PREDATORS, PREY, AND SLOPPY PARAMETERS: FINE-TUNING AMORTIZED INFERENCE FOR GENERALIZED LOTKA–VOLTERRA

Anonymous authors
Paper under double-blind review

ABSTRACT

Point estimation in nonlinear ordinary differential equation inverse problems remains challenging under practical non-identifiability. We conduct a controlled generalized Lotka–Volterra benchmark at species counts $S \in \{3, 6, 10\}$, using one trajectory per sampled parameter vector to compare amortized, posterior, and test-time parameter inference methods. Parameter accuracy can coexist with poor or numerically invalid trajectory recovery. Inferred-point FIM geometry and posterior diagnostics reveal sensitivity concentrated in a small fraction of parameter directions as system size increases. We introduce Optimizer-In-the-Loop (OIL) tuning, which learns a warm start with a three-term objective by differentiating through finite trajectory-matching (TM) updates, followed by TM at inference. Evaluation covers clean observations and 0.1%, 1%, and 5% additive noise, emphasizing simulation validity and upper-tail errors. To further generalize these results, we also compare neural posterior estimation (NPE) posteriors, and conduct separate FitzHugh–Nagumo and Lorenz-63 evaluations. OIL improves trajectory reliability and typically yields Fisher Information Matrix (FIM) orientations more closely aligned with those at the true parameters.

1 INTRODUCTION

The governing parameters of a dynamical system are rarely observable directly, so they must be recovered from measured trajectories. This inverse problem is central to mechanistic forecasting, hypothesis-driven discovery, and digital twins in biology and engineering (Henneking et al., 2026; Egan et al., 2024). For nonlinear, multi-component ordinary differential equation (ODE) systems, it remains difficult under practical non-identifiability, where the available observations constrain some parameter combinations strongly and leave others weakly determined. Classical test-time estimators optimize agreement with observations separately for each instance; physics-informed neural networks (PINNs) impose mechanistic constraints on learned trajectory representations (Raissi et al., 2019); amortized point estimators learn trajectory-to-parameter mappings (Kastoris et al., 2024; Pandey, 2023); and amortized posterior estimators additionally characterize parameter uncertainty. These objectives are related but distinct, and parameter accuracy alone does not establish reliable trajectory recovery. We investigate these distinctions through a controlled study of the generalized Lotka–Volterra (gLV) system, a canonical model of interacting populations in computational ecology and microbiome modeling (Stein et al., 2013; Bucci et al., 2016; Gibson & Gerber, 2018; Fontanarrosa & Barnes, 2025); its quadratic parameter growth and qualitatively distinct dynamical regimes make it a useful methodological testbed. At species counts $S \in \{3, 6, 10\}$, we use one trajectory per sampled parameter vector and hold the prior, trajectories, folds, and scoring fixed across methods within each benchmark condition, exposing discrepancies between parameter accuracy, trajectory recovery, and numerical-simulation validity as practical non-identifiability increases. Drawing on sloppy-models theory (Gutenkunst et al., 2007; Transtrum et al., 2015), we evaluate the trajectory-sensitivity Fisher information matrix (FIM) at each method's inferred parameters, using effective-rank and principal-subspace diagnostics to characterize local constraints and their alignment with those at the truth. We propose task-aligned fine-tuning of message-passing and temporally encoded Transformer estimators, in which we compare joint parameter-and-trajectory objectives with Optimizer-In-the-Loop (OIL) tuning. OIL is trained with a three-term objective function that at the optimum reduces the discrepancy

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

between the inferred and true parameters and is supervised by differentiating through deterministic fixed-step trajectory-matching optimizer that it would later see at inference. At inference, the initializer is refined by trajectory matching with the same downstream optimizer and call budget used for ordinary warm starts. We assess improvements through trajectory accuracy, simulation validity, and error tails, without assuming uniform superiority across conditions.

Contributions. (i) Controlled inverse-method benchmark. We hold the dynamical family, prior, trajectories, folds, and scoring fixed while comparing amortized point, posterior, and test-time estimators as practical non-identifiability increases. **(ii) Inferred-point geometry.** We evaluate the trajectory-sensitivity FIM at the parameter returned by each method using scale-free effective rank and truth-versus-inferred principal-subspace alignment. **(iii) Task-aligned learned warm start.** We combine two established ingredients, unrolled optimization and trajectory losses, into an initializer trained with a three-term objective through finite downstream trajectory-matching updates, and evaluate it in a controlled setting for this inverse problem. **(iv) Failure-aware evaluation.** We report matched call budgets, wall-clock time, simulation invalidity, upper-tail errors, additive and structured noise, neural posterior estimation (NPE) calibration and predictive checks, and separate evaluations on FitzHugh–Nagumo and Lorenz-63 systems.

2 Previous Work

Physics-informed neural networks. PINNs (Raissi et al., 2019) solve inverse problems via composite loss functions on neural surrogates. However, optimization frequently fails due to gradient imbalances (Wang et al., 2021) and ill-conditioned differential-operator penalties (Krishnapriyan et al., 2021). This motivates decoupling surrogate training limitations from parameter recovery failures.

Sloppy models and identifiability. "Sloppy" models exhibit highly unequal parameter sensitivities, reflected in broad FIM eigenvalue spectra (Gutenkunst et al., 2007; Transtrum et al., 2015). Because this local geometry depends on parameters and observation design, we evaluate where inverse methods land by comparing the effective rank and principal-subspace alignment of the trajectory-sensitivity FIM against the truth.

Amortized inference. While trajectory regression learns point-valued maps (Kastoris et al., 2024; Pandey, 2023), Neural Posterior Estimation (NPE) learns conditional parameter distributions (Gonçalves et al., 2020; Deistler et al., 2025). Using tools like BayesFlow (Radev et al., 2023) and `sbi` (Tejero-Cantero et al., 2020) with autoregressive flows (Papamakarios et al., 2017), amortized workflows characterize complex observation-parameter mappings. Following SBI benchmarking guidance (Lueckmann et al., 2021), we move beyond single point summaries to separately report parameter error, posterior calibration, and predictive trajectory recovery.

Learned warm starts and differentiable optimization. Learning to warm-start (Sambharya et al., 2024) and semi-amortized inference (Kim et al., 2018) combine amortized initialization with instance-specific refinement. Concurrently, methods like OptNet (Amos & Kolter, 2017) align predictive training with downstream task performance (Donti et al., 2017). OIL builds on these paradigms for mechanistic point-parameter estimation. By backpropagating through a finite unroll of trajectory-matching updates, OIL targets an initialization specifically compatible with the downstream optimizer, avoiding the need to differentiate through a converged solution.

3 The Inverse Problem on the gLV System

3.1 Problem Setup and Dataset

We study parameter inference for the generalized Lotka–Volterra (gLV) system,

$$\frac{dx_i(t)}{dt} = x_i(t)\left(\mu_i + \sum_{j=1}^{S} A_{ij}x_j(t)\right), \qquad i = 1, \ldots, S, \tag{1}$$

where $x_i(t) \geq 0$ is the abundance of species i, $\boldsymbol{\mu} \in \mathbb{R}^S$ contains growth rates, and $\mathbf{A} \in \mathbb{R}^{S\times S}$ contains interaction coefficients. The inferred parameter vector $\boldsymbol{\theta} = (\boldsymbol{\mu}, \mathbf{A})$ has dimension $P = S + S^2$, and we consider $S \in \{3, 6, 10\}$. For each S, we independently sample growth rates,

interaction coefficients, and initial abundances according to

$$\mu_i \sim \mathcal{U}(0.8, 1.6),\ A_{ii} \sim \mathcal{U}(-0.16, -0.04),\ A_{ij} \sim \mathcal{U}(-0.03, 0.06),\ x_i(0) \sim \mathcal{U}(5, 15);\ i \neq j. \quad (2)$$

All coordinates are sampled independently before rejection, including independence between the parameter and initial-state proposals. The supports are identical across species counts, with no species-count-dependent rescaling of the interaction coefficients. We retain 100,000 accepted parameter–initial-state pairs per S under the integration and rejection rules in Appendix A, each contributing exactly one trajectory. The benchmark therefore samples the joint proposal distribution conditioned on generation acceptance, which differs from the unconditioned product prior.

For each S, we use five deterministic, saved folds with a 70%/10%/20% training/validation/test split. The five test partitions are disjoint and exhaust the dataset, and every learned prediction comes from a checkpoint whose training and validation sets exclude that example, so pooling the five test folds yields exactly one out-of-fold prediction per system. Test-time estimators are evaluated on the same systems. Adding observation noise yields further measurements of this same underlying trajectory and supplies no new dynamical experiments. Full clean-data results pool all 100,000 out-of-fold systems per S. Matched clean/noisy analyses use fixed 1,000-system cohorts per S, drawn as 200 systems from each test fold, with system identities held constant across the methods and noise levels compared; all such summaries are labeled as matched-cohort. NPE calibration also uses 1,000 held-out systems per S, while its more expensive posterior-predictive and uncertainty-geometry checks use 200 systems per S (40 per fold); these diagnostic sample counts are reported separately.

We score every estimate $\widehat{\boldsymbol{\theta}}$ on three axes. First is parameter error in the native-coordinate $\mathrm{RMSE}(\widehat{\boldsymbol{\theta}}, \boldsymbol{\theta})$. Second is trajectory error in between the forward rollout $\widehat{\boldsymbol{X}}_m$ under $\widehat{\boldsymbol{\theta}}$ and the observation $\boldsymbol{Y}_m$. Third is rollout validity. A rollout is valid if the solve succeeds with finite values and the trajectory error is $< 10^{12}$; invalid rollouts are scored 10^{12}, and we report validity, medians, and upper quantiles over the complete cohort, with no filtering on success. Noise experiments add Gaussian noise with standard deviation 0.1%, 1%, or 5% of the pooled clean-abundance standard deviation for each S, train learned estimators at the matching level, and score trajectories against the clean latent state.

In addition to clean observations, we evaluate additive, homoscedastic Gaussian noise:

$$\mathbf{Y}_m^{(p)} = \mathbf{X}_m + \boldsymbol{\epsilon}_m^{(p)}, \qquad \boldsymbol{\epsilon}_m^{(p)} \sim \mathcal{N}\big(\mathbf{0}, \sigma_{S,p}^2 \mathbf{I}_{ST}\big), \qquad \sigma_{S,p} = \frac{p}{100} s_S, \quad (3)$$

where $p \in \{0, 0.1, 1, 5\}$ denotes the noise percentage. The dataset-level scale is

$$s_S = \mathrm{Std}(\{X_{m,i,k} : m \in \mathcal{R}_S,\ i = 1, \ldots, S,\ k = 1, \ldots, T\}), \quad (4)$$

where $\mathcal{R}_S$ contains all 100,000 accepted clean trajectories for that species count (pooled over systems, times, and species; population standard deviation). This scale is computed from pooled clean abundances and held fixed across trajectories and methods for a given dataset, so it is never recomputed per trajectory. Learned estimators are trained at the corresponding noise level, and all methods receive identical observations within each comparison. Under additive noise, L_m^{obs} measures fit to the noisy observation; results labeled as latent-trajectory recovery compare the rollout with $\mathbf{X}_m$. Appendix K describes a separate structured observation-shift experiment that applies temporally correlated multiplicative noise without retraining and evaluates recovery against the clean latent trajectory.

3.2 Inverse Methods and Benchmarking

We benchmark test-time inference methods trajectory matching (TM) via `diffrax` (Kidger, 2022) and L-BFGS-B, gradient matching (GM) via estimated derivatives, and inverse PINNs (Raissi et al., 2019) along with amortized MLP, Transformer, and MPNN regressors (both with and without temporal encoding). Architecture definitions, test-time methods, and the original derivative analysis appear in Appendices C, D, and B; architecture results are retained in Table 10. Additionally we compared an NPE baseline with its posterior mean and assessed the distribution separately. To isolate variables, prior-initialized iterative comparisons reuse a fixed prior draw per system, whereas learned-warm-start tests vary the initializer but strictly fix the downstream TM optimizer and call budget. We consistently separate raw network predictions from optimizer-refined outputs, independently reporting call budgets, wall-clock time, and offline training costs. Architectural comparisons reveal a mismatch between parameter-space accuracy and forward-simulation reliability: at $S = 10$, Transformer and

MPNN variants attain parameter RMSE 0.056–0.062 but only 72.5–82.1% valid rollouts. These diagnostics motivate evaluating forward validity and optimization compatibility alongside parameter error; the current matched-budget results follow below.

3.3 Inferred-Point Sensitivity Geometry

We examine local trajectory-sensitivity geometry to characterize the mismatch between parameter accuracy and forward recovery. In line with the sloppiness work on dynamical systems of biology (Gutenkunst et al., 2007; Transtrum et al., 2015), we asked geometric questions about the parameter-trajectory function, like how sensitivity are certain parameter directions, and how those sensitive directions are oriented at the predicted parameter values.

For system m, let $f_m(\boldsymbol{\theta}) = \mathbf{X}(\boldsymbol{\theta}; \mathbf{x}_{0,m})$ denote the forward trajectory at the prescribed observation times with the initial state held fixed, and let $J_{\theta,m} = \partial f_m / \partial \boldsymbol{\theta}$. Prior-whitened parameter coordinates are

$$\mathbf{z} = L_S^{-1}(\boldsymbol{\theta} - \mathbf{m}_S), \qquad J_{z,m} = J_{\theta,m} L_S, \tag{5}$$

where $\mathbf{m}_S$ and L_S are the empirical mean and covariance square root of the corresponding fold's clean training parameters, fixed across methods, evaluation points, and noise conditions within that fold (fold index suppressed). Whitening is a change of coordinates and introduces no prior-precision term into the information matrix. We evaluate the unweighted trajectory-sensitivity matrix

$$\mathcal{F}_m^{\mathrm{w}}(\boldsymbol{\theta}) = J_{z,m}^\top J_{z,m} = L_S^\top J_{\theta,m}^\top J_{\theta,m} L_S \tag{6}$$

at the true parameters $\boldsymbol{\theta}_m^\star$ and at each method's returned parameters $\widehat{\boldsymbol{\theta}}_m$. Under isotropic Gaussian observation noise with variance $\sigma^2 > 0$, the likelihood FIM in these coordinates is $\sigma^{-2}\mathcal{F}_m^{\mathrm{w}}$, a scalar rescaling that leaves the normalized spectrum and the eigenspaces unchanged at a fixed parameter vector. We therefore apply Eq. (6) directly to clean observations, where σ^{-2} diverges. Noise can still shift the inferred parameters and hence the geometry evaluated at them, and this invariance does not generally extend to non-isotropic observation covariance.

Let $\lambda_1 \geq \cdots \geq \lambda_P \geq 0$ be the eigenvalues of a finite, nonzero $\mathcal{F}^{\mathrm{w}}$. Our primary spectral summary is the threshold-free entropy effective rank,

$$q_i = \frac{\lambda_i}{\sum_{j=1}^P \lambda_j}, \qquad r_{\mathrm{ent}} = \exp\left(-\sum_{i=1}^P q_i \log q_i \right), \tag{7}$$

with $0 \log 0 = 0$. This quantity measures how broadly sensitivity is distributed across parameter directions and does not count structurally identifiable parameters; neither a larger nor a smaller value is intrinsically better for an estimator. Appendix I reports principal-angle distributions over the leading-subspace dimension and numerical-validity checks.

A registered relative threshold serves only to fix the comparison dimension at truth:

$$k_m = \#\left\{ i : \lambda_i(\mathcal{F}_m^{\mathrm{w}}(\boldsymbol{\theta}_m^\star)) > 0.01\, \lambda_1(\mathcal{F}_m^{\mathrm{w}}(\boldsymbol{\theta}_m^\star)) \right\}. \tag{8}$$

Let $U_m^\star$ and $\widehat{U}_m$ contain the leading k_m orthonormal eigenvectors of the truth and inferred FIMs in the same whitened coordinates. We report the largest principal angle,

$$\alpha_{\max,m} = \arccos\left[\sigma_{\min}\left((U_m^\star)^\top \widehat{U}_m \right) \right], \tag{9}$$

in degrees, with smaller angles indicating closer local orientation. The inferred subspace is truncated at the truth-selected dimension k_m, so its own threshold rank plays no role. This orientation diagnostic therefore retains one declared dimension choice, while the entropy effective rank and the complete spectrum remain cutoff-free.

The median truth-FIM entropy effective ranks are 2.140, 2.194, and 1.525 for $P = 12$, 42, and 110, respectively: sensitivity remains concentrated as dimension grows, but absolute rank need not decrease monotonically. Similar spectral concentration does not guarantee similar orientation. Table 2 reports the angle comparison; Appendix I gives the full summaries and their solver-validity qualifications.

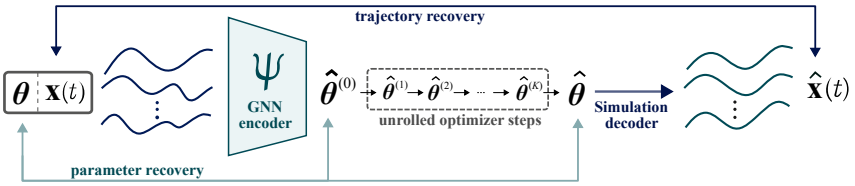


Figure 1: OIL training differentiates through five trajectory-matching updates. The objective supervises the initial parameters, refined parameters, and refined trajectory (Eq. (13)). At inference, the learned initialization instead enters a separate, budgeted L-BFGS-B solve.

For a sufficiently small displacement $\delta\mathbf{z}$ within a smooth, valid region of the forward model,

$$\frac{1}{2}\left\|f_m(\boldsymbol{\theta}_m^\star + L_S\delta\mathbf{z}) - f_m(\boldsymbol{\theta}_m^\star)\right\|_2^2 = \frac{1}{2}\delta\mathbf{z}^\top \mathcal{F}_m^{\mathrm{w}}(\boldsymbol{\theta}_m^\star)\delta\mathbf{z} + O(\|\delta\mathbf{z}\|^3), \tag{10}$$

which explains why parameter errors of similar magnitude can carry different trajectory consequences depending on their direction. FIM spectra and principal angles alone, however, cannot establish that an estimator's error lies entirely in weak directions, that different estimates occupy a shared globally flat subspace, or that non-identifiability is the sole cause of numerical failures. We therefore read inferred-point geometry as descriptive, interpreting it alongside trajectory recovery, simulation validity, and posterior diagnostics.

4 Differentiable Optimizer-In-the-Loop (OIL) Fine-Tuning

Per the geometric analysis in Section 3.3, we posit that if the relevant object is the trajectory-fitting basin, an amortized estimator can be steered toward it by training under a trajectory-matching objective. We pursue this by combining knowledge we gained from test-time and amortized inference methods we explored above. Thus, we introduce Optimizer-In-the-Loop (OIL) tuning, where the predictor's output is refined by a few fixed optimization steps on the trajectory loss before the loss is propagated back through the optimizer. Section 4.3 evaluates each intervention on trajectory accuracy, parameter accuracy, identifiability, and forward-simulation validity.

4.1 Method

Let $\mathbf{X}^\star$ denote the observed trajectory supplied in the relevant experimental condition, $\boldsymbol{\theta}^\star$ its generating parameter label, and $\mathbf{x}_0$ the initial state used by the simulator. In the additive-noise experiments, $\mathbf{X}^\star$ is the noisy observation rather than the unobserved clean trajectory. For gLV, the encoder g_ψ is an MPNN initialized from the corresponding parameter-supervised baseline, and including its bounded native-parameter output map it produces

$$\boldsymbol{\theta}^{(0)} = g_\psi(\mathbf{X}^\star). \tag{11}$$

OIL training applies exactly 5 differentiable trajectory-matching updates,

$$\boldsymbol{\theta}^{(k+1)} = \mathcal{U}_{\mathrm{train}}\left(\boldsymbol{\theta}^{(k)}; \mathbf{X}^\star, \mathbf{x}_0\right), \qquad k = 0, \ldots, 4, \tag{12}$$

where $\mathcal{U}_{\mathrm{train}}$ is the safeguarded inner update implemented in prior-logit coordinates, with native-coordinate endpoint $\boldsymbol{\theta}^{(5)}$. The outer objective is backpropagated through the simulator and these five updates to train the initialization. The unroll is finite by design and assumes no convergence of the inner optimization.

Writing $f(\boldsymbol{\theta}, \mathbf{x}_0)$ for a simulated trajectory, the per-example objective is

$$\mathcal{L}_{\mathrm{OIL}}(\psi) = \mathcal{L}_\theta\left(\boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^\star\right) + \mathcal{L}_\theta\left(\boldsymbol{\theta}^{(5)}, \boldsymbol{\theta}^\star\right) + \mathcal{L}_{\mathrm{traj}}\left(f(\boldsymbol{\theta}^{(5)}, \mathbf{x}_0), \mathbf{X}^\star\right). \tag{13}$$

The three terms carry equal 1:1:1 outer weights, where $\mathcal{L}_\theta$ operates on block-standardized growth and interaction parameters and $\mathcal{L}_{\mathrm{traj}}$ applies a Huber penalty to separately scaled, asinh-transformed simulated and target states. Equal weighting therefore applies to these implemented scaled losses,

and the terms are not three unscaled native-coordinate MSE terms. Each term serves a distinct role. The initial parameter term preserves a useful amortized estimate. The refined parameter term discourages uncontrolled drift in weakly constrained directions and retains supervision toward the labeled parameter representative. The refined trajectory term emphasizes observation-sensitive directions and shapes the initializer for the downstream optimizer. The roles are complementary, since reproducing an observation can leave the generating parameter vector unrecovered, while parameter accuracy alone can still yield an unreliable forward rollout. Direct parameter-and-trajectory fine-tuning (FT) is a separate reference procedure that applies supervision at the encoder output without differentiating through downstream optimizer updates. We specify its loss separately, and it should not be identified with Eq. (13) at $K = 0$.

At inference, the raw prediction $\boldsymbol{\theta}^{(0)} = g_\psi(\mathbf{X}^\star)$ initializes native-coordinate, bounded L-BFGS-B trajectory matching:

$$\widehat{\boldsymbol{\theta}}_{\mathrm{OIL}\to\mathrm{TM}} = \mathrm{TM}_B\left(g_\psi(\mathbf{X}^\star); \mathbf{X}^\star, \mathbf{x}_0\right). \tag{14}$$

The five differentiable training updates play no part at inference. The principal comparison uses the registered budget $B = 1{,}200$ combined objective/gradient calls, with additional budgets reported in the budget ablation, and ordinary and OIL-trained warm starts share the same downstream optimizer, parameter bounds, objective, and call budget. We report the raw OIL-trained prediction and OIL→TM separately throughout; unless explicitly labeled as raw, OIL inference results use the final optimizer-returned parameter and its independently scored rollout. The overall estimator thus combines an amortized initialization with instance-specific optimization, which places it outside the single-pass amortized class. Training losses, coordinate maps, numerical safeguards, and the separate training, validation, and scoring solvers are detailed in Appendix E.

4.2 Evaluation Protocol

We use the full and matched cohorts and failure-aware recovery scores defined in Section 3.1. The matched-budget ablation changes only the initializer (saved prior draw, MPNN, or OIL) while retaining the same downstream optimizer and combined objective/gradient-call budget. Inference time and offline training cost are reported separately. Geometric diagnostics follow Section 3.3, with their evaluation-validity criteria stated alongside each analysis.

4.3 Results

Throughout this section, TM denotes prior-initialized trajectory matching, MPNN denotes the raw parameter-supervised prediction, MPNN+TM denotes its trajectory-matching continuation, and OIL denotes the OIL-trained MPNN followed by the same downstream TM procedure. Optimized pipelines use the registered budget of 1,200 combined objective/gradient calls unless stated otherwise.

4.3.1 Clean Separation and Ablations

Figure 3 separates parameter and trajectory accuracy: at $S = 6, 10$, raw MPNN improves median parameter RMSE over TM but has much larger trajectory error. OIL combines near-MPNN parameter accuracy at $S = 6, 10$ with the lowest median trajectory error at all three sizes. Appendix Table 3 compares MPNN→TM with OIL→TM to isolate the initializer's impact under an identical downstream budget. Relative performance depends on system size: at $S = 3$, the two methods are nearly tied on terminal median objective; at $S = 6$, MPNN→TM achieves a strictly better objective; and at $S = 10$, OIL→TM attains the lowest terminal median objective and time. Because OIL includes downstream optimization, its end-to-end inference cost exceeds the raw encoder's forward pass.

Across 100,000 clean systems per S, direct TM validity remains 100%. Raw MPNN validity drops from 99.49% at $S = 3$ to 85.13% at $S = 10$. OIL improves this stability, achieving validities of 99.60%, 99.06%, and 93.62% for $S \in \{3, 6, 10\}$. While OIL improves validity over raw MPNN by 8.49 percentage points at $S = 10$, its 6.38% invalid rate (6,380 systems) remains a limitation. In valid-only tail analysis, OIL improves p95 and p99 relative to prior-initialized TM, though the worst 1% of valid OIL rollouts still accounts for $> 98\%$ of the valid-only mean error across all sizes (Appendix Table 4).

Table 1 reports median parameter RMSE and median log-trajectory MSE under noise. At higher noise levels, raw MPNN retains the best parameter RMSE but suffers in trajectory recovery. Conversely,

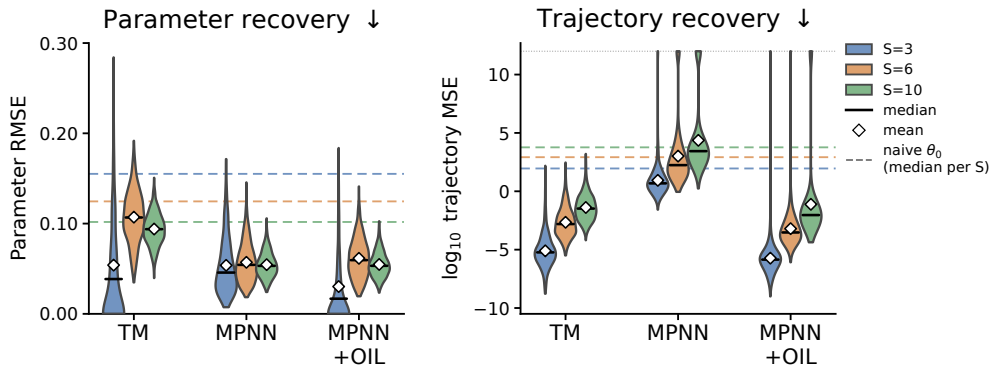


Figure 2: Clean recovery on 1,000 matched systems per species count S. Left: native parameter RMSE; right: $\log_{10}$ trajectory MSE. Lower is better in both panels. Smoothed violins retain all systems; bars/diamonds show medians/means of the displayed values. Colored dashed lines mark species-specific median errors of the saved naive prior draws θ_0. TM and OIL (shown as MPNN+OIL) use $B = 1{,}200$; MPNN is raw inference. Failed rollouts retain MSE 10^{12}. Full-cohort comparisons appear in Appendix F.

Table 1: Matched-cohort recovery: median parameter RMSE / median $\log_{10}$ trajectory MSE (1,000 systems per S, five folds; lower is better). Trajectories are scored against the clean observation in the clean column and the noisy observation otherwise. Invalid rollouts retain MSE 10^{12}; validity is reported in Table 5. TM and OIL use $B = 1{,}200$.

S	Method	Clean	Noise 0.1%	Noise 1%	Noise 5%
3	TM	$0.0386/-5.233$	$0.0464/-2.991$	$0.0872/-1.003$	$0.1277/0.396$
	MPNN	$0.0457/0.680$	$0.0553/0.718$	$\mathbf{0.0687}/0.651$	$\mathbf{0.0878}/0.996$
	OIL	$\mathbf{0.0168}/\mathbf{-5.852}$	$\mathbf{0.0302}/\mathbf{-3.002}$	$0.0792/\mathbf{-1.005}$	$0.1193/\mathbf{0.395}$
6	TM	$0.1068/-2.806$	$0.1067/-2.121$	$0.1079/-0.210$	$0.1116/1.182$
	MPNN	$\mathbf{0.0543}/2.241$	$\mathbf{0.0597}/2.621$	$\mathbf{0.0715}/2.271$	$\mathbf{0.0806}/2.604$
	OIL	$0.0595/\mathbf{-3.527}$	$0.0672/\mathbf{-2.182}$	$0.0768/\mathbf{-0.215}$	$0.0841/\mathbf{1.180}$
10	TM	$0.0938/-1.480$	$0.0936/-1.301$	$0.0939/0.181$	$0.0951/1.564$
	MPNN	$\mathbf{0.0532}/3.439$	$\mathbf{0.0580}/3.619$	$\mathbf{0.0652}/3.653$	$\mathbf{0.0698}/3.605$
	OIL	$0.0532/\mathbf{-2.037}$	$0.0606/\mathbf{-1.568}$	$0.0663/\mathbf{0.174}$	$0.0727/\mathbf{1.562}$

OIL→TM attains the lowest median noisy-observation trajectory error across all nine conditions, though its margin over direct TM becomes small at higher noise levels. Crucially, parameter proximity does not guarantee observation fit; the smallest trajectory error can coexist with larger parameter error, confirming these are distinct recovery axes.

4.4 FIM Under Noise

OIL yields the smallest median truth-versus-inferred principal angle in every cell (Table 2). However, high-dimensional alignment is challenging even on clean data, where median angles are $45.6°$ (OIL) and $59.9°$ (TM) at $S = 10$. Under the isotropic noise model, FIM spectral changes across noise levels stem strictly from shifts in the inferred parameter vectors, not from variance reweighting.

Truth-FIM entropy effective ranks are highly concentrated: 2.14, 2.19, and 1.53 for parameter dimensions 12, 42, and 110, respectively. While sensitivity remains isolated in a few dominant directions despite growing dimensionality, this spectral concentration does not guarantee orientational alignment, and OIL is not uniformly closest to the truth in entropy rank (Appendix I).

Table 2: Median largest principal angle $\alpha_{\max}$ (degrees) between the truth and inferred prior-whitened trajectory-sensitivity FIM stiff subspaces. Cells denote clean / 0.1% / 1% / 5% additive noise across 1,000 matched test systems per S. These finite-Jacobian summaries use the original evaluation gate; see the complete-solve audit in Appendix I.

Method	$S = 3$	$S = 6$	$S = 10$
TM	$0.080/0.128/0.344/0.820$	$2.851/2.891/3.892/9.634$	$59.878/59.062/59.312/66.281$
MPNN	$4.546/4.366/4.070/5.006$	$17.205/18.989/16.010/19.239$	$59.407/63.724/65.054/69.622$
OIL	$\mathbf{0.046/0.094/0.312/0.788}$	$\mathbf{2.212/2.228/3.712/9.585}$	$\mathbf{45.615/48.304/48.910/60.930}$

4.5 Posterior Baseline

We evaluate NPE-MAF as a conditional distribution rather than by point-summary RMSE. On clean systems, NPE-MAF exhibits near-nominal marginal coverage and improves the energy score relative to the prior, though its TARP MACE is worse. Predictive validity drops from 100% and 98.5% at $S \in \{3, 6\}$ to 79.8% at $S = 10$. Consequently, median posterior-draw trajectory MSE degrades drastically from 3.81×10^{-2} at $S = 3$ to 2.34×10^{3} at $S = 10$.

The posterior captures strong sloppy-versus-stiff uncertainty anisotropy at $S = 3, 6$. However, this alignment, measured as the angle between the truth-FIM stiff subspace and the posterior covariance's smallest-variance subspace, deteriorates to 83.4° at $S = 10$. These diagnostics reveal that while the posterior learns valuable updates beyond the prior, its predictive validity and uncertainty orientation struggle at larger system sizes (Appendix Table 7).

Going beyond additive gLV observations, Appendix K reports a zero-shot shift to correlated multiplicative observations, and Appendix L reports separately trained FHN and Lorenz–63 benchmarks. These targeted checks broaden the tested conditions without establishing general real-world transfer; their validity and tail results expose nonuniform gains.

5 Generalization to Real Data

We evaluated OIL, TM, and MPNN on real-world data from the Hudson's Bay hare–lynx dataset Carpenter (2018), which comprises 21 annual observations (1900–1920) of snowshoe hare and Canadian lynx pelt counts, expressed in thousands as proxies for population abundance. To train our models, we generated 100,000 synthetic two-species Lotka–Volterra trajectories by jointly sampling the four dynamical parameters, initial populations, and observation-noise scales from the Stan reference posterior fitted to the Hudson's Bay observations via posteriordb Magnusson et al. (2024). We applied correlated perturbations to double the posterior standard deviations in log space.

Training relied on noise-free trajectories sampled annually over a 20-year period, while a separately generated pool provided 1,000 validation and 1,000 test trajectories. We trained OIL on nested subsets ranging from 100 to 100,000 trajectories, consistently utilizing 3, 10, or 30 differentiable TM updates across both training and inference. The neural models were trained for up to 100 epochs, incorporating early stopping if validation performance plateaued for 10 epochs. Baseline models included parameter-error-only MPNNs, TM initialized at the full synthetic pool's mean parameters with matched update budgets, and unrefined mean parameters.

Performance was assessed using parameter MSE, trajectory RMSE, and simulation validity on the synthetic validation data. For the real-world observations, we evaluated trajectory RMSE and validity while fixing initial states to the first input observation; because the real data lacks ground-truth parameters, evaluations here were strictly limited to trajectory recovery. On the synthetic validation set, the MPNN narrowly outperformed OIL (trained on 10,000 samples with 30 unrolling steps) in parameter RMSE, but this same OIL configuration outperformed all other methods in trajectory MSE. This advantage held on the real-world data, where OIL (10,000 samples, 30 steps) best recovered the physical dynamics of the time series, outperforming TM given an equivalent step budget and avoiding the catastrophic failure observed with the MPNN.

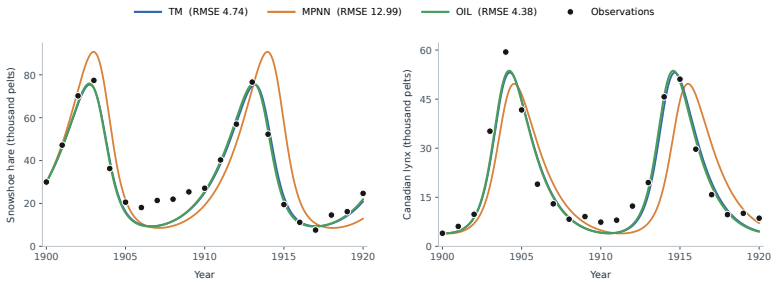


Figure 3: Trajectory alignment of predicted parameters from TM, MPNN, and OIL versus ground truth observations of the Hudson Bay lynx-hare dataset.

6 Discussion and Conclusion

Our gLV benchmark ($S \in \{3, 6, 10\}$) demonstrates that under practical non-identifiability, Euclidean parameter accuracy, trajectory fit, numerical validity, and posterior quality are strictly distinct evaluation targets. Accurate point predictions frequently produce invalid forward trajectories. The prior-whitened trajectory-sensitivity FIM complements these metrics by characterizing local sensitivity concentration (entropy rank) and subspace orientation (principal angles), though these diagnostics describe where estimates land rather than establishing global structural non-identifiability.

Implications for method selection. Method choice depends on the downstream task. Raw MPNN favors single-pass latency and Euclidean point recovery. TM and OIL are preferable when a numerically valid, trajectory-reproducing parameterization is required. NPE should be utilized and assessed for posterior uncertainty, not evaluated primarily by its posterior mean. OIL's primary advantage is not universal point accuracy, but rather yielding an optimization-compatible initialization for difficult high-dimensional regimes. Because OIL explicitly combines amortized prediction with instance-specific downstream optimization, its end-to-end computational cost inherently exceeds that of single-pass inference.

Limitations. Evaluations rely heavily on synthetic, fully observed, regularly sampled trajectories with known initial states. Crucially, difficult error tails and invalid rollouts persist: OIL's 6.38% invalid rate at $S = 10$ confirms that improved median trajectory fit and FIM alignment do not eliminate extreme failures. Consequently, valid-only tail summaries must be interpreted alongside complete-cohort validity scores. Finally, reported timings are implementation-specific operational measurements, requiring careful distinction between offline training cost, forward-pass latency, and optimizer-refined inference.

Future work. Priority directions include partial real-world observation designs, jointly trained mixed-S estimators, and refinement strategies to truncate high-dimensional error tails. Combining learned initializations with posterior diagnostics may help distinguish useful trajectory-fitting representatives from unresolved parameter uncertainty.

Conclusion. This work provides a controlled diagnosis of where inverse methods land under practical non-identifiability. By introducing a task-aligned initializer evaluated against ordinary warm starts under strictly matched downstream budgets, we demonstrate that competitive parameter accuracy does not guarantee reliable system simulation. We advocate choosing methods according to their downstream purpose and evaluating them holistically across typical accuracy, uncertainty geometry, numerical validity, and extreme error tails.

AI USE STATEMENT

Generative AI tools (OpenAI Codex) assisted with literature discovery, experimental planning, code implementation and debugging, result aggregation and interpretation, figure and table preparation, and manuscript drafting and editing. Numerical results were computed by the simulation and evaluation pipelines described in this paper, rather than generated as language-model responses. The authors take responsibility for the content, claims, and artifacts in the submission.

ETHICS STATEMENT

This study uses synthetic ODE trajectories and does not involve human participants or personal data. Its controlled benchmarks do not establish suitability for ecological or clinical decision-making with real observations; the scope and limitations are discussed in Section 6.

REPRODUCIBILITY STATEMENT

The observation models and failure-aware evaluation are specified in Section 3.1. Appendix A describes data generation, parameter priors, and fold construction. Appendix E details the OIL implementation and numerical safeguards; Appendix G reports inference budgets, timing conventions, validity, and error tails. Appendices J, K, and L specify the posterior, observation-shift, and cross-system evaluations. The captions distinguish full, matched, and sensitivity-valid cohorts and document the remaining numerical qualifications.

REFERENCES

Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International conference on machine learning*, pp. 136–145. PMLR, 2017.

Vanni Bucci, Belinda Tzen, Ning Li, Matt Simmons, Takeshi Tanoue, Elijah Bogart, Luxue Deng, Vladimir Yeliseyev, Mary L Delaney, Qing Liu, et al. Mdsine: Microbial dynamical systems inference engine for microbiome time-series analyses. *Genome biology*, 17(1):121, 2016.

Bob Carpenter. Predator-prey population dynamics: the lotka-volterra model in stan. *See https://mc-stan. org/users/documentation/case-studies/lotka-volterra-predator-prey. html (accessed 28 August 2019)*, 2018.

Michael Deistler, Jan Boelts, Peter Steinbach, Guy Moss, Thomas Moreau, Manuel Gloeckler, Pedro LC Rodrigues, Julia Linhart, Janne K Lappalainen, Benjamin Kurt Miller, et al. Simulation-based inference: A practical guide. *arXiv preprint arXiv:2508.12939*, 2025.

Priya Donti, Brandon Amos, and J Zico Kolter. Task-based end-to-end model learning in stochastic optimization. *Advances in neural information processing systems*, 30, 2017.

Kevin Egan, Weizhen Li, and Rui Carvalho. Automatically discovering ordinary differential equations from data with sparse regression. *Communications Physics*, 7(1):20, 2024.

Pedro Fontanarrosa and Chris P Barnes. Exploring accuracy and uncertainty quantification in physics-informed neural networks for inferring microbial community dynamics. *arXiv preprint arXiv:2509.02282*, 2025.

Travis Gibson and Georg Gerber. Robust and scalable models of microbiome dynamics. In *International conference on machine learning*, pp. 1763–1772. PMLR, 2018.

Pedro J Gonçalves, Jan-Matthis Lueckmann, Michael Deistler, Marcel Nonnenmacher, Kaan Öcal, Giacomo Bassetto, Chaitanya Chintaluri, William F Podlaski, Sara A Haddad, Tim P Vogels, et al. Training deep neural density estimators to identify mechanistic models of neural dynamics. *elife*, 9:e56261, 2020.

Ryan N Gutenkunst, Joshua J Waterfall, Fergal P Casey, Kevin S Brown, Christopher R Myers, and James P Sethna. Universally sloppy parameter sensitivities in systems biology models. *PLoS computational biology*, 3(10):e189, 2007.

Stefan Henneking, Sreeram Venkat, and Omar Ghattas. Goal-oriented real-time bayesian inference for linear autonomous dynamical systems with application to digital twins for tsunami early warning. *Journal of Computational Physics*, pp. 114682, 2026.

Dimitris Kastoris, Kostas Giotopoulos, and Dimitris Papadopoulos. Neural network-based parameter estimation in dynamical systems. *Information*, 15(12):809, 2024.

Patrick Kidger. On neural differential equations. *arXiv preprint arXiv:2202.02435*, 2022.

Yoon Kim, Sam Wiseman, Andrew Miller, David Sontag, and Alexander Rush. Semi-amortized variational autoencoders. In *International Conference on Machine Learning*, pp. 2678–2687. PMLR, 2018.

Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael W Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in neural information processing systems*, 34:26548–26560, 2021.

Jan-Matthis Lueckmann, Jan Boelts, David Greenberg, Pedro Goncalves, and Jakob Macke. Benchmarking simulation-based inference. In *International conference on artificial intelligence and statistics*, pp. 343–351. PMLR, 2021.

Måns Magnusson, Jakob Torgander, Paul-Christian Bürkner, Lu Zhang, Bob Carpenter, and Aki Vehtari. posteriordb: Testing, benchmarking and developing bayesian inference algorithms. *arXiv preprint arXiv:2407.04967*, 2024.

Amogh Pandey. *TranDynaMo: A comparative study of transformer and GRU performance in modeling dynamical systems*. PhD thesis, University of Illinois at Urbana-Champaign, 2023.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

Stefan T Radev, Ulf K Mertens, Andreas Voss, Lynton Ardizzone, and Ullrich Köthe. Bayesflow: Learning complex stochastic models with invertible neural networks. *IEEE transactions on neural networks and learning systems*, 33(4):1452–1466, 2020.

Stefan T Radev, Marvin Schmitt, Lukas Schumacher, Lasse Elsemüller, Valentin Pratz, Yannik Schälte, Ullrich Köthe, and Paul-Christian Bürkner. Bayesflow: Amortized bayesian workflows with neural networks. *arXiv preprint arXiv:2306.16015*, 2023.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. 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.

Rajiv Sambharya, Georgina Hall, Brandon Amos, and Bartolomeo Stellato. Learning to warm-start fixed-point optimization algorithms. *Journal of Machine Learning Research*, 25(166):1–46, 2024.

Richard R Stein, Vanni Bucci, Nora C Toussaint, Charlie G Buffie, Gunnar Rätsch, Eric G Pamer, Chris Sander, and Joao B Xavier. Ecological modeling from time-series inference: insight into dynamics and stability of intestinal microbiota. *PLoS computational biology*, 9(12):e1003388, 2013.

Alvaro Tejero-Cantero, Jan Boelts, Michael Deistler, Jan-Matthis Lueckmann, Conor Durkan, Pedro J Gonçalves, David S Greenberg, and Jakob H Macke. Sbi–a toolkit for simulation-based inference. *arXiv preprint arXiv:2007.09114*, 2020.

Mark K Transtrum, Benjamin B Machta, Kevin S Brown, Bryan C Daniels, Christopher R Myers, and James P Sethna. Perspective: Sloppiness and emergent theories in physics, biology, and beyond. *The Journal of chemical physics*, 143(1), 2015.

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

A gLV Data Generation and Rejection

We generated synthetic gLV trajectories independently for each system size $N \in \{3, 6, 10\}$. For each sample we drew parameters $\boldsymbol{\theta} = (\boldsymbol{\mu}, A)$ from independent uniform priors inspired from previous microbial gLV studies (Stein et al., 2013; Bucci et al., 2016): growth rates $\mu_i \sim \mathcal{U}(0.8, 1.6)$, off-diagonal interaction coefficients $M_{ij} \sim \mathcal{U}(-0.03, 0.06)$ for $i \neq j$, and self-interactions $M_{ii} \sim \mathcal{U}(-0.16, -0.04)$. The negative self-interaction prior ensures that, in expectation, each species has a logistic-type carrying capacity, while the off-diagonal prior assigns twice as much probability to positive as to negative interactions. For each sample we drew an initial state $N_0 \in \mathbb{R}^N$ with each component sampled independently from $\mathcal{U}(5, 15)$, so that initial-condition variability contributes to the inverse problem alongside parameter variability.

Each $(\boldsymbol{\theta}, N_0)$ pair was integrated over $t \in [0, 10]$ with 101 evenly spaced timepoints using SciPy's `odeint` (LSODA). Samples whose trajectories produced any non-finite value or whose maximum abundance exceeded 10^3 within the integration window were rejected and re-drawn from the prior. This rejection is rare in practice and applies only at data generation time and downstream methods are evaluated on the resulting numerically valid sample population. These clean trajectories also supply the additive-noise observations defined in Eq. (3). We generated $100{,}000$ samples per system size for $N \in \{3, 6, 10\}$, partitioned into consistent 5-fold cross-validation into 70,000 training, 10,000 validation, and 20,000 test instances per fold in downstream training and inference tasks.

The same prior bounds and integration settings were used for the parameter-MSE-trained baselines, the test-time inference benchmark, the FIM analysis, and the fine-tuning experiments, so all method comparisons are conducted on the same parameter and trajectory distributions.

B PINNs Fail at gLV Inversion

The inverse gLV problem is, at its core, a problem of relating trajectory time derivatives to species products: rearranging Eq. (1) gives $\dot{x}_i(t)/x_i(t) = \mu_i + \sum_j A_{ij} x_j(t)$, so any inverse method must, implicitly or explicitly, supply an estimate of $\dot{x}(t)$. PINN-based methods produce this estimate by autodifferentiating a learned surrogate $\tilde{x}_\phi(t)$, a step typically blamed for poor performance on gLV inversion via the multi-loss training problem (Wang et al., 2021; Krishnapriyan et al., 2021). We sharpen this characterization by comparing the derivative signals produced by PINN-style surrogates against a classical numerical alternative.

We compare three estimators of the gLV right-hand side $\dot{x}(t)$ against finite differences of the noiseless ground-truth trajectory: **Spline + FD** fits a smoothing spline to the observed trajectory and evaluates its derivative analytically; **PINN autodiff** autodifferentiates a surrogate trained jointly under the physics-residual loss in the standard inverse PINN configuration; and **MLP autodiff** autodifferentiates a surrogate trained only on observed trajectory data (equivalent to the asynchronous PINN's surrogate phase), removing the multi-loss training problem. For each estimator we report the cosine and Euclidean distances to the ground-truth derivative at each observed time point, aggregated across $10{,}000$ test samples for $N \in \{3, 6, 10\}$.

Figure 4 shows this behavior at $N \in \{3, 6, 10\}$. PINN autodiff is consistently the worst estimator, with mean Euclidean distance roughly an order of magnitude larger than spline+FD (5.6×10^{-2} vs 5.7×10^{-3} at N=3; 6.5×10^{-1} vs 4.8×10^{-2} at N=10); cosine distance shows the same separation. MLP autodiff substantially closes the gap, with mean Euclidean distance within $1.1\times$–$1.7\times$ of spline+FD, confirming that most of the standard PINN's derivative error is attributable to multi-loss training rather than to surrogate expressivity. However, the MLP distribution remains statistically distinguishable from spline+FD at every species count (paired t-test, $p < 0.05$ at N=3, $p < 10^{-100}$ at $N \in \{6, 10\}$), and the gap widens with N. The neural-surrogate route to derivative estimation is therefore, at best, a slightly degraded version of differentiating the trajectory directly; any inverse method that consumes such a derivative inherits this gap before parameter optimization begins, motivating the trajectory-direct baselines benchmarked in Section 3.3.

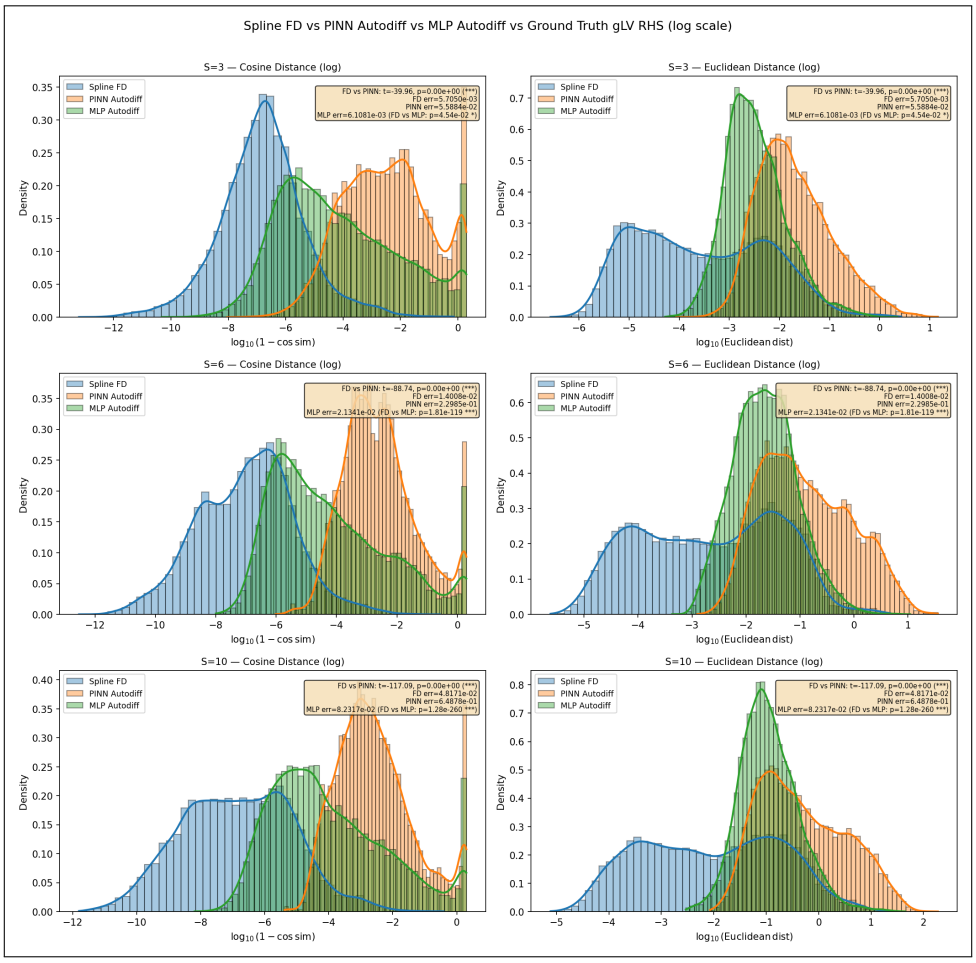


Figure 4: **Neural-surrogate derivatives are not a viable replacement for direct numerical differentiation in gLV inverse problems.** Any inverse method for the gLV system must, implicitly or explicitly, supply an estimate of $\dot{x}(t)$ to fit; PINNs supply this estimate by autodifferentiating a learned surrogate. We compare three estimators of the gLV right-hand side against finite differences of the noiseless ground-truth trajectory: **spline + FD** (blue), the autodifferentiated surrogate of a **standard inverse PINN** (orange), and the autodifferentiated surrogate of an **MLP** trained on trajectory data with no physics-residual loss (green). Distributions of cosine distance (left column) and Euclidean distance (right column) are shown on a log_{10} scale, for $S \in \{3, 6, 10\}$ species (top to bottom). All pairwise differences against spline + FD are statistically significant (paired t-test; inset values). Any inverse method that consumes a neural-surrogate derivative inherits this gap before parameter optimization begins.

C Test-time Inference Method Definitions

Direct trajectory matching. The most classical approach to ODE parameter inference, direct trajectory matching estimates $\boldsymbol{\theta} = (\mu, A)$ by minimizing the squared error between observed trajectories and forward simulations of Eq. (1) under candidate parameters,

$$\hat{\boldsymbol{\theta}}_{\mathrm{TM},m} = \arg\min_{\boldsymbol{\theta}} \sum_{k=1}^{T} \left\| x^{(m)}(t_k) - x^{(m)}_{\boldsymbol{\theta}}(t_k) \right\|_2^2, \tag{15}$$

where $x^{(m)}_{\boldsymbol{\theta}}(t)$ is the trajectory obtained by integrating Eq. (1) from $x^{(m)}(0)$ under parameters $\boldsymbol{\theta}$. We perform forward integration with `diffrax` (Kidger, 2022), optimizing $\boldsymbol{\theta}$ from a shared initialization $\boldsymbol{\theta}_0$ via L-BFGS-B with bound constraints set to the parameter prior support (Appendix A), with

gradients backpropagated through the ODE solver. Each system is optimized separately from its one observed trajectory. Trajectory matching requires no derivative estimate but is the most expensive at inference time, as each gradient step entails a forward ODE solve and a reverse-mode pass through it.

Gradient matching. The authenticated GM* results use central finite differences of the observations and iterative L-BFGS fitting of the vector-field residual, with a saved initialization per system. This avoids repeated forward simulation during fitting but inherits error in the derivative estimates. GM* is not closed-form spline regression; the spline-based derivative diagnostic in Appendix B is a separate historical comparison.

Inference time. Current matched-budget inference and training costs are reported separately in Appendix G; these are implementation-specific measurements.

D Amortized Method Architectures

MLP regressor. We flatten each single-trajectory observation to $\mathbb{R}^{T \cdot N}$ and train an MLP under MSE loss to regress directly to $\boldsymbol{\theta} \in \mathbb{R}^{N+N^2}$. This baseline ignores both the temporal structure of trajectories and the relational structure between species, and serves as a reference for the more structured architectures.

Transformer regressor. Trajectories are sequences, and transformer-based encoders are now standard for amortized inference on dynamical systems (Radev et al., 2020). We treat each time point $x(t_k) \in \mathbb{R}^N$ as a token, augment with sinusoidal positional encodings of t_k, and pool across time before a regression head emits $\hat{\boldsymbol{\theta}}$.

Temporal encoded MPNN with attention. This model is matched to the graph structure of pairwise species interactions, we propose a message-passing architecture composed of three modules: a temporal encoder for per-species trajectory embedding and an attention-weighted MPNN over the species-interaction graph.

Temporal encoder. Each observed species' trajectory $x_i^{(m)}(t_{1:T})$ is independently encoded into a fixed-dimensional embedding via a three-layer 1D convolutional network followed by masked average pooling. Under full-length observations, this reduces to standard mean pooling over the time axis. Under variable-length conditions)where trajectories shorter than T are zero-padded) the pooling is restricted to valid timesteps using a per-node length indicator, preventing dilution of the learned representation by uninformative padding.

Attention-weighted MPNN. Initial node embeddings (from either the temporal encoder or the learned prior) are refined through L rounds of message passing on a fully connected directed graph where each species is a node and all N^2 directed edges (including self-loops) represent candidate pairwise interactions. At each layer, a pairwise message $m_{ji} = \mathrm{MLP}_{\mathrm{msg}}(h_i \,\|\, h_j)$ is computed for every directed edge ($\|$ denotes concatenation). Messages are aggregated via learned attention weights $\alpha_{ji} = \mathrm{softmax}_j\big(\mathrm{MLP}_{\mathrm{attn}}(h_i \,\|\, h_j)\big)$, which allow the model to selectively upweight informative neighbors. Node embeddings are updated via

$$h_i^{(\ell+1)} = \mathrm{MLP}_{\mathrm{update}}\big(h_i^{(\ell)} \,\|\, a_i\big) + h_i^{(\ell)}, \qquad a_i = \sum_j \alpha_{ji}\, m_{ji}, \tag{16}$$

with the residual connection preserving information from earlier layers to mitigate over-smoothing on densely connected graphs. After L rounds, per-node growth rates are predicted via $\hat{\mu}_i = \mathrm{MLP}_{\mu}\big(h_i^{(L)}\big)$ and per-edge interaction coefficients via $\hat{A}_{ij} = \mathrm{MLP}_{A}\big(h_i^{(L)} \,\|\, h_j^{(L)}\big)$, directly producing the full N-dimensional growth-rate vector and $N \times N$ interaction matrix in a single forward pass.

Training. Training of these baseline, amortized models were done on a RTX Pro Blackwell 4500 GPU for a total of ~20 GPU hours total.

E OIL IMPLEMENTATION AND NUMERICAL SAFEGUARDS

Encoder and outer optimization. The gLV OIL encoder is a four-layer MPNN with hidden dimension 128. Fine-tuning starts from the corresponding parameter-supervised checkpoint and runs for 50 outer epochs with AdamW, learning rate 6.25×10^{-5}, weight decay 10^{-4}, and 200 warm-up steps. Checkpoint candidates are evaluated every five epochs on the saved validation split alone. These settings describe the gLV MPNN implementation and carry no implied transfer to other architectures or ODE families.

Bounded native outputs and prior-logit refinement. Let $\boldsymbol{\ell}$ and $\boldsymbol{u}$ contain the coordinatewise parameter bounds given by the gLV proposal supports in Eq. (2). The encoder output passes through `SafeTheta`, which produces bounded native parameters using the implementation's straight-through gradient. Inner refinement then operates in prior-logit coordinates, so the gradient steps act on unconstrained coordinates while the bounds are enforced by the coordinate map itself. Writing $\mathrm{sigmoid}$ elementwise, that map is

$$\boldsymbol{\theta} = b_S(\mathbf{z}) = \boldsymbol{\ell} + (\boldsymbol{u} - \boldsymbol{\ell}) \odot \mathrm{sigmoid}(\mathbf{z}), \tag{17}$$

with the initial prediction converted under the implementation's finite boundary handling. All parameter supervision is evaluated after mapping back to native parameters. Bounded parameters alone cannot guarantee a numerically valid forward trajectory, so rollout and gradient safeguards remain necessary.

Five inner updates and batch reduction. The training unroll contains five inner updates with learning rate $\eta = 3 \times 10^{-5}$. For a training batch of size B_{tr}, let $\ell_m^{(k)}$ denote the implemented per-sample trajectory loss at inner step k. The inner gradient is computed from the sum

$$\mathbf{G}^{(k)} = \nabla_{\mathbf{Z}^{(k)}} \sum_{m=1}^{B_{\mathrm{tr}}} \ell_m^{(k)}, \tag{18}$$

where $\mathbf{Z}^{(k)}$ stacks the per-system prior-logit coordinates. For an active, finite row, the update takes the form

$$\mathbf{z}_m^{(k+1)} = \mathbf{z}_m^{(k)} - \eta\, \mathbf{G}_m^{(k)}, \qquad k = 0, \ldots, 4, \tag{19}$$

subject to the safeguards below. Because each simulated system depends only on its own parameter row, sum reduction supplies each row's per-sample gradient without an additional $1/B_{\mathrm{tr}}$ factor; under a batch-mean inner loss, the effective inner step size would instead vary with batch size. This inner sum reduction is distinct from the fixed reductions used to aggregate the outer training losses.

Parameter and trajectory training losses. Growth rates and interaction coefficients are standardized as separate parameter blocks before computing $\mathcal{L}_\theta$, and the same block-standardized loss is applied at $\boldsymbol{\theta}^{(0)}$ and $\boldsymbol{\theta}^{(5)}$. For trajectory supervision, simulated and target states are processed by their respective scaling rules, transformed with asinh, and compared through a Huber penalty on the resulting residuals. The implementation uses fold-training-derived scales and a finite per-sample trajectory-loss cap, and the simulated-state and target-state scaling operations are specified separately, so a common normalization should not be assumed in their place. Together with the parameter standardization, these operations define the losses entering the equal-weight objective in Eq. (13). All of them are training conventions: the block-standardized parameter loss is distinct from the native-coordinate parameter RMSE used for evaluation, and the transformed, capped Huber penalty is distinct from the untransformed trajectory MSE reported in the benchmark. The registered recovery metric and invalid-rollout penalty remain as declared.

Training discretization. Differentiable training uses batched, fixed-grid fourth-order Runge–Kutta (RK4) integration, with the trajectory loss evaluated at 21 points and 20 RK4 substeps between successive loss points. The gLV right-hand sides are evaluated jointly across the batch using batched matrix operations, while the integration steps retain their sequential time dependence. The training loss grid is separate from the full-resolution validation and scoring grids.

Failure handling and gradient gates. The unroll maintains sticky per-sample failure masks: once a row fails the relevant numerical checks, its failure status persists through the remainder of the unroll, and last-finite row incumbents supply numerical fallbacks while leaving that status intact. Nonfinite adjoint rows are zeroed, and fail-closed gradient gates suppress unsafe gradient contributions or

Table 3: Matched-budget optimization on 100,000 clean out-of-fold systems per S. Entries at B are median retained optimizer objectives, not independently rescored canonical trajectory MSE. B counts combined objective/gradient calls. Time is median terminal optimization seconds per system; training is median successful-fold pipeline minutes.

S	Initializer	$B = 32$	$B = 128$	$B = 512$	$B = 1200$	Time (s)	Train (min)
3	Prior	1.06×10^{-2}	3.15×10^{-4}	2.16×10^{-5}	5.38×10^{-6}	1.58	–
3	MPNN	4.73×10^{-4}	1.98×10^{-5}	3.63×10^{-6}	1.42×10^{-6}	1.24	3.45
3	OIL	4.69×10^{-4}	1.94×10^{-5}	3.59×10^{-6}	1.40×10^{-6}	1.39	45.19
6	Prior	1.32	3.61×10^{-2}	4.52×10^{-3}	1.61×10^{-3}	8.04	–
6	MPNN	8.64×10^{-2}	4.81×10^{-3}	8.41×10^{-4}	2.90×10^{-4}	5.90	3.96
6	OIL	2.90×10^{-1}	7.61×10^{-3}	1.10×10^{-3}	3.61×10^{-4}	8.67	89.47
10	Prior	1.23×10^{2}	1.21	1.32×10^{-1}	3.69×10^{-2}	52.01	–
10	MPNN	1.80×10^{1}	2.72×10^{-1}	3.74×10^{-2}	1.29×10^{-2}	41.10	5.37
10	OIL	1.74×10^{1}	2.40×10^{-1}	3.39×10^{-2}	1.11×10^{-2}	36.66	327.36

updates before nonfinite values can propagate through the training computation. Loss reductions retain their prescribed fixed denominators and are never renormalized by the number of surviving rows, so a change in the surviving-row count leaves the remaining rows' losses unscaled. These safeguards stabilize the training computation without certifying that a predicted parameter vector will pass the independent evaluation solver, and a retained numerical fallback is never counted as a successful held-out rollout on the grounds of being finite.

Validation and checkpoint selection. Checkpoint candidates are assessed with full-resolution adaptive Dopri5 integration, which separates model selection from the fixed-grid RK4 training discretization and its loss grid. The validation split is the sole source of checkpoint-selection information, and test systems play no part in it.

Independent inference and scoring. Inference-time trajectory matching uses bounded L-BFGS-B directly in native parameter coordinates, and both inference and scoring use an independent float64 `diffrax`/`Tsit5` implementation. The trajectory-matching objective is the registered untransformed observation-space loss,

$$\mathcal{L}_{\mathrm{TM}}\left(\boldsymbol{\theta}; \mathbf{X}^{\star}, \mathbf{x}_0\right) = \frac{1}{ST}\left\| f_{\mathrm{test}}(\boldsymbol{\theta}, \mathbf{x}_0) - \mathbf{X}^{\star}\right\|_F^2 \tag{20}$$

for valid solves, with invalid solves handled by the common scoring protocol; the training-time state transformation and Huber penalty leave this test-time objective untouched. The primary inference budget is $B = 1{,}200$, and budget ablations use $B \in \{32, 128, 512, 1{,}200\}$. These quantities count combined objective/gradient calls, distinct from L-BFGS-B iterations, RK4 substeps, and differentiable training updates. Ordinary and OIL-trained warm starts are passed to the same downstream optimizer under the same registered budget, raw encoder outputs and optimizer-refined outputs are stored and reported separately, and offline training cost is kept distinct from inference wall-clock time.

F Expanded Clean Recovery

Figure 3 reports native parameter RMSE and absolute log-trajectory errors on 1,000 matched systems per species count. Figure 5 expands the method comparison to the full 100,000-system out-of-fold cohort. This companion retains percentage parameter improvement relative to the saved prior draw, whereas its trajectory panel uses the same log-error metric as the main figure.

G Matched Budgets, Compute, and Error Tails

Table 3 compares prior, ordinary MPNN, and OIL initializations under the same downstream optimizer and combined objective/gradient-call budgets. These are pipeline comparisons, not isolated ablations of the three loss terms or numerical safeguards. The milestone curves retain the optimizer objective; Table 4 instead uses independent canonical scoring of the final parameters.

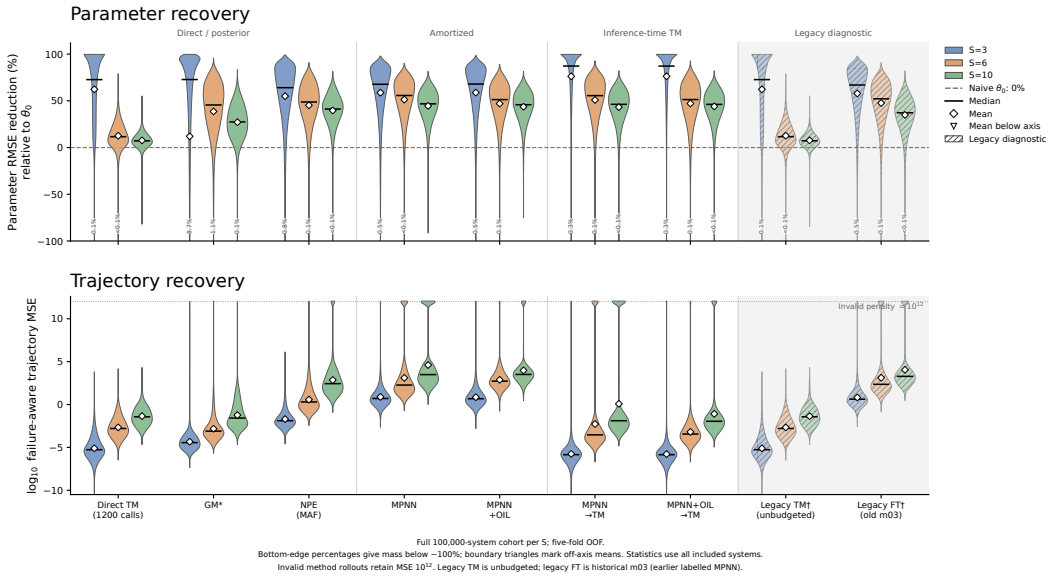


Figure 5: Expanded clean recovery. Parameter improvement is relative to the registered per-system prior draw; trajectory values are $\log_{10}$ failure-aware MSE. Black bars are medians and white diamonds are means of the displayed per-system quantity, not logarithms of mean MSE. Invalid rollouts retain 10^{12}. Raw learned predictions and their TM continuations are separate readouts; current TM continuations use $B = 1200$. Hatched historical columns are diagnostic, not matched-budget competitors.

Table 4: Clean canonical trajectory-error tails at $B = 1200$ for TM and OIL; MPNN is raw. Each row uses all 100,000 out-of-fold systems. The full-cohort columns retain invalid-rollout MSE 10^{12}. The last three columns are explicitly conditional on a valid canonical rollout.

			Full cohort				Valid only		
S	Method	Valid (%)	Mean	p95	p99	p99.9	Mean	p95	p99
3	TM	100.000	0.2648	2.70×10^{-3}	0.0846	3.384	0.2648	2.70×10^{-3}	0.0846
3	MPNN	99.485	5.15×10^{9}	558.1	4.02×10^{4}	10^{12}	3.91×10^{4}	461.7	1.19×10^{4}
3	OIL	99.602	3.98×10^{9}	8.03×10^{-5}	2.54×10^{-3}	10^{12}	3.12×10^{-3}	7.08×10^{-5}	1.02×10^{-3}
6	TM	100.000	1.413	0.3214	2.681	181.7	1.413	0.3214	2.681
6	MPNN	91.929	8.07×10^{10}	10^{12}	10^{12}	10^{12}	2.15×10^{5}	2.96×10^{4}	3.66×10^{5}
6	OIL	99.061	9.39×10^{9}	0.0486	261.7	10^{12}	2.503	0.0360	0.3253
10	TM	100.000	6.906	3.340	100.0	1175	6.906	3.340	100.0
10	MPNN	85.131	1.49×10^{11}	10^{12}	10^{12}	10^{12}	4.27×10^{5}	9.35×10^{4}	1.40×10^{6}
10	OIL	93.620	6.38×10^{10}	10^{12}	10^{12}	10^{12}	5.333	0.4364	5.230

The worst $\lceil 0.01 n_{\text{valid}} \rceil$ valid OIL rollouts account for 99.53%, 99.76%, and 98.47% of its valid-only total error at $S = 3, 6, 10$. This concentration does not remove the failures: at $S = 10$, 6,380 invalid systems dominate the full-cohort arithmetic mean through the fixed penalty. Valid-only and failure-aware summaries therefore answer different questions.

H Additive-Noise Recovery and Validity

All methods in a comparison receive the same noisy observation and the same 1,000 held-out systems per S. Trajectory error here measures observation fit, not latent denoising. Table 5 counts the registered failure-penalty scores directly; finite penalty values are not valid rollouts.

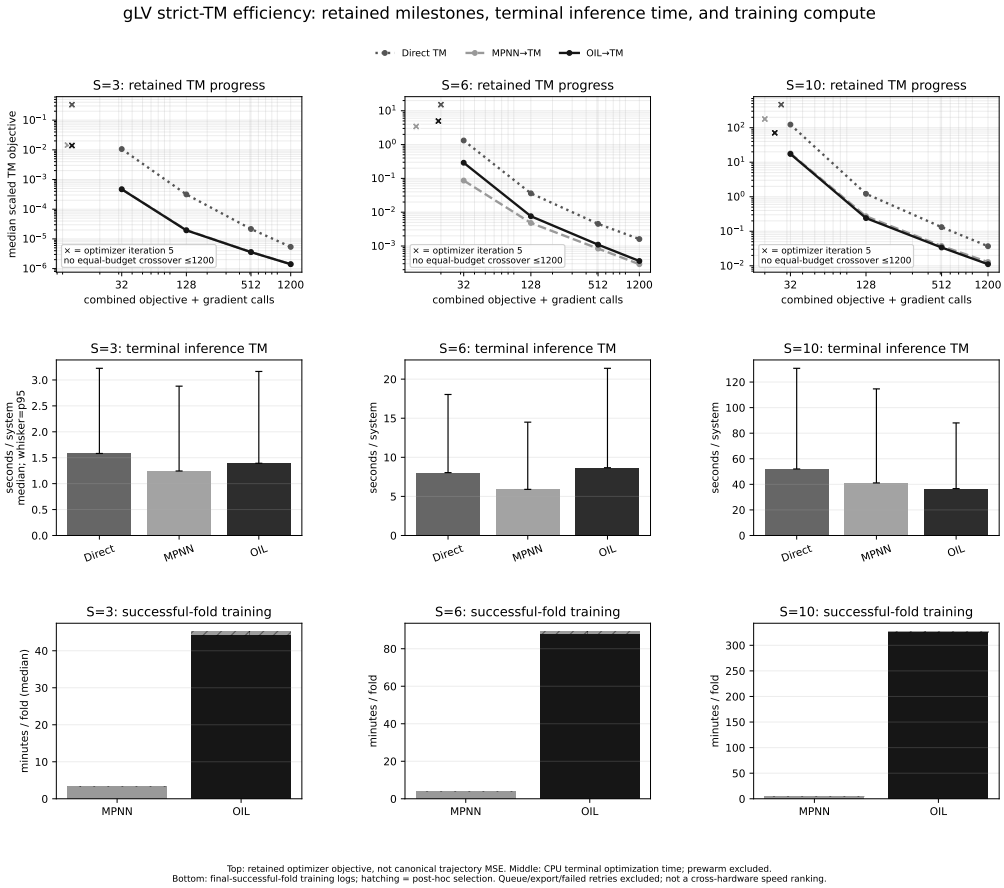


Figure 6: Optimization progress and operational cost. Top: retained optimizer objective at matched call budgets; crosses mark five optimizer iterations, which need not consume the same number of calls. Middle: terminal optimizer wall-clock median and p95, excluding worker prewarming; intermediate-budget times were not retained. Bottom: recorded training-stage costs with selection overhead hatched, excluding queue/export/failed retries. OIL fine-tuning does not include its pretrained MPNN stage. MPNN used RTX 3090 GPUs, OIL used H100 MIG allocations, and TM ran on CPUs; training bars are not a controlled cross-hardware speed ranking.

Table 5: Forward validity (%) on the matched clean/noisy cohort. Entries are ordered clean/0.1%/1%/5%, with 1,000 systems in every cell. Validity requires a finite registered trajectory score below 10^{12}; an optimizer completion flag alone is insufficient.

Method	$S = 3$	$S = 6$	$S = 10$
TM	100.0/100.0/100.0/100.0	100.0/100.0/100.0/100.0	100.0/100.0/100.0/100.0
MPNN	99.0/99.4/99.7/99.8	93.2/83.2/94.0/81.7	87.7/80.6/77.1/78.4
OIL	99.3/99.8/100.0/99.9	99.0/98.2/99.6/98.2	93.3/98.1/95.5/97.0

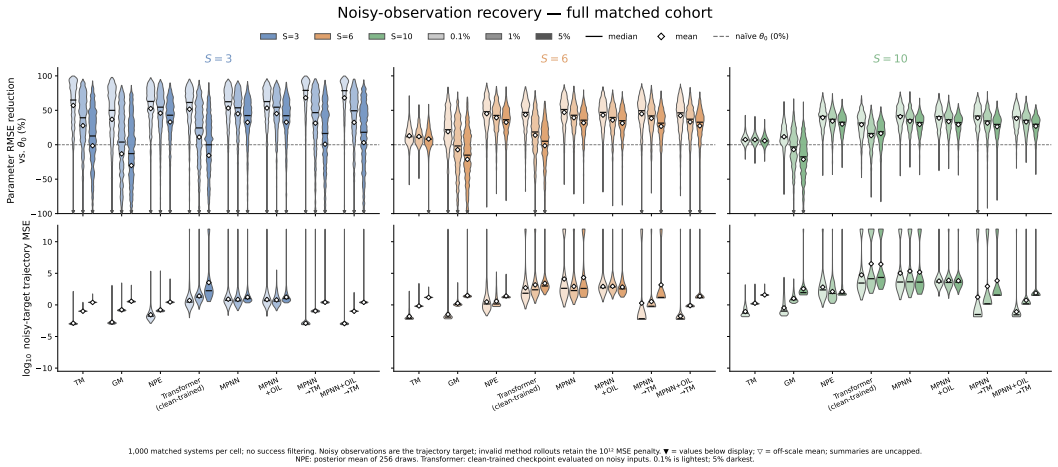


Figure 7: Matched noisy comparison: 1,000 systems per method, species count, and noise level, with no success filtering. Noise shades are lightest at 0.1% and darkest at 5%; species hues match the main figure. Black bars are medians and white diamonds are means of displayed values. Trajectory MSE targets the noisy observation and retains the 10^{12} failure atom. The Transformer is a clean-trained historical checkpoint tested under noise; NPE is the 256-draw posterior mean. Raw and TM-refined neural readouts are distinguished on the horizontal axis.

I Sensitivity Geometry and Subspace-Angle Distributions

Table 6 retains the earlier noise-dependent geometry summaries. Its finite-Jacobian screen did not certify complete sensitivity solves; it must not be interpreted as 1,000 solver-certified FIMs in every cell. The separate clean full-k analysis below applies a stricter complete-solve gate and has different, explicitly reported denominators.

For the full-k check, we integrate continuous forward sensitivities in float64 with DOP853 ($\mathrm{rtol} = 10^{-8}$, $\mathrm{atol} = 10^{-10}$) on the 101-point observation grid. Both truth and inferred FIMs use the same fold-training empirical-prior whitening. We require a complete finite state/sensitivity solve and a finite positive leading FIM eigenvalue. Truth and TM are valid on all 1,000 systems per S; MPNN is valid on $990/932/877$ and OIL on $993/990/933$ for $S = 3/6/10$. The shared comparison consequently contains $989/926/832$ systems.

A tighter-tolerance audit of 20 typical/hard systems per S changed the largest angle at the truth-selected stiff dimension by less than 3.92×10^{-6} degrees. Near-null high-k eigenspaces were substantially less stable (up to roughly 30° at $S = 10$); their fine structure is descriptive, not robust mechanistic evidence. Full-space and intersecting subspaces also force zero angles geometrically. Neither those zeros nor entropy effective rank establishes structural identifiability.

J Posterior Diagnostics

We assess NPE–MAF as a distribution, with the empirical accepted prior as a control. Marginal coverage alone does not establish conditional posterior correctness. TARP (Tests of Accuracy with Random Points) probes coverage through random reference points; its mean absolute coverage

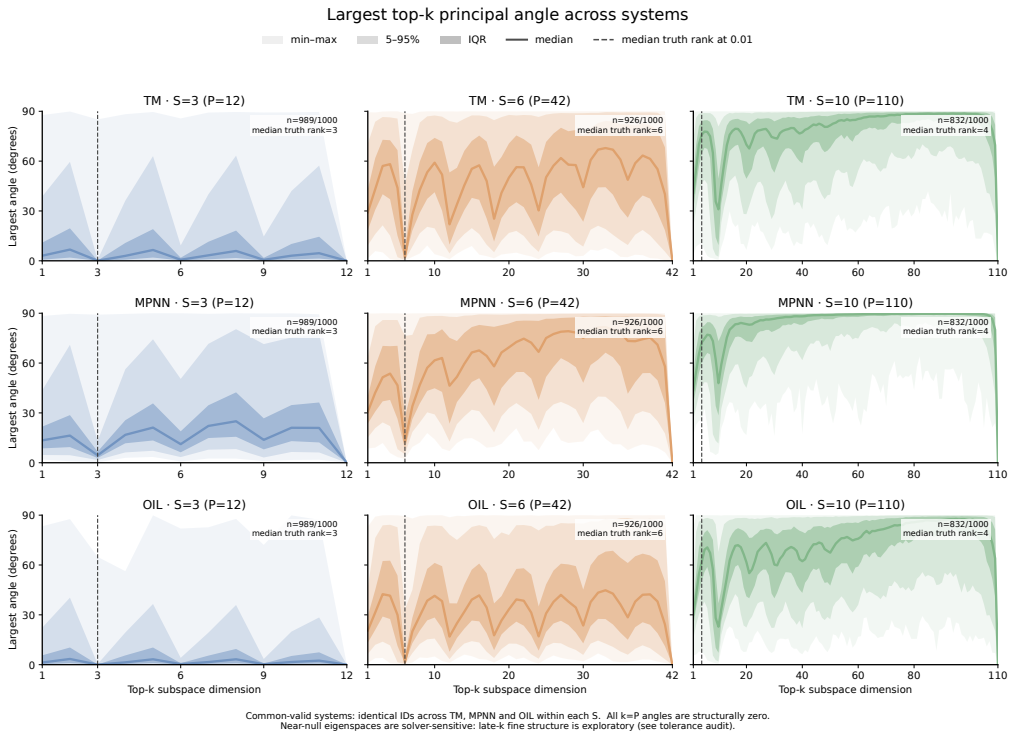


Figure 8: Largest principal angle between the top-k truth and inferred prior-whitened FIM subspaces, conditional on a valid FIM for all three methods. Columns use the same $989/926/832$ systems for $S = 3/6/10$; rows show TM, MPNN, and OIL. Lines are medians; bands show IQR, 5–95%, and min–max, not confidence intervals. Dashed guides mark median truth ranks $3/6/4$ at relative threshold 0.01, not a fixed dimension for every system. The full-space angle at $k = P$ is identically zero.

Table 6: Prior-whitened trajectory-sensitivity geometry on the matched 1,000-system cohort per S (200 systems per test fold). Entries within each cell are ordered clean / 0.1% / 1% / 5% additive noise. Panel A reports the median entropy effective rank; Panel B reports the across-system median largest truth-versus-inferred principal angle, evaluated at truth-selected dimension in Eq. (8). MPNN denotes the raw amortized prediction and OIL the OIL-trained MPNN initializer after downstream TM. These are the original finite-Jacobian summaries; the stricter complete-solve audit and its distinct denominators are reported in Appendix I.

Method	$S = 3$	$S = 6$	$S = 10$
A. Median entropy effective rank			
Truth	2.140 / 2.140 / 2.140 / 2.140	2.194 / 2.194 / 2.194 / 2.194	1.525 / 1.525 / 1.525 / 1.525
TM	2.162 / 2.127 / 2.079 / 1.981	2.329 / 2.335 / 2.227 / 1.981	1.720 / 1.740 / 1.712 / 1.587
MPNN	2.249 / 2.264 / 2.268 / 2.440	2.556 / 1.977 / 3.167 / 2.154	2.259 / 1.992 / 1.783 / 2.262
OIL	2.155 / 2.142 / 2.064 / 1.997	2.262 / 2.240 / 2.245 / 1.958	1.656 / 1.639 / 1.669 / 1.570
B. Median largest principal angle $\alpha_{\max}$ (degrees)			
TM	0.080 / 0.128 / 0.344 / 0.820	2.851 / 2.891 / 3.892 / 9.634	59.878 / 59.062 / 59.312 / 66.281
MPNN	4.546 / 4.366 / 4.070 / 5.006	17.205 / 18.989 / 16.010 / 19.239	59.407 / 63.724 / 65.054 / 69.622
OIL	0.046 / 0.094 / 0.312 / 0.788	2.212 / 2.228 / 3.712 / 9.585	45.615 / 48.304 / 48.910 / 60.930

Table 7: Clean NPE–MAF distributional diagnostics. Calibration uses 1,000 systems per S and 512 posterior draws per system. Posterior-predictive/FIM checks use 200 systems per S, 256 rollout draws per system, and 1,024/1,024/2,048 covariance draws for $S = 3/6/10$.

Metric	$S = 3$	$S = 6$	$S = 10$
50% marginal coverage	0.489	0.492	0.505
90% marginal coverage	0.892	0.898	0.901
95% marginal coverage	0.947	0.948	0.949
TARP MACE: NPE / prior	0.0293/0.0055	0.0573/0.0118	0.0233/0.0190
Energy score: NPE / prior	1.182/2.405	3.180/4.552	6.103/7.395
Valid predictive rollouts (%)	100.00	98.53	79.76
Median system-wise draw-median MSE	0.0381	11.96	2340.49
Median sloppy/stiff variance ratio	387.61	66.00	6.32
Median posterior–FIM angle (degrees)	21.23	30.57	83.36

error (MACE) is lower for the prior control here, whereas NPE improves the energy score. Both comparisons are retained in Table 7.

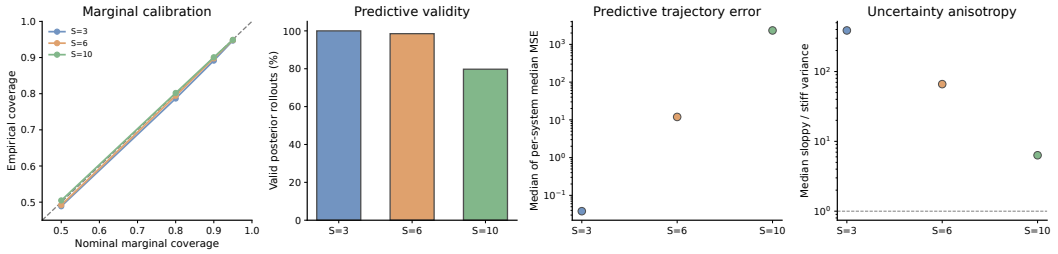


Figure 9: Clean NPE–MAF: marginal coverage, posterior-predictive validity, predictive error, and uncertainty anisotropy. Predictive error first takes the failure-aware median over 256 draws for each system, then the median across 200 systems; draws are not independent test systems. The variance ratio compares posterior variance per sloppy and per stiff direction of the truth FIM. Table 7 supplies the prior-control calibration and energy-score comparisons.

The posterior–FIM angle compares the truth stiff subspace with the posterior covariance's smallest-variance subspace, not with a point-estimate FIM. Near-nominal marginal coverage coexists with reduced predictive validity and weaker uncertainty orientation at $S = 10$.

K Zero-Shot Structured Observation Shift

Without retraining the 5%-Gaussian-trained estimators, we replace their test observations by mean-preserving multiplicative lognormal noise:

$$\eta_{0,s} \sim \mathcal{N}(0, \tau^2), \quad \eta_{t,s} = \rho\eta_{t-1,s} + \tau\sqrt{1-\rho^2}\, z_{t,s}, \quad y_{t,s} = x_{t,s}\exp(\eta_{t,s} - \tau^2/2), \tag{21}$$

where independent $z_{t,s} \sim \mathcal{N}(0,1)$, $\rho = 0.8$, and $\tau = \sqrt{\log(1+0.05^2)}$. This gives a 5% marginal coefficient of variation, heteroscedasticity, and temporal correlation; it is not exactly signal-to-noise matched to dataset-scaled additive Gaussian noise.

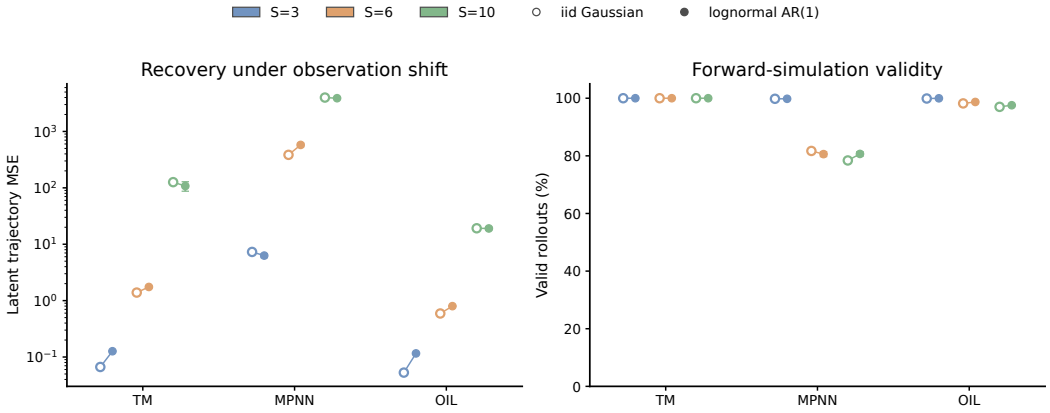


Figure 10: Zero-shot observation shift on the same 1,000 systems per S. Open markers show the iid-Gaussian control; filled markers show the mean and sample SD across three structured-noise seeds. Left: mean of seed-level medians of failure-aware clean-latent trajectory MSE. Right: seed-level valid percentages. Seeds reuse systems and do not constitute 3,000 independent examples. TM and OIL use $B = 32$; MPNN is the raw network, not MPNN+TM. All scores retain failed rollouts.

These checks broaden the observation conditions without establishing real-data generalization. Any unweighted trajectory-sensitivity geometry reported for this experiment is not the full likelihood FIM of the correlated lognormal observation model.

L Separately Trained Cross-System Benchmarks

For each family we evaluate 400 fixed held-out systems, 200 from each of folds 0 and 1 of the five-fold design. TM starts at the prior mean; OIL starts at its learned initializer. Both use native-coordinate bounded L-BFGS-B with strict $B = 32$ combined objective/gradient calls. MPNN is the raw estimate. These are separately trained clean benchmarks, not zero-shot transfer from gLV. All 400 final rollouts per method were valid.

L.1 FitzHugh–Nagumo

We infer (a, b, ϵ, I) in

$$\dot{v} = v - v^3/3 - w + I, \qquad \dot{w} = \epsilon(v + a - bw), \tag{22}$$

with $(v_0, w_0) = (-1, 1)$, independent uniform supports $[-1, 1] \times [0, 2] \times [0.01, 0.2] \times [0, 1.5]$, and 2,000 time points over $[0, 200]$. The 100,000-system dataset uses LSODA-generated clean trajectories; parameter RMSE is reported in native coordinates. OIL lowers the pooled median trajectory MSE by approximately $10\times$ relative to TM, but raw MPNN retains smaller p95 and p99 errors.

L.2 Lorenz–63

We infer (σ, ρ, β) in

$$\dot{x} = \sigma(y - x), \qquad \dot{y} = x(\rho - z) - y, \qquad \dot{z} = xy - \beta z, \tag{23}$$

Table 8: Separate clean cross-system benchmarks, each with 400 held-out systems (200 from each of folds 0 and 1). TM and OIL use $B = 32$; MPNN is raw. Every rollout was valid. Parameter RMSE is native for FHN and divided coordinatewise by prior ranges for Lorenz–63; the parameter columns are not directly comparable across families.

System	Method	Median parameter RMSE	Median trajectory MSE	p95 MSE	p99 MSE
FHN	TM	0.0044	1.757×10^{-6}	1.6988	2.0422
FHN	MPNN	0.0103	2.700×10^{-4}	0.3491	0.9869
FHN	OIL	0.0054	1.765×10^{-7}	1.5911	2.0513
Lorenz–63	TM	6.621×10^{-8}	1.621×10^{-13}	9.2763	17.7034
Lorenz–63	MPNN	7.140×10^{-4}	1.258×10^{-4}	0.0014	0.0072
Lorenz–63	OIL	6.450×10^{-8}	1.614×10^{-13}	3.032×10^{-13}	1.202×10^{-12}

Table 9: Cross-system geometry in fold-specific empirical-prior-whitened coordinates. All truth/method FIMs were valid (400/400 per family). Values are medians; angles are in degrees. FHN uses the per-system truth rank at $0.01\lambda_{\max}$. Lorenz uses fixed top-1/top-2 spaces; its truth rank is three in 80% of systems, making the full-space angle trivial. Displayed 0.00 values denote rounding, not exact identity.

System	Method	Entropy rank	FHN truth-rank angle	Lorenz top-1 / top-2 angle
FHN	Truth	1.541	–	–
FHN	TM	1.470	0.05	–
FHN	MPNN	1.543	0.56	–
FHN	OIL	1.591	0.07	–
Lorenz–63	Truth	1.546	–	–
Lorenz–63	TM	1.599	–	$0.00/0.00$
Lorenz–63	MPNN	1.546	–	$0.06/0.04$
Lorenz–63	OIL	1.546	–	$0.00/0.00$

with $(x_0, y_0, z_0) = (1, 1, 1)$ and independent uniform supports $[8, 12] \times [24, 32] \times [2, 10/3]$. The 100,000 clean targets use DOP853 at 401 points over $[0, 4]$; parameter RMSE divides each coordinate error by its prior range. Independent float64 LSODA scores final predictions. TM and OIL have practically indistinguishable median trajectory errors near the numerical floor; OIL's advantage is in the upper tail.

M Best/Worst Trajectories and Paired Rollout Examples

Error-tail membership uses canonical full-state, full-time trajectory MSE on the clean matched 1,000-system cohort. Each method independently selects its lowest- and highest-error 50 systems. Invalid predictions retain 10^{12}, and ties use a fixed hash ordering rather than visual selection. The tail groups therefore differ across methods and are not paired tests.

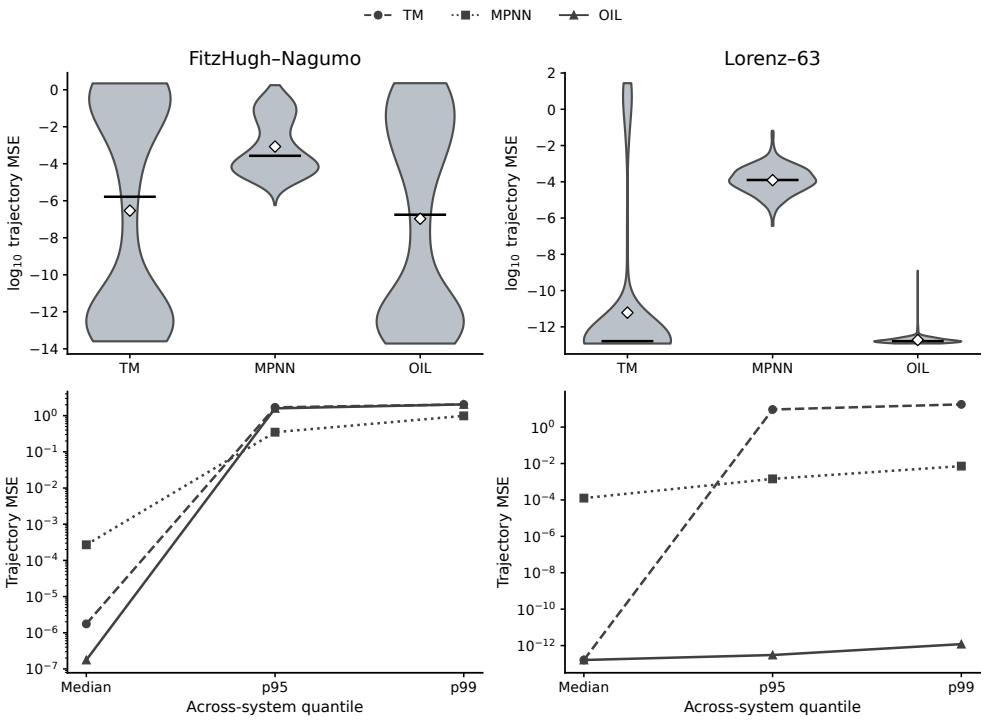


Figure 11: Cross-system recovery and tails on 400 paired systems per family. Top: distributions of $\log_{10}$ trajectory MSE, with black medians and white means of log-error. Bottom: median, p95, and p99 of untransformed MSE on logarithmic axes. Axis ranges differ across families. OIL improves typical FHN recovery but not the MPNN upper tail; on Lorenz–63 it removes the large TM tail without a meaningful median improvement at the scoring floor. Table 9 reports corresponding local geometry.

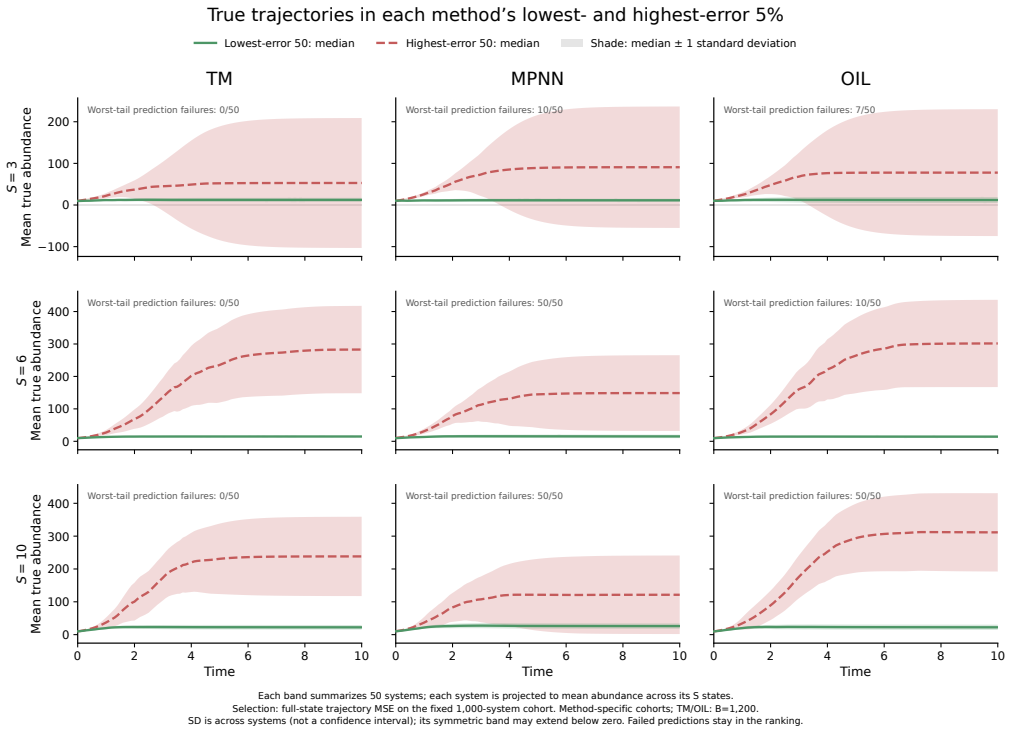


Figure 12: True trajectories in each method's lowest-error (green) and highest-error (red) 5%. Each system is projected to its mean abundance across states; lines are pointwise medians and bands are median $\pm$ one sample SD across 50 systems. These are neither prediction bands nor confidence intervals. Negative lower band edges can result from symmetric spread around a median despite nonnegative true abundances. Methods share row-wise scales, but select different systems. In the highest-error tails, all 50 selected MPNN systems at $S = 6, 10$ and OIL systems at $S = 10$ are invalid predictions; their valid true trajectories remain visible.

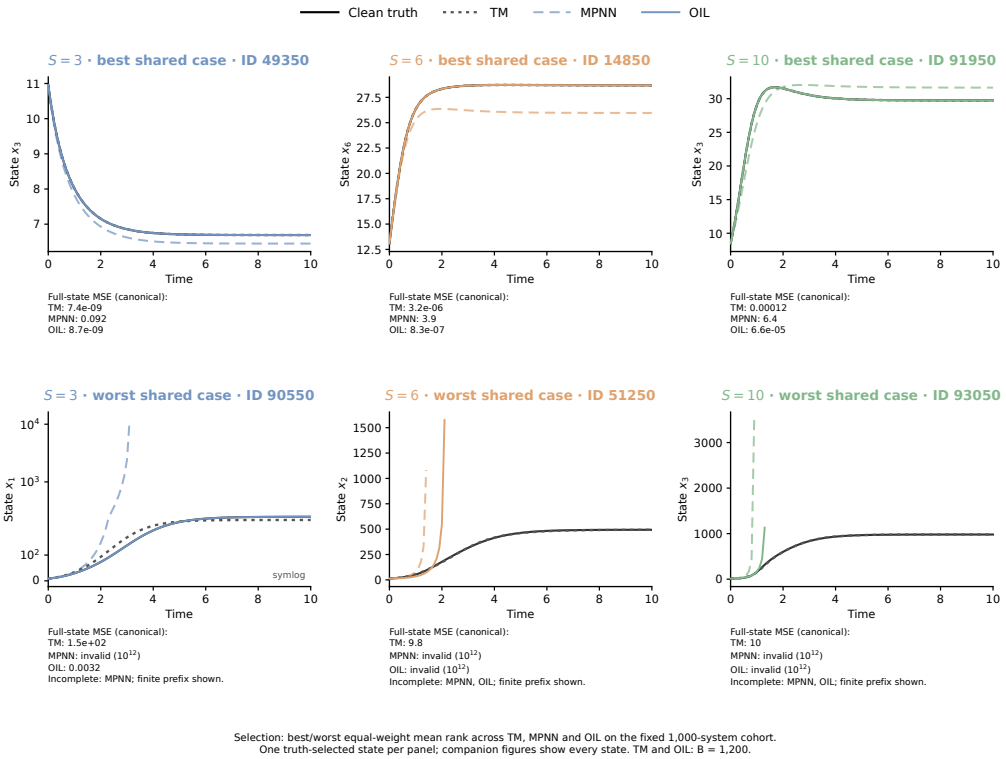


Figure 13: Paired truth/prediction overlays. For each S, the best and worst shared systems minimize/maximize the equal-weight mean of the three methods' error ranks. Every method is shown on the same selected system; these are paired extrema, not each method's individual extremum. The displayed state has the largest true temporal excursion. Independent float64 LSODA replays show only finite prefixes for invalid predictions; partial trajectories are not scored as successes. TM and OIL use $B = 1200$, and MPNN is raw.

N ARCHITECTURE AND RESULTS

Table 10: Full trained methods on gLV at $N \in \{3, 6, 10\}$, reported as median over 5-fold cross-validation across 100,000 sampled systems per N. Soft-DTW is computed on the intersection of instances where all methods in the original baseline comparison produced numerically valid forward simulations. Forward-sim success rate computed on all 100,000 instances per method. Best result per column per N in **bold**.

N	Method	RMSE$_{\boldsymbol{\theta}}$	MSE$_{\mathrm{traj}}$ (mean)	MSE$_{\mathrm{traj}}$ (med)	soft-DTW	sim. valid (%)
3	MLP	0.089	1.12×10^4	3.713	1.00×10^3	98.0
3	Transformer	0.069	8.48×10^3	3.244	8.68×10^2	98.5
3	MPNN	0.064	4.42×10^3	5.081	1.36×10^3	99.2
3	MLP + T	0.092	3.74×10^3	3.742	1.14×10^3	99.2
3	Transformer + T	**0.063**	4.87×10^3	**2.454**	$\mathbf{7.32 \times 10^2}$	99.0
3	MPNN + T	0.070	$\mathbf{5.88 \times 10^3}$	3.513	1.00×10^3	98.9
6	MLP	0.084	2.62×10^4	172.1	1.13×10^5	95.4
6	Transformer	0.061	4.33×10^4	94.82	5.76×10^4	92.1
6	MPNN	**0.060**	5.47×10^4	188.8	1.30×10^5	90.4
6	MLP + T	0.081	$\mathbf{1.03 \times 10^4}$	342.8	2.09×10^5	98.2
6	Transformer + T	0.065	4.61×10^4	**74.31**	$\mathbf{4.85 \times 10^4}$	91.3
6	MPNN + T	0.063	4.63×10^4	158.0	9.24×10^4	92.0
10	MLP	0.073	1.75×10^5	1.31×10^4	1.05×10^7	68.9
10	Transformer	0.057	1.57×10^5	4.97×10^3	5.92×10^6	72.5
10	MPNN	**0.056**	$\mathbf{1.03 \times 10^5}$	3.10×10^3	2.96×10^6	82.1
10	MLP + T	0.069	2.41×10^5	3.80×10^4	2.72×10^7	60.5
10	Transformer + T	0.062	1.10×10^5	$\mathbf{2.57 \times 10^3}$	$\mathbf{2.62 \times 10^6}$	79.1
10	MPNN + T	0.058	1.48×10^5	5.20×10^3	5.51×10^6	74.8