Poisson-Algebraic Parallel Scan: A Fast Symplectic Framework for Neural Hamiltonians

Jiwoong Kim, Erdembileg Davaasuren, Youngsuk Lee, Sungwoo Park*

*Corresponding author

Department of Computer Science and Engineering, Korea University

KOREA UNIVERSITY 1905

NEURAL INFORMATION PROCESSING SYSTEMS

NeurReps Workshop

Motivation

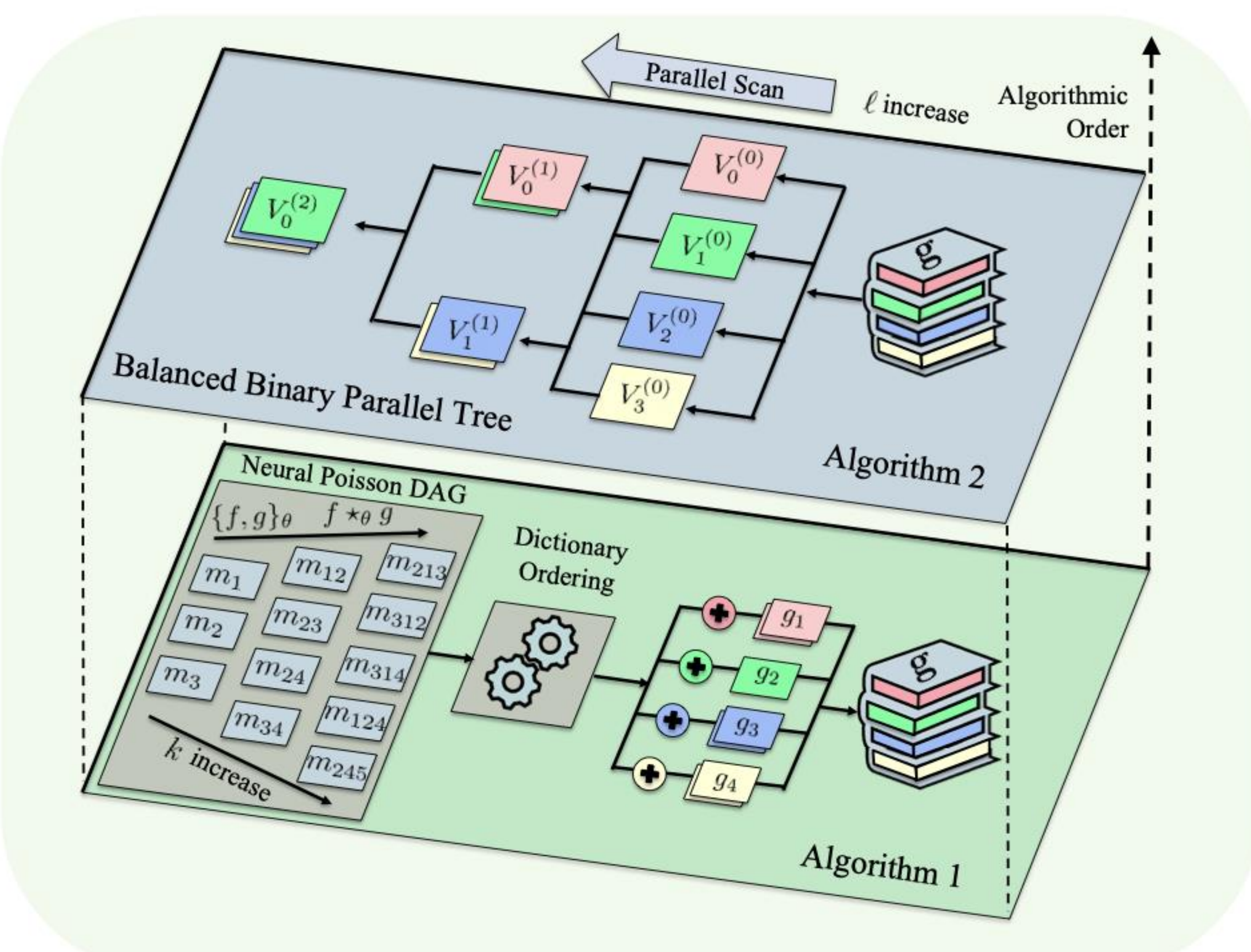
- HNNs preserve physics, but rely on sequential ODE integrators
- Sequential dependence follows number of steps M
- Long-horizon simulation: computationally prohibitive
- Need for parallelization of Hamiltonian flows, along the **timestep M**

- Goal: construct Hamiltonian flows compatible with parallel scan -> **How?**

- **Associativity of flow composition is the missing key!**

- Represent the Hamiltonian using polynomial generators in a finite Poisson algebra.
- Induced flows form a Lie group with strictly associative composition, enables parallel prefix-scan, Symplectic consistency is preserved by construction.

Pipeline



Methodology

Polynomial Poisson Algebra and DAG structure

We represent hamiltonian as truncated polynomial basis

$$P_{\le r} := \mathrm{span}\{\, q^\alpha p^\beta \mid |\alpha| + |\beta| \le r \,\} \quad g_\theta(z,t) = \sum_{|\alpha|+|\beta|\le r} c_{\alpha,\beta}(t;\theta)\, q^\alpha p^\beta$$

To keep the function space finite, the truncation operator with multiplication and Poisson bracket are replaced;

$$\pi_{\le r}(f) = \sum_{|\alpha|+|\beta|\le r} c_{\alpha,\beta} q^\alpha p^\beta \quad f \star g = \pi_{\le r}(fg),\ \{f,g\}_\theta = \pi_{\le r}(\{f,g\})$$

This induces a finite-dimensional Poisson algebra.

Starting from the base generator set $G(0) = P_{\le r}$, we iteratively construct $G(k) = \left\{ \pi_{\le r}(f \star g),\ \pi_{\le r}(\{f,g\}) \,\middle|\, f,g \in G(k-1) \right\}$, with removing redundant terms using algebraic identities: commutativity of the truncated product, antisymmetry of the poisson bracket, and Jacobi identity.

Through this closure process, the union $G(\le K) = \bigcup_{k=0}^{K} G(k)$ forms the **Neural Poisson Algebra**.

Lie Group Flows

From $g_m \in G(\le K)$ we define the truncated Lie operator, $\mathcal{L}_{g,\le r} f := \pi_{\le r}\{f,g\}$ and formally define a Lie group as

$$\mathscr{T}_{r,K} := \left\{ \Phi_{\epsilon,g} := \exp(\epsilon \mathcal{L}_{g,\le r}) = \sum_{j=0}^{r} \frac{\epsilon^j}{j!} \mathcal{L}^j_{g,\le r} \;\middle|\; g \in G^{(\le K)} \right\}$$

form an embedded Lie subgroup, inheriting **associativity** for elementary Lie group update $U_m = \Phi_{\epsilon,g_m} = \exp(\epsilon L_{g_m,\le r})$, in the walk $\mathcal{W}_{\mathbf{g}} : \{0,1,\dots,M\} \to \mathscr{T}_{r,K}$

Parallel Scan Algorithm

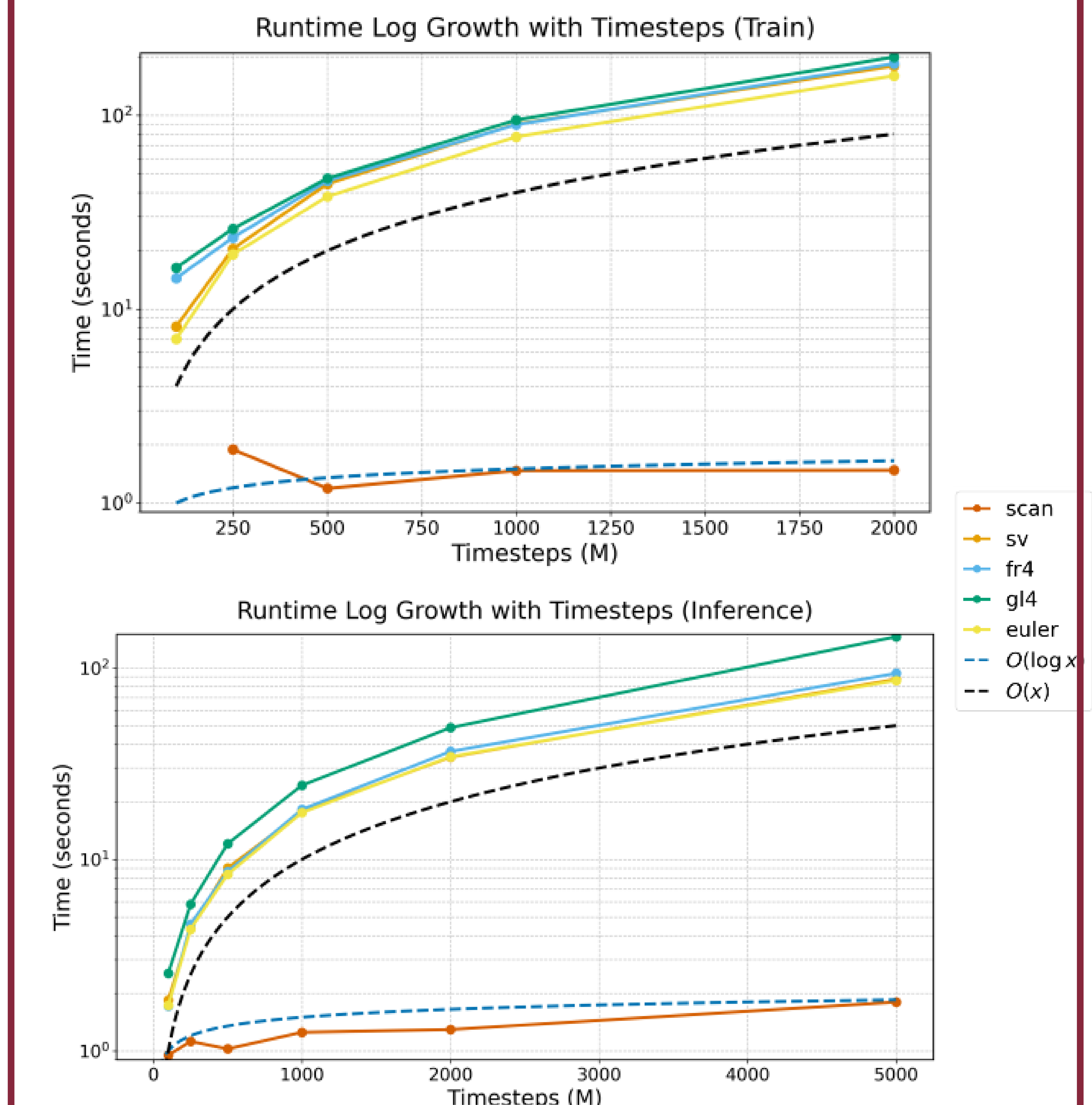
Thus, for given per-step flows $U_1,\dots,U_M$, we apply parallel scan algorithm with defining each steps,

$$V_m^{(0)} = U_m, \qquad V_m^{(\ell)} = V_{2m}^{(\ell-1)} \circ V_{2m-1}^{(\ell-1)}$$

This eventually provides the **acceleration along the timescale**, not dimensionality or model complexity that are largely treated before.

Experiments

We evaluate on Hamiltonian systems compared with Euler, Störmer–Verlet, Forest–Ruth 4th, and Gauss–Legendre 4th.



As seen in the graph, our model shows log-scale complexity along the timescale, offers a scalable path for fast neural simulators in large-scale physics.

References

Vladimir I Arnold (1989). “Mathematical Methods of Classical Mechanics.” In: Graduate Texts in Mathematics 60. Springer.

Guy E Blelloch (1990). “Prefix sums and their applications.” In: Technical Report CMU-CS-90-190, Computer Science Department, Carnegie Mellon University.

Ricky T Q Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud (2018). “Neural ordinary differential equations.” In: arXiv preprint arXiv:1806.07366.

Samuel Greydanus, Misko Dzamba, and David Yosinski (2019). “Hamiltonian neural networks.” In: Advances in Neural Information Processing Systems 32.

Jerrold E Marsden and Tudor S Ratiu (1999). “Introduction to Mechanics and Symmetry.” In: Texts in Applied Mathematics 17. Springer.