λ-Lipschitz physical transition, the state error ε_t obeys

$$\varepsilon_{t+1} \leq \Lambda\varepsilon_t + \eta_t + \gamma_t\zeta_t + (1 - \gamma_t)\chi_t. \tag{14}$$

Here ε_t is prediction error, distinct from the event label e_t in Equation (1). Memory transfer is intended to reduce the representation change ζ_t, but the bound does not guarantee that it does so. Damping scales the event-difference term while introducing the omitted-update term $(1-\gamma_t)\chi_t$. Thus unchanged particles and preserved aggregates do not imply unchanged predictions; event handoff must be evaluated over autonomous rollouts (Proposition A.14).

5 EXPERIMENTS

5.1 EXPERIMENTAL DESIGN

Benchmarks and protocols. We evaluate four benchmarks spanning long-range gravity, bonded interactions, free-surface fluids, and granular motion. Our DyWorlds-NBody suite uses two three-seed protocols, H3 and C3, with 300 test trajectories per seed; their selected fixed-cardinality baselines are Fixed-K16 and Fixed-K32, respectively. DyWorlds-Bonded uses three seeds and 256 test trajectories per seed. Public Dam2D (Toshev et al., 2023) and Sand2D (Sanchez-Gonzalez et al., 2020) use one hybrid training seed, 25 and 30 test trajectories, FP32 evaluation, and five causal velocity-history inputs. Thus Table 1 contains five evaluation settings, not five distinct benchmarks. Dataset generation, normalization, and the complementary diagnostic studies are described in Sections C, E and F.

Baselines and metrics. GNS-PF is a pair-force graph comparator, and STT-R is a compact recurrent-token baseline, not a reproduction of WorldParticle (Wang et al., 2026). For the public benchmarks, GNS-A is a GNS-style model retrained under the common protocol, and enlarged GNS-A provides an approximately parameter-matched comparison (Sanchez-Gonzalez et al., 2020). UPT and Transolver represent latent evolution and physical slicing (Alkin et al., 2024; Wu et al., 2024); we deploy a particle-adpated version of these two baselines. Fixed-K, memoryless, memory-shuffled, and regenerated variants examine different components of the neuronal-world state. We report endpoint position MSE or per-trajectory-normalized RMSE (NRMSE) after autonomous rollout, with seed-level and conditional trajectory-level analyses kept distinct. Physical diagnostics and runtime are evaluated separately from prediction error.

Table 1: **Cross-domain 100-step position NRMSE.** Lower is better. N-body/Bonded report mean ± standard deviation over three seeds; public-benchmark entries include evaluation dispersion. GNS denotes GNS-PF for N-body/Bonded and GNS-A for Dam2D/Sand2D.

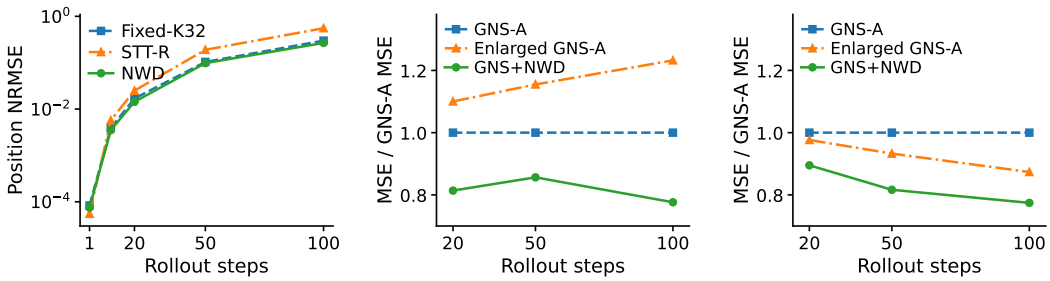
Benchmark	Proposed deployment	Proposed	Fixed-K	GNS	STT-R	UPT	Transolver
N-body (H3)	NWD	**0.259**±0.014	0.303±0.039	0.832±0.345	0.546±0.003	0.295±0.058	0.337±0.062
N-body (C3)	NWD	**0.267**±0.008	0.299±0.031	1.003±0.564	0.554±0.005	0.303±0.069	0.327±0.078
Bonded (H3)	NWD	**0.180**±0.007	0.197±0.025	0.184±0.022	0.214±0.027	0.206±0.046	0.202±0.062
Dam2D (T1)	GNS+NWD	**0.402**±0.012	0.467±0.022	0.457±0.031	0.489±0.014	0.454±0.030	0.433±0.044
Sand2D (T1)	GNS+NWD	**0.679**±0.018	0.732±0.035	0.749±0.026	0.825±0.036	0.704±0.061	0.683±0.078

5.2 NEURONAL-WORLD CORRECTION AT MATCHED PARAMETER COUNT

NWD or GNS+NWD attains the lowest reported mean 100-step NRMSE in all five settings of Table 1. On Dam2D and Sand2D, the jointly trained hybrid has lower mean position MSE than both graph baselines at every reported horizon (Table 2 and Figure 2). Its 100-step reductions are 22.4% and 22.6% relative to GNS-A, and 37.0% and 11.3% relative to enlarged GNS-A, respectively. The enlarged baseline approximately matches the hybrid's parameter count (1.72M versus 1.73M), so this tested widening does not reproduce the hybrid's mean accuracy. All methods receive 2,000 updates, but this comparison neither matches runtime nor characterizes performance after full convergence. These final mean comparisons use one training seed; the conditional paired analyses are reported separately in Section E.

Table 2: **Matched training-budget and capacity comparison.** Endpoint position MSE over 25/30 Dam2D/Sand2D test trajectories, conditional on one trained seed. All methods use 2,000 updates; parameters do not imply equal runtime. Bold marks lowest means, not statistical significance.

Benchmark	Method	Position MSE ↓ 20 steps	50 steps	100 steps	Parameters (M)
Dam2D	GNS	0.00419±0.00032	0.08872±0.018	0.68173±0.20	1.59
	Enlarged GNS	0.00461±0.00026	0.10246±0.020	0.84012±0.25	1.72
	GNS+NWD	**0.00341**±0.00024	**0.07598**±0.016	**0.52919**±0.15	1.73
Sand2D	GNS-A	0.00467±0.00070	0.04652±0.012	0.19892±0.070	1.59
	Enlarged GNS-A	0.00456±0.00068	0.04339±0.011	0.17375±0.060	1.72
	GNS+NWD	**0.00418**±0.00063	**0.03798**±0.001	**0.15404**±0.054	1.73



(a) DyWorlds-NBody. (b) Dam2D. (c) Sand2D.

Figure 2: **Accuracy across autonomous rollout lengths.** (a) Three-seed C3 horizon study (Table 7); NWD improves on STT-R from 10 steps onward. (b,c) Mean MSE relative to GNS-A (Table 2); the hybrid improves on both graph baselines at every horizon.

5.3 Persistent state over long autonomous rollouts

At 100 steps, NWD has the lowest reported mean NRMSE in both final H3/C3 comparisons, with reductions of 10.7–14.5% versus Fixed-K and 51.8–52.6% versus STT-R (Table 1). A five-seed diagnostic extension reports 0.2663 ± 0.0086 for NWD, 0.2989 ± 0.0320 for Fixed-K32, and 0.5540 ± 0.0004 for STT-R, with lower NWD error in every matched seed (Table 10). These results support a long-rollout benefit beyond a single favorable training run; the first three seeds overlap the C3 diagnostic series rather than provide independent additional replication.

5.4 Component ablations and attribution

The largest N-body contrasts concern persistent memory and its association with particle support (Table 3). The complete NWD reduces the 100-step NRMSE by 21.0% relative to the memoryless variant and 14.0% relative to shuffled memories. The latter is a fixed-checkpoint intervention, showing that the fitted predictor uses the correctly associated context rather than merely the same collection of memory vectors. The smaller contrasts with regenerated neuronal worlds (1.9%), world-local lifting (3.0%), and no handoff (2.8%) are not additive, and regeneration performs better at shorter horizons (Table 12). These results support retained, support-associated context, but do not isolate the benefit of memory transfer specifically at split–merge events.

The public-benchmark diagnostics distinguish the trained hybrid from the role of its active correction. On four reused Sand2D trajectories, removing the branch from a fixed hybrid checkpoint increases mean MSE, corresponding to full-model reductions of 6.0%, 10.6%, and 12.1% at 20, 50, and 100 steps. The Dam2D effects are smaller, and removing persistent memory slightly lowers the mean MSE in all six benchmark–horizon cells (Section E.3).

Table 3: **Component contrasts at 100 steps.** C3 diagnostic series: three training seeds and 900 trajectory–seed pairs. We report full NWD's relative NRMSE reduction and Bonferroni intervals for "ablation minus NWD," conditional on the fitted seeds. Shuffled memories is a fixed-checkpoint intervention; contrast families are specified in Section D.5.

Ablation	NWD gain	Paired wins	ΔNRMSE interval
No persistent memory	21.0%	99.7%	[0.06635, 0.07575]
Shuffled memories	14.0%	86.8%	[0.03973, 0.04749]
Regenerated neuronal worlds	1.9%	51.6%	[0.00164, 0.00876]
World-local lift; handoff retained	3.0%	55.4%	[0.00612, 0.01065]
No handoff; graph lift retained	2.8%	56.7%	[0.00533, 0.01018]

5.5 Robustness, physical behavior, and computational cost

Across 256–2,048 particles, NWD has lower NRMSE than all three tested comparators in every 50- and 100-step setting, with 100-step gains over Fixed-K16 increasing from 7.1% to 22.5% (Figure 3).

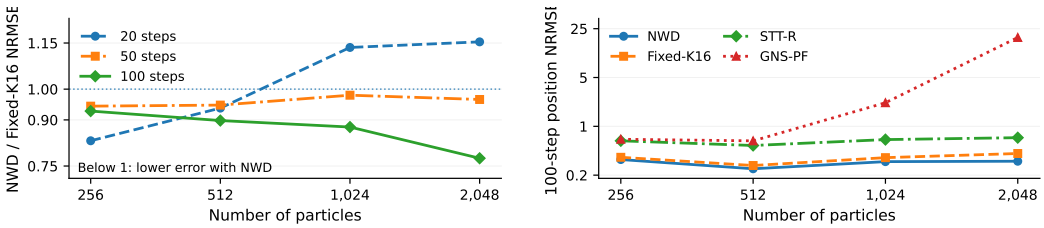
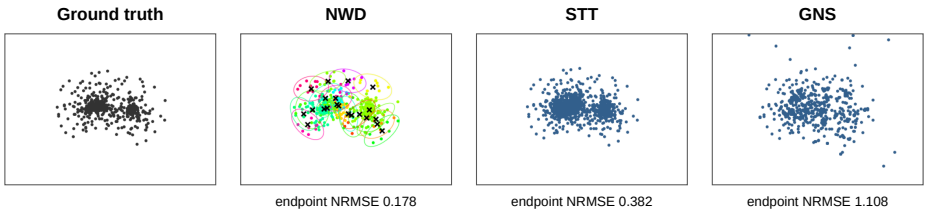
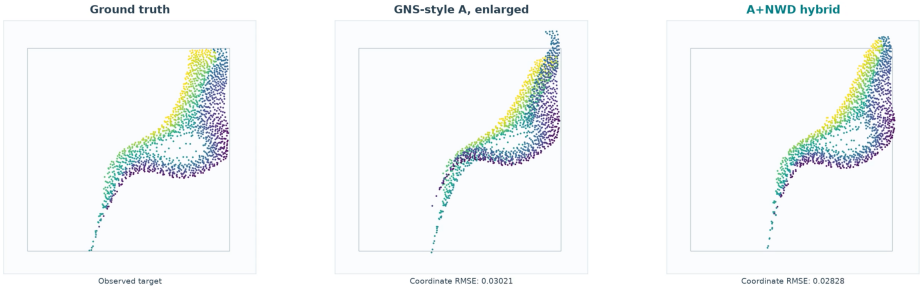


Figure 3: **Long-rollout robustness to particle count.** Two-seed validation: NWD/Fixed-K16 NRMSE (left); 100-step NRMSE (right). NWD wins every 50-/100-step setting; short-horizon rankings vary (Table 13).



(a) N-body: 1,024-particle cluster merger after 100 steps. In NWD, particle colors indicate neuronal-world membership; colored outlines show the worlds' spatial extent.



(b) Sand2D: A+NWD hybrid.

Figure 4: **Particle predictions across physical regimes.** Panel (a) shows x–y projections; STT/GNS denote STT-R/GNS-PF. In (b), A+NWD denotes the graph–NWD hybrid. Annotations report snapshot NRMSE in (a) and coordinate RMSE in (b). See Figure 9 for the full N-body gallery.

Two short-horizon reversals remain. The 4,096- and 8,192-particle examples preserve the ranking but are illustrative stress tests rather than a seed-complete generalization study (Table 15). Physical diagnostics do not share a uniform ranking. The hybrid lowers the logged Dam2D momentum/angular-momentum errors by 62.6%/38.6% at 20 steps and Sand2D maximum penetration by 15.7% at 100 steps. Its 100-step relative kinetic-energy diagnostic is worse than GNS-A's on both benchmarks, although Dam2D target-trajectory kinetic-energy MSE is lower (Tables 23 to 25). For forced or dissipative systems, deviation from an initial invariant and error relative to the target trajectory measure different properties. These observations are consistent with the limited guarantee in Proposition A.13. N-body inference takes 22.50 ms per step versus 20.01 ms for GNS-PF and 15.99 ms for STT-R on an uncontended RTX 4090 (Table 14). Figure 4 provides complementary visual comparisons.

6 Conclusion and Limitations

NWD separates remeasured physical moments from inherited recurrent context so that predictive units can reorganize without replaying trajectories. Long-rollout comparisons support this design, with N-body ablations providing the clearest evidence for memory tied to particle support. The scheduler remains discrete, inherited context may omit relevant history, and aggregate constraints do not guarantee energy fidelity or trajectory accuracy. Public-benchmark results support the hybrid, but not a general memory benefit. Future work should learn event utility, strengthen energy control, and extend multi-seed and controlled-runtime evaluation.

AI USE STATEMENT

Generative AI assisted with the literature comparison, manuscript editing, experimental data checking, code validation, and scientific figure editing. The authors take responsibility for the final claims, code, and reported evidence.

ETHICS STATEMENT

This work uses synthetic or public physical trajectories and does not involve human participants. Prediction error, invariant error, boundary behavior, and runtime are reported separately because improvements in one do not certify safety or physical validity. These learned surrogates require domain-specific verification before safety-critical use.

REPRODUCIBILITY STATEMENT

Sections B, C, E and E.3 specify model configurations, event rules, training, selection, interventions, and statistical procedures; proofs appear in Section A. The corresponding experiment artifact is uploaded in the supplementary material, including the anonymized code, the configuration, and several samples. Architectural and evaluation specifications are given in the appendices.

REFERENCES

Benedikt Alkin, Andreas Fürst, Simon Schmid, Lukas Gruber, Markus Holzleitner, and Johannes Brandstetter. Universal physics transformers: A framework for efficiently scaling neural operators. *Advances in Neural Information Processing Systems*, 37:25152–25194, 2024.

Josh Barnes and Piet Hut. A hierarchical o (n log n) force-calculation algorithm. *nature*, 324(6096): 446–449, 1986.

Lennart Bastian, Samuel Leventhal, Mustafa Hajij, and Tolga Birdal. Topological neural operators. *arXiv preprint arXiv:2606.09806*, 2026.

Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. *Advances in neural information processing systems*, 29, 2016.

Dominik Bauer, Zhenjia Xu, and Shuran Song. Doughnet: A visual predictive model for topological manipulation of deformable objects. In *European Conference on Computer Vision*, pp. 92–108. Springer, 2024.

Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J Bekkers, and Max Welling. Geometric and physical quantities improve e(3) equivariant message passing. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=_xwr8gOBeV1.

Christos Chatzisavvas, Panagiotis Rigas, George Ioannakis, Vassilis Katsouros, and Nikolaos Mitianoudis. When slots compete: Slot merging in object-centric learning. In *Greeks in AI*, 2026. URL https://openreview.net/forum?id=6wDCYsq1jJ.

Kyunghyun Cho, Bart Van Merriënboer, Çağlar Gulçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1724–1734, 2014.

Huayu Deng, Xiangming Zhu, Yunbo Wang, and Xiaokang Yang. EvoMesh: Adaptive physical simulation with hierarchical graph evolutions. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 13327–13346. PMLR, 13–19 Jul 2025.

Ryan D'Orazio, Danilo Vucetic, Zichu Liu, Junhyung Lyle Kim, Ioannis Mitliagkas, and Gauthier Gidel. Solving hidden monotone variational inequalities with surrogate losses. In *International Conference on Learning Representations*, volume 2025, pp. 77721–77749, 2025.

Leslie Greengard and Vladimir Rokhlin. A fast algorithm for particle simulations. *Journal of computational physics*, 73(2):325–348, 1987.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pp. 4651–4664. PMLR, 2021.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pp. 3744–3753. PMLR, 2019.

Zichu Liu and Lacra Pavel. Recursive reasoning in minimax games: A level k gradient play method. *Advances in Neural Information Processing Systems*, 35:16903–16917, 2022.

Karolis Martinkus, Aurelien Lucchi, and Nathanaël Perraudin. Scalable graph networks for particle simulations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 8912–8920, 2021.

Hazime Mori. Transport, collective motion, and brownian motion. *Progress of theoretical physics*, 33(3):423–455, 1965.

Ehsan Pajouheshgar, Hyunsoo Kim, Sabine Süsstrunk, Wenzel Jakob, and Jinah Park. Neural particle automata: Learning self-organizing particle dynamics. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–11, 2026.

Zhengqi Pei and Shuhui Wang. Dynamics-inspired neuromorphic visual representation learning. In *International Conference on Machine Learning*, pp. 27521–27541. PMLR, 2023.

Zhengqi Pei, Anran Zhang, Shuhui Wang, and Qingming Huang. Modeling language tokens as functionals of semantic fields. In *Forty-first International Conference on Machine Learning*, 2024a.

Zhengqi Pei, Anran Zhang, Shuhui Wang, Xiangyang Ji, and Qingming Huang. Data-free neural representation compression with riemannian neural dynamics. In *Forty-first International Conference on Machine Learning*, 2024b.

Zhengqi Pei, Qingming Huang, and Shuhui Wang. Neuronal group communication for efficient neural representation. *arXiv preprint arXiv:2510.16851*, 2025.

Zhengqi Pei, Qingming Huang, and Shuhui Wang. When LLMs develop languages: Symbolic communication for efficient multi-agent reasoning. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=ovpL0ujD6j.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=roNqYL0_XP.

Hanno Rein and S-F Liu. Rebound: an open-source multi-purpose n-body code for collisional dynamics. *Astronomy & Astrophysics*, 537:A128, 2012.

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*, pp. 8459–8468. Pmlr, 2020.

Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pp. 9323–9332. PMLR, 2021.

Omer Rochman Sharabi, Sacha Lewin, and Gilles Louppe. A neural material point method for particle-based emulation. *arXiv preprint arXiv:2408.15753*, 2024.

Vinay Sharma and Olga Fink. A physics-informed graph neural network conserving linear and angular momentum for dynamical systems. *Nature Communications*, 17(1):1045, 2026.

Karn Tiwari, Niladri Dutta, NM Krishnan, et al. Graph mamba operator: A latent simulator for interacting particle systems. *arXiv preprint arXiv:2606.09432*, 2026.

Artur Toshev, Gianluca Galletti, Fabian Fritz, Stefan Adami, and Nikolaus Adams. Lagrangebench: A lagrangian fluid mechanics benchmarking suite. *Advances in Neural Information Processing Systems*, 36:64857–64884, 2023.

Nghia Truong, Cem Yuksel, Chakrit Watcharopas, Joshua A Levine, and Robert M Kirby. Particle merging-and-splitting. *IEEE Transactions on Visualization and Computer Graphics*, 28(12): 4546–4557, 2021.

Caoliwen Wang, Minghao Guo, Siyuan Chen, Heng Zhang, Mengdi Wang, Xingyu Ni, Hanson Sun, Kunyi Wang, Zherong Pan, Kui Wu, et al. Worldparticle: Unified world simulation of lagrangian particle dynamics via transformer. *arXiv preprint arXiv:2605.15305*, 2026.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53681–53705. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/wu24r.html.

Yin Yu, John Harlim, Daning Huang, and Yan Li. Learning coarse-grained dynamics on graph. *Physica D: Nonlinear Phenomena*, 481:134801, 2025.

Yanshun Zhao, Xiaoyu Peng, Jiamin Jiang, Congcong Zhu, and Jingrun Chen. Meshtok: Efficient multi-scale tokenization for scalable PDE transformers. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=I8QGwNBO4T.

Robert Zwanzig. Memory effects in irreversible thermodynamics. *Physical Review*, 124(4):983, 1961.

A FORMAL THEORY

This appendix gives the formal statements behind Section 4, together with their consequences for state representation, projection, and rollout error. Neuronal worlds are predictive coarse units; the analysis does not require them to coincide with uniquely identifiable physical objects.

State spaces and lifecycle maps. With $\mathcal{P}_{N,K}$ the sparse-support space, active states occupy

$$\mathfrak{S}_N = \bigsqcup_{K=K_{\min}}^{K_{\max}} \underbrace{(\mathcal{X}^N \times \mathcal{W}^K \times \mathcal{P}_{N,K})}_{\mathfrak{S}_{N,K}}, \tag{15}$$

$$\Psi_{\text{split}} : \mathfrak{S}_{N,K} \to \mathfrak{S}_{N,K+1}, \qquad \Psi_{\text{merge}} : \mathfrak{S}_{N,K} \to \mathfrak{S}_{N,K-1}. \tag{16}$$

Here $\mathcal{X}$ and $\mathcal{W}$ are the single-particle and single-unit state spaces. The disjoint union expresses changing active state, not an impossibility of implementing it in a padded tensor.

A.1 A FIBER CRITERION FOR DETERMINISTIC COARSE CLOSURE

Let $\mathcal{X}$ be the microscopic state space, $F : \mathcal{X} \to \mathcal{X}$ a deterministic one-step map, and $C : \mathcal{X} \to \mathcal{Y}$ a coarse representation. Write $[X]_C = \{X' \in \mathcal{X} : C(X') = C(X)\}$ for the fiber of C through X.

Theorem A.1 (Deterministic memoryless closure)**.** *There exists a deterministic map $G : C(\mathcal{X}) \to C(\mathcal{X})$ satisfying*

$$G(C(X)) = C(F(X)) \qquad \textit{for every } X \in \mathcal{X} \tag{17}$$

if and only if $C \circ F$ is constant on every fiber of C; equivalently,

$$C(X) = C(X') \quad \Longrightarrow \quad C(F(X)) = C(F(X')). \tag{18}$$

Proof. If G satisfies (17) and $C(X) = C(X') = y$, then $C(F(X)) = G(y) = C(F(X'))$, proving necessity. Conversely, assume (18). For any $y \in C(\mathcal{X})$ choose an arbitrary representative X_y with $C(X_y) = y$ and define $G(y) = C(F(X_y))$. Equation (18) makes this definition independent of the representative, so G is well-defined and satisfies (17). □

Corollary A.2 (Closure obstruction)**.** *If there exist X, X' with $C(X) = C(X')$ but $C(F(X)) \neq C(F(X'))$, no deterministic memoryless coarse transition can predict both coarse futures. Exact deterministic closure must retain information that distinguishes the relevant microscopic states, for example through added coarse variables or sufficient history, or restrict the admissible states. A stochastic closure may instead describe a conditional distribution; it does not determine both distinct microscopic futures exactly.*

Corollary A.3 (Irreducible error of an aliased coarse state)**.** *Suppose $\mathcal{Y}$ is a normed vector space and $C(X) = C(X') = y$, with $\delta = \|C(F(X)) - C(F(X'))\| > 0$. Then every deterministic coarse predictor G satisfies*

$$\max\{\|G(y) - C(F(X))\|, \|G(y) - C(F(X'))\|\} \geq \delta/2. \tag{19}$$

Proof. The triangle inequality bounds δ by the sum of the two prediction errors, so at least one error is at least $\delta/2$. □

The bound holds for every capacity and parameterization of G using the same coarse observation. It can be avoided only by changing the information supplied to the predictor or the admissible state family.

For NWD, exact physical moments form part of C, recurrent neuronal world memory augments the state with history, and a split can enrich the instantaneous representation. The theorem does not say that these choices guarantee exact closure; it states why a fixed memoryless bottleneck can be structurally insufficient.

Table 4: **Related representations and their state transitions.** The comparison separates latent compression, temporal recurrence, spatial adaptation, and particle-supported state transfer. Entries describe the cited formulations rather than all possible extensions.

Method family	Established contribution	Relation to recurrent particle-group state
UPT / latent bottlenecks (Alkin et al., 2024)	Efficient dynamics in a fixed-size latent representation.	Latent rollout carries a compressed state; no particle-group split–merge inheritance is specified.
Transolver / EvoMesh / MeshTok (Wu et al., 2024; Deng et al., 2025; Zhao et al., 2026)	State-conditioned slices, input-adaptive hierarchy, or complexity-adaptive tokenization.	Spatial aggregation adapts, without specifying cross-step recurrent inheritance between reorganized particle groups.
WorldParticle (Wang et al., 2026)	Progressive merging into super-tokens regenerated from the current particle state at every step.	Tokens are re-induced each step; merging within a forward pass differs from recurrent inheritance across physical steps.
GraMO (Tiwari et al., 2026)	Graph-coupled recurrent latent dynamics for long-horizon particle systems.	Memory is indexed by graph nodes; NWD attaches it to an evolving population of supported particle groups.
DoughNet / adaptive slots (Bauer et al., 2024; Chatzisavvas et al., 2026)	Latent-set prediction of object topology, or variable and mergeable slots for scene decomposition.	Topology prediction and scene grouping have different targets from carrying physically grounded recurrent units over unchanged particles.
TNO / HTNO (Bastian et al., 2026)	Physical fields on cell complexes with cross-dimensional and learned coarse-complex operators.	Cell-complex operators relate spatial representations; NWD event maps transfer recurrent state across physical time.
NPA / numerical particle split–merge (Pajouheshgar et al., 2026; Truong et al., 2021)	Self-organizing microscopic agents or adaptive changes to physical particle resolution.	NWD changes a mesoscopic representation above fixed microscopic particles rather than changing the physical discretization itself.
NWD (ours)	Persistent particle-supported neuronal worlds with variable cardinality, exact moments, recurrent memory, and lifecycle lineage.	Physical moments are remeasured, fixed-width memories are inherited, and shared neural updates specialize the successors.

A.2 Separation from stateless tokenization

Let a microscopic history be $H_t = (X_0, \dots, X_t)$, and let $\Gamma(H_t)$ denote a desired mesoscopic state that may contain persistent identity, lineage, or unresolved-mode memory. A stateless tokenizer is any map $T : \mathcal{X} \to \mathcal{Y}$ that constructs its coarse state from the terminal microscopic state alone.

Proposition A.4 (Path-dependence separation)**.** *Suppose there exist histories H_t and H'_t with the same terminal microscopic state $X_t = X'_t$ but different desired mesoscopic states $\Gamma(H_t) \neq \Gamma(H'_t)$. Then no stateless tokenizer $T(X_t)$ can represent Γ on both histories. A recurrent mesoscopic update $W_{t+1} = \Psi(W_t, X_{t+1}, e_t)$ is not subject to this restriction.*

Proof. The equality $X_t = X'_t$ implies $T(X_t) = T(X'_t)$ for every stateless tokenizer, while the desired outputs differ by assumption. Therefore T fails on at least one history. A recurrent update can distinguish the histories through W_t or the event record e_t, so the impossibility does not apply. □

Remark A.5. The proposition does not claim that path dependence is always required to predict the next microscopic state. It isolates a representational capability: persistent lifecycle semantics cannot be reduced to per-step retokenization unless history enters through another state variable. A simulator that retokenizes X_t alone cannot encode the assumed path-dependent target. It can remove this restriction by receiving history through another state variable, including an appropriately sized fixed-dimensional recurrent state. The proposition therefore distinguishes stateless retokenization from carried history, not NWD from every possible recurrent architecture.

A.3 TOPOLOGY-INVARIANT GRAPH-LEVEL LIFTING

For one particle graph, enumerate support edges $e = (i, k)$ with nonnegative weight $w_e = P_{ik}$. Let $f_e \in \mathbb{R}^d$ be an edge force and let c_g be a graph reference point. Define the linear physical aggregation operator

$$A_S f = \begin{bmatrix} \sum_{e=(i,k)} w_e f_e \\ \sum_{e=(i,k)} w_e (x_i - c_g) \times f_e \end{bmatrix}, \tag{20}$$

where the second block is scalar in two dimensions and vector-valued in three dimensions. For prescribed $b = (F_g, \tau_g)$, NWD computes

$$\Pi_{S,b}(\widetilde{f}) = \arg\min_f \|f - \widetilde{f}\|_W^2 \quad \text{s.t.} \quad A_S f = b, \tag{21}$$

for the ideal exact constrained solve; the implementation uses the regularized variant quantified below. Here S denotes the current sparse support and W its diagonal weight matrix.

Proposition A.6 (Partition-independent global constraints)**.** *For any admissible support S for which (21) is feasible,*

$$A_S \Pi_{S,b}(\widetilde{f}) = b. \tag{22}$$

Consequently, if a split or merge replaces S by another admissible support S' but retains the same prescribed graph-level target b, the projected updates before and after the event obey the same total force and torque constraints, irrespective of K_t, neuronal world IDs, or the neuronal world partition.

Proof. The equality is the feasibility constraint defining the optimization problem. A split or merge changes the optimization variables and objective geometry, but not the physical target b. Therefore the two projected solutions may distribute force differently over support edges while sharing exactly the same global aggregate. The statement requires feasibility and an exact constrained solve. A damped solve generally leaves a residual and must not be identified with exact satisfaction of the equality constraint. □

Corollary A.7 (Topology-neutral representational transition)**.** *Let Ψ_e be any split or merge transition that maps an admissible support S to another admissible support S' while leaving the graph-level target b unchanged. Then the prescribed global force–torque aggregate is identical before and after the transition:*

$$A_S \Pi_{S,b}(\widetilde{f}) = b \qquad \text{and} \qquad A_{S'} \Pi_{S',b}(\widetilde{f}') = b. \tag{23}$$

Thus representational topology can change without changing the prescribed global force and torque, even though the distribution of force over particle–unit support edges may change.

Corollary A.8 (Closed internal system)**.** *If $F_g = 0$ and $\tau_g = 0$, the graph-level learned residual has zero net force and zero net torque at every step, including immediately before and after a lifecycle event. With no external impulse, the learned residual therefore preserves total linear and angular momentum up to numerical integration and floating-point error.*

The proposition is intentionally not an energy-conservation theorem. Dissipative tasks and the chosen integrator need not conserve energy; corresponding experiments report energy drift or decay error rather than asserting exact preservation.

Remark A.9 (Graph scope and target independence are different). The implemented graph lift aggregates learned neuronal world forces and their orbital torques to form b_t, with a reference center computed from the neuronal world centers. Consequently, the aggregate target itself can change with the neuronal world state. Proposition A.6 guarantees equality to a specified target, and its topology-neutral corollary requires that the target actually remain the same across the compared representations. The additional force-aware particle projection in the adapter uses the fixed zero-residual target and is the stronger instance of this separation. Neither construction certifies exact physical trajectories.

Proposition A.10 (Defect of the regularized lift)**.** *Let $G = AW^{-1}A^{\mathsf{T}}$ and $r = A\widetilde{f} - b$. For $\lambda > 0$, the implemented regularized correction*

$$f_\lambda = \widetilde{f} - W^{-1}A^{\mathsf{T}}(G + \lambda I)^{-1} r \tag{24}$$

has aggregate defect

$$A f_\lambda - b = \lambda (G + \lambda I)^{-1} r. \tag{25}$$

Thus exact constraint satisfaction is not implied by positive damping.

Proof. Substituting the correction gives $r - G(G + \lambda I)^{-1} r$. The identity $I - G(G + \lambda I)^{-1} = \lambda(G + \lambda I)^{-1}$ yields the result. □

A.4 CONSTRAINT-PRESERVING RESIDUAL COMPOSITION

Let $M = \mathrm{diag}(m_1 I_d, \ldots, m_n I_d)$ for the moving particles and let B_X map particle forces to their total force and torque about a fixed particle-state reference point. For any unprojected acceleration correction r, define

$$r^{\perp} = M^{-1} \left[I - B_X^{\mathsf{T}} (B_X B_X^{\mathsf{T}})^{\dagger} B_X \right] M r. \tag{26}$$

The pseudoinverse is the Moore–Penrose inverse for the exact-arithmetic statement.

Proposition A.11 (Baseline-relative aggregate preservation)**.** *For any baseline acceleration a_A and scalar α,*

$$B_X M(a_A + \alpha r^{\perp}) = B_X M a_A. \tag{27}$$

The statement is independent of the neuronal world partition that generated r. If a numerical implementation leaves $\delta = B_X M r^{\perp}$, its aggregate change is exactly $\alpha\delta$.

Proof. $B_X B_X^{\mathsf{T}} (B_X B_X^{\mathsf{T}})^{\dagger}$ projects onto the range of B_X, which contains $B_X M r$. Hence $B_X M r^{\perp} = 0$. Linearity gives both identities. □

The tested implementation applies a tolerance-truncated pseudoinverse to the physical-unit residual over the moving-particle mask. Its subsequent kinematic and boundary operations are outside this linear identity and can change the aggregate response. The theorem therefore preserves a baseline's aggregate before those operations, not necessarily the final boundary-conditioned output or the true physical target. A zero configured positive-work cap disables the optional work limiter in the implementation; it does not enforce zero work.

Corollary A.12 (Dimension of the admissible correction space)**.** *For n positive-mass particles in $d \in \{2, 3\}$ dimensions, define $\mathcal{R}_X = \{r \in \mathbb{R}^{dn} : B_X M r = 0\}$. With $s_d = d + d(d-1)/2$,*

$$\dim \mathcal{R}_X = dn - \mathrm{rank}(B_X) \geq dn - s_d. \tag{28}$$

If the force–torque constraints are independent, equality holds in the lower bound.

Proof. Positive masses make M invertible, hence $\mathrm{rank}(B_X M) = \mathrm{rank}(B_X)$. Rank–nullity gives the equality, and B_X has at most s_d independent rows. □

Thus preserving aggregate force and torque need not eliminate local or mesoscopic corrections. The statement is independent of the neuronal world partition and concerns the particle-level residual before boundary operations.

Proposition A.13 (Force–torque neutrality does not control energy)**.** *Under a velocity kick $v' = v_A + \Delta t\, \alpha r$, the kinetic-energy change relative to the baseline velocity v_A is*

$$E(v') - E(v_A) = \Delta t\, \alpha v_A^{\mathsf{T}} M r + \tfrac{1}{2} (\Delta t\, \alpha)^2 r^{\mathsf{T}} M r. \tag{29}$$

Even when $B_X M r = 0$, this expression can be positive.

Proof. Expand $\frac{1}{2}(v_A + \Delta t\, \alpha r)^{\mathsf{T}} M (v_A + \Delta t\, \alpha r)$ and subtract $\frac{1}{2} v_A^{\mathsf{T}} M v_A$. For a counterexample, take two equal masses at opposite positions on one axis, opposite outward velocities, and equal-and-opposite outward accelerations. The residual has zero total force and zero torque, but both its work term and its squared-norm term are positive. □

A.5 ROLLOUT ERROR WITH REPRESENTATION SHOCK

Let F_t be the true one-step map, Q_t the known predictor, and u_t^- and u_t^+ the learned state increments before and after a representational event at the same predicted particle state. The gated prediction

is $\widehat{x}_{t+1} = Q_t(\widehat{x}_t) + \gamma_t u_t^+(\widehat{x}_t)$. Assume F_t is Λ-Lipschitz on visited states and define nonnegative bounds

$$\|F_t(\widehat{x}_t) - Q_t(\widehat{x}_t) - u_t^-(\widehat{x}_t)\| \le \eta_t, \tag{30}$$

$$\|u_t^+(\widehat{x}_t) - u_t^-(\widehat{x}_t)\| \le \zeta_t, \tag{31}$$

$$\|u_t^-(\widehat{x}_t)\| \le \chi_t. \tag{32}$$

Without an event, take $u_t^+ = u_t^-$ and $\gamma_t = 1$. The increments can include an integrator's Lipschitz response to acceleration so that all terms are in the same state norm.

Proposition A.14 (Event attenuation trades shock against omission)**.** *With $e_t = \|x_t - \widehat{x}_t\|$ and $0 \le \gamma_t \le 1$,*

$$e_{t+H} \le \Lambda^H e_t + \sum_{s=0}^{H-1} \Lambda^{H-1-s} \left[\eta_{t+s} + \gamma_{t+s}\zeta_{t+s} + (1-\gamma_{t+s})\chi_{t+s}\right]. \tag{33}$$

Setting $\gamma_t = 0$ removes the instantaneous event-difference term but retains the omitted-update term χ_t.

Proof. The identity

$$\gamma_t u_t^+ - u_t^- = \gamma_t(u_t^+ - u_t^-) - (1-\gamma_t)u_t^- \tag{34}$$

and the triangle inequality give $e_{t+1} \le \Lambda e_t + \eta_t + \gamma_t\zeta_t + (1-\gamma_t)\chi_t$. Repeated substitution yields (33). □

This result does not guarantee that a handoff improves accuracy. It separates event-induced representation error from ordinary approximation error and explains the role of the matched no-handoff comparison.

Prediction stability versus equilibrium convergence. Surrogate-loss methods establish convergence for hidden monotone variational inequalities under interpolation and sufficiently accurate surrogate optimization (D'Orazio et al., 2025), while level-k gradient play uses recursive opponent prediction to stabilize minimax learning (Liu & Pavel, 2022). These analyses concern equilibrium-seeking updates; Proposition A.14 instead controls deviation from a moving physical trajectory without assuming hidden monotonicity or a game structure. Its finite-horizon error bound does not imply asymptotic convergence or uniformly bounded error as the horizon grows.

A.6 Adaptive allocation under an aggregate neuronal world budget

Let $\epsilon_t(k)$ be the minimum attainable prediction loss at step t within a fixed hypothesis class when k neuronal worlds are active. For integer schedules $k_t \ge 1$, define

$$\mathcal{E}_{\text{adapt}}(B) = \min_{\sum_{t=1}^T k_t \le B} \sum_{t=1}^T \epsilon_t(k_t). \tag{35}$$

Proposition A.15 (Weak dominance of adaptive schedules)**.** *If $B = Tk$ for an integer k, then*

$$\mathcal{E}_{\text{adapt}}(B) \le \sum_{t=1}^T \epsilon_t(k). \tag{36}$$

The inequality is strict whenever there is a feasible reallocation of neuronal worlds across time that strictly lowers the summed loss.

Proof. The uniform schedule $k_t = k$ is feasible for the adaptive optimization and is therefore an upper bound on its minimum. Strictness follows whenever another feasible schedule has lower objective value. □

This proposition is a budget-allocation statement, not a guarantee that a learned scheduler discovers the optimum. The experiments compare realized schedules with fixed-K baselines; similar cardinality need not imply equal measured runtime.

B Implementation Details

B.1 Exact selected configuration

Table 5: Selected N-body configuration. The model configuration fixes both active-state limits and event frequency.

Setting	Value	Setting	Value
Particle/unit feature width	128	Initial neuronal worlds K_0	16
Minimum/maximum neuronal worlds	8 / 96	Lifecycle target	20
Candidate neuronal worlds q	8	Sparse support r	2
Neuronal world neighbors	10	Event interval	4
Minimum age/cooldown	8 / 8	Conservative scope	graph
Event residual scale	0	Mass-normalized forces	true
Parameters	526,858	Training updates	2,000

The implementation is PyTorch-based and uses BF16 autocast during training. Routing normalization, moment induction, invariant calculations, and the small conservative-projection solves remain in FP32. Standalone latency measurements disable BF16 and use uncontended FP32 inference after warm-up. Dam2D/Sand2D evaluation also uses FP32; its shared-resource timing is reported separately.

B.2 Canonical one-step operator

One NWD step follows the operator order below.

1. Encode the current particles and local radius graph, using normalized velocity/history features when configured.
2. Initialize or advect neuronal world centers and memory proposals, preserving the previous primary support as a routing candidate.
3. Score $q = 8$ candidates, retain top-$r = 2$ support, and re-induce physical moments and closure descriptors.
4. At the event interval, apply a legal split or merge and re-induce moments before predicting the neuronal world interaction.
5. Predict antisymmetric neuronal world forces and update refreshed recurrent memory from moment descriptors and current support features.
6. Decode local and neuronal world forces, divide by particle masses, and apply the configured event attenuation.
7. Convert normalized output to physical units if configured, apply the force-aware residual projection if enabled, add known forcing, and apply shared boundary constraints and integration.
8. Increment neuronal world age, decrease cooldown, and return particles, carried neuronal world state, and event diagnostics.

The hybrid separately evaluates its graph backbone, subtracts known acceleration from the NWD output, adds the bounded residual to the backbone acceleration, and applies the common final boundary/integration interface. It does not integrate the backbone and branch particle trajectories independently and average the resulting positions.

B.3 Descriptor-conditioned lifecycle

The scheduler uses descriptors of the current neuronal worlds, with discrete event selection and no counterfactual event-utility loss. Closure descriptors enter split/merge scoring modules followed by discrete argmax/threshold decisions; gradients do not pass through the chosen event. Event timing is therefore descriptor-conditioned rather than learned by differentiating an event-utility objective. The continuous router, particle and neuronal-world dynamics, recurrent memory, and decoder are optimized end-to-end through autonomous rollout losses.

Starting from 16 neuronal worlds, the target-cardinality trust region permits natural splits until the count exceeds 20 and then admits the first legal merge before another unconstrained split. Legal splits

require minimum age, zero cooldown, and at least twice the minimum support. The split plane is the principal eigenvector of the support covariance. Merge proposals use the nearest legal pair. Each new neuronal world receives a new integer ID; split children record one parent and merged neuronal worlds record both parents.

B.4 Moment induction and event transformations

Support weights are coalesced after every lifecycle event and normalized per particle. Exact moments are then recomputed from the microscopic particle state rather than predicted by a neural network. For a split $k \to \{a, b\}$,

$$M_k = M_a + M_b, \qquad p_k = p_a + p_b, \tag{37}$$

and angular momentum is reconciled by

$$L_k = L_a + L_b + (c_a - c_k) \times p_a + (c_b - c_k) \times p_b. \tag{38}$$

A merge uses the reverse aggregation and mass-weighted memory combination. Event memories are intentionally conservative: split children inherit the parent memory, while the next recurrent update differentiates them through their distinct supports.

B.5 Shared neural coordinates and inheritance cost

Every neuronal world uses the same routing, recurrent, and decoding parameters. The unit-specific quantities are its support, moments, hidden state, identity, age, and cooldown. Splitting therefore allocates state for two applications of an existing neural module; it does not create a new set of learned weights. The fixed-width GRU state summarizes history without retaining the sequence of past inputs. Its representation is learned through the autonomous prediction loss, while Equation (7) defines how that representation is initialized after an event.

For an algebraic view, stack the pre-event memories as $H_t \in \mathbb{R}^{K_t \times d_h}$. The event can be represented as $H_t^- = T_{e_t} H_t$, with

$$T_{e_t} \in \mathbb{R}^{K_t' \times K_t}, \qquad T_{e_t} \geq 0, \qquad T_{e_t} \mathbf{1} = \mathbf{1}. \tag{39}$$

Unchanged units have identity rows; split children have the same one-parent row; a merged unit has a two-parent row weighted by the parent masses. This is a mathematical description of local copies and averaging, not a dense matrix required by the implementation. For E disjoint two-parent/two-child events, the changed-vector arithmetic is $O(Ed_h)$; copying an entire state tensor, if used, additionally costs $O(K_t d_h)$. The active hidden-state storage is $O(K_t d_h)$. Moment re-induction costs $O(Nr)$ at fixed physical descriptor dimension, and a dense width-d_h GRU update costs $O(K_t d_h^2)$ per physical step. Candidate search and message computation have the separate costs stated in Section 3.2. These bounds exclude training activation storage and do not claim that the full simulator runs in linear time.

Each successor starts from a convex combination of its parent memories, hence

$$\max_k \|h_{k,t}^-\| \leq \max_j \|h_{j,t}\| \tag{40}$$

for any vector norm. Inheritance alone therefore does not amplify the largest individual hidden-state norm, although a split increases total stored state and subsequent neural updates can change its norm. Shared initialization does not force children to remain identical: their different support features and physical moments enter the next recurrent update. Nor does mass-weighted averaging recover a separate history for every particle; it defines a bounded-size contextual summary whose predictive utility is evaluated by the memory and identity ablations.

Neuronal dynamics and communication. Dynamics-inspired visual learning represents neural weights through interactions between dynamical sub-models (Pei & Wang, 2023), while RieM parameterizes neuronal interactions with Riemannian metrics for data-free compression (Pei et al., 2024b). Neuronal Group Communication uses low-dimensional neuronal states and group communication to represent neural computation (Pei et al., 2025). NWD applies a state-based perspective to particle simulation: its recurrent units track changing particle groups rather than replacing the weights of a pretrained network. Related language representations are discussed in Section B.5.

Related representation interfaces. LasF represents language tokens as functionals of semantic fields (Pei et al., 2024a), while CLSR evolves and routes compact symbolic protocols among LLM agents (Pei et al., 2026). Here, communication and inheritance operate on numerical recurrent states associated with physical particles, rather than on language tokens or symbolic protocols.

B.6 Conservative projection

Let $\widetilde{f}_e$ be raw edge forces and A_S the force/torque aggregation matrix in (20). The damped projection is implemented as

$$f^\star = \widetilde{f} - W^{-1}A_S^\mathsf{T}\left(A_S W^{-1} A_S^\mathsf{T} + \lambda I\right)^{-1}(A_S\widetilde{f} - b), \tag{41}$$

with $\lambda = 10^{-5}$ and graph-batched 3×3 (2D) or 6×6 (3D) systems. Debug validators check support normalization, finite moments, non-reused IDs, and post-projection aggregate force/torque.

B.7 Training

The loss is (11). Position, velocity, and acceleration are supervised over autonomous windows with rollout schedule R2: horizon 1 initially, then samples from $\{1,2\}$, $\{2,4\}$, and finally $\{4,8\}$. A ground-truth context burn-in of at most 16 steps initializes recurrent state before the loss window. AdamW uses a base learning rate of 2×10^{-4}, gradient clipping at 1, and checkpoint selection on validation rollouts. Adapter hyperparameters are specified in Table 16.

C Data, Baselines, and Evaluation Protocol

C.1 DyWorlds-NBody

DyWorlds-NBody is our controlled 3D gravitational suite, with cluster merger, tidal split, collapse/capture, and non-merging flyby. Learned N-body prediction has precedents in Interaction Networks, hierarchical particle simulators, and EGNN (Battaglia et al., 2016; Martinkus et al., 2021; Satorras et al., 2021); our trajectories are generated separately rather than taken from those datasets. Principal train/validation/test manifests contain 3,000/300/300 trajectories and particle counts in $\{256, 512, 1024, 2048\}$. Reference trajectories use tree-based gravity with leapfrog integration, standard choices in numerical N-body simulation (Barnes & Hut, 1986; Rein & Liu, 2012). The opening angle is 0.25 and softening is 0.03; SHA-256 identifies the generator and selected manifests. Each trajectory stores 512 physical steps. The test matrix evaluates every model at $L \in \{1, 10, 20, 50, 100\}$ and treats missing or nonfinite cells as failures.

The eight-case visualization manifest is independent of model error. It contains six fixed test trajectories and two OOD trajectories: cluster merger, tidal split, collapse/capture, and flyby cases at 256–1,024 particles, plus 4,096-particle cluster merger and 8,192-particle tidal split. Every manifest entry is rendered and tabulated in Section D.

C.2 DyWorlds-Bonded

The Bonded suite models repeated fusion–fission trajectories with breakable and reformable local interactions. Frozen manifests contain 2,000/256/256 train/validation/test trajectories. In addition to position and velocity error, evaluation records bond precision/recall, connected-component agreement, event timing, and invariant drift. The fixed-K and adaptive variants share the same local radius, integrator, rollout curriculum, and prediction target in the final comparison.

C.3 LagrangeBench fluids and material datasets

Dam2D is the two-dimensional free-surface dam-break case from LagrangeBench (Toshev et al., 2023). The adapter evaluation uses all 25 test trajectories; it is distinct from the five-seed validation of the standalone NWD configuration. Sand2D is a granular-particle benchmark from the Learning-to-Simulate suite (Sanchez-Gonzalez et al., 2020). The Sand2D hybrid evaluation uses all 30 test trajectories in two spatial dimensions. Adapter models, splits, and statistics are specified in Section E.

C.4 Comparator identities

Standalone baselines. The standalone GNS-style comparator is a pair-force graph baseline, denoted GNS-PF; its results are not attributed to the published GNS reference architecture. The recurrent super-token-style baseline, denoted STT-R, is a compact matched implementation and not a reproduction of the full WorldParticle model. Its record supports a comparison against that implementation, not a published WorldParticle performance claim (Wang et al., 2026). Fixed-K retains the corresponding NWD encoder, routing, moments, memory, and training interface but disables lifecycle adaptation. The local-lift comparator changes the lifting and handoff interfaces together; it is a compound architectural ablation rather than a one-factor intervention.

Protocol-matched graph backbone. The GNS-A comparator follows the GNS-style encode–process–decode architecture with five velocity-history inputs, particle-type embeddings, boundary features, and normalized acceleration outputs (Sanchez-Gonzalez et al., 2020). All adapter-comparison methods are retrained for the same 2,000-update budget. These are common-protocol retrainings rather than official published benchmark scores. GNS-A-large raises hidden width from 128 to 133 and supplies a capacity baseline within 0.7% of the hybrid parameter count.

Latent and physical-slice baselines. We compare particle-adapted implementations of UPT and Transolver, retaining their respective computational mechanisms rather than substituting a shared generic latent model. UPT retains its point-cloud encoder, fixed-size latent propagator, and query decoder, together with inverse-encoding and inverse-decoding consistency objectives for latent rollout (Alkin et al., 2024). Transolver retains learned physical slicing, attention between slice tokens, and redistribution to particle features (Wu et al., 2024). For autonomous position prediction, task-specific readouts predict accelerations at the current predicted particle locations and use the benchmark's common forcing, boundary, and integration interface. Future reference positions are never supplied as decoder queries. Both methods use the same benchmark-specific causal inputs, data splits, and endpoint metrics as NWD, with normalization statistics estimated from training data. The fixed-budget comparison uses 2,000 optimizer updates per model and validation-based configuration and checkpoint selection; it does not assume equal runtime or convergence under this budget. The run manifest records the implementation revision, architecture, parameter count, batch exposure, optimizer settings, selected checkpoint, training seeds, and evaluation records for each table entry. These are task-specific retrainings, not scores copied from the original papers. SEGNN, GraMO, and NeuralMPM provide additional architectural context for equivariant messages, recurrent graph-state dynamics, and particle–grid emulation (Brandstetter et al., 2022; Tiwari et al., 2026; Sharabi et al., 2024).

C.5 Metric definitions and aggregation

Let $\mathcal{M}$ be the benchmark's moving-particle loss mask, d the coordinate dimension, and δ_i the minimum-image displacement between predicted and target positions when boundaries are periodic. At an endpoint of L autonomous steps,

$$\mathrm{MSE}_L = \frac{1}{d|\mathcal{M}|}\sum_{i\in\mathcal{M}}\|\delta_i\|^2, \qquad \mathrm{NRMSE}_L = \frac{\sqrt{\mathrm{MSE}_L}}{\max\{\sqrt{(d|\mathcal{M}|)^{-1}\sum_{i\in\mathcal{M}}\|x_i\|^2}, 10^{-8}\}}. \tag{42}$$

The target normalization is computed per trajectory before averaging, so aggregate MSE and aggregate NRMSE cannot generally be converted using a single common denominator. The adapter primary metric is the endpoint field `position_mse`, not the within-rollout average field `position_mse_mean`. Velocity NRMSE uses the analogous velocity normalization. The reported relative invariant error compares an invariant against its initial reference; kinetic-energy MSE is a different, target-trajectory diagnostic and must not be substituted for it. For forced or dissipative systems, conservation relative to an initial value is not by itself a complete physical-accuracy criterion. An omitted contact, energy, or distribution diagnostic is not interpreted as zero error.

For multiseed tables, means are averaged within each seed and then across seeds; $\pm$ denotes the sample standard deviation of seed means. Dispersion for a one-training-seed evaluation must be distinguished from variation across independently trained seeds. Paired intervals in the confirmation and adapter studies resample trajectories conditional on the trained seeds; they do not estimate

training-seed variability. Standalone latency uses synchronized uncontended FP32 inference; adapter latency uses shared-resource logs. No missing or nonfinite result is silently discarded from a reported cohort.

D Additional N-Body Results

Evaluation-series scope. Tables 1 and 2 report the final benchmark comparisons. The horizon-resolved, paired-trajectory, and physical N-body analyses below retain the earlier diagnostic series, with their own reported aggregates and uncertainty estimates. H3 and C3 denote protocol families rather than unique checkpoints or evaluation records. The diagnostic tables are not higher-precision expansions of Table 1; their paired intervals apply only to the corresponding diagnostic comparisons. The five-seed extension, particle-count study, and fixed-checkpoint interventions are interpreted within their separately stated scopes, not pooled as independent replications of the final comparison.

D.1 Horizon-resolved diagnostic series

Table 6: H3 diagnostic series: autonomous position NRMSE, retained separately from the final comparison in Table 1. Means and standard deviations are computed across three independently trained seeds; each seed contains 300 held-out trajectories. All reported endpoints are finite. Lower is better.

Rollout steps	NWD	Fixed-K16	STT-R	GNS-PF	Local lift
1	$7.50\times10^{-5}\pm2.17\times10^{-5}$	$5.36\times10^{-5}\pm2.47\times10^{-5}$	$5.50\times10^{-5}\pm2.39\times10^{-8}$	$2.15\times10^{-3}\pm1.00\times10^{-3}$	$5.99\times10^{-5}\pm2.87\times10^{-5}$
10	0.003399 ± 0.000671	0.003145 ± 0.000447	0.005707 ± 0.000001	0.166862 ± 0.208385	0.003358 ± 0.000535
20	0.013788 ± 0.002152	0.013785 ± 0.001758	0.024619 ± 0.000006	0.442884 ± 0.573310	0.014761 ± 0.001584
50	$\mathbf{0.097439\pm0.006275}$	0.103256 ± 0.014670	0.191511 ± 0.000045	1.511625 ± 1.887012	0.111708 ± 0.014233
100	$\mathbf{0.259807\pm0.004025}$	0.303317 ± 0.038764	0.545973 ± 0.000337	3.723887 ± 4.283767	0.301405 ± 0.045774

Prediction errors accumulate differently across rollout lengths. Fixed-K16 and STT-R have lower one-step position error than NWD, and Fixed-K16 remains slightly better in the aggregate after 10 and 20 steps. NWD overtakes Fixed-K16 after 50 steps and expands the reduction to 14.3% after 100 steps. Against STT-R and GNS-PF, the improvement becomes positive by 10 steps and grows with rollout depth. The full horizon matrix distinguishes one-step fit from sustained autonomous prediction.

D.2 Paired trajectory evidence

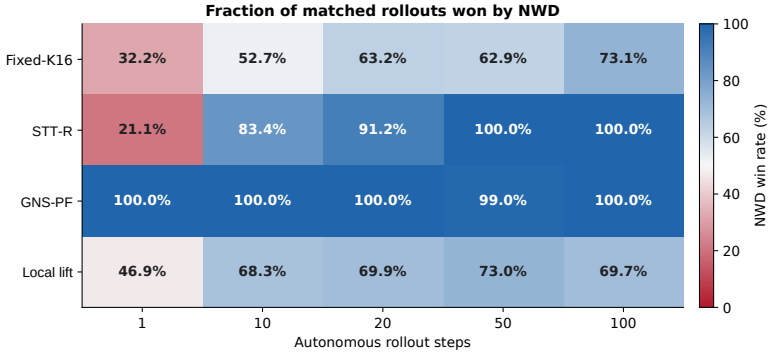


Figure 5: **Full-test paired win rates.** Fraction of the 900 matched trajectory–seed pairs for which NWD has lower position NRMSE than each comparator at every registered rollout length. All pairs are included. After 100 steps, NWD wins 73.1%/100%/100%/69.7% of pairs against Fixed-K16/STT-R/GNS-PF/local lift.

Paired statistics expose the corresponding trajectory-level behavior. For one-step prediction, NWD wins only 32.2% of pairs against Fixed-K16 and 21.1% against STT-R. Against Fixed-K16, its win rate rises to 52.7% at 10 steps, 63.2% at 20 steps, 62.9% after 50 steps, and 73.1% at 100

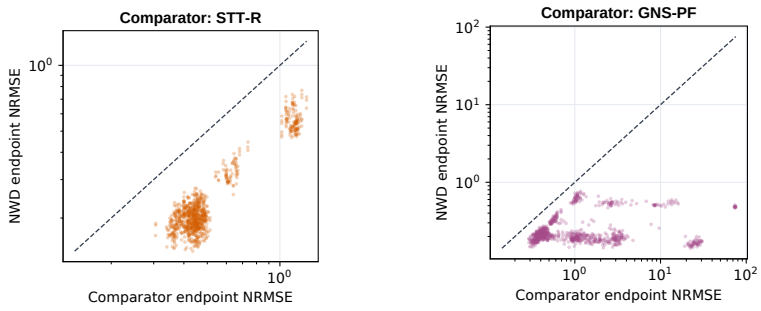
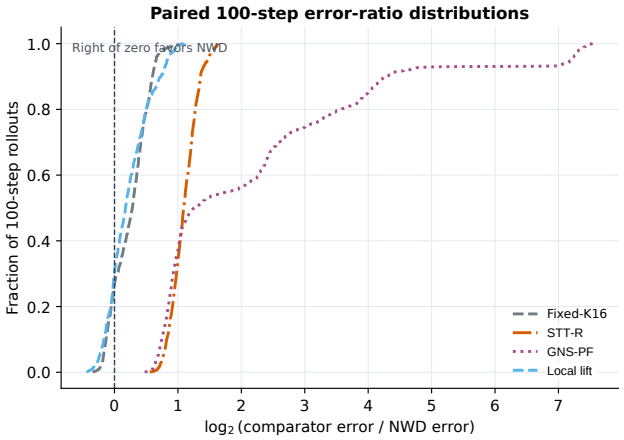
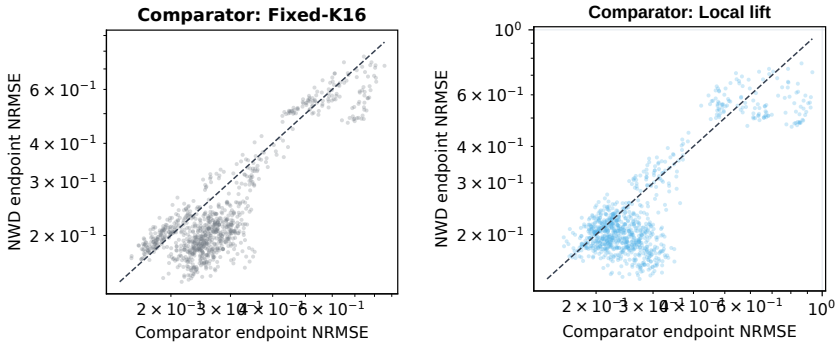


Figure 6: **Gains across individual trajectories.** All 900 pairs per baseline in the complementary 100-step N-body cohort; log-scale axes show every point below the equal-error diagonal. Figure 7 gives the remaining baselines and error-ratio distributions.



(a) All four comparator-to-NWD error-ratio distributions.



(b) Fixed-cardinality baseline.

(c) Compound local-lift ablation.

Figure 7: **Paired distributions for the complete N-body cohort.** Every panel includes all 900 trajectory–seed pairs at 100 steps. Positive log-ratios in (a) and points below the diagonal in (b,c) favor NWD. The corresponding STT-R and GNS-PF scatterplots appear in Figure 6. Local lift changes the lifting and handoff interfaces jointly.

steps. Although the 20-step aggregate mean is almost tied, the median paired reduction is positive (15.1%), indicating a small number of difficult trajectories that affect the mean. Seed-wise variation is substantial for some baselines, so the paper reports both seed-level uncertainty and full paired distributions.

D.3 C3 HORIZON AND PAIRED DIAGNOSTICS

The C3 diagnostic series contains three independently trained seeds and 300 trajectories per seed. Its horizon, paired, and physical summaries are retained together from the earlier analysis, separately from the final comparison in Table 1. One teacher-forced/autonomous one-step agreement failure occurs in the topology-local ablation; full NWD, Fixed-K32, STT-R, and GNS-PF pass these checks. Physical metrics are reported alongside position error. H3 and C3 have separate fixed-cardinality baseline selection and are analyzed independently.

Table 7: C3 diagnostic series over five autonomous rollout lengths, retained separately from the final comparison in Table 1. Values are mean ± sample standard deviation across three training seeds, with 300 trajectories per seed. Lower is better.

Rollout steps	Full NWD	Fixed-K32	STT-R	GNS-PF
1	0.000075 ± 0.000022	0.000083 ± 0.000026	$\mathbf{(5.4965 \pm 0.0024) \times 10^{-5}}$	0.002268 ± 0.001101
10	$\mathbf{0.003506 \pm 0.000688}$	0.003924 ± 0.001142	0.005727 ± 0.000001	0.173360 ± 0.214641
20	$\mathbf{0.014345 \pm 0.002154}$	0.016902 ± 0.005091	0.024902 ± 0.000005	0.456655 ± 0.586577
50	$\mathbf{0.097902 \pm 0.006477}$	0.103702 ± 0.011521	0.189621 ± 0.000037	1.550443 ± 1.915089
100	$\mathbf{0.267071 \pm 0.003550}$	0.298743 ± 0.031308	0.554186 ± 0.000291	4.003323 ± 4.564016

The confirmation retains the short-rollout ranking reversal: STT-R has lower one-step error, whereas NWD has lower error at longer lengths. From 10 through 100 steps, NWD is lower than every direct matched comparator, and its reductions at 100 steps are 10.6% versus Fixed-K32, 51.8% versus STT-R, and 93.3% versus GNS-PF.

Table 8: Paired 100-step statistics for the C3 diagnostic series in Table 7, not for the final comparison in Table 1. The simultaneous interval is a Bonferroni interval for the paired difference "comparator minus NWD," so a positive interval favors NWD. The 10,000-resample procedure resamples trajectories within the fixed training seeds; these intervals do not include uncertainty over retraining.

Comparator	Relative reduction	NWD paired wins	Simultaneous interval
Fixed-K32	10.6%	71.3%	[0.02789, 0.03555]
STT-R	51.8%	100.0%	[0.27898, 0.29580]
GNS-PF	93.3%	100.0%	[3.10943, 4.42921]

NWD improves all three listed diagnostics relative to Fixed-K32 but has higher kinetic-energy error than STT-R and higher linear-momentum error than GNS-PF. The position ranking therefore does not establish a universal ordering of physical accuracy. The reported trajectory records are finite; the physical statistics retain their original reference normalization.

D.4 EXTENSION TO FIVE MATCHED TRAINING SEEDS

The N-body position analysis additionally evaluates NWD, Fixed-K32, and STT-R over five matched training seeds, each with 300 trajectories at 100 steps. Seeds 0–2 overlap the C3 diagnostic cohort; seeds 3–4 extend the training-seed coverage rather than constitute an independent second cohort. All three methods are compared on the same five seeds. The GNS-PF comparison remains restricted to its common three-seed cohort.

NWD reduces the five-seed mean by 10.9% relative to Fixed-K32 and 51.9% relative to STT-R. Its lower error in all five matched seeds complements the trajectory-paired intervals, which condition on the originally selected three seeds and do not quantify uncertainty over retraining. No confidence interval or new significance test is inferred from these five-seed summaries.

Table 9: C3 mean 100-step physical diagnostics, common training seeds 0–2. These invariant errors accompany the position confirmation; they do not show universal physical superiority.

Model	Linear momentum	Angular momentum	Kinetic energy
NWD	0.001107	0.029107	26.8066
Fixed-K32	0.001989	0.082756	29.1950
STT-R	1.857305	0.098173	0.0334
GNS-PF	0.000024	1.375763	3897.2323

Table 10: **N-body robustness across five matched training seeds.** Each seed entry is its mean 100-step position NRMSE over 300 trajectories. The last column gives the mean and sample standard deviation across the five seed means. NWD has the lowest error in every column.

Method	Seed 0	Seed 1	Seed 2	Seed 3	Seed 4	Mean $\pm$ SD
NWD	0.2633	0.2704	0.2676	0.2767	0.2535	0.2663 ± 0.0086
Fixed-K32	0.2667	0.3002	0.3293	0.3318	0.2665	0.2989 ± 0.0320
STT-R	0.5543	0.5538	0.5544	0.5540	0.5534	0.5540 ± 0.0004

D.5 State-semantics ablations

Hidden-state/support association intervention. We test whether a trained model uses the association between the carried context and particle support by permuting the incoming memories while retaining their numerical values. For each of the three fitted NWD seeds, the intervention uses the same checkpoint, causal initialization, and test trajectories as its unmodified reference, without retraining. At every autonomous step, after routing and any split–merge inheritance but before the recurrent update, draw a permutation π_t of the K'_t active units within each physical system and set

$$\widetilde{h}^{-}_{k,t} = h^{-}_{\pi_t(k),t}.$$

Current supports, moments, identifiers, lineage, age, and cooldown are not permuted in this intervention. Thus, the operation preserves the multiset of incoming memory vectors while disrupting their association with current particle groups. Memory is not exchanged between different systems in a batch. The unmodified context initialization is shared, and subsequent particles, routing, and events evolve autonomously rather than being forced to match the reference rollout. The permutation generator and seed are recorded, and permutation randomness is not presented as additional training-seed replication. The intervention measures reliance on correctly associated context in a fitted model; it does not establish that an independently retrained memoryless architecture must perform worse.

The C3 component ablations share the same evaluation protocol; the main contrasts change one component at a time. These comparisons isolate carried memory, identity continuity, regeneration, and the output interface. The lifting contrast retains the handoff, and the handoff contrast retains graph lifting, so neither requires interpreting a joint interface change as a single-component effect. The memory, identity, and regeneration intervals belong to the state-semantics contrast family; the lifting and handoff intervals belong to the factorial interface family. Within each family, Bonferroni intervals target 95% simultaneous coverage using 10,000 paired-trajectory bootstrap resamples within each fixed training seed, followed by averaging over seeds. These intervals condition on the fitted models, do not resample training seeds, and do not provide simultaneous coverage across the two families jointly.

Removing memory and shuffling identities produce the largest unflagged error increases. Regenerated neuronal worlds and removal of the handoff have smaller mean penalties, with conditional paired intervals above zero. The topology-local result has a positive 100-step contrast but a one-step output mismatch and cannot alone certify a clean causal effect of topology separation. The reported mismatch is one of 10,800 teacher/autonomous one-step pairs, at topology-local seed 0, with absolute difference 1.47×10^{-5}. This exception limits the causal interpretation of the topology-local comparison.

Table 11: C3 mechanism ablations after 100 autonomous steps. The reduction is the relative decrease in position NRMSE achieved by full NWD, and the interval is for "ablation minus full NWD." All comparisons contain 900 paired endpoints. † marks the ablation with one one-step agreement mismatch, which limits its use for direct mechanism attribution.

Ablated state semantic	Ablation NRMSE	NWD reduction	NWD paired wins	Simultaneous interval
No persistent memory	0.338026	21.0%	99.7%	[0.06635, 0.07575]
Shuffled neuronal-world memories	0.310442	14.0%	86.8%	[0.03973, 0.04749]
Topology-local lifecycle†	0.305282	12.5%	69.7%	[0.03257, 0.04402]
World-local lift, handoff retained	0.275457	3.0%	55.4%	[0.00612, 0.01065]
Graph lift without handoff	0.274793	2.8%	56.7%	[0.00533, 0.01018]
Regenerated neuronal worlds	0.272205	1.9%	51.6%	[0.00164, 0.00876]

Table 12: **Horizon-resolved component study.** Mean position NRMSE in the same three-seed C3 diagnostic cohort. Short-horizon reversals are retained; the primary component comparison in Table 3 uses the 100-step endpoint. † marks the compound ablation with the one-step agreement mismatch described above.

Variant	1 step	10 steps	20 steps	50 steps	100 steps
Full NWD	0.000075	0.003506	0.014345	0.097902	0.267071
No persistent memory	0.000079	0.003788	0.015799	0.112224	0.338026
Shuffled memories	0.000087	0.003835	0.015472	0.105140	0.310442
Regenerated neuronal worlds	0.000051	0.002860	0.012336	0.089174	0.272205
World-local lift, handoff retained	0.000079	0.003469	0.014033	0.098897	0.275457
Graph lift, no handoff	0.000052	0.003050	0.013437	0.096710	0.274793
World-local lift, no handoff†	0.000060	0.003393	0.015054	0.109857	0.305282

D.6 Scale stratification

The scale-stratified comparison measures robustness across both particle count and rollout length. NWD does not uniformly dominate a fixed neuronal world count at short horizons, but its 50- and 100-step advantage persists over an $8\times$ particle-count range. The matched GNS-PF baseline degrades sharply at 2,048 particles in this long-range setting. A limited local communication budget is a plausible explanation, but the present comparison does not isolate that cause.

D.7 Measured efficiency

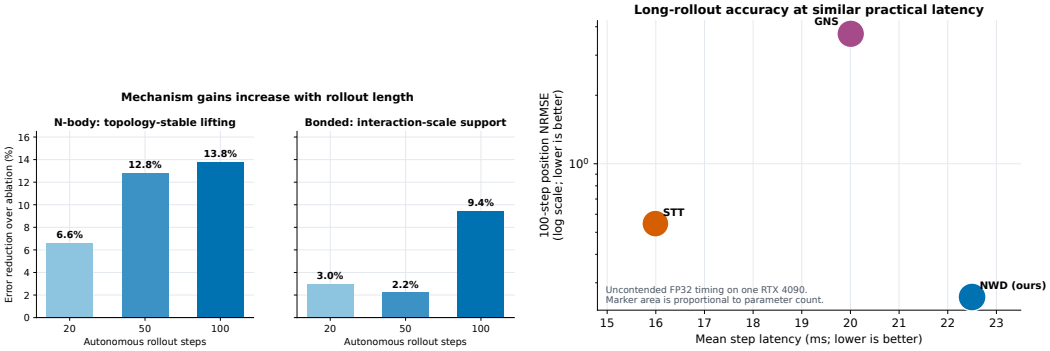
NWD is not presented as the fastest model. Its step latency remains in the same tens-of-milliseconds regime as the matched GNS and STT baselines, while persistent neuronal world state, sparse-support bookkeeping, lifecycle operations, and graph-level projection require additional memory. The

Table 13: Scale-stratified N-body position NRMSE on validation replication seeds 1–2. Every STT/GNS comparison is favorable to NWD. Fixed-K16 is retained as better after 20 steps for 1,024 and 2,048 particles, while NWD is better in every 50- and 100-step cell.

N	Rollout steps	NWD	Fixed-K16	STT	GNS
256	20	**0.01482**	0.01781	0.03175	0.03753
	50	**0.13788**	0.14595	0.24046	0.24930
	100	**0.33457**	0.36025	0.62026	0.65297
512	20	**0.00928**	0.00989	0.02146	0.05045
	50	**0.06652**	0.07016	0.16076	0.18433
	100	**0.24678**	0.27485	0.53166	0.61940
1024	20	0.01960	**0.01726**	0.02991	0.10415
	50	**0.12549**	0.12801	0.23620	0.58324
	100	**0.31229**	0.35619	0.64592	2.17657
2048	20	0.02581	**0.02237**	0.03233	1.46089
	50	**0.15884**	0.16436	0.27262	7.29888
	100	**0.31643**	0.40817	0.68786	18.70478

Table 14: Uncontended FP32 inference on the same RTX 4090 profiling device. Latency uses 50 warm-up and 500 timed repetitions in each of two runs. Memory is peak allocated GPU memory.

Family	Model	Mean (ms)	p50 (ms)	p95 (ms)	Peak MiB
N-body	NWD	22.499	21.774	25.449	52.369
	GNS	20.006	19.753	21.731	29.252
	STT	**15.992**	15.788	17.017	23.011
Bonded	NWD	21.532	21.332	23.026	44.307
	GNS	19.617	19.398	20.813	41.614
	STT	**16.014**	15.768	17.095	27.086



(a) Gain over topology-local lifting.

(b) 100-step accuracy–latency frontier.

Figure 8: **Mechanism and measured efficiency on the N-body suite.** Graph-level lifting with the event-step handoff improves over the local-lift comparator by 6.6%, 12.8%, and 13.8% after 20, 50, and 100 predicted steps, while NWD and the matched graph/token baselines remain in the same tens-of-milliseconds latency regime.

contribution is therefore a long-horizon accuracy/stability gain at comparable inference latency rather than a new throughput record; exact profiling values are reported in the table.

D.8 Qualitative gallery and particle-count stress tests

Across the eight fixed trajectories, NWD is more accurate than all three baselines in every case. Reductions versus Fixed-K16 range from 1.0% to 31.5%, versus STT from 42.2% to 60.7%, and versus GNS from 37.6% to 99.9%. The gallery is used to inspect geometry and failure modes, while the principal claims remain based on the complete multiseed test matrix.

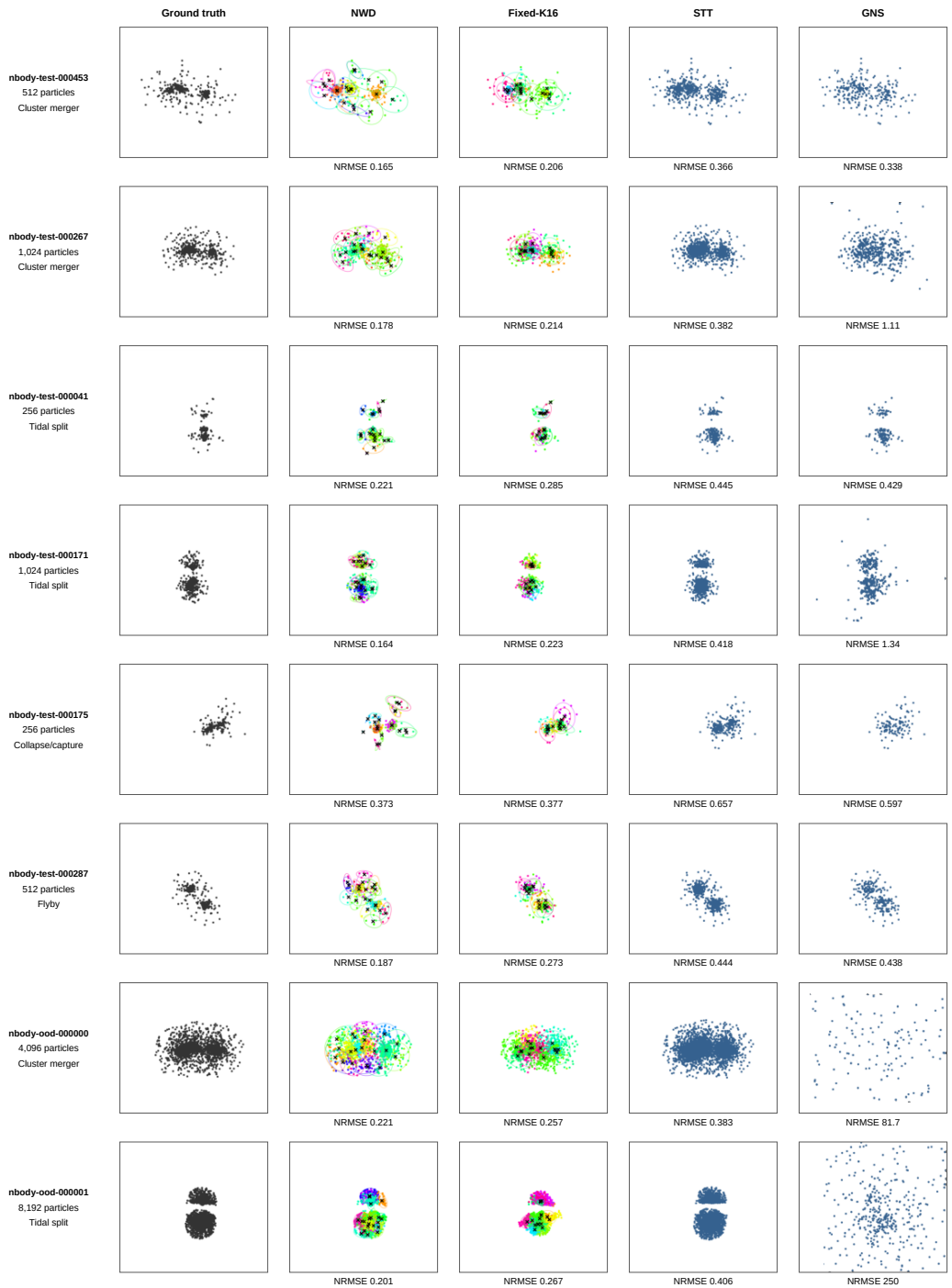


Figure 9: **Complete 100-step visualization gallery.** All six test and two OOD entries in the preregistered manifest are displayed; no trajectory was removed after evaluation. Columns compare ground truth, NWD, Fixed-K16, STT-R (STT), and GNS-PF (GNS). All panels show x–y projections; their frames delimit the views, not confining physical boundaries.

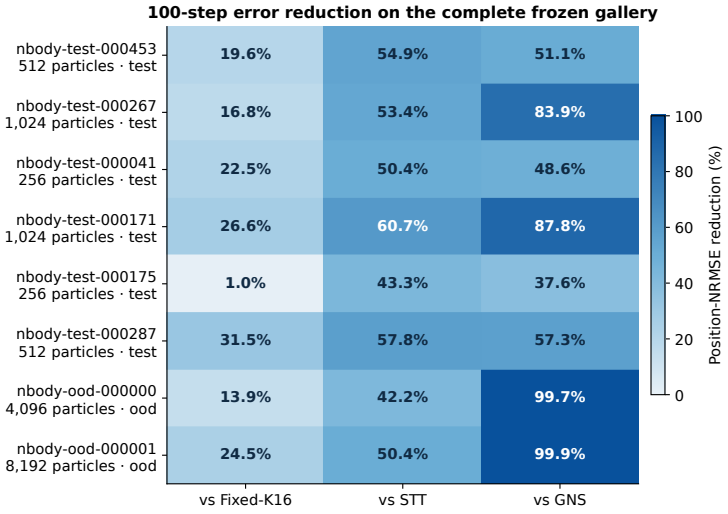


Figure 10: NWD 100-step error reduction on each fixed visualization trajectory. All 24 trajectory–comparator cells are positive.

Table 15: 100-step position NRMSE on the two fixed particle-count OOD trajectories. These cases are illustrative stress tests rather than replacements for the seed-complete principal test.

N	Scenario	NWD	Fixed-K16	STT	GNS
4096	Cluster merger	**0.2210**	0.2568	0.3827	81.7048
8192	Tidal split	**0.2014**	0.2668	0.4058	250.4061

E FLUID AND GRANULAR RESIDUAL-ADAPTER BENCHMARKS

E.1 COHORT, SELECTION, AND ARCHITECTURE

The Dam2D and Sand2D benchmarks compare four methods per benchmark: GNS-A, its enlarged graph baseline, the standalone NWD branch, and GNS+NWD. All eight jobs use training seed 0, FP32 evaluation, five causal velocity-history inputs, and prediction start step 5. The cohort contains 25 Dam2D and 30 Sand2D trajectories at three separately initialized rollout lengths, giving 300 and 360 endpoint rows, respectively. All 660 endpoint records are complete and finite. Checkpoint selection uses eight development trajectories and the registered 100-step score at the 1,000- and 2,000-update checkpoints; test trajectories are not used for checkpoint selection. This evaluation examines the trained hybrid independently of the standalone validation studies.

The hybrid and standalone GNS-A share backbone initialization and training budget, but the hybrid jointly updates both backbone and neuronal-world branch parameters. Consequently, this experiment tests a trained composite model, not a frozen pretrained graph model with a post-hoc correction. The enlarged graph baseline is important because added parameters alone need not explain the hybrid's mean advantage. This width-matched comparison does not characterize the full graph-model capacity frontier.

Table 16: Exact adapter configurations; settings are common across matched methods unless explicitly branch-specific.

Setting	Dam2D	Sand2D
GNS hidden width / processing steps	128 / 10	128 / 10
Enlarged GNS hidden width	133	133
Graph neighbor radius	0.029	0.015
NWD branch width	64	64
Branch initial / target neuronal worlds	8 / 8	16 / 20
Branch minimum / maximum neuronal worlds	8 / 32	8 / 32
Support candidates / retained support	8 / 2	8 / 2
Event interval / minimum age / cooldown	4 / 8 / 8	4 / 8 / 8
Event residual multiplier	0.25	0.25
Residual scale initialization / maximum	0.01 / 0.25	0.00125 / 0.03125
Learning rate / training-update budget	3×10^{-4} / 2,000	3×10^{-4} / 2,000
Velocity loss weight	1	2
Contact penalty weight	0	3

The branch uses direct lifting with mass-normalized forces, persistent memory, descriptor-conditioned lifecycle, and force-aware projection in physical units. Per-unit conservative lifting is disabled in these adapter configurations. The projection's configured relative pseudoinverse tolerance is 10^{-5}; the positive-work cap is zero, which disables that optional limiter. For Sand2D, all four models share the registered contact penalty and velocity weight 2; these are not NWD-only advantages. The task-specific settings in Table 16 define these comparisons. GNS-A contains 1,591,826 parameters, GNS-A-large 1,717,841, the standalone neuronal world branch 137,354, and the hybrid 1,729,181. The hybrid therefore adds 8.63% parameters to GNS-A and 0.66% relative to GNS-A-large.

E.2 ENDPOINT ERRORS AND COMPLEMENTARY EFFECTS

The graph-backbone and hybrid position-MSE comparisons are reported in Table 2 at all three lengths. The hybrid's mean improvement over GNS-A is 18.5%, 14.4%, and 22.4% for Dam2D and 10.4%, 18.3%, and 22.6% for Sand2D, at 20, 50, and 100 steps. At 100 steps, improvement over GNS-A-large is 37.0% and 11.3%, respectively. The standalone neuronal world branch is worse than GNS-A at 100 steps on both datasets. Thus the fluid/granular result supports complementary mesoscopic correction of a local backbone, not standalone superiority. The fixed-checkpoint intervention below isolates the trained branch's inference-time contribution. The comparison measures the trained composite; it does not isolate every mechanism responsible for joint-training gains.

E.3 FIXED-CHECKPOINT CONTRIBUTION OF THE NEURONAL WORLD BRANCH

We evaluate inference-time interventions on four fixed trajectories per benchmark, selected at equally spaced indices after lexically sorting the original test IDs. The Dam2D suffixes are 00000, 00008, 00016, and 00024; the Sand2D suffixes are 000000, 000009, 000019, and 000029. The panel was specified after the earlier test outcomes were available and is therefore a reused-test sensitivity diagnostic. Each benchmark uses one seed-0 hybrid checkpoint selected within the same 2,000-update training budget: update 1,000 for Dam2D and update 2,000 for Sand2D, as recorded in the endpoint metadata. All interventions retain the learned weights, FP32 precision, prediction start step 5, and matched initial states and targets; no new training is performed. The 25-trajectory Dam2D and 30-trajectory Sand2D tests remain distinct from this four-trajectory panel.

For branch removal, the original correction in Equation (10) is omitted while the jointly trained graph backbone and outer physical constraint are retained:

$$a_{\text{off}} = \mathcal{B}_{X_t}[a_A(X_t)]. \tag{43}$$

Consequently, this intervention differs from the independently trained GNS-A baseline. It isolates the correction's contribution to a given trained hybrid, rather than the effect of learning a different architecture.

Table 17: **Inference-time contribution of the trained mesoscopic correction.** Mean position MSE on four fixed, previously evaluated test trajectories per benchmark; one training seed and unchanged hybrid weights. Branch off removes the NWD correction while retaining the jointly trained backbone and the outer physical constraint. Reduction is $100(\bar{e}_{\text{off}} - \bar{e}_{\text{full}})/\bar{e}_{\text{off}}$; W/T/L counts trajectories on which full has lower/equal/higher MSE. These are fixed-checkpoint sensitivity measurements, not retrained baselines or an independent confirmation.

Benchmark	Steps	Full MSE	Branch-off MSE	Reduction (%)	W/T/L
Dam2D	20	0.00341651	0.00343053	0.4086	4/0/0
Dam2D	50	0.07599461	0.07645862	0.6069	4/0/0
Dam2D	100	0.52938788	0.52948721	0.0188	3/0/1
Sand2D	20	0.01023681	0.01088738	5.9755	3/0/1
Sand2D	50	0.09023653	0.10091240	10.5793	3/0/1
Sand2D	100	0.34168988	0.38859538	12.0705	2/0/2

The full model lowers mean position MSE in all six benchmark–horizon comparisons in Table 17. The Sand2D reductions increase from 5.98% at 20 steps to 10.58% at 50 steps and 12.07% at 100 steps. The longer-horizon mean differences are concentrated in trajectory 000009, with 3/4 and 2/4 paired wins at 50 and 100 steps; the mean advantage is not a uniform trajectory-level effect. Dam2D exhibits smaller reductions of 0.41%, 0.61%, and 0.02%. Retaining carried neuronal world context also lowers Dam2D mean MSE relative to regenerating neuronal worlds by 0.39%, 0.74%, and 0.73%, with four paired wins at each horizon. The latter intervention changes carried context jointly, so it does not isolate recurrent memory from support and state continuity.

Repeating the full hybrid on the same IDs produces maximum absolute per-trajectory MSE changes of 6.97×10^{-6} on Dam2D and 1.13×10^{-5} on Sand2D despite matched targets and unchanged learned weights. These repeat-run differences are not a statistical noise bound. No confidence intervals, significance tests, or independent-training-seed claims are attached to this panel. Its branch-removal evidence concerns prediction at fixed weights, separately from physical fidelity, lifecycle attribution, and the benefit of a retrained memory mechanism.

Separating the branch from its memory state. Table 18 also removes recurrent memory or regenerates neuronal worlds while keeping the learned weights unchanged. Memory-off slightly lowers mean position MSE in every cell; regeneration is slightly unfavorable on Dam2D and mixed on Sand2D. These small effects, together with repeat-run variation, do not establish a persistent-memory advantage in this fixed-checkpoint fluid/granular panel. They differ from the trained N-body component comparisons in Table 3.

The Sand2D branch improves velocity NRMSE by 8.9%, 12.5%, and 7.2% across the three horizons, alongside its increasing mean position-MSE reduction. Dam2D velocity effects are smaller, 0.2–0.4%.

Table 18: **Complete fixed-checkpoint state interventions.** Mean position MSE on four reused test trajectories per benchmark, conditional on one trained hybrid. All arms share learned weights and all six benchmark–horizon cells are retained. Memory off disables persistent recurrence; regeneration removes carried neuronal-world context; branch off removes the correction.

Benchmark	Steps	Full	Memory off	Regenerated	Branch off
Dam2D	20	0.00341651	0.00341184	0.00342990	0.00343053
Dam2D	50	0.07599461	0.07593453	0.07655832	0.07645862
Dam2D	100	0.52938788	0.52859485	0.53327346	0.52948721
Sand2D	20	0.01023681	0.01023662	0.01023676	0.01088738
Sand2D	50	0.09023653	0.09023623	0.09023796	0.10091240
Sand2D	100	0.34168988	0.34167512	0.34168314	0.38859538

Table 19: **The trained branch improves both position and velocity.** Same fixed-checkpoint cohort as Table 18. The last two columns give full-hybrid relative reductions versus branch removal in position MSE and velocity NRMSE, respectively; they are distinct metrics. All six mean contrasts favor the full hybrid, without implying uniform per-trajectory gains or multi-seed significance.

		Velocity NRMSE		Full-hybrid reduction	
Benchmark	Steps	Full	Branch off	Position MSE	Velocity NRMSE
Dam2D	20	0.750877	0.753043	0.41%	0.29%
Dam2D	50	0.783859	0.785481	0.61%	0.21%
Dam2D	100	0.743578	0.746321	0.02%	0.37%
Sand2D	20	1.077031	1.181714	5.98%	8.86%
Sand2D	50	1.955222	2.234117	10.58%	12.48%
Sand2D	100	1.826345	1.969013	12.07%	7.25%

This supports a contribution to predicted motion at fixed weights rather than interpreting the branch as a position-only adjustment.

E.4 Primary and secondary statistical families

The following statistics retain the earlier one-seed paired-analysis series. They are not recomputed from the rounded means or dispersion entries in Table 2, and are not used to assign significance to unmatched final-table entries. The reported primary family consists of two paired endpoint contrasts: hybrid versus GNS-A on Dam2D at 20 steps and Sand2D at 50 steps. Intervals are percentile bootstrap intervals from 20,000 paired-trajectory resamples. Two-sided paired sign-flip tests use 100,000 Monte Carlo draws with the plus-one correction, PCG64 seed 20260906, and lexicographically ordered trajectory IDs. Holm adjustment is applied to the two primary p-values as one family and to all 36 secondary pair/length contrasts as a separate family. Reported intervals are marginal 95% intervals, not simultaneous familywise intervals. They condition on one trained model seed and do not estimate retraining variability.

Table 20: Primary hybrid-versus-GNS-A contrasts in the earlier one-seed paired-analysis series; positive differences favor the hybrid. Intervals and tests condition on the fitted models and evaluated trajectories.

Endpoint	Mean reduction	Paired 95% interval	Wins	Primary Holm p
Dam2D, 20 steps	0.0007760	$[0.0007754, 0.0007766]$	25/25	2.00×10^{-5}
Sand2D, 50 steps	0.0085299	$[0.0032620, 0.0142457]$	19/30	0.00405

The Sand2D 50-step contrast has secondary-family Holm $p = 0.0648$ when assessed inside the larger 36-comparison family. These values answer different multiplicity questions and are not interchangeable. The hybrid-versus-enlarged Sand2D contrast at 50 steps has secondary Holm $p = 0.2849$ despite a favorable mean. Accordingly, the main text describes improvement in mean error across lengths and reserves a significance statement for the declared primary tests.

In this earlier paired-analysis series, the 20-step Dam2D GNS-A values range from 0.004188 to 0.004196 and the hybrid values from 0.003411 to 0.003419. These ranges describe that series, not the dispersion entries in the final comparison of Table 2. The 25 target trajectories have distinct hashes, but distinct hashes do not establish broad initial-condition diversity or independent population sampling. The narrow conditional interval therefore does not establish population-wide or training-seed robustness.

E.5 Physical diagnostics and runtime

The physical matrix separates linear momentum, angular momentum, relative kinetic-energy invariant error, and target kinetic-energy MSE. The latter two metrics answer different questions and can rank a model differently. All models have logged mass error zero in these fixed-mass benchmarks. Dam2D penetration and Sand2D target kinetic-energy MSE are not part of the reported diagnostic matrix. The logs report 100-step completion for all trajectories; this is a finite-rollout observation rather than a validated long-term stability guarantee.

At 20 steps on Dam2D, the hybrid lowers the logged momentum and angular-momentum errors relative to GNS-A by 62.6% and 38.6%. At 100 steps, however, its relative kinetic-energy error is 10.3342 versus 3.5496, while target kinetic-energy MSE is lower, 0.2993 versus 0.3172. On Sand2D at 100 steps, momentum and angular-momentum errors decrease, but relative kinetic-energy error is 0.8267 versus 0.7434. The same Sand2D endpoint lowers maximum penetration by 15.7% and mean outside fraction by 7.7%. These mixed physical changes are consistent with Proposition A.13 and limit any claim of uniformly improved physical fidelity.

Logged hybrid latency is about 3.26 times GNS-A on Dam2D and 4.82 times GNS-A on Sand2D at 100 steps. These shared-resource measurements do not establish a controlled runtime ratio, but they also provide no evidence for a speedup. A small parameter increment therefore must not be described as a small computational increment. The reported accuracy comparison is matched in optimization budget and approximately in parameter count, rather than in measured execution time.

Table 21: Dam2D secondary position-MSE contrasts at all rollout lengths. Δ is reference minus candidate; a positive value favors the candidate. A denotes GNS-A, A+width the enlarged graph baseline, W the standalone neuronal world branch, and A+NWD the hybrid. Holm adjustment includes all 36 secondary contrasts across both benchmarks.

Steps	Reference	Candidate	Δ MSE	Marginal 95% interval	Holm p
20	A	A+width	-0.000425	$[-0.000425, -0.000425]$	0.00036
20	A	A+NWD	0.000776	$[0.000775, 0.000777]$	0.00036
20	A	W	-0.010628	$[-0.010632, -0.010623]$	0.00036
20	A+width	A+NWD	0.001201	$[0.001200, 0.001202]$	0.00036
20	A+width	W	-0.010203	$[-0.010208, -0.010198]$	0.00036
20	W	A+NWD	0.011404	$[0.011400, 0.011408]$	0.00036
50	A	A+width	-0.013739	$[-0.013743, -0.013734]$	0.00036
50	A	A+NWD	0.012736	$[0.012713, 0.012755]$	0.00036
50	A	W	-0.496190	$[-0.496454, -0.495920]$	0.00036
50	A+width	A+NWD	0.026475	$[0.026451, 0.026495]$	0.00036
50	A+width	W	-0.482452	$[-0.482715, -0.482182]$	0.00036
50	W	A+NWD	0.508927	$[0.508650, 0.509196]$	0.00036
100	A	A+width	-0.158387	$[-0.158444, -0.158327]$	0.00036
100	A	A+NWD	0.152533	$[0.152375, 0.152678]$	0.00036
100	A	W	-0.088076	$[-0.089064, -0.087113]$	0.00036
100	A+width	A+NWD	0.310920	$[0.310746, 0.311091]$	0.00036
100	A+width	W	0.070311	$[0.069319, 0.071265]$	0.00036
100	W	A+NWD	0.240609	$[0.239655, 0.241597]$	0.00036

Table 22: Sand2D secondary position-MSE contrasts at all rollout lengths. Δ is reference minus candidate; a positive value favors the candidate. A denotes GNS-A, A+width the enlarged graph baseline, W the standalone neuronal world branch, and A+NWD the hybrid. Holm adjustment includes all 36 secondary contrasts across both benchmarks.

Steps	Reference	Candidate	Δ MSE	Marginal 95% interval	Holm p
20	A	A+width	0.000105	$[0.000006, 0.000216]$	0.58778
20	A	A+NWD	0.000484	$[0.000123, 0.000890]$	0.27209
20	A	W	0.000476	$[-0.000225, 0.001381]$	1.00000
20	A+width	A+NWD	0.000379	$[0.000079, 0.000715]$	0.31636
20	A+width	W	0.000372	$[-0.000352, 0.001255]$	1.00000
20	W	A+NWD	0.000007	$[-0.000716, 0.000713]$	1.00000
50	A	A+width	0.003130	$[0.001400, 0.005141]$	0.00697
50	A	A+NWD	0.008530	$[0.003262, 0.014246]$	0.06480
50	A	W	0.002470	$[-0.006493, 0.013338]$	1.00000
50	A+width	A+NWD	0.005399	$[0.001250, 0.010056]$	0.28488
50	A+width	W	-0.000661	$[-0.010426, 0.010140]$	1.00000
50	W	A+NWD	0.006060	$[-0.003658, 0.016425]$	1.00000
100	A	A+width	0.025173	$[0.011330, 0.041946]$	0.00216
100	A	A+NWD	0.044876	$[0.013943, 0.078991]$	0.13370
100	A	W	-0.036355	$[-0.084820, 0.013965]$	1.00000
100	A+width	A+NWD	0.019704	$[-0.001784, 0.041961]$	0.76431
100	A+width	W	-0.061528	$[-0.118452, -0.008378]$	0.35730
100	W	A+NWD	0.081232	$[0.024631, 0.148726]$	0.08325

Table 23: Dam2D physical diagnostics and shared-resource runtime for the single-seed test. Momentum and energy columns are relative invariant errors, not exact conservation guarantees. V is endpoint velocity NRMSE. Latency is logged mean per-step milliseconds, not a controlled exclusive-device comparison.

Steps	Model	V	Linear mom.	Angular mom.	Kinetic energy	ms/step
20	A	0.7985	1.7709	0.6941	1.2419	13.09
20	A+width	0.8371	2.5225	1.0826	2.3436	17.13
20	W	1.7410	13.2507	10.2607	83.8278	27.43
20	A+NWD	0.7508	0.6631	0.4261	0.8929	41.00
50	A	0.8381	2.8055	1.3174	2.9345	12.36
50	A+width	0.9085	4.7983	2.7033	8.1269	16.45
50	W	2.8294	33.1267	27.2910	494.5588	25.79
50	A+NWD	0.7839	1.5865	1.0198	3.0888	39.51
100	A	0.9015	3.3189	1.9630	3.5496	11.83
100	A+width	0.9983	7.3205	5.4637	18.9102	16.10
100	W	4.6626	66.2533	79.4303	2000.9623	25.16
100	A+NWD	0.7435	3.8352	2.3479	10.3342	38.50

Table 24: Sand2D physical diagnostics and shared-resource runtime for the single-seed test. Momentum and energy columns are relative invariant errors, not exact conservation guarantees. V is endpoint velocity NRMSE. Latency is logged mean per-step milliseconds, not a controlled exclusive-device comparison.

Steps	Model	V	Linear mom.	Angular mom.	Kinetic energy	ms/step
20	A	0.8382	0.2125	1.1314	0.2493	7.31
20	A+width	0.8332	0.2357	1.1445	0.2990	7.55
20	W	0.7946	0.0000	0.0000	0.2078	25.02
20	A+NWD	0.7939	0.2415	0.7882	0.3785	33.50
50	A	1.7702	0.4791	2.3449	0.5127	7.00
50	A+width	1.6965	0.5115	1.9906	0.5791	7.15
50	W	1.8961	0.0000	0.0000	0.3233	23.63
50	A+NWD	1.5895	0.4572	1.4672	0.6462	33.15
100	A	2.1588	0.7907	3.1415	0.7434	6.92
100	A+width	1.9840	0.8167	2.1165	0.8017	6.93
100	W	3.0585	0.0000	0.0000	0.4829	23.42
100	A+NWD	1.8675	0.6329	2.1683	0.8267	33.38

Table 25: Additional physical metrics, one-seed test means. Dam2D uses target-trajectory kinetic-energy MSE; Sand2D uses maximum penetration over rollout and mean outside fraction. These are distinct from the relative invariant errors above.

Steps	Model	Dam2D KE MSE	Sand2D penetration	Sand2D outside fraction
20	A	0.005180	0.062795	0.071434
20	A+width	0.004715	0.060450	0.069940
20	W	0.020632	0.055284	0.071941
20	A+NWD	0.005257	0.057482	0.067222
50	A	0.081709	0.239203	0.216076
50	A+width	0.070533	0.221310	0.207743
50	W	2.354616	0.221898	0.218679
50	A+NWD	0.083261	0.206633	0.201954
100	A	0.317234	0.511644	0.398919
100	A+width	0.245480	0.459024	0.383027
100	W	54.497072	0.759185	0.435556
100	A+NWD	0.299272	0.431367	0.368256

F BONDED-SYSTEM TRANSFER

F.1 DYWORLDS-BONDED LOCAL-RADIUS STUDY

The local-radius sensitivity study below is distinct from the cross-benchmark comparison in Table 1. It uses three seeds and 256 held-out trajectories per seed, with the integrator, rollout curriculum, and prediction target held fixed. NWD at radius 0.35 reduces position NRMSE relative to STT-R by 2.5%, 1.9%, and 7.8% at 20, 50, and 100 steps. Within this radius study, the pair-force graph baseline is more accurate at those endpoints; increasing NWD's local radius improves over the radius-0.15 configuration but does not close that gap. The radius-0.15 model is not a Fixed-K ablation.

Table 26: DyWorlds-Bonded local-radius sensitivity study: position NRMSE, mean $\pm$ sample standard deviation across three training seeds and 256 test trajectories per seed. The radius parameter denotes local graph connectivity, not sparse neuronal world membership.

Rollout steps	NWD (radius 0.35)	NWD (radius 0.15)	STT-R	GNS-PF
20	0.034590 ± 0.000175	0.035654 ± 0.001065	0.035476 ± 0.001009	0.034299 ± 0.000132
50	0.240733 ± 0.000908	0.246193 ± 0.005505	0.245342 ± 0.005280	0.239252 ± 0.000685
100	0.196734 ± 0.004041	0.217182 ± 0.027025	0.213453 ± 0.026985	0.183868 ± 0.002200