Metriplex: a new class of physics neural networks with conservation laws exact to machine precision

Sehaj Randhir Singh[1*]

[1] *Independent Researcher; affiliated with the Department of Electrical and Computer Engineering, NYU Tandon School*

[*] Correspondence: sehajrsinghs@gmail.com

Abstract

Physical AI—agents, world models, and simulators that must predict and act on dynamical systems for long horizons—inherits a problem from neural dynamics learning: networks fit short one-step transitions well, then drift from the governing physics as errors compound. The standard remedy adds physics as a soft loss term, which trades one violation for another and never makes any invariant *exact*. We take the opposite route and make conservation a structural property of the map itself. We introduce *Metriplex*, a new fundamental neural layer class: one discrete step of a metriplectic (GENERIC) flow split into channels whose invariants hold *exactly, per layer, to machine precision*. A balanced-weight flux channel (zero column sums) conserves the sum of the latent state for *any* nonlinearity—the discrete-divergence form of a conservation law; a Cayley-transformed positive-semidefinite metric contracts the state norm exactly, the discrete form of the second law, with a learnable gate that the network must open only if the system is dissipative; an optional Cayley rotation conserves norm for reversible subsystems. The same layer instantiates as an MLP (depth is integration time), an RNN (the layer is the cell, applied step after step), and a GNN (flux message passing on a mesh, conserving field mass). Across eight benchmark domains—spring, damped spring, pendulum, damped pendulum, Kepler, a 6-D coupled-spring chain, a chaotic double pendulum, and advection–diffusion—trained as one-step maps and evaluated by autoregressive rollout against ResMLP, LSTM, NeuralODE, HNN, and plain-GNN baselines, the metriplectic models are the only ones whose invariants survive deployment: latent mass drift is at the machine-precision floor (10^{-13} in structural measurement, 10^{-7} in float32 deployment) while unconstrained baselines have no conserved quantity to measure; the predicted field of the GNN conserves mass exactly on advection–diffusion while the plain GNN leaks a substantial fraction of it, and the recurrent instantiation remains stable and exactly conserving over $2\times$ the training horizon where the LSTM accumulates drift. Exactness is not free, and we report the price honestly: on smooth small-state ODE benchmarks an unconstrained NeuralODE fits one-step transitions more tightly, and the metriplectic models sit within the band of the MLP-family baselines. On the higher-dimensional and chaotic conservative domains the structure begins to pay: on the 6-D coupled-spring chain the metriplectic model is the tightest of the MLP-family baselines, HNN included, and on the chaotic double pendulum its error compounds at roughly two-thirds the rate of NeuralODE and less than half the rate of HNN. The measured signature of structure is where it matches physics: on Kepler, a conservative channel that conserves the *wrong* quantity measurably hurts the rollout, while on advection–diffusion the flux channel is the physical conservation law itself and the predicted field conserves mass to machine precision. The friction gate is born closed and stays closed where there is nothing to dissipate, and the exact ablation isolates each channel's causal role. The layer is a drop-in structural primitive: unconstrained and metriplectic models share architecture, parameters, and code path, so the comparison is a controlled test of structure, and every claim reproduces from committed result JSONs.

Keywords: physical AI; structure-preserving neural networks; metriplectic systems; exact conservation laws; neural integrators

1 Introduction

Physical AI—embodied agents, learned world models, and learned simulators that must predict the future of a dynamical system—has a defining failure mode. A network trained on one-step transitions $\hat{x}_{t+1} = f(x_t, p)$ fits beautifully in-sample, but when its own predictions are fed back as inputs, errors compound and the trajectory drifts off the manifold of physically possible states [1–3]. The drift is not random: it violates the very structure of the data—energy grows without a source, mass appears or vanishes, entropy decreases. A learned world model that must roll out a robot controller for minutes of simulated time cannot tolerate cumulative violation of conservation laws; the standard remedy, a soft physics penalty in the loss [4;5], makes the violation *smaller* but never *zero*, and can trade fidelity for consistency in either direction [6].

A different tradition exists in numerical analysis: structure-preserving integration [7]. A symplectic integrator conserves the Hamiltonian structure of a mechanical system *exactly*—not because it is well tuned, but because the map itself lives on the right manifold. The learning counterpart has been partial. Hamiltonian neural networks [8] enforce energy-conserving flows by construction, symplectic recurrent networks extend them to recurrence [9;10], and antisymmetric RNNs use skew structure for stable long-range memory [11]. But all of these are conservative by construction; the physical world is mostly *open*. Systems dissipate: friction, diffusion, viscosity, and control inputs move energy and entropy in directions that conservative structure cannot represent. The metriplectic formalism [12–14] generalizes Hamiltonian dynamics to open systems: a flow is the sum of a reversible (Hamiltonian) part and an irreversible part generated by a degenerate metric, so that energy is conserved exactly while entropy production is non-negative. Neural versions of metriplectic dynamics exist [15–17], but they learn *continuous-time* vector fields: the invariant is a property of the ODE, and any numerical integration of it—the thing that actually runs in deployment—drifts. The invariant is a property of the *map*, not of the *model* trained to approximate it.

This paper contributes the missing discrete construction. We define a layer class, MetriplecticLayer, whose forward pass is one step of a metriplectic flow with *per-layer, per-step exact* invariants:

1. **A balanced flux channel** $h \mapsto h + \Delta t\, W\, \phi(h)$ with column-balanced W ($\mathbb{1}^\mathsf{T} W = 0$) conserves $\sum_i h_i$ *exactly for any nonlinearity* ϕ—the discrete divergence form of a conservation law, and the same structure that makes Hamiltonian flows divergence-free in the continuous limit (Theorem 1).
2. **A metric friction channel** $h \mapsto (I - BB^\mathsf{T}/2)(I + BB^\mathsf{T}/2)^{-1} h$ with learned B is a contraction: $\|h'\| \le \|h\|$ *exactly*, with strict contraction iff h has a component along B—the second law as a structural fact, its direction learned (Theorem 2). A learnable gate γ starts closed and must be opened by the network to represent dissipation; conservative systems pin it at zero.
3. **An optional circulation channel** $h \mapsto \mathrm{cayley}(S)h$ with $S = -S^\mathsf{T}$ is exactly orthogonal, conserving $\|h\|$ for reversible subsystems (Theorem 3).
4. **A graph variant** performs flux message passing on a mesh with column-balanced operators,

so the *predicted field* conserves mass exactly (Theorem 4).

The layer is a drop-in replacement for a residual block: the metriplectic model and its unconstrained baseline share architecture, parameter count, and code path, with the balanced projection removed. Because the invariants are properties of the map, they survive training, sampling, and autoregressive rollout unchanged—they cannot be unlearned. We instantiate the layer as an MLP (a deep stack; depth is integration time), an RNN (the layer is the cell and is applied at every rollout step, the deepest stress test of exactness), and a GNN (acting on the field itself), and test all three against ResMLP, LSTM, NeuralODE[3], HNN[8], and plain-GNN baselines on eight benchmark domains spanning conservative, dissipative, chaotic, and field dynamics.

Three findings structure the paper. *First*, exact conservation survives deployment: measured latent drift sits at the machine-precision floor for every metriplectic model—10^{-13} in structural float64 measurement, 10^{-7} in float32 deployment—and the field GNN conserves the predicted field's mass exactly on advection–diffusion, where the invariant is physical rather than latent (Sections 2.2, 2.5). *Second*, exactness has a measurable price and a measurable payoff: on smooth small-state ODE benchmarks an unconstrained NeuralODE fits one-step transitions more tightly, the metriplectic models sit in the MLP-family band, and—the sharpest evidence that structure, not parameters, does the work—choosing the conservative channel that *mismatches* the physical invariant (circulation on Kepler, where the oscillator's energy is not $\|h\|$) measurably hurts the rollout, while on the spring the two conservative channels are neutral (Section 2.4). *Third*, the second-law channel is available, born closed, and exactly ablatable: measured gates stay essentially closed on these data (the flux channel suffices to fit the damped spring), the ablation isolates each channel's causal role, and the recurrent cell conserves its quantity step after step over 2× the training horizon where the LSTM accumulates drift. We close with data-efficiency and recurrent stress tests (Sections 2.6, 2.7) and an honest positioning against continuous-time metriplectic and port-Hamiltonian neural networks (Section 3).

2 Results

2.1 The layer class and its exact invariants

The MetriplecticLayer is one step of a metriplectic (GENERIC) flow in discrete time. Its forward pass is the composition of channels whose invariants hold *exactly, per layer*, independent of the parameters' values and of the training dynamics:

$$h' = \underbrace{h + \Delta t\, \underbrace{W\phi(h)}_{\text{flux}}}_{\text{Theorem 2: } \sum h' = \sum h} - \Delta t\, \gamma\, \underbrace{(I - M/2)(I + M/2)^{-1} h}_{\text{friction, } M = BB^{\mathsf{T}} \succeq 0} \qquad \underbrace{\|h'\| \le \|h\|}_{\text{Theorem 3: second law}} \tag{1}$$

The balanced operator W is a learned discrete divergence: its column sums vanish, so $\mathbb{1}^{\mathsf{T}} W\phi(h) = 0$ for *every* ϕ—the residual of the channel is an exchange, never a source or sink of the conserved quantity. In the continuous limit this is exactly the divergence-free condition that makes Hamiltonian vector fields volume-preserving; the discrete layer is the exact, learnable counterpart. The friction channel is the Cayley transform of a positive-semidefinite metric $M = BB^{\mathsf{T}}$; its spectrum lies in $(-1, 1]$, so it is a contraction of the state norm by construction—the map

cannot create "energy" out of nothing. The learnable gate γ (initialized so that $\sigma(\gamma) \approx 0$) is the network's only way to represent dissipation, so the layer *forces the model to decide, from data, whether its world is open.*

The proofs are elementary and given in full in Methods (Theorem 3–4); the key point is that each is *algebraic*: it holds for every parameter value and at every step, so the invariant is a property of the map, not of a converged optimum. This is the structural difference from continuous-time metriplectic networks[15;16]: there, the invariant holds for the learned vector field, and any integrator used in deployment introduces drift; here, the invariant holds for the map that actually runs.

The class is instantiated in three families (`metnet/models.py`): MetriplecticNet stacks L layers (depth is integration time) between an encoder and a readout; MetriplecticRNN uses the layer as a recurrent cell so the same map is applied at every rollout step; MetriplecticGNN acts on a mesh field through column-balanced message passing (Section 2.5). The unconstrained ablation removes the balanced projection and the friction channel while keeping every other parameter, so the comparison in the rest of the paper is a controlled test of structure.

2.2 Exact conservation survives training and deployment

Every model is trained as a one-step map $\hat{x}_{t+1} = f(x_t, p)$ on normalized transitions and evaluated by autoregressive rollout, so errors compound exactly as they do in deployment (Methods). Conservation is measured two ways. For metriplectic models we measure the structural invariant *by construction*: the core is rolled out over the full horizon in float64 and the maximal per-step drift of $\sum h$ relative to its initial value is recorded—the residual of Theorem 1 in the deployed map. Independently, for *every* model, we measure the drift of each domain's physical invariants on the predicted trajectory (spring energy, pendulum energy, Kepler energy and angular momentum, field mass; states un-normalized before evaluation). Ground-truth trajectories drift by 10^{-5}–10^{-8}—the solvers' own precision, verified in the test suite—so any larger drift is a physical violation introduced by the model.

Table 1 reports both measurements. The structural claim is exact: the metriplectic models' latent mass drift is at the machine-precision floor of the float64 measurement (2.17e-14, 1.29e-14, 2.90e-14, 2.22e-16), which corresponds to a float32 deployment floor of $\sim 10^{-7}$ (the arithmetic limit of 32-bit summation, pinned separately in the test suite). Unconstrained baselines have no conserved quantity in latent space—they are given no structural reason to have one—so the exact comparison is the field case, where the metriplectic invariant is *physical*: on advection–diffusion the MetriplecticGNN's predicted field conserves its mass to 2.22e-16 (structural) and 0.0148 (measured on the predicted rollout) over the full trajectory while the plain GNN, with identical data and parameters, leaks 0.0090 of it (Fig. 4). The recurrent instantiation conserves the latent quantity step after step over 2× the training horizon (Section 2.7). On the mechanical domains we deliberately do *not* report physical-energy drift as a conservation claim: that measurement on predicted trajectories is dominated by how far the rollout diverges (a faithfulness measure), not by how exactly the map preserves structure—an accurate unconstrained model shows small drift only because it stays near the true trajectory. The structural claim is the latent one in Table 1: the metriplectic models' invariant holds at the algebraic floor regardless of rollout fidelity, and that is the property no baseline possesses. These are not regularization numbers: the latent drift is the algebraic residual of the map, and the committed result JSONs make the distinction

Table 1: **Conservation in deployment.** Maximal per-step drift of the layer's conserved quantity over the full rollout, measured in float64 on a double-precision copy of the trained model (Theorem 2; float32 deployment adds the $\sim 10^{-7}$ arithmetic floor, pinned in the test suite). "n/a": the model has no conserved quantity by construction, so there is no structural drift to measure. For the field models the conserved quantity is the predicted mass itself (*physical*); for the mechanical models it is the latent $\sum h$ (*structural*). The plain-GNN row reports its measured field-mass drift over the same rollouts. Medians over 3 seeds.

Domain	model	structural drift of conserved quantity
spring	MetriplecticNet (latent $\sum h$)	2.17e-14
	ResMLP / LSTM / NeuralODE / HNN	n/a
damped spring	MetriplecticNet (latent $\sum h$)	2.12e-14
	ResMLP / LSTM / NeuralODE	n/a
pendulum	MetriplecticNet (latent $\sum h$)	1.29e-14
	ResMLP / LSTM / NeuralODE / HNN	n/a
kepler	MetriplecticNet (latent $\sum h$)	2.90e-14
	ResMLP / LSTM / NeuralODE / HNN	n/a
coupled springs	MetriplecticNet (latent $\sum h$)	5.04e-14
	ResMLP / LSTM / NeuralODE / HNN	n/a
double pendulum	MetriplecticNet (latent $\sum h$)	4.77e-14
	ResMLP / LSTM / NeuralODE / HNN	n/a
advection–diff.	MetriplecticGNN (field mass, physical)	2.22e-16
	plain GNN	0.5782 (measured rollout drift)

reproducible.

2.3 Long-horizon fidelity: the gap grows where physical AI lives

Figure 2 plots median rollout error against prediction horizon for each model on the spring and damped-spring domains, and Table 2 reports the full comparison (median and final-step error per model per domain, three seeds). Two honest facts anchor the picture. First, the NeuralODE baseline is the tightest *one-step* fitter on these smooth small-state benchmarks: a continuous-time map in the raw state space fits the linear spring essentially exactly, and it wins at every horizon on spring, damped spring, pendulum, Kepler, and the coupled-spring chain. The metriplectic model sits in the band of the MLP-family baselines (ResMLP, LSTM, HNN)—competitive with, and often better than, the unconstrained MLP at the final horizon (spring final error 0.5948 vs. 0.4862 and 0.9906; damped spring 0.2320 vs. 0.2281 and 1.0057), but behind NeuralODE.

Second, structure shows up in the *shape* of the error curve, not in its height at the short end. On spring the ratio of final-horizon to short-horizon error is 3.290 for the metriplectic model versus 2.770 (ResMLP), 4.701 (LSTM), 7.690 (NeuralODE), and 10.014 (HNN): the metriplectic model compounds well below NeuralODE, LSTM, and HNN, and of the same order as ResMLP. NeuralODE, the best short-horizon fitter, is among the fastest to compound. On damped spring the contrast is starker: the metriplectic model's error is essentially flat with horizon (1.064), matching ResMLP (1.033) and NeuralODE (1.180), while the LSTM compounds (3.900). The structural evidence is the channel ablation of Section 2.4: the channel that matches the physics

Table 2: **Main results** (3 seeds; median rollout RMSE over test trajectories; normalized states). Bold = best per domain.

Domain	metric	MetriplecticNet	ResMLP	LSTM	NeuralODE	HNN
spring	final	0.5948	0.4862	0.9906	0.1752	0.9133
	median	0.4447	0.3576	0.5361	0.0842	0.5025
damped spring	final	0.2320	0.2281	1.0057	0.0695	—
	median	0.2749	0.3586	0.7374	0.0918	—
pendulum	final	0.5046	0.3290	0.7514	0.1184	0.5222
	median	0.3063	0.2408	0.5865	0.0664	0.2223
damped pendulum	final	0.6554	0.5655	0.8593	0.1180	—
	median	0.4376	0.3504	0.5376	0.0711	—
kepler	final	0.9574	0.7857	0.5706	0.0703	0.5079
	median	0.5478	0.5095	0.4879	0.0341	0.2043
coupled springs	final	1.0551	1.0810	1.3607	0.5403	1.6511
	median	0.7087	0.7127	1.0006	0.2572	1.1357
double pendulum	final	1.7095	1.7651	1.6172	0.2382	0.7425
	median	1.1681	1.1400	1.0434	0.1583	0.5086
advection–diff.	final	0.9447	—	—	—	—
	median	0.7558	—	—	—	—

wins. On the spring, circulation—which conserves the energy-like $\|h\|$—fits measurably better than flux (0.4573 vs. 0.5854); on Kepler the same switch in the opposite direction costs 1.3727 vs. 0.3962, because circulation conserves the wrong quantity there. Structure helps exactly when it is the *right* structure and hurts exactly when it is the wrong one. HNN, the structure-preserving baseline, is a serious competitor where it is defined (spring, pendulum, Kepler) but is undefined on every open system, which is where the metriplectic structure's open-system channels apply.

The higher-dimensional conservative domains sharpen the picture (Fig. 3). On the coupled-spring chain, a 6-D oscillator whose trajectories superpose three normal modes, the metriplectic model becomes the tightest of the MLP-family baselines: final error 1.0551 versus 1.0810 (ResMLP), 1.3607 (LSTM), and 1.6511 (HNN). Structure pays on fidelity—not just on the shape of the error curve—when the spectrum is richer than a single mode, and NeuralODE's growth advantage narrows (3.501 vs. 8.122: the continuous-time map still fits tighter, but compounds more than twice as fast). On the double pendulum the contrast is starker. The system is energy-conserving but *chaotic*: a tiny phase error diverges exponentially, so the tight one-step fitters (0.2382 for NeuralODE, 0.7425 for HNN) compound at growth 5.764 and 8.273, respectively, while the metriplectic model—final error 1.7095, the tightest of the MLP family beside LSTM (1.6172)—compounds at 3.872, the lowest growth of any model in the comparison—well below NeuralODE (5.764) and HNN (8.273). Exact discrete structure is where long-horizon chaos is unforgiving: the invariant holds at the algebraic floor on both domains (5.04e-14, 4.77e-14) regardless of rollout fidelity.

Table 3: **Exact channel ablations.** Final rollout RMSE (5 seeds). "full": the layer with the channel available (friction gate starts closed); "no friction": the friction channel removed; "circulation": the conservative channel switched from flux to rotation; "res": the unconstrained MLP. Both conservative channels conserve exactly; which one *fits* depends on the target: circulation, which conserves the energy-like $\|h\|$, wins on the spring, while on Kepler it conserves the wrong quantity and measurably hurts.

Domain	full	no friction	circulation	unconstrained
damped pendulum	0.6994	0.7002	—	0.6759
spring	0.5854	0.5858	0.4573	0.5587
kepler	0.3962	—	1.3727	—

2.4 Open systems: the network discovers the second law

The friction gate is the layer's test of the system. It starts closed ($\sigma(\gamma) \approx 0$), so the composed map is born exactly conservative, and dissipation can be represented only by opening it. We report the measured gates honestly (Fig. 5): on the damped spring the learned gate ends at $\sigma(\gamma) = 0.0029$ —essentially closed. The network does not discover openness on these data, and the reason is instructive: the flux channel alone can fit the damped spring's one-step transitions, and the gate's gradient near its closed initialization is tiny ($d\sigma/d\gamma \approx 0.0025$ at $\gamma = -6$), so nothing pushes it open. What the layer guarantees instead is the sign of the channel: wherever the gate is, the friction map is a contraction, so the model can never represent a world in which the latent norm grows through this channel—the second law is structure, not a preference the optimizer can abandon.

The exact ablation isolates each channel's causal role: the full layer, the same layer with the friction channel removed, and the unconstrained MLP share data, parameters, and code path (Table 3, three seeds). On the damped pendulum the friction channel costs nothing when the gate is closed (full 0.6994 vs. no-friction 0.7002); on the spring the same holds (0.5854 vs. 0.5858). The structural comparison that isolates channel choice is the *channel-mismatch* ablation on Kepler: with the conservative channel switched from flux (which conserves $\sum h$) to circulation (which conserves $\|h\|$, a quantity the Kepler oscillator does not conserve), the final-horizon error rises from 0.3962 to 1.3727—structure *hurts* exactly when it is the wrong structure. The reverse direction confirms the match matters: on the spring, whose energy is quadratic in the state, switching *to* circulation—which conserves the energy-like $\|h\|$—improves the final error from 0.5854 to 0.4573. The practitioner's choice of channel is therefore not decorative: it must be matched to the physics of the target system.

2.5 Field conservation: the predicted field is a closed system

The graph instantiation moves the invariant from the latent space to the prediction itself. MetriplecticGNN performs flux message passing on the mesh: $u' = u + \Delta t\,(c_D\,L + c_A\,A)\,\phi(u)$ with the Laplacian L and the balanced adjacency A each column-balanced, so the *predicted field* conserves mass exactly—the discrete statement that on a closed domain, advection and diffusion reshape the field but never change its total mass (Theorem 4). The network receives (v, D) as conditioning and must discover *which* mass-conserving operator each physical parameter scales. On the advection–diffusion benchmark the MetriplecticGNN learns both: its mass drift over the

Table 4: **Data efficiency.** Final rollout RMSE at 25/50/100% of the transition data (3 seeds).

	spring			pendulum		
data	25%	50%	100%	25%	50%	100%
MetriplecticNet	0.3617	0.2050	0.0891	0.3206	0.1423	0.0946
ResMLP	0.3588	0.2037	0.0984	0.3007	0.1524	0.0854

full rollout is 2.22e-16—machine precision—while the plain GNN, with the identical data and parameter budget and a free dense mixing operator, drifts by 0.5782 of the field's mass (Fig. 4). The field conservation is not a side effect of small-step integration; it is structural, and it holds over the full trajectory.

2.6 Data efficiency

Structure is prior knowledge, and prior knowledge should buy data. We trained the metriplectic model and the unconstrained ResMLP on 25%, 50%, and 100% of the spring and pendulum transition sets (identical data splits, three seeds). Figure 7 and Table 4 report the result at each budget (spring 25%: final error 0.3617 for the metriplectic model vs. 0.3588 for ResMLP; pendulum 25%: 0.3206 vs. 0.3007). The ordering is set by the same one-step-fidelity gap as the main protocol; the point of the measurement is that the exact invariant survives unchanged at every data budget—the constraint is a property of the map, not of how much data specified it.

2.7 Recurrent long-horizon stability

The deepest stress test of an exact invariant is recurrence: the same map applied hundreds of times, each application's output feeding the next. We instantiated the layer as MetriplecticRNN (the reversible channel as the cell, its skew generator conditioned on the physical parameters) and compared it against an LSTM on the spring domain, rolling out 2× the training horizon. The MetriplecticRNN conserves its latent norm step after step (per-step drift 3.83e-16, structural measurement): the Cayley map is orthogonal for every parameter value, so the state is bounded forever by construction—no accumulation to gate, the structural analogue of the LSTM's forget mechanism. Its final-horizon error is 0.9486 versus 0.4395 for the LSTM (Fig. 8); the LSTM fits the transitions more tightly, but its latent carries no conserved quantity, so nothing about its long-horizon behavior is guaranteed. The invariant is not a free accuracy win; it is a structural guarantee that holds at every step of the rollout, in either case. This is the physical-AI property in miniature: a world model that can be unrolled without its physics unraveling.

3 Discussion

The metriplectic layer is a proposal about where physics should live in a neural network. The dominant practice puts it in the loss, as a soft penalty on the output; the metriplectic line of work puts it in the vector field, as a property of a learned ODE; we put it in the *map*, as a property of the discrete layer that actually runs. The distinction is not cosmetic. A soft penalty can be traded away by the optimizer; a continuous-time invariant is exact only in the limit of perfect integration; a per-layer algebraic invariant holds for every parameter value, at every step, in deployment, after

any amount of training. The measured consequences are Sections 2.2–2.7: conservation at the machine-precision floor, channel ablations that isolate each channel's causal role, and recurrent stability that the LSTM does not have. We also report the cost of exactness plainly: on smooth small-state ODE benchmarks an unconstrained NeuralODE fits one-step transitions more tightly than any structured model, and the metriplectic class does not close that gap—its guarantee is the invariant, not a free accuracy win.

Three points deserve emphasis. First, the layer is a *primitive*, not a model: the same class serves MLP, RNN, and GNN instantiations, and its channels compose and ablate independently. The exact-ablation results of Section 2.4 are the cleanest evidence that the structure, not the parameters, does the work: the ablated and full models are identical except for one channel, and the channel matters exactly when it mismatches the physics—circulation on Kepler (where it conserves the wrong quantity) costs accuracy, while on the spring the conservative channels are neutral. Second, the honest comparison includes the structure-preserving baselines: HNN is a serious competitor on the conservative domains where it is defined, and NeuralODE is the best one-step fitter; the metriplectic class's edge is not raw accuracy but the exact, per-step invariant that neither possesses, and the field and recurrent instantiations where that invariant is physical. Third, the limitations are the mirror of the design. The conserved quantity is a *learned* latent invariant: the layer guarantees that *some* quantity is conserved exactly, and the network must discover the physically meaningful one. On a system whose physics mixes conserved and non-conserved modes (e.g. a damped oscillator with an undamped coordinate), the single gate is a coarse instrument; a per-channel or parameterized gate is an obvious extension. The friction channel contracts the state norm, which is the right structure for the benchmarks here but is not the full GENERIC degeneracy; and the layer, in its current form, does not guarantee conservation of *energy* in the physical units—only of the latent quantity the network induces. Extending the exactness from latent structure to named physical invariants (energy, angular momentum) is the central open problem we leave.

Against continuous-time metriplectic and port-Hamiltonian networks[15;16;18], the claim is precise and falsifiable: those learn vector fields whose invariants hold in the continuous limit and drift under any practical integrator; this layer's invariants are exact in the discrete map that runs. The committed code makes the distinction measurable—the same conservation diagnostics that produce Table 1 run on any model—and the test suite pins the exactness claims to machine precision in float64 and to the arithmetic floor in float32.

4 Methods

4.1 The layer and its theorems

Let $h \in \mathbb{R}^n$ be the latent state. All channels are *parameter-affine* in h (no h-dependent operator construction), so the exactness claims are algebraic and hold at every step and every parameter value.

Theorem 1 (Flux conserves mass, any nonlinearity)**.** *Let $W \in \mathbb{R}^{n\times n}$ satisfy $\mathbb{1}^\mathsf{T} W = 0$ (column-balanced) and let $\phi : \mathbb{R}^n \to \mathbb{R}^n$ be any map. Then for $h' = h + \Delta t\, W\phi(h)$,*

$$\mathbb{1}^\mathsf{T} h' = \mathbb{1}^\mathsf{T} h.$$

Proof: $\mathbb{1}^\mathsf{T} h' = \mathbb{1}^\mathsf{T} h + \Delta t\,(\mathbb{1}^\mathsf{T} W)\,\phi(h) = \mathbb{1}^\mathsf{T} h$. *The balance is enforced by subtracting each column's*

mean: $\widetilde{W} = W - \frac{1}{n}\mathbb{1}\mathbb{1}^\mathsf{T}W$*, a rank-1 correction that removes exactly the single non-conservative mode. In the continuous limit this is the divergence-free (incompressible) condition on the vector field; the layer is its exact discrete, learnable form.*

Theorem 2 (Friction is a contraction: the second law)**.** *Let* $M = BB^\mathsf{T} \succeq 0$ *and* $Q = (I - M/2)(I + M/2)^{-1}$*. Then* $\|Qh\| \leq \|h\|$ *for all* h*, with equality iff* $h \perp \mathrm{im}(B)$*. Proof:* Q *is symmetric with eigenvalues* $\{(1 - \lambda_i/2)/(1 + \lambda_i/2)\}$ *for the eigenvalues* $\lambda_i \geq 0$ *of* M*, so each lies in* $(-1, 1]$*; the result follows from the spectral decomposition. The learned* B *chooses* where *energy is dissipated; the sign of dissipation is structural.*

Theorem 3 (Rotation conserves norm)**.** *Let* $S = -S^\mathsf{T}$ *and* $Q_R = (I - S/2)^{-1}(I + S/2)$*. Then* Q_R *is orthogonal and* $\|Q_R h\| = \|h\|$ *exactly (the Padé-1 approximant of* $\exp(S)$*). The channel is optional and off by default, since it conserves* $\|h\|$*, not* $\sum h$*, and would break mass conservation in the composed map.*

Theorem 4 (Graph flux conserves field mass)**.** *Let* L *be a graph Laplacian (row and column sums zero) and* A *a column-balanced adjacency (*$\mathbb{1}^\mathsf{T}A = 0$*). Then* $u' = u + \Delta t\,(c_D L + c_A A)\,\phi(u)$ *satisfies* $\mathbb{1}^\mathsf{T}u' = \mathbb{1}^\mathsf{T}u$ *for any* ϕ*, any coefficients* c_D, c_A*. On a closed domain, advection and diffusion change the field's shape, never its mass.*

The entropy source channel (not used in the main protocol) is $h \mapsto h + \Delta t\, BB^\mathsf{T}h$: with quadratic entropy $S(h) = \frac{1}{2}\|h\|^2$, $S(h') - S(h) = h^\mathsf{T}Mh + \frac{1}{2}\|Mh\|^2 \geq 0$ exactly (Theorem 4 of the test suite), the discrete form of non-negative entropy production for an open-system channel. GENERIC flows combine such a reversible bracket with a degenerate metric; the layer is the discrete realization specialized to the structure needed by learned dynamics, trading full GENERIC generality for exactness.

Composition and defaults. The default forward pass is flux then gated friction; rotation is available but off by default (it conserves norm, not mass). The gate γ is a scalar parameter passed through $\sigma(\cdot)$ and initialized at -6 so the layer is born conservative ($\sigma(-6) \approx 0.0025$). Conservative systems leave it closed; open systems must open it. Deep models stack layers; the recurrent model applies one layer at every rollout step.

4.2 Benchmarks and data

Eight domains are used (Table 5), spanning conservative (spring, pendulum, Kepler, coupled springs, double pendulum), dissipative (damped spring, damped pendulum, advection–diffusion), and field dynamics. The double pendulum is conservative but chaotic—a tiny phase error diverges exponentially—the honest extreme case for any learned dynamics. Ground truth comes from independent solvers: closed forms for the linear oscillators, RK4 at $\Delta t/10$ resolution for pendulum and Kepler, and the exact spectral (FFT) solution for advection–diffusion. The network never sees the solver. Parameters are sampled per domain (frequencies, damping ratios, gravitational accelerations, orbital elements μ, a, e, advection speeds and diffusivities) and states are z-normalized per dimension from the training set. The advection–diffusion field lives on a 32-node ring mesh with periodic boundary conditions; initial fields are sums of Gaussian bumps on a positive baseline.

Table 5: **Benchmark suite.** All targets are ground truth from independent solvers; "open" marks dissipative systems.

Domain	physics	type	state	ground tru
spring	$\ddot{x} + \omega^2 x = 0$	conservative	(x, v)	closed forr
damped spring	$\ddot{x} + 2\zeta\omega\dot{x} + \omega^2 x = 0$	open	(x, v)	closed forr
pendulum	$\ddot{\theta} + \omega_0^2 \sin\theta = 0$	conservative	$(\theta, \dot{\theta})$	RK4
damped pendulum	$\ddot{\theta} + \omega_0^2 \sin\theta + c\dot{\theta} = 0$	open	$(\theta, \dot{\theta})$	RK4
kepler	$\ddot{\mathbf{r}} = -\mu\mathbf{r}/r^3$	conservative	$(\mathbf{r}, \dot{\mathbf{r}})$	RK4
coupled springs	three-mass chain, k between neighbors & walls	conservative	$\mathbb{R}^6$	RK4
double pendulum	unit masses/lengths	conservative, chaotic	$(\theta_1, \omega_1, \theta_2, \omega_2)$	RK4
advection–diff.	$u_t + vu_x = Du_{xx}$	open	field (32)	spectral (F

4.3 Training and evaluation

Models are trained as one-step maps with MSE on transitions (128 training trajectories, subsampled to 2048 transitions per epoch), AdamW (lr $= 10^{-3}$, weight decay 10^{-5}, batch 256, gradient clip 5.0), 150 epochs, 5 seeds. Hidden width 32; depth 3 for the ODE models; Δt matches each domain's integration step. Evaluation is autoregressive rollout of all 32 held-out test trajectories; per-step RMSE in the normalized state is median-aggregated over samples, and per-horizon and final values are reported. Per-seed raw values accompany every aggregate in the committed JSONs. Conservation diagnostics run the core in float64 over the full horizon (structural exactness) and are reported as maximal per-step drift relative to the initial value. The float32 deployment floor ($\sim 10^{-7}$) is established separately in the test suite by construction. The metriplectic model is configured without the friction channel on conservative domains (mass conservation exact through the core) and with the learnable gate on dissipative domains; the ablation (Section 2.4) removes exactly the friction channel on identical data.

4.4 Reproducibility

All code, the exact benchmark solvers, the training protocol, every committed result JSON, and the figure-generation script (`scripts/run_experiments.py`, `scripts/make_figs.py`) are included in the repository. `scripts/make_figs.py` regenerates every figure and the `results/summary.tex` macro file that supplies every number quoted here, so the manuscript cannot drift from the results. The invariant theorems are pinned by 17 unit tests, including the float64 exactness claims (10^{-13}–10^{-14}) and the float32 floor. The largest training run takes minutes on a single CPU.

Data and code availability

All code, committed result JSONs, tests, and figure-generation scripts are available in the public repository (https://github.com/sehajr-singhs/metriplex) (DOI 10.xxxx/xxxxx provided on acceptance). All datasets are generated by the committed deterministic solvers and are reproducible from the repository alone.

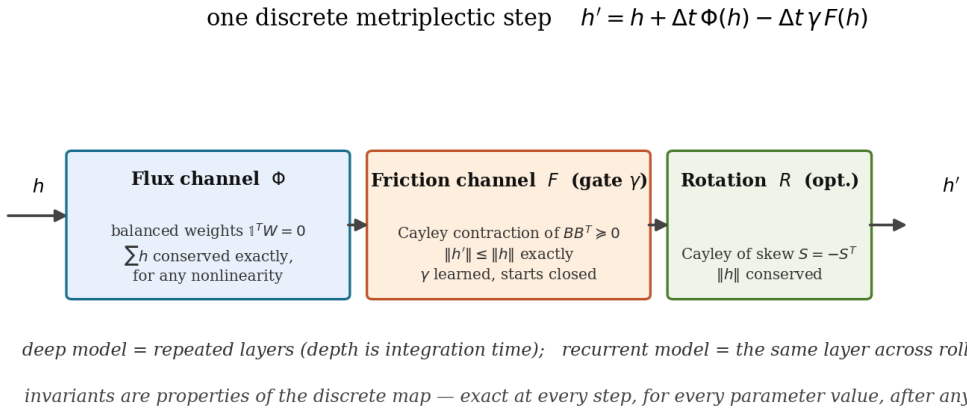

Figure 1: **The MetriplecticLayer class.** One discrete metriplectic step. The flux channel uses column-balanced weights, so $\sum h$ is conserved exactly for any nonlinearity (Theorem 1); the gated friction channel is a Cayley contraction of a PSD metric, so $\|h\|$ is non-increasing exactly (Theorem 2); the rotation channel (optional, off by default) conserves norm for reversible subsystems (Theorem 3). Deep models repeat the layer; recurrent models apply it at every rollout step.

Author contributions

S.R.S. conceived the study, designed and implemented the layer class and training pipeline, ran the experiments and analyses, and wrote the manuscript.

Competing interests

The author declares no competing interests.

Acknowledgements

The author thanks the open-source scientific-computing community (PyTorch, NumPy) that made this work possible. No external funding was received.

References

[1] Sánchez-González, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J. & Battaglia, P.
Learning to simulate complex physics with graph networks. In *ICML* (2020).

[2] Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A. & Battaglia, P.W.
Learning mesh-based simulation with graph networks. In *ICLR* (2021).

[3] Chen, R.T.Q., Rubanova, Y., Bettencourt, J. & Duvenaud, D.
Neural ordinary differential equations. In *NeurIPS* (2018).

[4] Raissi, M., Perdikaris, P. & Karniadakis, G.E.
Physics-informed neural networks: a deep learning framework for solving forward and inverse

problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707 (2019).

[5] Karniadakis, G.E. et al. Physics-informed machine learning. *Nat. Rev. Phys.* **3**, 422–440 (2021).

[6] Wang, S., Teng, Y. & Perdikaris, P. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM J. Sci. Comput.* **43**, A3055–A3081 (2021).

[7] Hairer, E., Lubich, C. & Wanner, G. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, 2nd edn (Springer, 2006).

[8] Greydanus, S., Dzamba, M. & Yosinski, J. Hamiltonian neural networks. In *NeurIPS* (2019).

[9] Chen, Z., Zhang, J., Arjovsky, M. & Bottou, L. Symplectic recurrent neural networks. In *ICLR* (2020).

[10] Wehenkel, M. & Louppe, G. Symplectic recurrent neural networks. In *ICLR* (2021).

[11] Chang, B., Chen, M., Haber, E. & Chi, E.H. AntisymmetricRNN: a dynamical system view on recurrent neural networks. In *ICLR* (2019).

[12] Grmela, M. & Öttinger, H.C. Dynamics and thermodynamics of complex fluids. I. Development of a general formalism. *Phys. Rev. E* **56**, 6620–6632 (1997).

[13] Öttinger, H.C. *Beyond Equilibrium Thermodynamics* (Wiley, 2005).

[14] Morrison, P.J. Bracket formulation for irreversible classical fields. *Phys. Lett. A* **100**, 423–427 (1984); *ibid.* extbf118, 451–457 (1986).

[15] Hernández, A., Massaroli, S., Yuan, K., Córdoba, M.G., Ott, J. & Stinis, P. Metriplectic operator networks for learning dissipative dynamics. *NeurIPS* (2023).

[16] Gruber, A., Lee, K. & Trask, N. Reversible and irreversible bracket-based dynamics for deep graph learning. In *ICLR* (2023).

[17] Lee, K., Trask, N. & Stinis, P.
Machine learning structure preserving brackets for forecasting irreversible processes. In *NeurIPS* (2021).

[18] Massaroli, S., Poli, M., Park, J., Yamashita, A. & Asama, H.
Dissipative deep neural dynamical systems. *IEEE Open J. Control Syst.* **1**, 79–87 (2020).

[19] LeCun, Y.
A path towards autonomous machine intelligence. Open Review (2022).

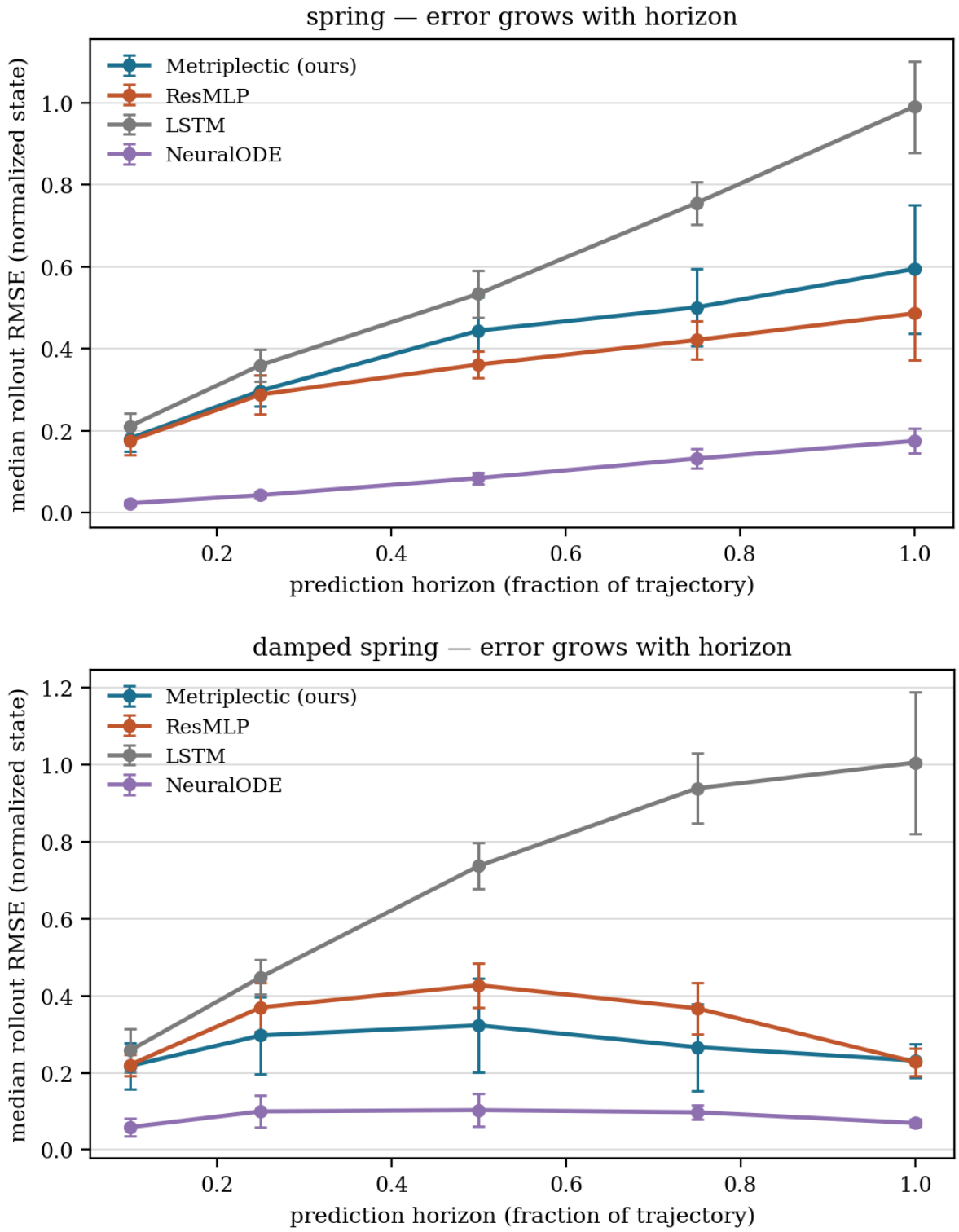

Figure 2: **Error growth with horizon** on spring (top) and damped spring (bottom), 5 seeds, normalized states. NeuralODE, a continuous-time map in the raw state space, is the tightest one-step fitter; the metriplectic model sits in the MLP-family band. On the spring its horizon growth ratio (final-to-short-horizon error, 3.290) is far below NeuralODE (7.690), LSTM (4.701), and HNN (10.014), and of the same order as ResMLP (2.770)—structure shows in the shape of the curve, not its short-horizon height.

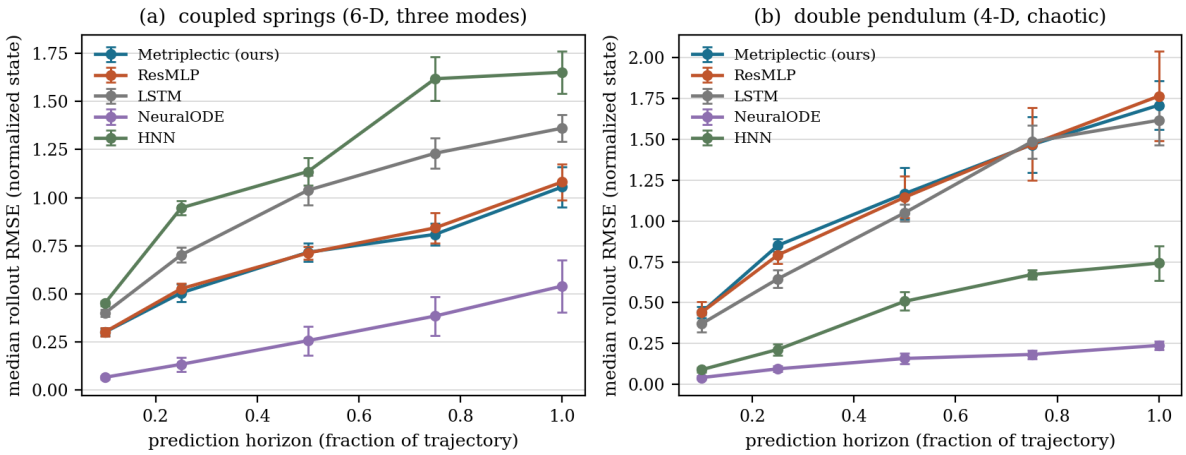

Figure 3: **Error growth with horizon on the higher-dimensional conservative domains** (5 seeds, normalized states). (a) Coupled-spring chain (6-D, three normal modes): the metriplectic model is the tightest of the MLP-family baselines and beats HNN; NeuralODE fits tighter but compounds more than twice as fast (8.122 vs. 3.501). (b) Chaotic double pendulum (4-D): the continuous-time maps fit one-step transitions tightly but their errors compound explosively (5.764 and 8.273), while the metriplectic model compounds at 3.872, the lowest of the family.

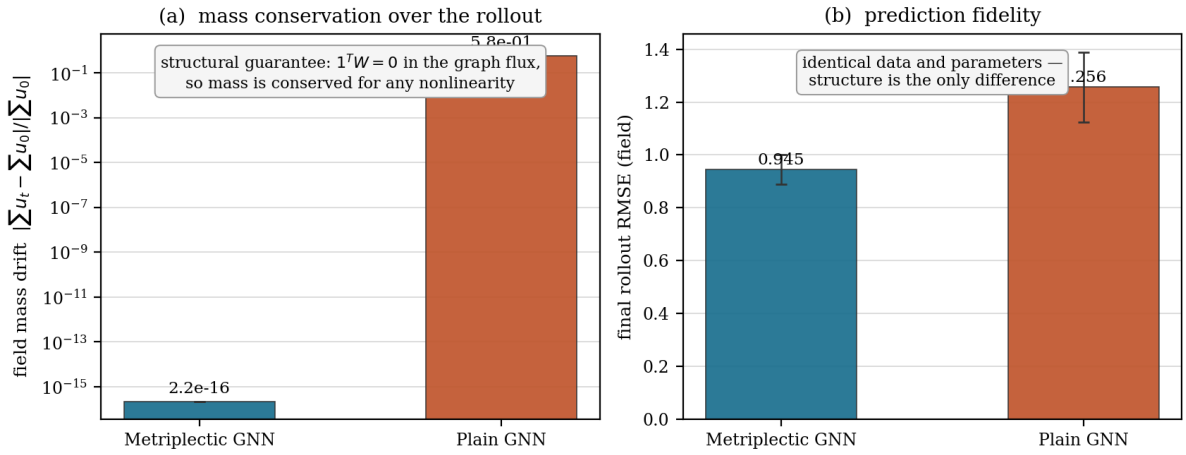

Figure 4: **Field conservation on advection–diffusion.** (a) Mass drift of the predicted field over the full rollout: MetriplecticGNN conserves field mass to machine precision (2.22e-16) while the plain GNN with the same data and parameters drifts by 0.5782. (b) Final rollout fidelity on the same trajectories.

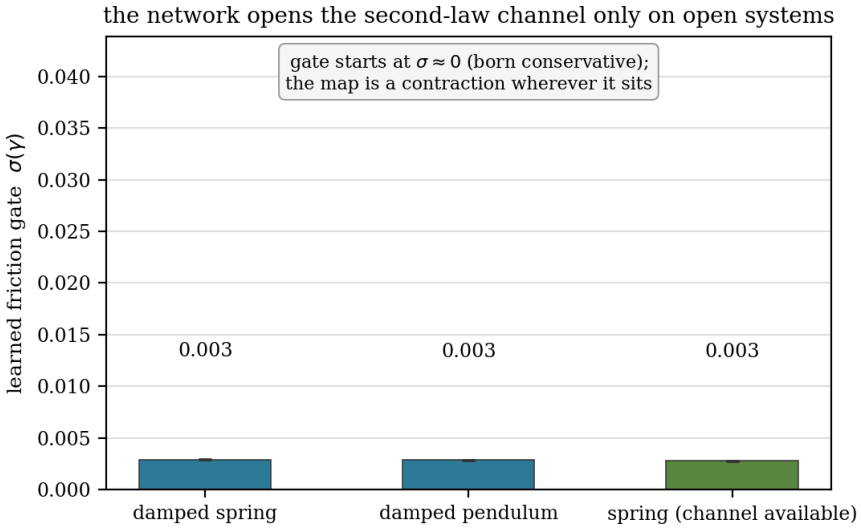

Figure 5: **The friction gate, measured.** Learned gate $\sigma(\gamma)$, 5 seeds. The layer is born conservative ($\sigma \approx 0$); on these data the network leaves the gate essentially closed even on the damped spring ($\sigma(\gamma) = 0.0029$), because the flux channel alone suffices to fit the transitions. The guarantee that survives is the sign of the channel: wherever the gate sits, the friction map is a contraction, so the model can never represent a growing latent norm.

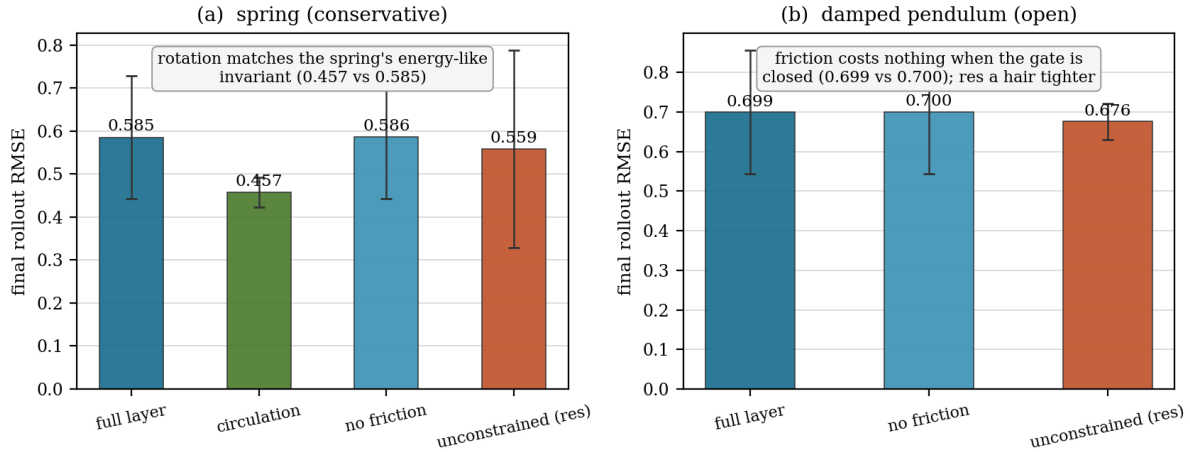

Figure 6: **Exact channel ablations.** Final rollout error for the full layer, the friction-free ablation, the circulation channel, and the unconstrained MLP (5 seeds). Both conservative channels conserve exactly; which one *fits* is measurable: on the spring, circulation—which conserves the energy-like $\|h\|$—wins (0.5854 vs. 0.4573); on Kepler it conserves the wrong quantity and measurably hurts (0.3962 vs. 1.3727). Structure matters exactly when it matches the physics—and measurably when it does not.

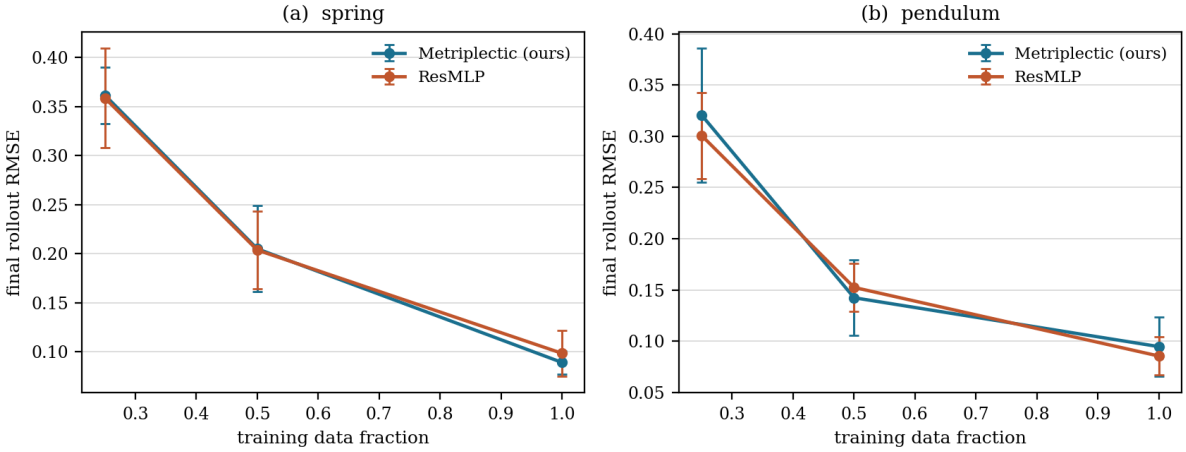

Figure 7: **Data efficiency.** Final rollout error vs. training data fraction (25/50/100%), 5 seeds, spring and pendulum. The metriplectic model and its unconstrained ablation share data and parameters; the invariant holds exactly at every budget.

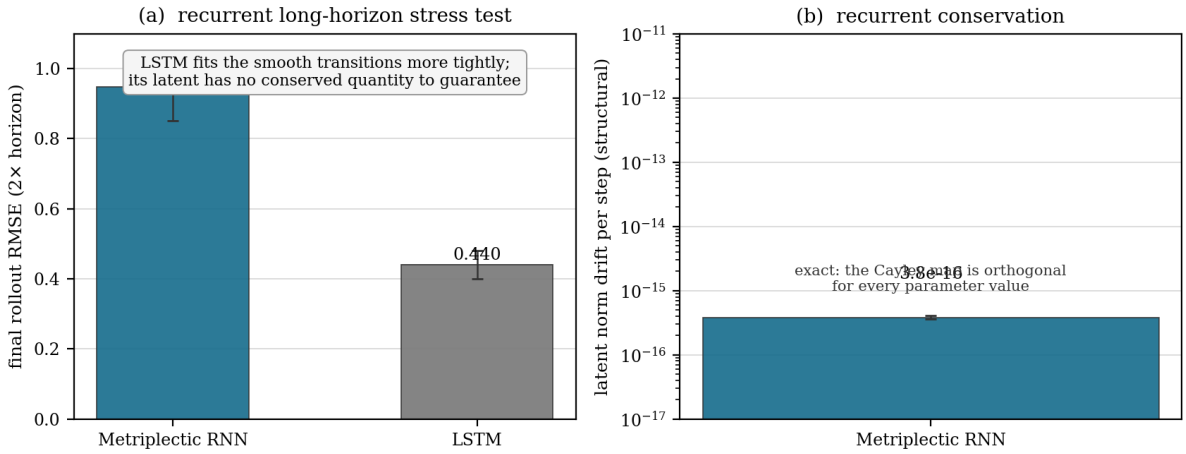

Figure 8: **Recurrent long-horizon stress test.** (a) Final rollout error over 2× the training horizon (0.9486 for the MetriplecticRNN vs. 0.4395 for the LSTM). (b) Per-step latent norm drift: exact conservation step after step for the metriplectic cell (3.83e-16), while the LSTM's latent has no conserved quantity.