Learning Verifiable Mathematical Laws for Scientific Agents via Graph-Hamiltonian World Models

Hrishi Sunder [1]

Abstract

Scientific agents need learned mathematical laws, not only next-state predictors. We study how an agent can learn a compact mathematical law, verify its invariants during deployment, and use it for intervention. The learned object is a graph-local Hamiltonian H_G: a known interaction graph decomposes the scalar energy into node and edge terms, and its gradients define dynamics through $f = J\nabla H_G$. This representation yields mathematical verification conditions for planned rollouts, including symplecticity, Hamiltonian integrability, and energy drift. For finite linear Hamiltonian bases, law learning reduces to ridge regression in the graph-local Hamiltonian dimension $P_G = |V|p_v + |E|p_e$, giving a finite-horizon planning guarantee with statistical term $\widetilde{O}(\sqrt{P_G/n})$ and matching lower bounds. Experiments on graph-coupled oscillators and nonlinear pendula validate n/P_G scaling, edge dependence, geometric verification, transition-only behavior, MPC intervention gains, and a minimal residual-guided refinement loop. Graph-Hamiltonian world models therefore provide verifiable learned laws as primitives for scientific agents.

1. Introduction: Scientific Agents Need Verifiable Mathematical Laws

Scientific agents need more than next-state predictors. An agent that plans an intervention in a physical system needs a mathematical law it can roll out, check, and refine when the law fails its own verification conditions. This paper studies a complementary AI4Math problem: *how a scientific agent can learn, verify, and use a mathematical law while planning interventions*. AI-for-math systems often emphasize theorem proving, autoformalization, and natural-language mathematical reasoning. Scientific intervention also needs learned mathematical structures that remain checkable during deployment. This paper treats AI4Math not only as theorem proving or symbolic derivation, but also as the problem of learning mathematical objects that scientific agents can use, check, and refine while acting in the world.

Why this is AI4Math. The mathematical object learned by the agent is a scalar Hamiltonian H_G. The reasoning procedure is rollout under the induced law $f = J\nabla H_G$. The verifier checks whether planned rollouts obey mathematical invariants: symplecticity, Hamiltonian integrability, and energy conservation under fixed actions. The refinement mechanism collects data where these mathematical checks fail. Thus the contribution is not a generic simulator, but a verifiable learned law for scientific reasoning and intervention.

Our thesis is that a learned Hamiltonian can function as a mathematical-law module: a compact object whose gradients generate dynamics, whose invariants are checkable, and whose failures provide refinement signals.

Hamiltonians are compact mathematical objects whose gradients define physical dynamics. Graph locality makes such laws statistically learnable in systems with interacting parts. Hamiltonian geometry supplies deployment-time verification conditions: symplecticity checks preservation of the rollout geometry, integrability checks whether a vector field could arise from a scalar Hamiltonian, and energy drift checks artificial dissipation or expansion under fixed-action rollouts. When these checks fail, the agent has a concrete reason to reject the rollout or collect data near the failure. We propose graph-local Hamiltonians as learned mathematical-law modules that connect data, physical dynamics, planning, verification, and refinement.

Scope and assumptions. The main theorem isolates a finite-basis, known-graph, fixed-action Hamiltonian setting so the statistical dimension and verification conditions are explicit. These assumptions are deliberate isolation choices. Section 8 discusses graph discovery, port-Hamiltonian forcing, neural or adaptive bases,

[1]University of Illinois Urbana-Champaign, Mathematics & Computer Science Undergraduate. Correspondence to: Hrishi Sunder <hsund@illinois.edu>.

The 3rd *AI for Math Workshop* at the 43rd International Conference on Machine Learning (ICML), Seoul, South Korea, 2026.
Copyright 2026 by the author(s).

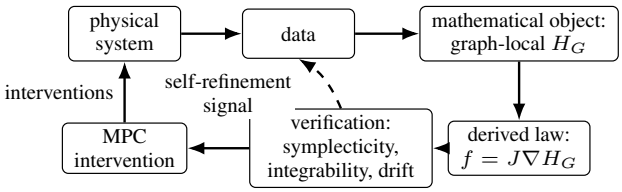


Figure 1. A verifier-guided scientific-agent loop with a graph-Hamiltonian world model. The learned Hamiltonian is the mathematical object, its gradients define the dynamics, verification residuals check planned rollouts, and high-residual trajectories trigger law-refinement data acquisition.

transition-data corrections, and stability-sensitive long-horizon bounds.

Figure 1 and Algorithm 1 summarize the central object of the paper: a verifier-guided scientific-agent loop in which a learned mathematical law supports intervention and gets refined when its verification checks fail. Figure 3 instantiates this loop empirically: verification residuals identify suspicious rollout states, acquisition targets those states, the law is refit, and MPC performance improves.

Several lines of work motivate this law module. LLM-centered systems for mathematical reasoning and scientific discovery use search, self-improvement, and verification to solve formal or natural-language problems (Trinh et al., 2024; Romera-Paredes et al., 2024). Hamiltonian and symplectic learning encode energy geometry (Greydanus et al., 2019; Zhong et al., 2020; Hairer et al., 2006); graph simulators encode relational locality (Battaglia et al., 2016; 2018; Sanchez-Gonzalez et al., 2020); factored MDPs and model-based RL provide sample-complexity templates for planning from learned models (Koller & Parr, 1999; Guestrin et al., 2003; Kearns & Singh, 2002). We combine these ingredients in a finite-dimensional law class where the statistical dimension is the graph-local Hamiltonian dimension.

Novelty. The novelty is the closed loop: a graph-local scalar mathematical law, its induced Hamiltonian dynamics, deployment-time verification residuals, finite-sample planning guarantees, and residual-guided law refinement. Prior Hamiltonian neural networks give structure; prior graph simulators give locality; prior factored MDPs give sample-complexity templates. This paper turns these ingredients into a verifiable mathematical-law module for a scientific agent.

Contributions.

1. **Learned mathematical-law module:** graph-local Hamiltonians H_G as compact scalar laws for scientific agents.
2. **Deployment-time verifier:** symplecticity, integrability, and energy-drift residuals for checking planned rollouts.
3. **Finite-sample law-to-intervention theory:** planning guarantees with statistical complexity governed by $P_G = |V|p_v + |E|p_e$, plus matching lower bounds.
4. **Verifier-guided refinement:** an agent loop that rejects or refines high-residual rollouts.
5. **Theorem-aligned validation:** experiments showing n/P_G scaling, edge dependence, geometric consistency, MPC gains, transition-data behavior, and residual-guided improvement.

What we claim. Graph-local Hamiltonian laws provide compact, verifiable mathematical objects for scientific agents. In finite bases, they yield sample-complexity guarantees controlled by P_G. Their residuals provide deployment-time checks and refinement signals.
What we do not claim. The residuals prove ground-truth correctness; the graph is discovered automatically; the method solves arbitrary long-horizon chaotic prediction; or the current experiments constitute broad physical simulation benchmarking.

2. Learning a Checkable Mathematical Law

Let $G = (V, E)$ be a known undirected interaction graph with $|V| = N$. Each node has position and momentum variables $q_i, p_i \in \mathbb{R}^d$ and state $x_i = (q_i, p_i) \in \mathbb{R}^{2d}$. The full state is $x = (q, p) \in \mathcal{X} \subset \mathbb{R}^{2dN}$. Actions are local:

$$a = ((a_i)_{i\in V}, (a_{ij})_{(i,j)\in E}) \in \mathcal{A}.$$

Let

$$J = \begin{pmatrix} 0 & I_{dN} \\ -I_{dN} & 0 \end{pmatrix}$$

be the canonical symplectic matrix.

Definition 2.1 (Graph-local Hamiltonian)**.** A Hamiltonian $H_G : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ is graph-local when

$$H_G(x, a) = \sum_{i\in V} H_i(x_i, a_i) + \sum_{(i,j)\in E} H_{ij}(x_i, x_j, a_{ij}). \quad (1)$$

Definition 2.2 (Graph-Hamiltonian law module)**.** A finite-horizon Graph-Hamiltonian law module has transition law

$$P(\cdot \mid x, a) = \mathrm{Law}(\Phi_h^a(x) + \xi),$$

where Φ_h^a is the time-h flow of the fixed-action system

$$\dot{x} = J\nabla_x H_G(x, a). \quad (2)$$

The definition covers controlled potentials, where

$$H_i(x_i, a_i) = T_i(p_i) + U_i(q_i, a_i),$$
$$H_{ij}(x_i, x_j, a_{ij}) = V_{ij}(q_i, q_j, a_{ij}).$$

Table 1. Contrast with nearby research areas. The paper adds learned mathematical laws with graph-local sample complexity, deployment-time verification, and intervention-oriented refinement.

Area	What it gives	What this paper adds
HNN/SympNet-style models	Hamiltonian structure and structure-preserving rollout	Graph-local dimension, deployment-time verifier, and planning theory
Graph simulators	Relational inductive bias for physical prediction	Scalar energy law and verifiable Hamiltonian geometry
Factored MDPs	Sample-efficient structured planning	Hamiltonian physical law with residual verification checks
AI4Math theorem proving	Formal symbolic verification	Learned mathematical laws for scientific agents acting in physical systems

Table 2. Claim-to-evidence map for the paper. This table makes explicit how the model, verifier, theorem, agent loop, and experiments support the AI4Math claim.

Claim	Theoretical object	Empirical evidence
Graph locality reduces sample complexity	$P_G = \lvert V\rvert p_v + \lvert E\rvert p_e$ graph-Hamiltonian dimension	n/P_G error collapse and edge-scaling threshold
Hamiltonian structure gives verification	Symplecticity, integrability, and energy-drift residuals	Near-zero verification residuals for Hamiltonian rollouts
Verified models improve intervention	Finite-horizon planning theorem and residual-gated corollary	MPC policy-gap improvements over unstructured vector fields
Residuals drive self-refinement	Verifier-guided law acquisition in Algorithm 1	Figure 3 refinement loop

Then node i evolves as

$$\dot{q}_i = \nabla_{p_i} T_i(p_i), \tag{3}$$

$$\dot{p}_i = -\nabla_{q_i} U_i(q_i, a_i) - \sum_{j:(i,j)\in E} \nabla_{q_i} V_{ij}(q_i, q_j, a_{ij}). \tag{4}$$

Thus local graph structure becomes a mathematical constraint on the vector field.

Proposition 2.3 (Fixed-action symplecticity)**.** *For every fixed action a, the deterministic flow Φ_h^a of (2) satisfies*

$$(D\Phi_h^a(x))^\top J D\Phi_h^a(x) = J.$$

The main theory assumes fixed-action Hamiltonian flows. Non-conservative actuation can be represented by an additional graph-local forcing term, discussed in Section 8.

3. A Verifier-Guided Scientific-Agent Loop

The goal is not only to minimize one-step prediction error. A learned law must support intervention, expose mathematical consistency checks, and provide failure signals when planned rollouts leave the regime where the law is reliable. This distinguishes the proposed law module from black-box transition models and from graph simulators optimized only for predictive accuracy.

The law module sits inside a scientific-agent loop rather than serving only as a one-step predictor. Algorithm 1 formalizes one component required by fuller scientific-agent systems: fit a graph-local Hamiltonian, derive its dynamics, use mathematical verification residuals to filter candidate rollouts, execute an MPC intervention, and acquire new data near high-residual states.

In this protocol the residuals are mathematical verification signals. They do not prove that a rollout is pointwise accurate, but they give the agent a deployment-time reason to distrust a mathematically inconsistent plan and gather law-refinement data near the states that triggered the alarm.

Algorithm 1 Verifier-Guided Graph-Hamiltonian Scientific Agent

Require: graph G, initial data $\mathcal{D}_0$, thresholds (τ_s, τ_i, τ_e), acquisition budget b, MPC horizon H

for round $r = 0, 1, 2, \ldots$ **do**
  fit $\widehat{H}_G$ on $\mathcal{D}_r$ by graph-local regression
  derive $\widehat{f} = J\nabla\widehat{H}_G$
  generate candidate action sequences and learned rollouts $\widehat{\tau}$
  compute verification residuals $\rho_{\text{symp}}(\widehat{\tau})$, $\rho_{\text{int}}(\widehat{\tau})$, and $\rho_{\text{energy}}(\widehat{\tau})$
  keep rollouts satisfying $\rho_{\text{symp}} \le \tau_s$, $\rho_{\text{int}} \le \tau_i$, $\rho_{\text{energy}} \le \tau_e$
  choose and execute the first MPC action from the best verified rollout
  append the resulting physical observation to $\mathcal{D}_r$
  if candidate rollouts exceed verifier thresholds **then**
    acquire b law-refinement samples near states with largest ρ_{tot}
  end if
  set $\mathcal{D}_{r+1}$ to the augmented data set
end for

For a planned rollout $\widehat{\tau}$, we summarize the verifier by

$$\rho_{\text{tot}}(\widehat{\tau}) = \rho_{\text{symp}}(\widehat{\tau}) + \alpha\rho_{\text{int}}(\widehat{\tau}) + \beta\rho_{\text{energy}}(\widehat{\tau}),$$

with nonnegative weights chosen so the three residuals are on comparable scales. This score can be used either as a hard gate, $\rho_{\text{tot}} \le \tau$, or as an acquisition signal for refinement: collect additional samples near states on high-

residual candidate trajectories, refit $\widehat{H}_G$, and replan.

4. Learning Laws Rather Than Predictors

Let node features $\phi_i : \mathbb{R}^{2d} \times \mathcal{A}_i \to \mathbb{R}^{p_v}$ and edge features $\phi_{ij} : \mathbb{R}^{2d} \times \mathbb{R}^{2d} \times \mathcal{A}_{ij} \to \mathbb{R}^{p_e}$ define the finite linear Hamiltonian class

$$H_\theta(x,a) = \sum_{i\in V} \theta_i^\top \phi_i(x_i,a_i) + \sum_{(i,j)\in E} \theta_{ij}^\top \phi_{ij}(x_i,x_j,a_{ij}). \tag{5}$$

Stack parameters into $\theta \in \mathbb{R}^{P_G}$ and concatenate features into $\Phi_G(x,a)$, so

$$H_\theta(x,a) = \theta^\top \Phi_G(x,a), \qquad P_G = |V|p_v + |E|p_e.$$

The induced vector field is linear in the unknown energy coefficients:

$$f_\theta(x,a) = J\nabla_x H_\theta(x,a) \tag{6}$$
$$= J(\nabla_x \Phi_G(x,a))^\top \theta = A_G(x,a)\theta. \tag{7}$$

The agent therefore does not learn a global vector field. It learns graph-local energy coefficients, and the Hamiltonian constraint determines the dynamics. With derivative observations

$$y_k = \dot{x}_k = f_\star(x_k,a_k) + \epsilon_k,$$

the ridge estimator is

$$\widehat{\theta} = \arg\min_{\theta\in\mathbb{R}^{P_G}} \sum_{k=1}^{n} \|y_k - A_G(x_k,a_k)\theta\|^2 + \lambda\|\theta\|^2. \tag{8}$$

Transition data $x'_k = \Phi_h^{a_k}(x_k) + \xi_k$ can be used through finite differences $y_k = (x'_k - x_k)/h$, which adds $1/h$ noise amplification and $O(h)$ small-step bias.

Comparison with less structured predictors. Let

$$\mathcal{F}_{\text{global}} = \{\text{generic global vector fields}\},$$
$$\mathcal{F}_{\text{graph}} = \{\text{generic graph-local vector fields}\},$$
$$\mathcal{F}_{\text{Ham},G} = \{J\nabla H_G : H_G \text{ graph-local}\}.$$

Then $\mathcal{F}_{\text{Ham},G} \subsetneq \mathcal{F}_{\text{graph}} \subsetneq \mathcal{F}_{\text{global}}$. The first inclusion is strict because Hamiltonian vector fields must satisfy cross-partial consistency after multiplication by $-J$; the second is strict because a global vector field can include non-neighbor or higher-order interactions absent from G.

This restriction changes the number of coefficients an agent must estimate. For polynomial bases of degree at most m and local dimension $2d$, the global vector-field class scales as

$$P_{\text{global}} \asymp 2d|V|\binom{2d|V|+m}{m},$$

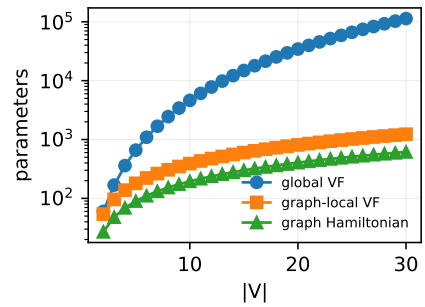


Figure 2. Parameter counts for global vector fields, graph-local vector fields, and graph-Hamiltonian models. A graph-Hamiltonian agent pays for local node and edge energy terms, not for an unconstrained global black-box dynamics model.

whereas graph-local Hamiltonians use

$$P_{\text{Ham},G} = |V|\binom{2d+m}{m} + |E|\binom{4d+m}{m}.$$

For fixed d, m and sparse graphs, the graph-Hamiltonian dimension is linear in graph size while the global polynomial vector-field dimension grows polynomially in $|V|$.

5. Sample Complexity of Learning a Checkable Law

We state the main result in the form needed for the scientific-agent interpretation. The theorem is a sample-complexity guarantee for learning a checkable mathematical law whose induced dynamics can be used for planning. Proof details and constants are in the appendix.

Assumption 5.1 (Finite-basis planning setup)**.** The learner observes derivative data $y_k = f_\star(x_k,a_k) + \epsilon_k$ with conditionally σ-sub-Gaussian noise. The graph-Hamiltonian design matrices satisfy $\|A_G(x,a)\|_{\text{op}} \le B_A$ and empirical conditioning $n^{-1}\sum_k A_k^\top A_k \succeq \kappa I$. The true vector field has approximate graph-Hamiltonian realizability error ϵ_{approx}, rollouts remain in a compact set, the true and learned flows and rewards satisfy the Lipschitz conditions used by the simulation lemma, and the learned model is rolled out with one-step local discretization error ϵ_{disc}.

The theory formalizes the benefit of learning a mathematical law rather than a generic predictor. The agent estimates $P_G = |V|p_v + |E|p_e$ scalar energy coefficients. Hamiltonian differentiation converts this scalar law into a vector field. Flow perturbation converts law-learning error into transition error. A simulation lemma converts transition error into intervention error. The agent pays for local node and edge energy terms, not for a global black-box dynamics model.

Theorem map: law learning to intervention
finite-basis law regression
$\Rightarrow$ vector-field error $\widetilde{O}(\sqrt{P_G/n})$
$\Rightarrow$ transition error by flow perturbation
$\Rightarrow$ value error by simulation lemma
$\Rightarrow$ planning guarantee
statistical + approximation + discretization + planning error

This chain is the bridge from learned mathematical law to agentic intervention.

Theorem 5.2 (Finite-sample planning guarantee)**.** *Under Assumption 5.1, let $\widehat{H}$ be fitted by (8), let $\widehat{f} = J\nabla\widehat{H}$, and let $\widehat{M}$ be the finite-horizon learned model induced by rolling out $\widehat{f}$. If $\widehat{\pi}$ is ϵ_{plan}-optimal in $\widehat{M}$ over a policy class Π, then there is a horizon-sensitivity constant C_H such that, with probability at least $1-\delta$,*

$$V_M^{\pi^\star}(x) - V_M^{\widehat{\pi}}(x) \le \widetilde{O}\left(C_H\sigma\sqrt{\frac{P_G + \log(1/\delta)}{n}}\right) + \epsilon_{\text{approx}} + \epsilon_{\text{disc}} + \epsilon_{\text{plan}}.$$

Here $P_G = |V|p_v + |E|p_e$.

The bound separates four error sources:

$$\text{statistical} + \text{approximation} + \text{discretization} + \text{planning}.$$

The statistical term is $\widetilde{O}(C_H\sigma\sqrt{(P_G + \log(1/\delta))/n})$; the other terms are ϵ_{approx}, ϵ_{disc}, and ϵ_{plan}. The theorem says that the agent can learn a mathematical law with sample complexity controlled by the graph-local Hamiltonian dimension, not by the dimension of an unconstrained global vector-field class. The proof has four steps. First, graph-Hamiltonian regression gives a uniform vector-field error of order $\widetilde{O}(\sigma\sqrt{P_G/n}) + \epsilon_{\text{approx}}$. Second, a Gronwall flow perturbation bound converts vector-field error into one-step transition error. Third, a finite-horizon simulation lemma turns one-step error into value error with sensitivity factor C_H. Fourth, a planning reduction compares the learned-model optimizer with the best policy in Π.

Theorem 5.3 (Finite-basis regression, compressed)**.** *Under the bounded-feature, sub-Gaussian-noise, conditioning, and approximate-realizability assumptions above, ridge regression satisfies*

$$\sup_z \|\widehat{f}(z) - f_\star(z)\| \le \frac{B_A}{\sqrt{\lambda + n\kappa}}\Big[\beta_n(\delta) + \sqrt{n}\epsilon_{\text{approx}} + \sqrt{\lambda}S\Big] + \epsilon_{\text{approx}},$$

where $\beta_n(\delta) = \widetilde{O}(\sigma\sqrt{P_G + \log(1/\delta)})$. In the exact finite-basis case this becomes $\widetilde{O}(\sigma\sqrt{P_G/n})$.

Transition observations. An agent usually observes transitions, not derivatives. If $x'_k = \Phi_h^{a_k}(x_k) + \xi_k$ and the learner fits $y_k = (x'_k - x_k)/h$, then $y_k = f_\star(x_k, a_k) + \xi_k/h + b_h(x_k, a_k)$. Thus the same theorem applies with a $1/h$ noise factor and an $O(h)$ finite-difference bias. This is why the derivative-data plots serve as the clean scaling test, while transition-data results are reported as a sanity check.

Corollary 5.4 (Sample complexity)**.** *For target suboptimality $\epsilon > \epsilon_{\text{approx}} + \epsilon_{\text{disc}} + \epsilon_{\text{plan}}$, it suffices to take*

$$n = \widetilde{O}\left(\frac{C_H^2\sigma^2(|V|p_v + |E|p_e + \log(1/\delta))}{(\epsilon - \epsilon_{\text{approx}} - \epsilon_{\text{disc}} - \epsilon_{\text{plan}})^2}\right)$$

samples. For fixed local basis sizes and sparse graphs, this scales linearly in the number of nodes, up to horizon, conditioning, and logarithmic factors.

Corollary 5.5 (Residual-gated planning)**.** *Let $\Pi_\tau \subseteq \Pi$ be the policies whose learned rollouts satisfy verifier thresholds, for example $\rho_{\text{symp}} \le \tau_s$, $\rho_{\text{int}} \le \tau_i$, and $\rho_{\text{energy}} \le \tau_e$. If $\widehat{\pi}_\tau$ is ϵ_{plan}-optimal in the learned model over Π_τ, then under the assumptions of Theorem 5.2,*

$$V_M^{\pi^\star}(x) - V_M^{\widehat{\pi}_\tau}(x) \le \widetilde{O}\left(C_H\sigma\sqrt{\frac{P_G + \log(1/\delta)}{n}}\right) + \epsilon_{\text{approx}} + \epsilon_{\text{disc}} + \epsilon_{\text{plan}} + \epsilon_{\text{gate}}(\tau),$$

where $\epsilon_{\text{gate}}(\tau) = V_M^{\pi^\star}(x) - \sup_{\pi\in\Pi_\tau} V_M^{\pi}(x)$ is the cost of rejecting or truncating plans that fail the verifier.

The corollary does not claim that a small residual certifies optimality. Instead, it separates the usual statistical and planning errors from the agentic cost of enforcing mathematical consistency. In deployment, this term is the price paid for refusing mathematically suspicious rollouts.

Theorem 5.6 (Sharpness)**.** *For every P, there exists a graph G, a P-dimensional linear graph-Hamiltonian class, and a design distribution such that any estimator from n derivative observations satisfies*

$$\inf_{\widehat{f}} \sup_{\theta\in\Theta} \mathbb{E}_\theta\|\widehat{f} - f_\theta\|^2_{L^2(\mu)} \ge c\frac{\sigma^2 P}{n}.$$

Thus the root prediction error is at least $\Omega(\sigma\sqrt{P_G/n})$ in the graph-Hamiltonian class.

6. Verification Conditions for Learned Laws

Graph-Hamiltonian law modules expose mathematical verification conditions that arbitrary black-box transition models do not. These residuals form the verifier used in Section 3. They catch planned rollouts that violate the mathematical structure the model claims to represent before the agent uses those rollouts for intervention.

Certification scope. These residuals certify consistency with the claimed mathematical structure, not ground-truth physical correctness.

Symplecticity: rollout geometry. For a learned one-step map $\widehat{\Phi}_h$, compute

$$\rho_{\text{symp}}(x) = \left\| (D\widehat{\Phi}_h(x))^\top J D\widehat{\Phi}_h(x) - J \right\|_F .$$

Hamiltonian flows satisfy this identity exactly, and symplectic integrators preserve it up to numerical precision for separable Hamiltonians. A large value flags a rollout map that does not preserve Hamiltonian geometry.

Integrability: Hamiltonian law consistency. A vector field is Hamiltonian only if $-Jf$ is a gradient field. A local check is

$$\rho_{\text{int}}(x, a) = \left\| \nabla_x(-Jf_\theta(x, a)) - \nabla_x(-Jf_\theta(x, a))^\top \right\|_F .$$

This residual tests cross-partial consistency. For scalar Hamiltonian models, integrability is zero by construction up to automatic-differentiation or numerical error. Its main value is comparative and diagnostic: it distinguishes Hamiltonian models from non-Hamiltonian baselines and catches implementation or numerical failures.

Energy drift: artificial dissipation or expansion. For fixed action and deterministic rollout, evaluate

$$\rho_{\text{energy}} = |H_\theta(x_T, a) - H_\theta(x_0, a)|.$$

Energy drift does not replace prediction error, but it detects artificial dissipation or expansion introduced by the learned rollout.

These residuals are not prediction metrics. They are mathematical verification conditions that a scientific agent can run while using a learned law for reasoning and control. Symplecticity does not guarantee pointwise long-horizon accuracy in unstable systems; it verifies that the learned map respects the geometry it claims to encode.

A small residual is not a proof of physical correctness. Rather, the residuals test whether the learned model remains consistent with the mathematical law class it claims to represent. This distinction matters in deployment: prediction error may be unavailable on planned counterfactual rollouts, but structural residuals can still be evaluated before the agent acts.

The verifier is therefore most useful for comparing Hamiltonian and non-Hamiltonian rollouts, detecting numerical failures, identifying extrapolative trajectories, and triggering refinement when candidate interventions leave the trusted regime.

How an agent uses the verifier. The finite-basis workflow is simple. The agent fits $\widehat{H}$ by ridge regression, derives $\widehat{f} = J\nabla\widehat{H}$, rolls out the learned dynamics with a symplectic integrator such as velocity Verlet, and plans with MPC or another shooting optimizer. Before trusting a rollout, it evaluates the residuals above on states encountered by the planner. A large residual is a mathematical failure signal: it does not prove the model is useless, but it gives a concrete reason to gather new data, refine the learned law, or reject a proposed extrapolation.

Table 3. Experiment-to-claim map.

Experiment	Claim tested
n/P_G scaling	Sample complexity follows graph-local Hamiltonian dimension
Edge sweep	Sample need grows with $\lvert E\rvert p_e$
Residuals	Learned rollouts preserve or violate mathematical structure in detectable ways
MPC	Verified structure improves intervention quality
Refinement demo	Residual failures guide refinement

7. Theorem-to-Agent Validation

These experiments are controlled mathematical validation experiments designed to test the theorem-to-agent chain: graph-local law learning, geometric verification, intervention through MPC, and residual-guided refinement. They are not intended as broad simulator benchmarks. The experiments follow the theorem-to-agent chain:

$$\text{energy estimation} \Rightarrow \text{transition accuracy} \Rightarrow \text{geometry} \Rightarrow \text{control}.$$

We use synthetic physical systems where the graph, Hamiltonian basis, and controlled dynamics are known. The empirical goal is to validate the complete chain: graph-local energy estimation reduces sample complexity; the induced dynamics preserve checkable geometry; the resulting law module improves intervention quality under MPC; and residual failures guide a minimal refinement loop.

Environments. Env 1 is a controlled harmonic oscillator used as an implementation check. Env 2 is a graph-coupled linear oscillator with

$$H_G = \sum_i \frac{1}{2}p_i^2 + \sum_i \frac{1}{2}\omega_i^2 q_i^2 + \sum_{(i,j)\in E} \frac{1}{2}k_{ij}(q_i - q_j)^2 - \sum_i a_i q_i.$$

Env 3 is a nonlinear graph pendulum with

$$H_G = \sum_i \frac{1}{2}p_i^2 + \sum_i m_i(1 - \cos q_i) + \sum_{(i,j)\in E} k_{ij}(1 - \cos(q_i - q_j)) - \sum_i a_i q_i.$$

Both graph environments use the exact basis $P_G = 3|V| + |E|$. We compare global vector fields, graph-local vector

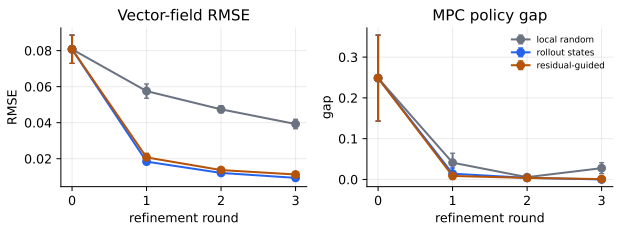
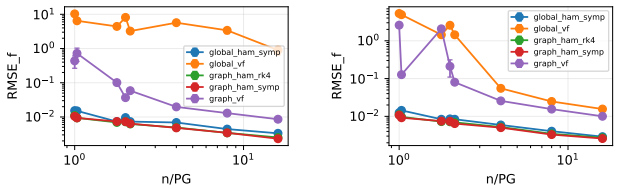


Figure 3. Residual-guided self-refinement on a small Env 3-style graph pendulum with $P_G = 11$. Starting from seven local samples, each round adds six samples. Residual-guided acquisition selects high energy-drift rollout states; rollout-state acquisition samples rollout states uniformly; local random continues sampling near the initial distribution. Residual-guided refinement reduces RMSE from 0.0808 to 0.0112 and MPC gap from 0.248 to 7.94×10^{-4}, outperforming local random exploration and closely tracking the rollout-state baseline.



(a) Linear graph oscillators. (b) Nonlinear graph pendula.

Figure 4. Vector-field error versus normalized sample size n/P_G. Env 2 graph-Hamiltonian RMSE drops from 0.01046 at $n/P_G = 1$ to 0.00232 at $n/P_G = 16$; Env 3 drops from 0.01081 to 0.00259.

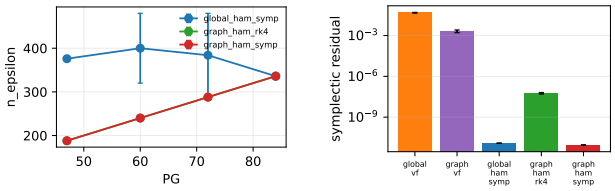
fields, global Hamiltonians, graph-local Hamiltonians with RK4 rollout, and graph-local Hamiltonians with velocity Verlet.

Minimal verifier-guided refinement loop. Figure 3 instantiates the protocol in Algorithm 1. This is a minimal self-refinement loop: mathematical verification identifies suspicious rollouts, produces an actionable data-acquisition signal, the model is refit, and MPC replans. We train an Env 3-style graph pendulum from $n = 7 \approx 0.6P_G$ local samples, generate candidate rollouts, and add six new samples per round. Because the fitted model is a scalar Hamiltonian, integrability is zero by construction; in this exact finite-basis demo the verifier score is dominated by energy drift along learned rollouts. Residual-guided acquisition selects the highest-drift rollout states, refits $\widehat{H}_G$, and replans. Across 12 seeds, residual-guided refinement reduces vector-field RMSE by 7.2× and nearly closes the MPC policy gap. This instantiates one component required by fuller scientific-agent systems; it does not claim autonomous hypothesis generation, graph discovery, or experiment design.

Sample complexity follows P_G. Figure 4 gives the main statistical validation. Increasing n/P_G from 1 to 16 reduces RMSE by 4.51× in Env 2 and 4.17× in Env 3. Endpoint slopes are 0.543 and 0.515, close to the predicted $n^{-1/2}$ behavior.

Table 4. Transition-observation sanity check on an Env 2-style graph oscillator with $P_G = 31$ and $n = 8P_G$. Transition rows use finite differences with $\sigma_x = 10^{-2}h$ state noise. The trend matches the theory: transition-only learning remains usable, with the expected finite-difference bias/noise tradeoff.

Observation	h	State noise	RMSE_f
Derivative	–	–	0.00112 ± 0.00003
Transition	0.02	2×10^{-4}	0.00185 ± 0.00005
Transition	0.05	5×10^{-4}	0.00405 ± 0.00010
Transition	0.10	10^{-3}	0.00942 ± 0.00021



(a) Edge-scaling threshold. (b) Symplectic residual.

Figure 5. Edge and verification checks. With $|V| = 12$, varying $|E|$ gives $P_G \in \{47, 60, 72, 84\}$. At threshold $\epsilon = 0.006$, the graph-Hamiltonian model reaches the threshold at $n_\epsilon = 4P_G$ for every tested edge count. Symplectic Hamiltonian rollouts have residuals near floating-point precision.

Edge sweep tests the $|E|p_e$ term. Figure 5(a) fixes $|V| = 12$ and varies edge count. The graph-Hamiltonian model reaches the RMSE threshold at $188, 240, 288, 336$ samples for $P_G = 47, 60, 72, 84$, exactly $n_\epsilon = 4P_G$ across the tested grid. This directly validates the $|E|p_e$ term.

Residuals verify geometric consistency. Figure 5(b) and Table 5 show that Hamiltonian symplectic rollouts have verification residuals near numerical precision. In Env 2, graph-Hamiltonian Verlet has residual 9.02×10^{-12}, while the global vector-field baseline has 4.76×10^{-2}. In Env 3, the corresponding values are 1.65×10^{-11} and 8.74×10^{-3}.

Verified structure improves intervention quality. We also run CEM-MPC with 64 fixed initial states and episode horizon 50. The policy-gap curves do not collapse as cleanly as vector-field error because intervention quality depends on topology, horizon, oracle quality, and planner optimization. The separation persists: in Env 2, the global vector-field policy gap is 248.2, while structured models are around 106.4–106.7; in Env 3, graph-Hamiltonian models reach gaps 155.8–156.4, compared with 170.8 for graph-local vector fields and 260.7 for global vector fields. Full control tables appear in Appendix F.

Table 5. Noisy derivative-data verification metrics. Lower is better. The full graph-Hamiltonian model combines low RMSE with near-zero symplectic residual.

Env.	Model	RMSE_f	RMSE_1	Symp. residual
2	Global VF	4.923	0.2458	4.76×10^{-2}
2	Graph-local VF	0.09395	0.004688	2.09×10^{-3}
2	Global Ham.	0.007275	0.000717	1.22×10^{-11}
2	Graph Ham. RK4	0.005379	**0.000269**	5.74×10^{-8}
2	Graph Ham. Verlet	**0.005279**	0.000659	9.02×10^{-12}
3	Global VF	1.113	0.03340	8.74×10^{-3}
3	Graph-local VF	0.3005	0.009018	3.24×10^{-3}
3	Global Ham.	0.006361	0.000193	2.00×10^{-11}
3	Graph Ham. RK4	0.005375	0.000161	2.27×10^{-9}
3	Graph Ham. Verlet	**0.005248**	**0.000160**	1.65×10^{-11}

8. Limitations and Extensions

Known graph. We assume the interaction graph is known as a scope boundary, not as a claim that future scientific agents should be given the graph. This isolates the statistical role of $P_G = |V|p_v + |E|p_e$. Learning the graph would add a structure-learning problem and a model-selection term. The edge-scaling experiment shows that the term matters because sample need tracks $|E|$.

Fixed-action Hamiltonian dynamics. We focus the main theorem on fixed-action Hamiltonian flows deliberately, so that the geometric verification layer is mathematically clean. Real actuators may inject non-conservative forces. A natural extension is

$$\dot{x} = J\nabla H_G(x, a) + g_G(x, a),$$

where g_G is a graph-local forcing or port-Hamiltonian residual. If the residual contributes error ϵ_{port}, the value bound gains an additional term of order $C_H h e^{Lh} \epsilon_{\text{port}}$.

Finite basis and derivative data. We use finite linear Hamiltonian bases and derivative observations to expose sample complexity without neural approximation ambiguity. This is a modeling lens for the theorem, not a restriction on the agent protocol. Transition observations introduce the expected $1/h$ noise amplification and $O(h)$ finite-difference bias, and Table 4 gives a main-text sanity check. Neural Graph-Hamiltonian law modules, adaptive basis selection, and active experiment design for better conditioning are natural extensions.

Long horizons. The finite-horizon theorem makes horizon sensitivity explicit through C_H. In stable or nonexpansive systems $C_H = O(H^2)$; in unstable or chaotic systems it can grow quickly. The claim is therefore geometric fidelity and verifier-guided caution, not arbitrary long-horizon pointwise prediction. Symplecticity improves geometric fidelity, not worst-case pointwise closeness.

9. Conclusion

We propose graph-local Hamiltonians as checkable mathematical laws that scientific agents can learn from data, verify during planning, and refine when verification fails. The agent estimates a scalar graph-local energy instead of an arbitrary global vector field; the Hamiltonian constraint supplies dynamics, symplectic structure, and integrability checks. In the AI4Math perspective, the learned Hamiltonian is the maintained mathematical object and the agent reasons by rollout, verification, rejection or refinement, and replanning. For finite linear bases, this yields planning guarantees with statistical complexity $\widetilde{O}(\sqrt{(|V|p_v + |E|p_e)/n})$, and lower bounds show that the graph-local Hamiltonian dimension is intrinsic. The residual-gated corollary separates statistical model error from the cost of enforcing mathematical consistency during planning. The experiments support the theorem chain through a minimal residual-guided refinement loop, n/P_G error collapse, transition-only sanity checks, explicit edge scaling, near-zero symplectic residuals, and MPC intervention gains. A next step is to connect this law module to fuller scientific-agent systems that conjecture graph structure, choose experiments to improve conditioning, verify learned dynamics through geometric residuals, and use the law for intervention.

Impact Statement

This work aims to improve the reliability of scientific agents by making learned dynamics more mathematically checkable during planning and intervention. The potential benefits include safer model-based experimentation, clearer failure signals for learned physical laws, and more transparent links between data-driven models and classical geometric structure. Possible risks include over-trusting structural residuals as guarantees of real-world correctness, or deploying learned laws outside the regimes where their assumptions and validation data apply. The paper therefore emphasizes that verification residuals certify consistency with the claimed Hamiltonian structure, not ground-truth physical correctness, and that residual failures should trigger caution, rejection, or data collection rather than automatic intervention.

References

Battaglia, P. W., Pascanu, R., Lai, M., Rezende, D. J., and Kavukcuoglu, K. Interaction networks for learning about objects, relations and physics. In *Advances in Neural Information Processing Systems*, 2016.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gulcehre, C., Song, F., Ballard, A., Gilmer, J., Dahl, G., Vaswani, A., Allen, K., Nash, C., Langston, V., Dyer, C., Heess, N., Wierstra, D., Kohli, P., Botvinick, M., Vinyals, O., Li, Y., and Pascanu, R. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Greydanus, S., Dzamba, M., and Yosinski, J. Hamiltonian neural networks. In *Advances in Neural Information Processing Systems*, 2019.

Guestrin, C., Koller, D., Parr, R., and Venkataraman, S. Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research*, 19:399–468, 2003.

Hairer, E., Lubich, C., and Wanner, G. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer, 2 edition, 2006.

Kearns, M. and Singh, S. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49: 209–232, 2002.

Koller, D. and Parr, R. Computing factored value functions for policies in structured MDPs. In *International Joint Conference on Artificial Intelligence*, 1999.

Romera-Paredes, B., Barekatain, M., Novikov, A., Balog, M., Kumar, M. P., Dupont, E., Ruiz, F. J. R., Ellenberg, J. S., Wang, P., Fawzi, O., Kohli, P., and Fawzi, A. Mathematical discoveries from program search with large language models. *Nature*, 625:468–475, 2024.

Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., and Battaglia, P. W. Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning*, 2020.

Trinh, T. H., Wu, Y., Le, Q. V., He, H., and Luong, T. Solving olympiad geometry without human demonstrations. *Nature*, 625:476–482, 2024.

Zhong, Y. D., Dey, B., and Chakraborty, A. Symplectic ODE-net: Learning hamiltonian dynamics with control. In *International Conference on Learning Representations*, 2020.

A. Full Assumptions

Let $z = (x, a)$ and $A(z) = A_G(x, a) \in \mathbb{R}^{D\times P}$, where $D = 2d|V|$ and $P = P_G$. The detailed theorem assumes predictable designs, bounded features $\sup_z \|A(z)\|_{\mathrm{op}} \le B_A$, conditionally σ-sub-Gaussian derivative noise, approximate realizability $\sup_z \|f_\star(z) - A(z)\theta^\dagger\| \le \epsilon_{\mathrm{app}}$ with $\|\theta^\dagger\| \le S$, and empirical conditioning $n^{-1}\sum_k A(z_k)^\top A(z_k) \succeq \kappa I_P$.

B. Regression Proof

The ridge estimator satisfies

$$\widehat{\theta} = \arg\min_{\theta\in\mathbb{R}^P} \sum_{k=1}^{n} \|y_k - A_k\theta\|^2 + \lambda\|\theta\|^2,$$

$$V_n = \lambda I + \sum_{k=1}^{n} A_k^\top A_k.$$

Write $f_\star(z_k) = A_k\theta^\dagger + b_k$ with $\|b_k\| \le \epsilon_{\mathrm{app}}$. The normal equations give

$$V_n(\widehat{\theta} - \theta^\dagger) = \sum_{k=1}^{n} A_k^\top \epsilon_k + \sum_{k=1}^{n} A_k^\top b_k - \lambda\theta^\dagger.$$

A self-normalized martingale bound controls the noise term:

$$\left\|\sum_{k=1}^{n} A_k^\top \epsilon_k\right\|_{V_n^{-1}} \le \sigma\sqrt{P\log(1 + nB_A^2/\lambda) + 2\log(1/\delta)}.$$

The approximation term is at most $\sqrt{n}\epsilon_{\mathrm{app}}$ by duality, and the ridge-bias term is at most $\sqrt{\lambda}S$. Since $V_n \succeq (\lambda+n\kappa)I$, multiplying the resulting parameter bound by B_A gives the uniform vector-field bound.

C. Flow, Value, and Planning Proof

If $\sup_{x,a} \|\widehat{f}(x, a) - f_\star(x, a)\| \le \epsilon_f$ and $f_\star$ is L-Lipschitz, then for coupled flows $x_\star(t)$ and $\widehat{x}(t)$,

$$\|\widehat{x}(t) - x_\star(t)\| \le \int_0^t \epsilon_f\, ds + L\int_0^t \|\widehat{x}(s) - x_\star(s)\|\, ds.$$

Gronwall's inequality gives

$$\|\widehat{\Phi}_h^a(x) - \Phi_h^a(x)\| \le \begin{cases} (e^{Lh} - 1)\epsilon_f/L, & L > 0, \\ h\epsilon_f, & L = 0. \end{cases}$$

Adding the local discretization error gives one-step learned-model error ϵ_Φ. For any policy π, if the true closed-loop maps are L_Φ-Lipschitz and rewards are L_R-Lipschitz, then

$$|V_M^\pi(x) - V_{\widehat{M}}^\pi(x)| \le L_R\epsilon_\Phi \sum_{t=0}^{H-1}\sum_{s=0}^{t-1} L_\Phi^s.$$

The planning reduction follows by adding and subtracting learned-model values of $\pi^\star$ and $\widehat{\pi}$.

D. Lower Bounds

Let G have P isolated one-dimensional nodes and define

$$H_\theta(q,p) = \sum_{j=1}^{P} \theta_j q_j.$$

Then $\dot{q}_j = 0$ and $\dot{p}_j = -\theta_j$, so the regression design is identity in P dimensions. A hypercube packing with side length proportional to $\sigma/\sqrt{n}$ and Assouad's lemma give

$$\inf_{\widehat{f}} \sup_{\theta\in\Theta} \mathbb{E}_\theta \|\widehat{f} - f_\theta\|^2_{L^2(\mu)} \geq c\sigma^2 P/n.$$

A horizon-one value lower bound follows by choosing terminal rewards that read out the momentum shift after one Hamiltonian step.

E. Transition Observations

For transition observations $x'_k = \Phi_h^{a_k}(x_k) + \xi_k$, define $y_k = (x'_k - x_k)/h$. Then

$$y_k = f_\star(x_k, a_k) + \eta_k + b_h(x_k, a_k), \qquad \eta_k = \xi_k/h.$$

If ξ_k is conditionally σ_x-sub-Gaussian and $\|b_h\| \leq C_h h$, the same regression proof gives

$$\sup_z \|\widehat{f}(z) - f_\star(z)\| = \widetilde{O}\left(\frac{\sigma_x}{h}\sqrt{\frac{P_G + \log(1/\delta)}{n}}\right) + O(h) + O(\epsilon_{\text{app}}).$$

F. Experimental Details and Full Control Results

The main derivative-data experiments use $h = 0.05$ for Env 2, $h = 0.03$ for Env 3, derivative noise $\sigma = 0.01$, train/test splits, and ridge parameter $\lambda = 10^{-4}$ in noisy runs. Env 2 tests path, star, and connected random sparse graphs with $|V| \in \{4, 8, 16\}$ and sample sizes $n = \max(32, cP_G)$ for $c \in \{1, 2, 4, 8, 16\}$. The edge sweep fixes $|V| = 12$ and varies $|E| \in \{11, 24, 36, 48\}$.

The residual-guided refinement demo uses a three-node path graph pendulum with $P_G = 11$, $n_0 = 7$ initial local samples, three refinement rounds, and six added samples per round. The script reports means and standard errors over 12 random seeds for local random sampling, uniform rollout-state sampling, and residual-guided rollout-state sampling.

The transition-observation sanity check uses an eight-node path graph oscillator with $P_G = 31$ and $n = 8P_G$. Transition rows simulate $x' = \Phi_h^a(x) + \xi$ with $\sigma_x = 10^{-2}h$, fit finite differences $(x' - x)/h$, and report vector-field RMSE over 20 seeds. The generated summary is written to `figures/transition_data_sanity.csv`.

G. Reproducibility Notes

The experiment driver, configs, and generated CSV summaries accompany the paper source. The residual-refinement figure is generated by the included refinement-demo script, which also writes the corresponding CSV summary. The camera-ready figures are regenerated as vector PDFs where the underlying tabular data are available.

Table 6. Full CEM-MPC control metrics averaged over 1320 completed control rows. Policy gap is oracle return minus learned-MPC true return, so lower is better.

Env.	Model	Policy return	Oracle return	Gap	MPC value err.
1	Global VF	-65.56	-39.83	25.73	0.341
1	Graph-local VF	-65.54	-39.83	25.71	0.300
1	Global Ham.	-65.47	-39.83	25.64	0.061
1	Graph Ham. RK4	-65.53	-39.83	25.70	0.058
1	Graph Ham. Verlet	-65.53	-39.83	25.70	0.054
2	Global VF	-364.3	-116.2	248.2	1.27×10^{12}
2	Graph-local VF	-222.9	-116.2	106.7	0.799
2	Global Ham.	-222.6	-116.2	106.4	0.123
2	Graph Ham. RK4	-222.7	-116.2	106.5	0.070
2	Graph Ham. Verlet	-222.7	-116.2	106.6	0.093
2 edge	Global VF	-438.1	-199.5	238.6	7.39×10^{12}
2 edge	Graph-local VF	-323.1	-199.5	123.6	0.401
2 edge	Global Ham.	-322.9	-199.5	123.4	0.113
2 edge	Graph Ham. RK4	-323.3	-199.5	123.7	0.057
2 edge	Graph Ham. Verlet	-323.1	-199.5	123.6	0.109
3	Global VF	-1008.0	-747.1	260.7	104.0
3	Graph-local VF	-917.9	-747.1	170.8	8.44×10^{10}
3	Global Ham.	-903.6	-747.1	156.4	0.348
3	Graph Ham. RK4	-903.0	-747.1	155.8	0.284
3	Graph Ham. Verlet	-903.0	-747.1	155.9	0.291