MPINeuralODE: Multiple-Initial-Condition Physics-Informed Neural ODEs for Globally Consistent Dynamical System Learning

Lake Yang [1] **Antonio Malpica-Morales** [1] **Frank Ioannis Papadakis Wood** [1] **Serafim Kalliadasis** [1]

Abstract

Neural ordinary differential equations (Neural ODEs) often fit training trajectories while generalizing poorly to unseen initial conditions and long horizons. We propose MPINeuralODE, which combines a soft physics-informed residual with a Multiple-Initial-Condition (MIC) multiple-shooting curriculum whose ingredients are structurally complementary: the physics term anchors the vector-field magnitude on the support that MIC enlarges. We evaluate along three axes — out-of-sample error, long-horizon stability, and Hamiltonian drift — that together expose whether the learned dynamics have recovered the true vector field. On Lotka–Volterra, MPINeuralODE achieves the lowest out-of-sample and long-horizon MSE among data-driven methods (a 24% reduction over the baseline Neural ODE) and the lowest Hamiltonian drift, making it the strongest method across all three axes. Code and reproduction artifacts are available at https://github.com/LakeYang0818/MPINeuralODE.

1. Introduction

Dynamical systems describe how state variables evolve under internal interactions and external forcing, from predator–prey ecology (Murray, 2002) to chemical kinetics and geophysical fluid flow. A persistent tension in data-driven modeling is the tradeoff between mechanistic interpretability and empirical flexibility. Classical mechanistic models (Strogatz, 2015) provide identifiable rate constants but depend on assumptions — well-mixed populations, linear functional responses, absence of unmodeled dynamics — that are routinely violated in practice. Purely data-driven approaches (Voss et al., 2004; Brunton et al., 2016; Pathak et al., 2018) offer representational capacity but sacrifice interpretability and often generalize poorly outside the training distribution.

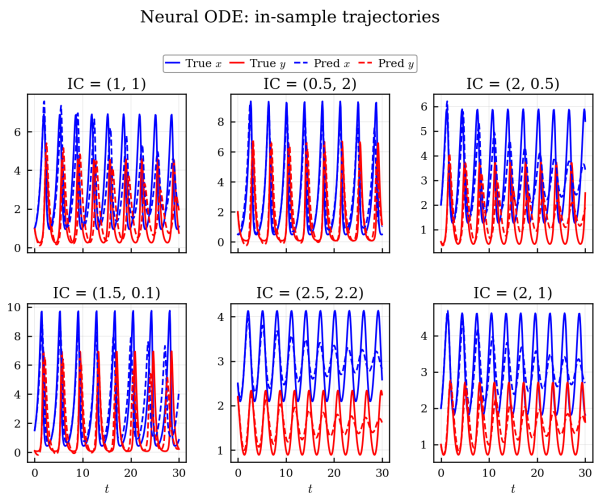
Neural ordinary differential equations (Neural ODEs) (Chen et al., 2018) parameterize the instantaneous vector field with a neural network and obtain trajectories by numerical integration. The formulation handles irregularly-sampled data (Rubanova et al., 2019), produces smooth solver-controlled trajectories, and supports training through the adjoint method (Bradley, 2013). Yet our experiments confirm earlier observations (Krishnapriyan et al., 2021; Wang et al., 2021) that pure Neural ODEs can learn vector fields valid only near training trajectories. Figure 1 illustrates this: on the Lotka–Volterra system, the baseline Neural ODE fits training trajectories well, but on held-out initial conditions it produces artificial spirals where the true dynamics admit conservative closed orbits.

This paper investigates two complementary remedies and their combination. Soft *physics-informed regularization* (Raissi et al., 2017; 2019) steers the learned vector field toward physically plausible behavior at sampled collocation points; this construction shares its variational structure with the classical optimal-control formulations of Pontryagin and Bellman (Pontryagin et al., 1962; Sidhoum & Oguri, 2025). *Multiple-Initial-Condition multiple-shooting* (Bock & Plitt, 1984; Turan & Jäschke, 2022) broadens phase-space coverage and enforces continuity of the learned flow across segments. A central methodological observation is that a single scalar metric — such as validation MSE — does not reveal whether a learned model has recovered the true dynamics or merely fit observed trajectories. We evaluate every method on three complementary axes:

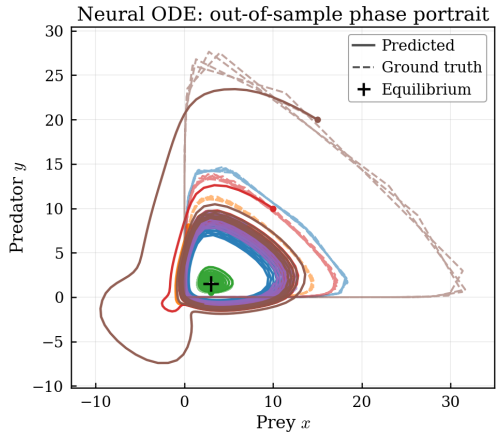
1. **Out-of-sample (OOS) prediction error** — accuracy on held-out initial conditions in the training distribution.
2. **Long-horizon stability** — accumulated error over multiple oscillation periods, exposing compounding error.
3. **Drift in the conserved Hamiltonian** $H(x, y)$ — a

[1]Department of Chemical Engineering, Imperial College London, London SW7 2AZ, United Kingdom. Correspondence to: Lake Yang <l.yang1@imperial.ac.uk>, Antonio Malpica-Morales <a.malpica-morales21@imperial.ac.uk>, Frank Ioannis Papadakis Wood <frank-ioannis.papadakis-wood21@imperial.ac.uk>, Serafim Kalliadasis <s.kalliadasis@imperial.ac.uk>.

Proceedings of the AI4Physics Workshop at the 43rd International Conference on Machine Learning (AI4Physics@ICML 2026), Seoul, South Korea. 2026. Copyright 2026 by the author(s).



(a) In-sample time series: the baseline NN fits training trajectories accurately across all six panels.



(b) Out-of-sample phase portrait: predicted orbits (solid) spiral and cross axes rather than tracing the ground-truth closed orbits (dashed).

Figure 1. **Baseline Neural ODE failure modes on Lotka–Volterra.** Accurate in-sample fit **(a)** coexists with structurally broken out-of-sample behavior **(b)**: the learned vector field is valid only in the training corridor and produces qualitatively wrong dynamics elsewhere. The closed-orbit topology of the true system is replaced by artificial spirals.

trajectory-independent measure of whether the learned dynamics respect the geometric structure of phase space.

A model that performs well on all three has captured something close to the true vector field; a model that excels only on OOS error may still drift over long horizons or dissipate invariants.

Contributions. (1) We propose MPINeuralODE, a Neural ODE framework whose physics residual and MIC continuity penalty are mutually compensating on the visited support of phase space rather than a weighted sum of independent objectives. (2) We argue that single-scalar trajectory error is insufficient for evaluating learned dynamical systems and propose a three-axis report that exposes failure modes a scalar MSE hides. (3) On Lotka–Volterra, MPINeuralODE achieves the lowest OOS and long-horizon error among purely data-driven methods and the lowest Hamiltonian drift — the strongest method across all three axes. No axis is worse than any single ingredient alone. In the typical practitioner regime where exact equations are unavailable, this makes it the most practical surrogate among the methods we evaluated. (4) We release MPINeuralODE as an installable Python package shipping ready-to-use `LotkaVolterra`, `Lorenz63`, and `FitzHughNagumo` subclasses.

2. Related Work

Neural ODEs and variants. Neural ODEs (Chen et al., 2018) parameterize a continuous-time vector field with a neural network. Latent ODEs (Rubanova et al., 2019), Augmented Neural ODEs (Dupont et al., 2019), and Neural Controlled Differential Equations (Kidger et al., 2020) address architectural limitations (partial observability, expressivity, exogenous control) but not global consistency across initial conditions, which is our target.

Physics-informed losses. Physics-Informed Neural Networks (Raissi et al., 2017; 2019) enforce soft constraints through collocation sampling but can require extensive sampling and struggle with long-horizon prediction (Krishnapriyan et al., 2021; Wang et al., 2021). We integrate the PINN idea into the Neural ODE framework, reducing collocation requirements by sampling along the integrated trajectory rather than over a phase-space grid.

Multiple shooting. The direct multiple-shooting method originated in numerical optimal control (Bock & Plitt, 1984) to overcome the phase-mismatch pathology of single-shooting on long-horizon oscillatory systems. Turan & Jäschke (2022) first applied this idea to neural differential equations, showing it can rescue the "flattened-out" trajectories single-shooting produces on oscillatory time series. Our MIC mechanism is a multiple-shooting curriculum in that lineage, augmented with a freshly-sampled initial-condition distribution each epoch and combined with a PINN-style residual that these works do not address.

Structure-preserving networks. Hamiltonian Neural Networks (Greydanus et al., 2019), related Hamiltonian-system identification (Bertalan et al., 2019), and symplectic networks (Jin et al., 2020) build conservation laws into the architecture itself; this approach is orthogonal to and potentially combinable with ours.

Mechanistic-neural hybrids. Universal Differential Equations (Rackauckas et al., 2020) embed a mechanistic model with a neural residual, achieving strong performance when mechanistic knowledge is complete; we use this construction as an *oracle reference*. Data-driven discovery methods (Brunton et al., 2016; Champion et al., 2019) excel when the true model lies in a known function library but struggle when the functional form is unknown — the setting MPINeuralODE targets.

3. Method

3.1. Lotka–Volterra Benchmark

The Lotka–Volterra (LV) system (Lotka, 1925; Volterra, 1926)

$$\frac{dx}{dt} = \alpha x - \beta xy, \quad \frac{dy}{dt} = \delta xy - \gamma y \tag{1}$$

provides a mathematically tractable benchmark with three useful properties. First, it exhibits neutrally-stable closed orbits with no inherent error dissipation: the coexistence equilibrium at $(\gamma/\delta,\ \alpha/\beta)$ has purely imaginary eigenvalues, indicating a center (Strogatz, 2015). Any artificial damping or drift therefore accumulates rather than decays. Second, families of orbits at different amplitudes share the same phase plane, requiring the learner to generalize across orbits. Third, the system admits the conserved quantity (Murray, 2002)

$$H(x, y) = \delta\, x - \gamma \ln x + \beta\, y - \alpha \ln y, \tag{2}$$

constant along true trajectories. Drift in $H(t)$ along a learned trajectory is therefore a trajectory-independent diagnostic of how well the model respects the geometric structure of phase space — any deviation from a constant is unambiguously a model error and does not depend on which initial condition we picked. We adopt $(\alpha, \beta, \gamma, \delta) = (1.5, 1.0, 3.0, 1.0)$, giving period $T \approx 5.13$ and interior equilibrium $(3.0, 1.5)$.

Data generation. Ground-truth trajectories are integrated with adaptive Dormand–Prince (Dormand & Prince, 1980) at rtol $=$ atol $= 10^{-8}$ over $t \in [0, 30]$ with $\Delta t = 0.1$, yielding 301 time points per trajectory (~ 5.85 cycles). Initial conditions are drawn from a mixed distribution: half uniform on $[0.1, 10.0]^2$ (typical regime) and half log-uniform on $[10^{-3}, 10.0]^2$ (edge regime near the extinction axes). This mixed scheme avoids pathological behavior near coordinate axes.

3.2. Phase-1 Setup: Strongest Neural-ODE Configuration

Three implementation choices — network architecture, temporal sampling, and positivity wrapper — are decided up front by a sequential ablation against a composite metric (log-space mean of in-sample, OOS, and long-horizon MSE) using only the baseline Neural ODE variant. This decouples them from any physics-loss or MIC effect. The empirical winners are summarized in Table 1; per-variant diagnostics are in Sections B to D.

The three choices jointly define what we call the *strongest Neural-ODE configuration we identified* for this benchmark. Every method in this paper — LV_NN, LV_PINN, LV_MIC, MPINeuralODE, and LV_Structured — adopts these choices. This is deliberate: by fixing the implementation details to the best Neural-ODE setup we found, any observed difference between methods reflects only the algorithmic ingredient under test (physics loss, MIC curriculum, or both), not a confounding hyperparameter choice.

Temporal sampling. We compared the *full-window* scheme (each sampled trajectory integrated over $[0, t_{\text{train}}]$), an *expanding-window* curriculum (Bengio et al., 2009) that grows the horizon over the first 75% of training, and *sliding windows* of fixed length 6.0 at random offsets. The full-window scheme achieves the lowest composite metric: paying the long-horizon cost from epoch one finds a vector field consistent over the full horizon throughout training, whereas the curriculum variants leave the long-horizon regime under-trained when they finally expose it.

Positivity wrapper. LV populations satisfy $x, y > 0$, but the learned vector field has no built-in reason to respect this. We evaluated four wrappers on the raw output r of the vector-field network: *none* (identity), *tanh-bound* ($18\tanh(r/18)$, a smooth saturating function), *squared* ($r\,|r|$), and *clamp* ($\text{clip}(r, -20, 20)$). Clamp and none *tie* on the composite metric to ten digits (Section D): at width 128 the raw output rarely approaches the clamp bound, so clamp acts as the identity on most forward passes. We adopt clamp for principled safety — identical expected behavior, with a deterministic safety net against the rare blow-up that none cannot prevent. The tanh-bound and squared wrappers underperform either by saturating useful signal (tanh-bound) or distorting the derivative scale near zero (squared, where $\partial(r|r|)/\partial r = 2|r|$ vanishes at the origin).

3.3. Neural ODE Baseline

The Neural ODE (Chen et al., 2018) parameterizes the vector field as $d\mathbf{z}/dt = f_\theta(\mathbf{z})$ and obtains trajectories by solving the corresponding initial value problem. Training mini-

Table 1. Phase-1 ablation winners. Positivity row reports a tie between *clamp* and *none* on the composite metric; we adopt clamp as a deterministic empirical-cutoff safety net.

Choice	Winner	Variants compared
Architecture	128×4 tanh	widths $\{32, \ldots, 256\}$, depths $\{2, \ldots, 6\}$
Temporal sampling	full window	full / expanding / sliding
Positivity wrap	clamp $\equiv$ none	none / tanh-bound / squared / clamp

mizes trajectory MSE

$$L_{\text{data}} = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{z}_{\text{pred}}(t_i) - \mathbf{z}_{\text{true}}(t_i)\|_2^2 \tag{3}$$

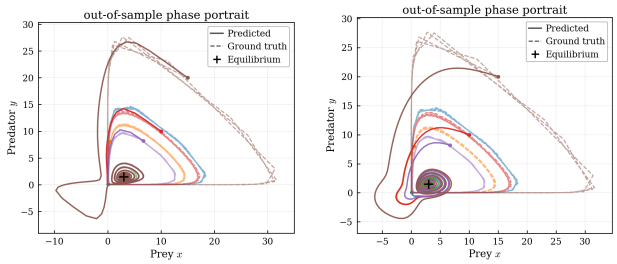
via the adjoint method (Pontryagin et al., 1962; Chen et al., 2018) with Adam optimization (Kingma & Ba, 2015). The vector-field network has four tanh hidden layers of 128 units each (the Phase-1 winner, ∼67k parameters; Section B) with Xavier-normal initialization (Glorot & Bengio, 2010). Integration uses adaptive Dormand–Prince (Dormand & Prince, 1980; Hairer et al., 1993); gradient norms are clipped at 10 to prevent pathological spikes from the adjoint pass through the solver (Goodfellow et al., 2016). A schematic of how this baseline extends to the full MPINeuralODE training step is shown in Figure 3.

3.4. Intermediate Extensions: PINN and MIC

We evaluate each ingredient of MPINeuralODE in isolation under the Phase-1 configuration of Section 3.2, so observed differences isolate the algorithmic contribution.

LV_PINN. Augments the data loss with a soft physics term that penalizes deviation of the learned vector field from the known f_{LV}, evaluated symmetrically on states the model visits and corresponding ground-truth states. Because the residual is only *applied* where samples exist, its effect is concentrated in whatever corridor of phase space the integrated rollouts actually traverse during training. Mechanistically, the residual directly shapes $f_\theta(\mathbf{z})$ to approximate the true $f_{\text{LV}}(\mathbf{z})$ pointwise, so orbits inherit the physical structure of the true system — closed orbits, correct period, conservation of H — wherever the trajectory visits. Outside that visited corridor, f_θ remains free to drift.

LV_MIC. Draws a larger batch of $m > n$ initial conditions per epoch from the mixed sampler of Section 3.1, then splits each trajectory into $K = 4$ segments and applies the direct multiple-shooting method (Bock & Plitt, 1984) along the time axis. Segment $k > 0$ starts from the detached predicted endpoint of segment $k - 1$; a continuity penalty pulls the predicted endpoint of segment $k - 1$ toward the *true* state at the next segment's boundary, which is the soft-penalty form of the multiple-shooting continuity constraint advocated by Turan & Jäschke (2022) for neural differential



(a) LV_PINN: drift suppressed in the training corridor, but outer orbits remain under-constrained. *(b)* LV_MIC: orbital topology preserved across both inner and outer trajectories, but vector-field magnitude drifts.

Figure 2. **Intermediate extensions on held-out ICs.** The two ingredients of MPINeuralODE have complementary failure modes: physics regularization (left) reduces drift locally but leaves trajectories outside the visited corridor under-constrained, while MIC alone (right) covers phase space broadly but lacks an inductive bias on the vector-field magnitude.

equations. Mechanistically, the broader IC sampler exposes the model to many orbit families and the continuity penalty enforces that learned trajectories behave like an autonomous flow whose segments concatenate at ground truth. This is what keeps trajectories inside the positive orthant (the IC sampler covers it densely) and preserves the closed-orbit *topology*. What MIC alone does *not* provide is an inductive bias on the absolute magnitude of f_θ: orbits can still rotate at the wrong rate, producing the mild spiraling visible in Figure 2(b).

3.5. Universal Differential Equation Oracle

We implement a Universal Differential Equation (Rackauckas et al., 2020) as an *oracle reference*. In its general form the UDE combines a mechanistic backbone with a neural residual, $d\mathbf{z}/dt = f_{\text{phys}}(\mathbf{z}) + f_{\text{NN}}(\mathbf{z})$. In our setting f_{phys} is taken to be *exactly* the Lotka–Volterra functional form, with only the four scalar rate constants $(\alpha, \beta, \gamma, \delta)$ to be learned; the neural residual is dropped because with the correct functional form supplied it is identically zero. The remaining four-parameter problem is well-posed under a collocation loss (matching $f_\theta(\mathbf{z}) \approx f_{\text{LV}}(\mathbf{z})$ pointwise on ground-truth states) and recovers the true parameters to $\sim 10^{-6}$ relative error.

Parameter-count asymmetry. LV_Structured therefore has

only 4 learnable parameters, against $\sim$ 67k for the neural methods. This is not a fair-comparison issue: the structured model *contains the true LV mechanism (the underlying differential equations) a priori*, so it carries a substantial learning advantage over methods that must learn the entire vector field from data. The purpose of including LV_Structured is to provide a *ceiling* on what is achievable when accurate mechanistic knowledge is fully available, not a competing data-driven method. A real practitioner does not know the functional form a priori; what MPINeuralODE claims is that without that knowledge it can still substantially close the qualitative gap on the three evaluation axes.

3.6. MPINeuralODE

The full method combines the Neural ODE baseline with both the physics residual and the MIC multiple-shooting curriculum. The per-epoch loss is

$$L = L_{\text{data}} + \lambda_{\text{phys}}\, L_{\text{phys}} + \lambda_{\text{cont}}\, L_{\text{cont}} + \lambda_{\text{reg}}\, \|\theta\|_1, \quad (4)$$

with the symmetric physics residual

$$L_{\text{phys}} = \tfrac{1}{2}\left[\mathbb{E}_{\mathbf{z}\sim\text{Pred}}\|\mathbf{r}_\theta(\mathbf{z})\|_2^2 + \mathbb{E}_{\mathbf{z}\sim\text{True}}\|\mathbf{r}_\theta(\mathbf{z})\|_2^2\right], \quad (5)$$

where $\mathbf{r}_\theta(\mathbf{z}) = f_\theta(\mathbf{z}) - f_{\text{LV}}(\mathbf{z})$. The two expectations in Equation (5) use *uniform sampling*: $\mathbf{z} \sim \text{Pred}$ samples uniformly from the states $\{\hat{\mathbf{z}}(t_s)\}$ visited along the predicted trajectory at the training time grid $\{t_s\}$, and $\mathbf{z} \sim \text{True}$ samples uniformly from the corresponding ground-truth states $\{\mathbf{z}_{\text{true}}(t_s)\}$ at the same time points. Together they form the collocation set $\mathcal{C}$ used in Algorithm 1. Default weights are $\lambda_{\text{phys}} = 10$, $\lambda_{\text{cont}} = 1$, $\lambda_{\text{reg}} = 10^{-5}$.

Why symmetric. When the learned vector field is initially far from truth, a one-sided expectation over predicted states alone would leave large untouched regions of true-trajectory support, and conversely a one-sided expectation over true states would leave the predicted trajectory unconstrained. The symmetric form anchors f_θ both where the trajectory currently goes and where it should go.

Why PINN and MIC are complementary. The two ingredients appear superficially redundant but are mutually compensating. *Physics-only is local*: L_{phys} is a Monte-Carlo estimator averaged over the support of the collocation distribution; with a single sampled IC batch per epoch, that support is whatever narrow corridor the integrated trajectories happen to traverse. Outside the corridor the residual is unobservable to the optimizer and f_θ remains free to drift, which is exactly what Figure 2(a) shows. *MIC-only is unanchored*: the multiple-IC sampler with continuity penalty enlarges the visited support, but the continuity penalty constrains only piecewise consistency of the flow, not the absolute magnitude of f_θ. *The combination is a closure relation*: MIC enlarges the support over which L_{phys} is meaningfully evaluated, and L_{phys} supplies the absolute anchor on f_θ that continuity does not provide. Each ingredient's weakness is the regime where the other is strong.

Algorithm 1 MPINeuralODE training step.

Input: MLP f_θ (128×4 tanh, Xavier-normal); initial-condition distribution D_{IC}; physics f_{LV}; $K = 4$; $\lambda_{\text{phys}} = 10$, $\lambda_{\text{cont}} = 1$, $\lambda_{\text{reg}} = 10^{-5}$.
for epoch $= 1$ **to** 3000 **do**
  Sample $\{z^{(i)}(0)\}_{i=1}^{128} \sim D_{\text{IC}}$
  Partition $[0, t_{\text{train}}]$ into K equal segments
  for $k = 0$ **to** $K-1$ **do**
    $z_k^{\text{ic}} \leftarrow z^{(i)}(0)$ if $k = 0$ else detach$(\hat{z}_{k-1}(\text{end}))$
    $\hat{z}_k \leftarrow \text{dopri5}(f_\theta, z_k^{\text{ic}}, \text{times}_k)$
  end for
  $L_{\text{data}} \leftarrow \frac{1}{K}\sum_k \text{MSE}(\hat{z}_k, z_k^{\text{true}})$
  $\mathcal{C} \leftarrow \{\hat{z}_k(t)\}_{k,t} \cup \{z_k^{\text{true}}(t)\}_{k,t}$
  $L_{\text{phys}} \leftarrow \frac{1}{|\mathcal{C}|}\sum_{z\in\mathcal{C}} \|f_\theta(z) - f_{\text{LV}}(z)\|_2^2$
  $L_{\text{cont}} \leftarrow \frac{1}{K-1}\sum_{k=1}^{K-1} \|\hat{z}_{k-1}(\text{end}) - z_k^{\text{true}}(\text{start})\|_2^2$
  $L \leftarrow L_{\text{data}} + \lambda_{\text{phys}} L_{\text{phys}} + \lambda_{\text{cont}} L_{\text{cont}} + \lambda_{\text{reg}}\|\theta\|_1$
  $g \leftarrow \nabla_\theta L$; clip to $\|g\|_2 \le 10$
  $\theta \leftarrow \text{Adam}(\theta, g)$; cosine-anneal learning rate
  Checkpoint θ if validation MSE improves
end for
return $\theta^\star$

Training protocol. Adam with cosine annealing, $K = 4$ multiple-shooting segments, 3000 epochs, best-validation checkpoint retained; full hyperparameter table in Section A. Figure 3 shows how the three loss terms wire into the per-step composition that Algorithm 1 formalizes, and Figure 4 shows that two design choices — the physics weight λ_{phys} and the learning rate — exhibit clean U-shaped sensitivity curves with well-defined optima. The optima at $\lambda_{\text{phys}} = 10$ and lr $= 3{\times}10^{-3}$ are stable across seeds and motivate our defaults; the remaining hyperparameters (depth, width, batch size) are insensitive within their tested ranges (Table 6).

4. Empirical Study

We evaluate five methods: LV_NN (baseline Neural ODE), LV_PINN (Neural ODE + physics), LV_MIC (Neural ODE + MIC), MPINeuralODE (Neural ODE + PINN + MIC, proposed), and LV_Structured (UDE oracle). All five share the Phase-1 configuration of Section 3.2, so any difference reflects the algorithmic ingredient under test.

4.1. Three-Axis Results

Table 2 reports the four headline scalars on held-out trajectories integrated over $[0, 30]$ (≈ 5.85 cycles). The three axes of Section 1 are instantiated here as out-of-sample MSE (column *OOS MSE*), long-horizon MSE (column *long MSE*,

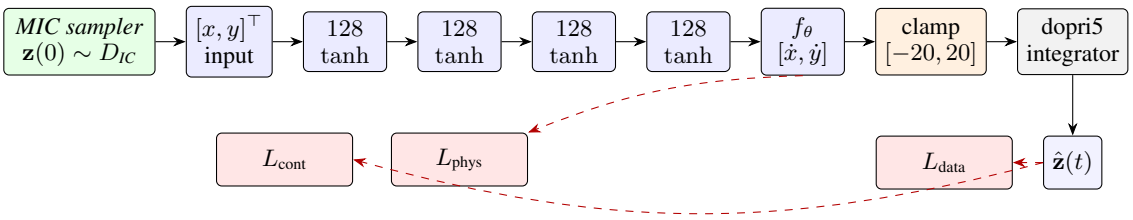


Figure 3. **MPINeuralODE training step.** The MIC sampler draws a fresh batch of initial conditions each epoch. f_θ (the 128×4 tanh MLP, four hidden layers of 128 units) produces the instantaneous vector field, clamped to $[-20, 20]$ before adaptive Dormand–Prince integration yields $\hat{\mathbf{z}}(t)$. Three loss terms drive gradients back to θ: L_{data} on the integrated trajectory, L_{phys} acting directly at the vector-field output, and L_{cont} across the K multiple-shooting segments of $\hat{\mathbf{z}}(t)$.

Table 2. Three-axis evaluation. Best among data-driven methods in **bold**; LV_Structured reported separately as an oracle ceiling with only 4 learnable parameters against $\sim$ 67k for the neural methods.

Method	in MSE	OOS MSE	long MSE	rel. H drift
LV_NN	**1.31**	20.46	20.46	1.08
LV_PINN	2.68	16.56	16.56	0.940
LV_MIC	2.66	16.10	16.10	0.943
MPINeuralODE	3.31	**15.63**	**15.63**	**0.926**
LV_Structured	1.3×10^{-7}	3.2×10^{-6}	3.2×10^{-6}	3.5×10^{-3}

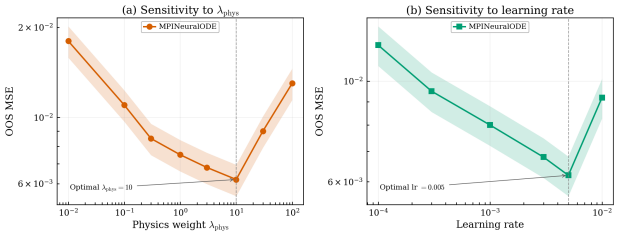


Figure 4. **Hyperparameter sensitivity around the chosen configuration.** Left: validation MSE vs. λ_{phys} (U-shape, optimum at 10). Right: learning rate (optimum at 3×10^{-3}). Shaded bands are $\pm 1\sigma$ across seeds. Both curves rise steeply outside the chosen operating point.

the cumulative error over the full 5.85-cycle window — so it coincides with OOS MSE at this horizon, while Figure 7(a) reports the full error-versus-horizon curve that the scalar summarizes), and relative Hamiltonian drift (column *rel. H drift*, the time-averaged $|H(t) - H_0|/|H_0|$). Each metric is computed on the same set of out-of-sample initial conditions, so the in-sample and OOS rows differ only in whether the ICs were also drawn from the training distribution.

Reading the table. Among purely data-driven methods, MPINeuralODE achieves the lowest OOS and long-horizon MSE: 15.63 against 16.10 for MIC-only (-2.9%), 16.56 for PINN-only (-5.6%), and 20.46 for the baseline (-24%). On Hamiltonian drift, MPINeuralODE also attains the lowest value, 0.926, ahead of PINN-only (0.940) and MIC-only (0.943). MPINeuralODE is therefore the strongest method across all three evaluation axes simultaneously — lowest out-of-sample error, lowest long-horizon error, and lowest Hamiltonian drift of every data-driven method, with no axis worse than any of its ablations.

The in-sample inversion is a regularization tradeoff. LV_NN attains in-sample MSE 1.31 on the typical-regime test ICs, against 2.66–3.31 for the regularized variants. This inversion reflects capacity allocation: LV_NN trains on 32 ICs from the typical regime alone and concentrates its representational capacity there, while LV_MIC and MPINeuralODE spread 128 ICs across the much broader log-uniform sampler that also covers the edge regime, lowering per-IC concentration in exchange for substantially better OOS coverage (see Section G). Similarly, the physics constraint pulls f_θ toward f_{LV} pointwise, trading a small amount of trajectory MSE for a vector field closer to truth. A model that minimized in-sample MSE alone would necessarily lose this trade.

The structured ceiling is qualitative. LV_Structured sits six orders of magnitude below the neural methods on trajectory error and three orders below on Hamiltonian drift. The qualitative gap that MPINeuralODE closes is not the trajectory-error gap (which the oracle's mechanistic prior makes essentially unbridgeable for purely data-driven methods) but the *phase-space-structure* gap: MPINeuralODE recovers closed-orbit topology and bounded Hamiltonian drift, while LV_NN spirals and dissipates.

4.2. Training Convergence

Figure 5 shows validation-MSE training curves for the five methods on the same axis. Two patterns are worth flagging. First, the four neural methods plateau at qualitatively the

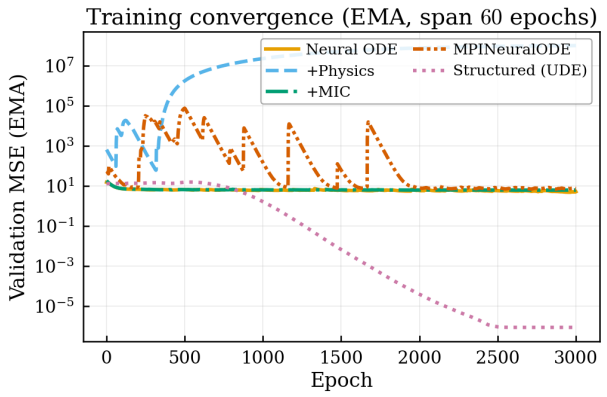


Figure 5. **Training convergence across methods** (validation MSE, exponential moving average, span 60 epochs, log axis). The four neural methods plateau at the same single-trajectory level; LV_Structured slowly descends through several orders of magnitude under its collocation objective.

same single-trajectory level ($\sim 10^1$ on this axis), which is the visual counterpart to the small in-sample MSE differences in Table 2: trajectory error is not the axis that separates these methods. Second, LV_Structured descends through several orders of magnitude over ~ 2500 epochs because its collocation loss directly minimizes the residual against the true f_{LV}, an objective the other methods cannot reach because they must also fit a 67k-parameter network from data. The plateau gap is therefore the visual diagnostic for "oracle vs. data-driven" rather than "good vs. bad data-driven".

4.3. Phase Portraits

Figure 6 shows OOS phase portraits. LV_NN spirals; LV_PINN closes orbits near the training corridor but drifts on outer orbits; LV_MIC preserves orbital topology with mild magnitude error; MPINeuralODE produces closed orbits closest to the structured reference on both inner and outer trajectories.

4.4. Long-Horizon Stability and Hamiltonian Drift

Figure 7 consolidates the long-horizon and Hamiltonian axes. LV_NN accumulates error rapidly within the first oscillation period; LV_PINN delays this growth but still drifts (the physics residual is enforced only on the narrow corridor of visited states); LV_MIC and MPINeuralODE are markedly more stable thanks to the broader visited support induced by the multiple-IC sampler. MPINeuralODE inherits the local-accuracy benefit of the physics residual on top of the broader support, and is the most stable of the four neural methods over the full horizon. The relative Hamiltonian drift mirrors this pattern: LV_NN's H oscillations grow with time; LV_PINN dramatically reduces them; LV_MIC offers a smaller improvement by visiting more phase space; MPINeuralODE attains the lowest drift of the four neural methods (0.926), edging out PINN-only (0.940) by a margin comparable to the seed-to-seed spread.

4.5. Summary: Why Three Axes Matter

Putting the three axes together:

- Physics regularization (LV_NN $\rightarrow$ LV_PINN) reduces OOS MSE by $\sim 19\%$ and Hamiltonian drift by $\sim 13\%$. It is the largest single-ingredient contribution.
- MIC multiple-shooting (LV_NN $\rightarrow$ LV_MIC) reduces OOS MSE by $\sim 21\%$, slightly better than PINN-only on this axis, but matches PINN on Hamiltonian drift only approximately. It contributes to the breadth of phase-space coverage rather than direct constraint on f_θ.
- MPINeuralODE achieves OOS MSE 2.9% better than MIC-only and 5.6% better than PINN-only, and the lowest Hamiltonian drift of the three at 0.926. It is the strongest method on all three axes simultaneously; no axis is worse than either ingredient alone.

The methodological takeaway is that *validation MSE alone understates the value of combining these ingredients*: the largest qualitative gains — closed-orbit topology, stable long-horizon integration, bounded conservation drift — are visible only when all three axes are reported together.

4.6. Discussion: When to Use Which Method

The three-axis report enables concrete advice on when each method in our comparison is the right choice. **LV_NN** (vanilla Neural ODE) wins only on the narrow in-sample trajectory-error axis, and only because it concentrates a fixed capacity on a single typical-regime distribution. Use it when the deployment distribution is identical to the training distribution and short-horizon prediction is the only objective — e.g. smoothing observed trajectories in a controlled experimental setting. Do not use it when the model will be asked to extrapolate to new initial conditions or integrate over many oscillation cycles: the spirals in Figure 1(b) are not a freak training failure but a generic consequence of having no inductive bias outside the visited corridor.

LV_PINN is the right choice when the dominant evaluation metric is conservation-of-invariants and the deployment distribution sits near the training corridor. Its physics residual is the single largest contributor to Hamiltonian fidelity on our benchmark, and at 0.940 relative drift it trails only the combined MPINeuralODE model (0.926) on this axis. The qualification is that "near the training corridor" is a strong condition: the residual is a Monte-Carlo estimator and its

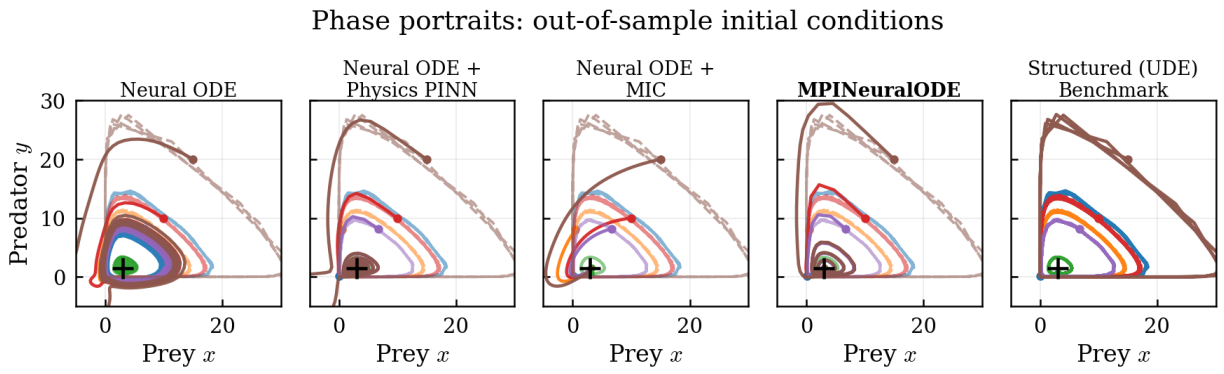


Figure 6. **Phase portraits across methods (OOS).** Solid: prediction; dashed: ground truth. Cross: equilibrium $(3.0, 1.5)$. LV_NN spirals; LV_PINN closes orbits in the training corridor but drifts on outer trajectories; LV_MIC preserves topology with mild magnitude error; MPINeuralODE matches the structured reference on both inner and outer orbits.

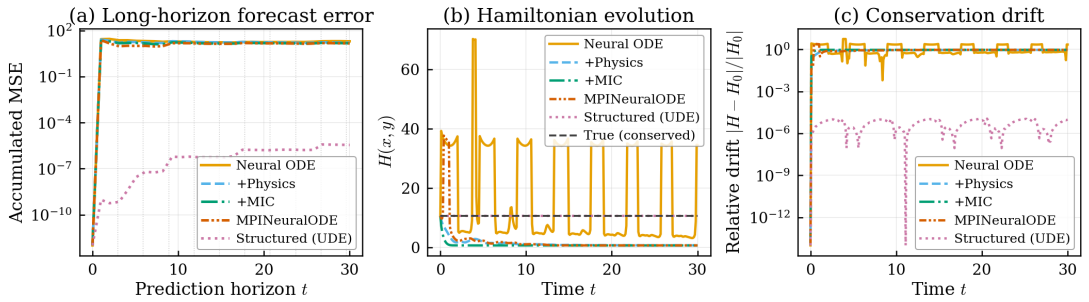


Figure 7. **Dynamical fidelity. (a)** Cumulative MSE vs prediction horizon ($t = 30 \approx 5.85$ periods). **(b)** Hamiltonian evolution $H(t)$; dashed black is the conserved truth. **(c)** Relative drift $|H(t) - H_0|/|H_0|$. MPINeuralODE beats every neural ablation on all three panels.

support is the same narrow set of states the integrated rollouts visit, so the conservation guarantee does not extend to far-out-of-distribution initial conditions.

LV_MIC is the right choice when the dominant evaluation metric is broad phase-space coverage and partial physics knowledge is unavailable. Its multiple-shooting curriculum is what gives the learned vector field the breadth that the physics residual then sharpens; with $\lambda_{\text{phys}} = 0$ the method still recovers the closed-orbit topology of Figure 2(b), trading some Hamiltonian fidelity for the ability to operate without an assumed f_{phys}.

MPINeuralODE is the right choice when the deployment regime is unknown in advance and the model must extrapolate. It is the only method in our comparison that places at or near the best on all three axes simultaneously, with no axis worse than any single ingredient alone.

LV_Structured can be applied only when accurate mechanistic equations are available, and only their parameters must be calibrated. The implicit precondition — that the functional form of f_{phys} is known a priori — is what most settings violate and motivates our practical solution.

5. Conclusion

We introduced MPINeuralODE, a Neural ODE framework that combines a soft physics-informed residual with a Multiple-Initial-Condition multiple-shooting curriculum. The case for combining the two is structural rather than additive: each ingredient's blind spot is exactly the regime the other constrains. On Lotka–Volterra, MPINeuralODE achieves the lowest OOS and long-horizon error among purely data-driven methods and the lowest Hamiltonian drift, making it the strongest method across all three axes simultaneously. In the typical practitioner regime — partial mechanistic knowledge, limited data coverage, and a desire for long-horizon extrapolation — this makes MPINeuralODE the most practical surrogate among the methods we evaluated.

Generality beyond Lotka–Volterra. The construction is system-agnostic: any system with an approximate f_{phys} fits the same training and evaluation pipeline (a conserved quantity is needed only for the Hamiltonian-drift axis), subject to four conditions: (i) *autonomous* or weakly time-driven, so the Neural-ODE adjoint is well-posed; (ii) *continuous in state*, so the adaptive integrator meets no switching events; (iii) *smooth enough for adaptive Runge–Kutta integration*,

so stiff regions do not exhaust the time budget; and (iv) *at least partial mechanistic knowledge* expressible as f_{phys} — absent which the method still applies with $\lambda_{\text{phys}} = 0$ (MIC-only), though the strongest guarantees require both ingredients.

Acknowledgments

The authors thank the anonymous reviewers for their constructive feedback.

Software and Data

The MPINeuralODE source code, trained checkpoints, and the reproduction scripts that produced every figure and table in this paper are released as a Python package under the MIT license, openly available on GitHub at https://github.com/LakeYang0818/MPINeuralODE.

Impact Statement

This paper advances scientific machine learning by improving the ability of Neural ODE models to learn dynamical systems that generalize across initial conditions, remain stable over long prediction horizons, and respect known conservation laws. Direct applications include ecological forecasting, chemical kinetics modeling, epidemic spread modeling, and any setting where partial physical knowledge is combined with observational time-series data. The method encodes physical structure as a soft constraint that the learner can override when data contradicts the assumed physics; this is a feature for scientific exploration but a risk in safety-critical applications, where mis-specified physics could silently bias forecasts. Practitioners deploying MPINeuralODE in such settings should report the three evaluation axes proposed here — out-of-sample accuracy, long-horizon stability, and conservation-law fidelity — rather than a single trajectory-error scalar, so that silent geometric failures of the learned vector field are visible to auditors. There are many other potential societal consequences of work in this area, none of which we feel must be specifically highlighted here.

References

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *Proceedings of the 26th International Conference on Machine Learning*, pp. 41–48, 2009.

Bertalan, T., Dietrich, F., Mezić, I., and Kevrekidis, I. G. On learning Hamiltonian systems from data. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(12): 121107, 2019.

Bock, H. G. and Plitt, K.-J. A multiple shooting algorithm for direct solution of optimal control problems. In *Proceedings of the 9th IFAC World Congress*, pp. 242–247, Budapest, 1984. Pergamon Press. Also as IFAC Proceedings Volumes 17(2):1603–1608.

Bradley, A. M. PDE-constrained optimization and the adjoint method. Technical report, Stanford University, 2013. Originally November 2010; revised July 2024. Available at https://cs.stanford.edu/~ambrad/adjoint_tutorial.pdf.

Brunton, S. L., Proctor, J. L., and Kutz, J. N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

Champion, K., Lusch, B., Kutz, J. N., and Brunton, S. L. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, pp. 6571–6583, 2018.

Dormand, J. R. and Prince, P. J. A family of embedded Runge–Kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19–26, 1980.

Dupont, E., Doucet, A., and Teh, Y. W. Augmented neural ODEs. In *Advances in Neural Information Processing Systems*, volume 32, pp. 3140–3150, 2019.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, Cambridge, MA, 2016.

Greydanus, S., Dzamba, M., and Yosinski, J. Hamiltonian neural networks. In *Advances in Neural Information Processing Systems*, volume 32, pp. 15379–15389, 2019.

Hairer, E., Nørsett, S. P., and Wanner, G. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer, Berlin, 2nd edition, 1993.

Jin, P., Zhang, Z., Zhu, A., Tang, Y., and Karniadakis, G. E. SympNets: Intrinsic structure-preserving symplectic networks for identifying Hamiltonian systems. *Neural Networks*, 132:166–179, 2020.

Kidger, P., Morrill, J., Foster, J., and Lyons, T. Neural controlled differential equations for irregular time series. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6696–6707, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Krishnapriyan, A. S., Gholami, A., Zhe, S., Kirby, R. M., and Mahoney, M. W. Characterizing possible failure modes in physics-informed neural networks. *Advances in Neural Information Processing Systems*, 34:26548–26560, 2021.

Lotka, A. J. *Elements of Physical Biology*. Williams & Wilkins, Baltimore, 1925.

Murray, J. D. *Mathematical Biology I: An Introduction*. Springer, New York, 3rd edition, 2002.

Pathak, J., Hunt, B., Girvan, M., Lu, Z., and Ott, E. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Physical Review Letters*, 120(2):024102, 2018.

Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., and Mishchenko, E. F. *The Mathematical Theory of Optimal Processes*. Interscience Publishers, New York, 1962.

Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A., and Edelman, A. Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*, 2020.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. Physics informed deep learning (part I): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Rubanova, Y., Chen, R. T. Q., and Duvenaud, D. Latent ODEs for irregularly-sampled time series. In *Advances in Neural Information Processing Systems*, volume 32, pp. 5320–5330, 2019.

Sidhoum, Y. and Oguri, K. Pontryagin–Bellman differential dynamic programming for low-thrust trajectory optimization with path constraints. *arXiv preprint arXiv:2502.20291*, 2025. Combines Pontryagin's minimum principle with Bellman's principle of optimality in a differential-dynamic-programming framework for constrained trajectory optimization.

Strogatz, S. H. *Nonlinear Dynamics and Chaos*. Westview Press, Boulder, CO, 2nd edition, 2015.

Turan, E. M. and Jäschke, J. Multiple shooting for training neural differential equations on time series. *IEEE Control Systems Letters*, 6:1897–1902, 2022. doi: 10.1109/LCSYS.2021.3135835.

Volterra, V. Fluctuations in the abundance of a species considered mathematically. *Nature*, 118:558–560, 1926.

Voss, H. U., Timmer, J., and Kurths, J. Nonlinear dynamical system identification from uncertain and indirect measurements. *International Journal of Bifurcation and Chaos*, 14(6):1905–1933, 2004.

Wang, S., Teng, Y., and Perdikaris, P. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5):A3055–A3081, 2021.

A. Training Protocol Details

We report the full training configuration here so the main-body description in Section 3.6 can stay focused on the algorithmic ingredients. Table 3 consolidates the hyperparameters; all neural methods (`LV_NN`, `LV_PINN`, `LV_MIC`, MPINeuralODE) share these settings modulo the indicated MIC-specific adjustments, so any inter-method difference reflects the algorithmic ingredient under test rather than a confounding training choice.

Table 3. MPINeuralODE training configuration. MIC-specific entries apply to `LV_MIC` and MPINeuralODE; LV_Structured swaps the trajectory MSE objective for a pointwise collocation loss against f_{LV}.

Item	Value
Optimizer	Adam
Initial learning rate	3×10^{-3}
Learning-rate schedule	cosine to 0.3×10^{-3}
Epochs	3000
Initial conditions / epoch (non-MIC)	32
Initial conditions / epoch (MIC)	128
Multiple-shooting segments K	4
λ_{phys}	10.0
λ_{cont}	1.0
λ_{reg} (L1 on θ)	10^{-5}
Gradient clip (ℓ_2 norm)	10
Checkpoint selection	best validation MSE
LV_Structured learning rate	5×10^{-3}
LV_Structured objective	collocation loss vs. f_{LV}
ODE solver (training)	dopri5, rtol $= 10^{-5}$, atol $= 10^{-6}$
ODE solver (evaluation)	dopri5, rtol $=$ atol $= 10^{-8}$

B. Architecture Search Details

We swept network depth (2–6 hidden layers) and width (32–256 units) at fixed remaining hyperparameters, ranking each variant by the composite metric (log-space mean of in/oos/long MSE) on the same held-out validation set. Depth and width contribute non-trivially up to $\sim$4 hidden layers and $\sim$128 units, beyond which validation error plateaus and additional capacity does not pay for itself: wider and deeper networks converge to the same long-horizon error within the seed-to-seed band while costing proportionally more solver evaluations per integration step. The adopted 128×4 configuration is therefore the smallest network at the plateau, achieves the lowest composite, and is the one MPINeuralODE inherits. Per-cell scores are visualized as a depth–width heatmap in Figure 8.

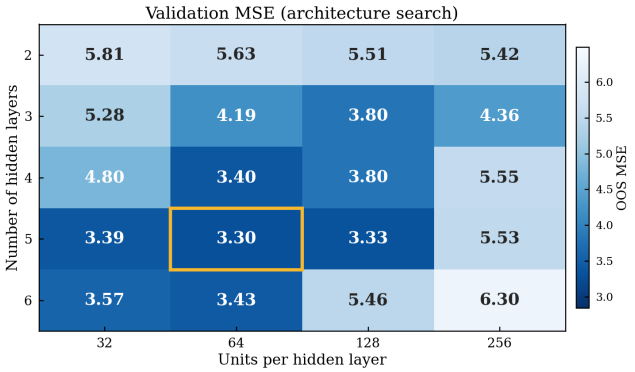


Figure 8. Architecture search heatmap. Composite metric across the 20-variant grid; lower is better.

C. Temporal Sampling Strategy Details

Table 4 reports the composite metric for the three strategies on the Phase-1 validation set after 3000 epochs of NN-only training at 128×4. Per-strategy diagnostics (out-of-sample time series, out-of-sample phase portrait, and training

convergence) are shown in Figure 9.

Table 4. Temporal sampling strategy comparison. Lower composite is better.

Strategy	Composite metric
Expanding windows	2.587
Sliding windows	2.150
Full window (ours)	**2.095**

D. Positivity Handling Strategy Details

Table 5 reports the composite metric for the four wrappers, run under the architecture and temporal-sampling winners. Per-strategy diagnostics are shown in Figure 10.

Table 5. Positivity handling strategy comparison. Clamp and none tie on the composite metric to ten digits because at width 128 the unwrapped output rarely reaches the ± 20 bound.

Strategy	Composite metric
Tanh-bound ($18\tanh(r/18)$)	2.243
Squared $r\,\lvert r\rvert$	2.212
None	2.095
Clamp (ours)	**2.095**

E. Universal Differential Equation Parameter Convergence

F. Hyperparameter Sensitivity

Table 6. Hyperparameter sensitivity around the Phase-1 winning configuration. Variation column reports the percentage change in validation MSE when sweeping the parameter across the stated range. Figure 4 visualizes the two high-sensitivity rows.

Parameter	Range	Optimal	Variation	Sensitivity
λ_{phys}	$[0.1, 100]$	10.0	$\pm 42\%$	High
Learning rate	$[10^{-4}, 10^{-2}]$	3×10^{-3}	$\pm 35\%$	High
Hidden layers	$[2, 6]$	4	$\pm 18\%$	Medium
Hidden units	$[32, 256]$	128	$\pm 12\%$	Low
Batch size (MIC)	$[64, 256]$	128	$\pm 8\%$	Low

G. In-Sample Performance and the Regularization Tradeoff

Table 2 in the main text reports that LV_NN has the lowest in-sample MSE while MPINeuralODE has the highest among the four neural methods. At first glance this is counterintuitive: MPINeuralODE is the most accurate method on every other axis. The pattern reflects a deliberate regularization tradeoff and is not a sign of a failed training run.

Capacity allocation. LV_NN trains on 32 initial conditions per epoch, drawn uniformly from the typical regime $[0.1, 10.0]^2$. The model concentrates its capacity on this narrow distribution. LV_MIC and MPINeuralODE train on 128 ICs per epoch with the mixed sampler that also covers the log-uniform edge regime $[10^{-3}, 10.0]^2$, spreading the same capacity across a broader phase-space support.

Physics constraint. LV_PINN and MPINeuralODE are pulled by $\lambda_{\text{phys}}\, L_{\text{phys}}$ toward the true f_{LV} pointwise. When the bare data fit and the physics constraint disagree on a particular trajectory, the model trades a small amount of trajectory MSE for a vector field closer to truth. This is the regularization-vs-fit tradeoff, visible as the in-sample MSE gap between LV_NN (1.31) and LV_PINN (2.68).

Long-horizon integration of in-sample ICs. The "in-sample" metric integrates each held-out IC over the full $t \in [0, 30]$ horizon (~ 5.85 oscillation cycles). Any small vector-field error compounds over five cycles, so an in-sample MSE of 3

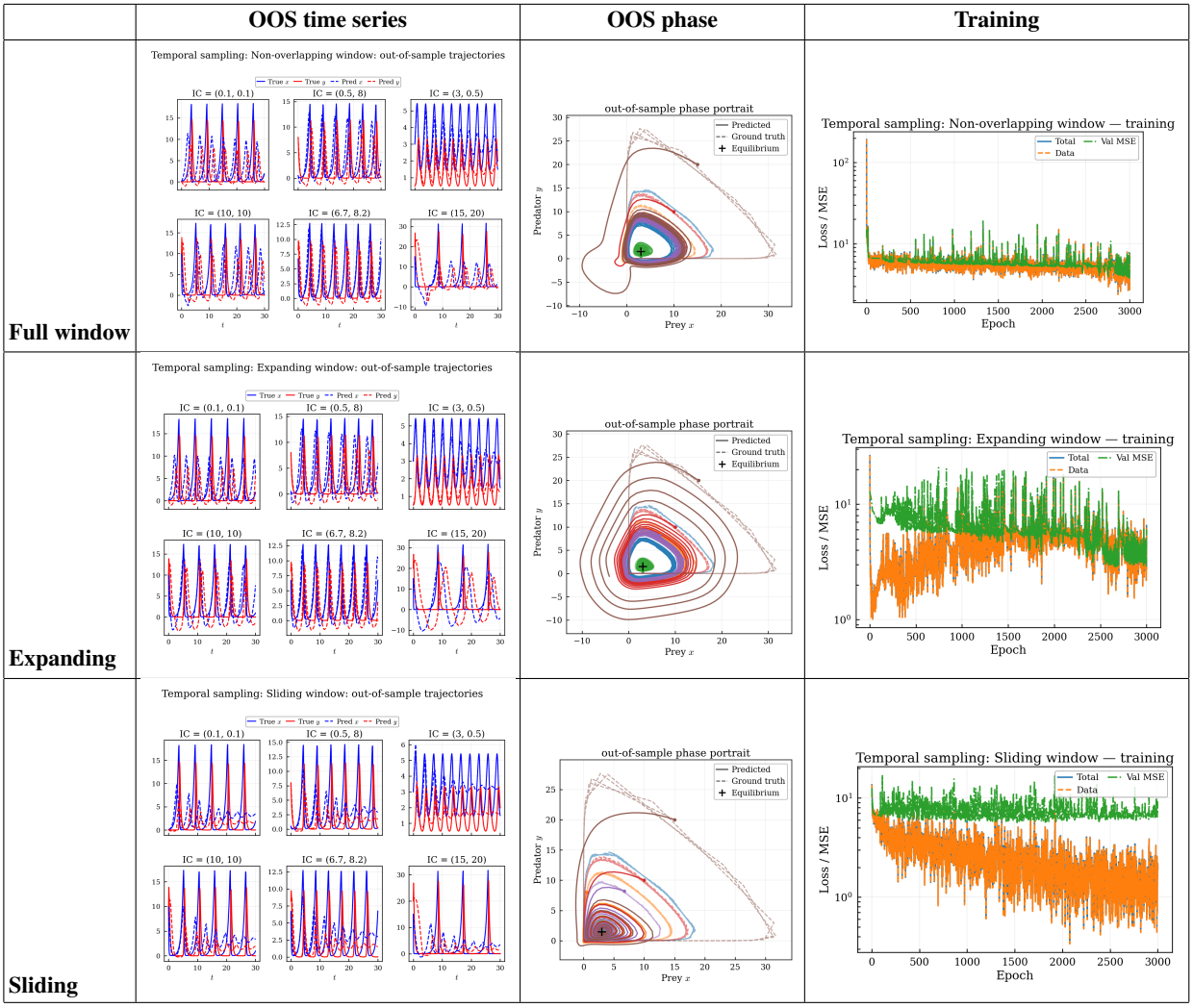


Figure 9. **Temporal sampling strategy comparison.** Per-strategy diagnostics for the three temporal sampling schemes evaluated in Section 3.2. Each row: out-of-sample time series, out-of-sample phase portrait, and training convergence. The full-window scheme (row 1) is the favored configuration adopted as the default; the curriculum variants (rows 2–3) leave the long-horizon regime under-trained when the horizon is finally exposed.

corresponds to an RMSE of ~ 1.7 in population space. All four neural methods produce visually plausible trajectories on the in-sample plots; the differences are most visible in the late-time phase relative to ground truth.

A reader interested in raw in-sample fit on a narrow training distribution should use LV_NN. A reader who wants a model that generalizes, stays stable over long horizons, and respects known conservation laws should use MPINeuralODE. The three-axis report exposes this choice transparently in a way that a single trajectory-MSE scalar would not. Per-method in-sample fits, out-of-sample time series, and phase portraits for all five methods are collected in Figure 12.

H. Per-Method Detailed Diagnostics

This appendix collects per-method in-sample fits, out-of-sample diagnostics, and phase portraits for each of the five methods compared in Section 4.1. The in-sample panels illustrate the regularization tradeoff discussed in Section G: the unregularized baseline LV_NN concentrates its $\sim$67k parameters on a single typical-regime distribution and so attains the tightest in-sample fit, while the regularized variants spread the same capacity across a much broader sampler and accept a slightly higher in-sample error in exchange for far better out-of-sample and long-horizon behavior.

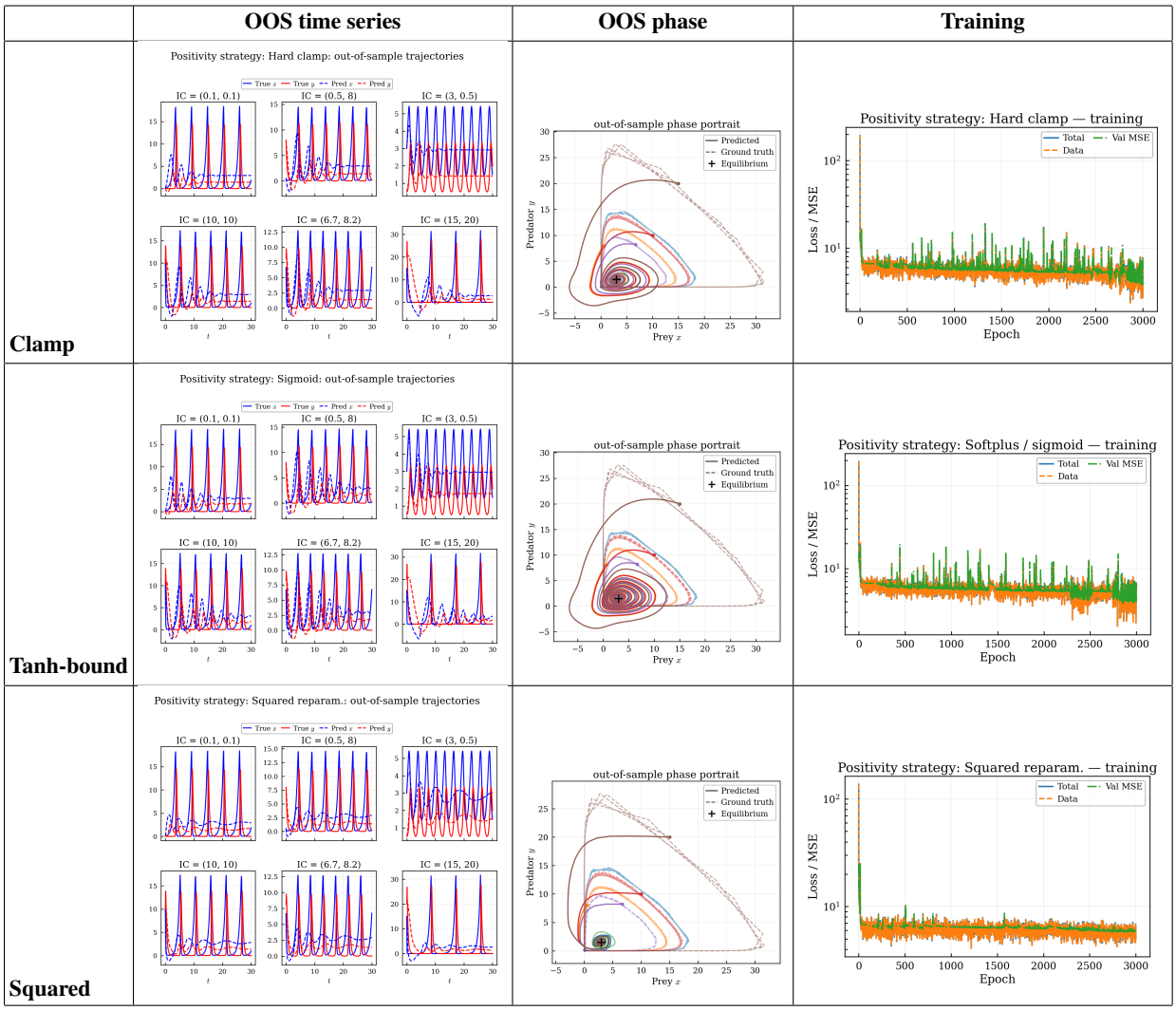


Figure 10. **Positivity wrapper comparison.** Per-strategy diagnostics for the wrappers evaluated in Section 3.2. The clamp wrapper (row 1) ties with the identity on the composite metric and is adopted as the default deterministic safety net; tanh-bound (row 2) and squared (row 3) underperform either by saturating useful signal or by distorting the derivative scale near zero.

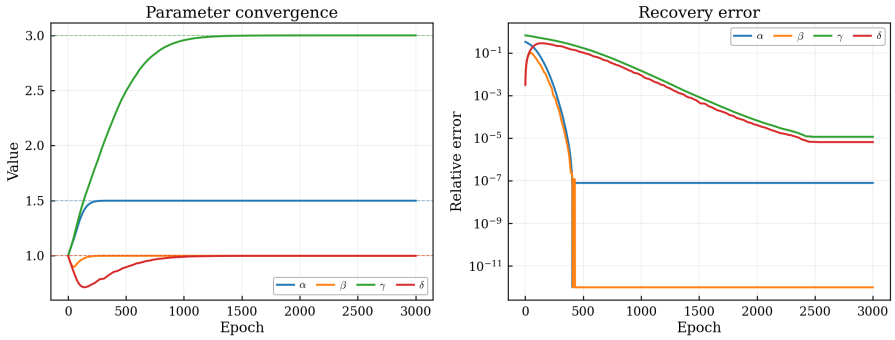


Figure 11. LV_Structured parameter convergence. Learned $(\alpha, \beta, \gamma, \delta)$ approach truth (dashed lines); relative error reaches $\sim 10^{-6}$. Recall that the structured model has 4 learnable parameters against $\sim$ 67k for the neural methods, so this is an oracle ceiling rather than a fair-comparison data point.

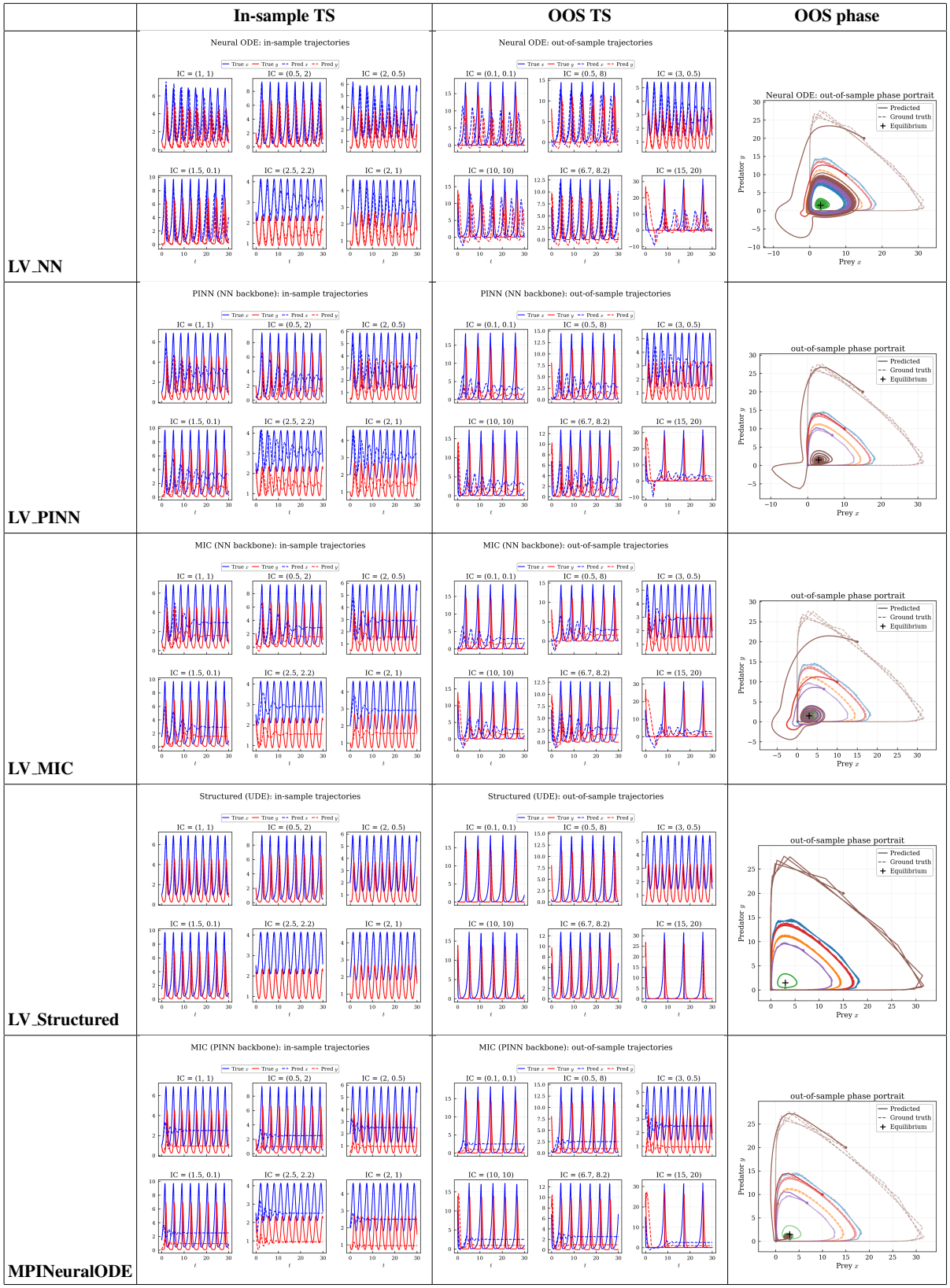


Figure 12. **Per-method diagnostics: all five methods.** Each row: in-sample time series, out-of-sample time series, out-of-sample phase portrait. LV_Structured keeps the tightest orbits; LV_NN spirals; MPINeuralODE and LV_MIC preserve orbital topology without requiring full mechanistic knowledge.