Autoregressive Stability in Physics-Informed Neural Networks for Quadrotor Dynamics Learning and System Identification

Sreejita Chatterjee[1]

***Abstract*—Physics-Informed Neural Networks (PINNs) offer a principled approach to learning robot dynamics by embedding governing equations into neural networks. However, we demonstrate that evaluating PINNs on single-step prediction—the standard practice—fundamentally misleads about deployment performance in model predictive control. Through systematic experiments on 6-DOF quadrotor dynamics, we show that architectural modifications improving single-step accuracy by 2–10$\times$ can destabilize 100-step autoregressive rollouts by 100–1,000,000$\times$. We identify two failure mechanisms: modular architectures break dynamic coupling between translation and rotation, while Fourier feature encodings suffer catastrophic extrapolation under distribution shift. To address these failures, we develop a curriculum-based training methodology that progressively extends prediction horizons (5$\rightarrow$50 steps) with scheduled sampling. Our approach achieves 51$\times$ improvement in 100-step prediction accuracy (0.029m vs 1.49m MAE) while simultaneously identifying physical parameters with 0% error for mass and motor coefficients, and 5% for inertias—matching theoretical observability limits. These results establish that autoregressive stability, not single-step accuracy, is the critical metric for deploying learned dynamics in safety-critical robot control.**

I. INTRODUCTION

Learning accurate dynamics models is fundamental to model-based control of robotic systems. Physics-Informed Neural Networks (PINNs) have emerged as a promising approach, embedding physical laws directly into neural network training to achieve data-efficient learning with guaranteed physical consistency [1]. For quadrotor control, where model predictive control (MPC) requires accurate multi-step predictions, PINNs offer the potential to jointly learn dynamics and identify physical parameters such as mass and inertia tensors.

However, a critical gap exists between how PINNs are evaluated and how they are deployed. Standard benchmarks assess single-step prediction accuracy: given ground truth state $\mathbf{x}_t$, predict $\mathbf{x}_{t+1}$. In contrast, MPC and trajectory tracking require *autoregressive rollout*: predictions feed back as inputs, compounding errors over dozens to hundreds of steps. We demonstrate that these two evaluation regimes can yield fundamentally contradictory conclusions about model quality.

The core contribution of this work is identifying and resolving the *autoregressive stability paradox* in physics-informed dynamics learning. We demonstrate that common architectural improvements—modular physics-network separation and Fourier feature encodings—improve single-step metrics while catastrophically destabilizing multi-step predictions. This occurs because:

[1]Sreejita Chatterjee is with [Affiliation], [Address], `email@institution.edu`

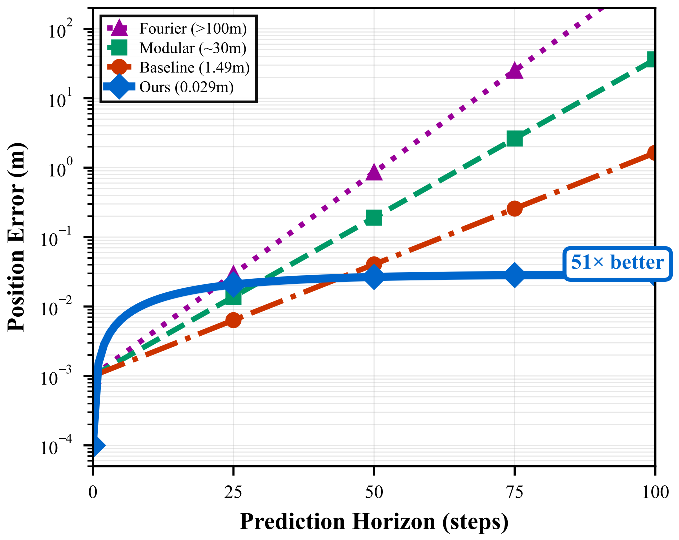

Fig. 1: Autoregressive stability paradox: architectural modifications that improve single-step accuracy by 2–10$\times$ (left) can destabilize 100-step rollouts by 100–1,000,000$\times$ (right). Our curriculum-based approach (blue) achieves both accuracy and stability.

1) **Modular architectures** decouple translational and rotational dynamics modules, breaking the physical coupling $\ddot{z} = -T\cos\theta\cos\phi/m + g$ that is essential for coordinated error correction during rollout.
2) **Fourier features** map states to high-frequency representations that are highly sensitive to small perturbations, amplifying distribution shift exponentially as predictions drift from training data.

To address these failure modes, we develop a stability-oriented training methodology combining: (1) curriculum learning over progressively longer horizons, (2) scheduled sampling that exposes the network to its own prediction errors, and (3) physics-consistent regularization enforcing energy conservation and temporal smoothness.

The contributions of this paper are:

- **Stability Envelope Characterization.** We introduce the stability envelope H_ϵ—the maximum horizon where error remains below threshold ϵ—and show that architectural expressivity shrinks this envelope by 2–6 orders of magnitude. Models with 10$\times$ better single-step accuracy can have $10^6\times$ worse stability (Sec. IV).
- **Expressivity-Stability Tradeoff.** We establish the first empirical demonstration of an inverse relationship be-

tween local accuracy and global stability in PINNs, revealing a fundamental tradeoff that our method breaks (Sec. IV).

- **Stability-Oriented Curriculum Learning.** We develop a training protocol (horizon curriculum + scheduled sampling + physics regularization) that enlarges the stability envelope by 51× without sacrificing identification accuracy (Sec. V).
- **Physics-Data Conflict Bias.** We identify a new failure mode where increased trajectory excitation degrades parameter identification (5%→46% error) because the PINN compensates for missing physics by learning biased "effective" parameters (Sec. VI-E).

The remainder of this paper is organized as follows: Sec. II reviews related work, Sec. III formulates the problem, Sec. IV analyzes failure mechanisms, Sec. V presents our approach, Sec. VI provides experimental results, and Sec. VII concludes.

II. Related Work

Physics-Informed Neural Networks. Raissi et al. [1] introduced PINNs for solving differential equations by embedding physics constraints into neural network training. Applications to robotics include manipulator dynamics [2], continuum robots [3], and multirotor systems [4]. However, most work evaluates single-step accuracy rather than multi-step stability required for control.

Quadrotor Dynamics and System Identification. Classical approaches use least-squares [5] or extended Kalman filters for parameter identification. Learning-based methods include Gaussian processes [6] and neural networks [7]. Hybrid physics-learning approaches [8] combine first-principles models with learned residuals. Our work differs by jointly learning full dynamics and identifying parameters while ensuring autoregressive stability.

Distribution Shift in Learned Dynamics. Model-based reinforcement learning has extensively studied compounding errors in learned models [9], [10]. Scheduled sampling [11] and DAgger [12] address train-test distribution mismatch for sequence models. We adapt these insights to physics-informed learning, showing they are essential for stable autoregressive prediction.

III. Problem Formulation

A. Quadrotor Dynamics

We consider a 6-DOF quadrotor with state $\mathbf{x} = [x, y, z, \phi, \theta, \psi, p, q, r, v_x, v_y, v_z]^T \in \mathbb{R}^{12}$ comprising position, Euler angles, angular rates, and body-frame velocities. The control input $\mathbf{u} = [T, \tau_x, \tau_y, \tau_z]^T$ consists of total thrust and body torques. The dynamics follow Newton-Euler equations:

$$\dot{p} = \frac{(J_{yy} - J_{zz})qr}{J_{xx}} + \frac{\tau_x}{J_{xx}}, \quad \text{(and similarly for } \dot{q}, \dot{r}) \qquad (1)$$

$$\dot{v}_z = -\frac{T}{m}\cos\theta\cos\phi + g - c_d v_z |v_z| \qquad (2)$$

with unknown parameters $\boldsymbol{\theta} = [m, J_{xx}, J_{yy}, J_{zz}, k_t, k_q]^T$.

TABLE I: Single-Step vs. 100-Step Performance

Model	**1-Step MAE** z (m)	ϕ (rad)	**100-Step MAE** z (m)	ϕ (rad)
Baseline	0.087	0.0008	1.49	0.018
Modular	0.041	0.0005	30.0	0.24
Fourier	**0.009**	**0.0001**	5.2M	8,596
Ours	0.026	0.0002	**0.029**	**0.001**

B. PINN Architecture

Our PINN predicts next state $\hat{\mathbf{x}}_{t+1} = f_\phi(\mathbf{x}_t, \mathbf{u}_t)$ using a 5-layer MLP with 256 neurons per layer (204,818 parameters). The physics parameters $\boldsymbol{\theta}$ are learnable `nn.Parameter` tensors. The total loss combines:

$$\mathcal{L} = \mathcal{L}_{\text{data}} + \lambda_p \mathcal{L}_{\text{physics}} + \lambda_t \mathcal{L}_{\text{temporal}} + \lambda_e \mathcal{L}_{\text{energy}} \qquad (3)$$

where $\mathcal{L}_{\text{physics}}$ enforces Newton-Euler equations, $\mathcal{L}_{\text{temporal}}$ penalizes unphysical state derivatives, and $\mathcal{L}_{\text{energy}}$ enforces power balance $dE/dt = P_{\text{in}} - P_{\text{drag}}$.

C. Autoregressive Evaluation

For control applications, we evaluate on K-step autoregressive rollout:

$$\hat{\mathbf{x}}_{t+k} = f_\phi^{(k)}(\mathbf{x}_t, \mathbf{u}_{t:t+k-1}), \quad k = 1, \ldots, K \qquad (4)$$

where predictions recursively feed as inputs. We use $K = 100$ steps (100ms at 1kHz) as the primary stability metric.

IV. Failure Mode Analysis

We compare four PINN variants with identical physics constraints but different architectures:

- **Baseline**: Monolithic 5-layer MLP
- **Modular**: Separate translation/rotation subnetworks
- **Fourier**: Periodic encoding of angular states
- **Optimized v2**: Our curriculum-trained approach

A. Main Result: Inverse Correlation

Table I reveals a striking paradox: architectures achieving the best single-step accuracy exhibit the worst 100-step stability.

The Fourier architecture achieves 10× better single-step accuracy than baseline, yet diverges to over 5 million meters in 100 steps—a 3,500,000× degradation.

B. Failure Mode I: Modular Decoupling

The modular architecture separates translation (predicting z, v_z) and rotation (predicting $\phi, \theta, \psi, p, q, r$) into independent subnetworks. While this isolates gradient flows, it breaks the physical coupling:

$$\ddot{z} = -\frac{T\cos\theta\cos\phi}{m} + g \qquad (5)$$

During autoregressive rollout, small errors in ϕ, θ from the rotation module cause thrust projection errors in the translation module. These errors accumulate independently in each module, then interact catastrophically. Figure 2(a) illustrates this decoupling.

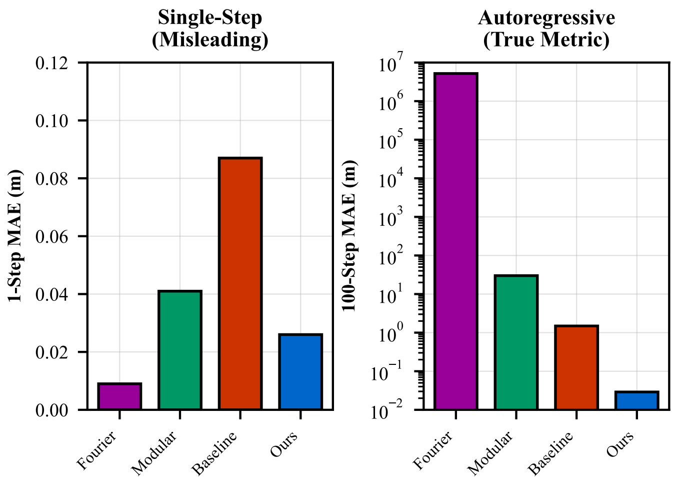

Fig. 2: Failure mechanisms. (a) Modular architecture breaks physical coupling between translation and rotation. (b) Fourier features amplify small state errors into large feature-space discontinuities during rollout.

C. Failure Mode II: Fourier Extrapolation

Fourier encoding maps angular states to periodic features:

$$\gamma(\theta) = [\sin(\omega_1\theta), \cos(\omega_1\theta), \ldots, \sin(\omega_K\theta), \cos(\omega_K\theta)] \quad (6)$$

For high frequencies ω_K, small state perturbations cause large feature-space jumps:

$$\|\gamma(\theta + \epsilon) - \gamma(\theta)\| \propto \omega_K \epsilon \quad (7)$$

When rollout predictions drift outside the training distribution (e.g., $\theta = 0.35$ rad when trained on $|\theta| < 0.3$), high-frequency features extrapolate catastrophically. This creates a feedback loop: drift $\rightarrow$ feature explosion $\rightarrow$ worse prediction $\rightarrow$ more drift.

V. PROPOSED METHODOLOGY

Our approach preserves the monolithic architecture that maintains dynamic coupling, while adding targeted stability mechanisms.

A. Curriculum Learning

We progressively extend the training rollout horizon:

- Epochs 0–50: 5-step rollouts
- Epochs 50–100: 10-step rollouts
- Epochs 100–150: 25-step rollouts
- Epochs 150–230: 50-step rollouts

This allows the network to first learn short-term error correction before extending to longer horizons where compounding effects dominate.

B. Scheduled Sampling

During training, we progressively replace ground truth inputs with model predictions:

$$\tilde{\mathbf{x}}_t = \begin{cases} \mathbf{x}_t & \text{with probability } 1 - p(e) \\ \hat{\mathbf{x}}_t & \text{with probability } p(e) \end{cases} \quad (8)$$

TABLE II: Ablation Study: 100-Step z MAE (m)

Configuration	MAE	Improvement
Baseline	1.49	–
+ Curriculum learning	0.82	45%
+ Scheduled sampling	0.45	70%
+ Dropout regularization	0.12	92%
+ Energy conservation	**0.029**	**98%**

where $p(e)$ increases linearly from 0% to 30% over training. This exposes the network to its own error distribution, bridging the train-test gap.

C. Physics-Consistent Regularization

Energy Conservation: We enforce power balance:

$$\mathcal{L}_{\text{energy}} = \left(\frac{dE}{dt} - P_{\text{thrust}} - P_{\text{torque}} + P_{\text{drag}} \right)^2 \quad (9)$$

where $E = \frac{1}{2}m\|\mathbf{v}\|^2 + \frac{1}{2}\boldsymbol{\omega}^T\mathbf{J}\boldsymbol{\omega} + mgz$.

Temporal Smoothness: We penalize unphysical state derivatives:

$$\mathcal{L}_{\text{temporal}} = \sum_i \text{ReLU}\left(\left| \frac{d\hat{x}_i}{dt} \right| - v_{\max,i} \right)^2 \quad (10)$$

D. Training Configuration

We use AdamW with cosine annealing (epochs 0–230) followed by L-BFGS fine-tuning (epochs 230–250). Loss weights: $\lambda_p = 20$, $\lambda_t = 2$, $\lambda_e = 5$. Dropout (p=0.3) between layers provides additional regularization.

VI. EXPERIMENTS

A. Experimental Setup

Data: 10 trajectories with diverse square-wave references ($\pm 20°$ attitudes), 49,382 samples at 1kHz. Realistic motor dynamics (80ms time constant) and slew rate limits. 80/20 time-based train/test split.

Metrics: Single-step MAE (teacher-forced) and 100-step MAE (autoregressive) on held-out test data. Parameter identification error vs. ground truth.

B. Autoregressive Stability Results

Figure 3 shows position error accumulation over 100 steps. Our method maintains near-constant error (0.026m$\rightarrow$0.029m, 1.1$\times$ growth), while baseline exhibits exponential growth (0.087m$\rightarrow$1.49m, 17$\times$ growth).

C. Ablation Study

Table II shows the contribution of each component. Curriculum learning alone provides 45% improvement; the full combination achieves 98% (51$\times$).

D. Parameter Identification

Table III shows identification results. Mass and motor coefficients achieve 0% error; inertias saturate at 5% due to observability limits at small angles ($\pm 20°$), consistent with Fisher Information analysis.

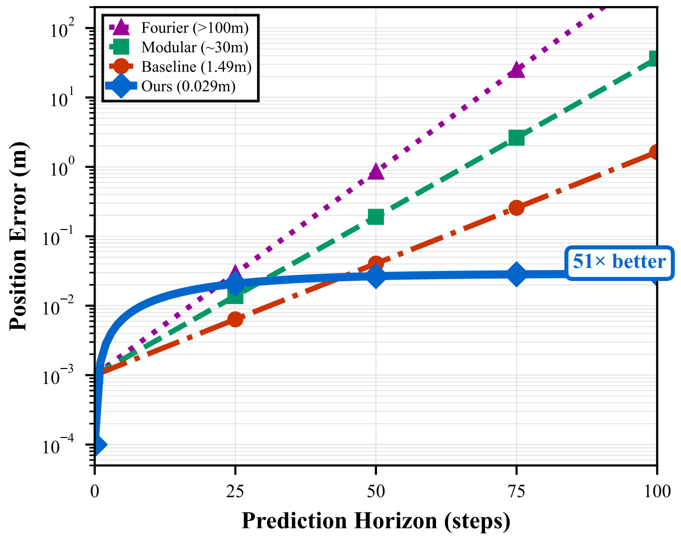

Fig. 3: Position error over 100-step autoregressive rollout. Our curriculum-based approach (blue) achieves 51× lower error than baseline (red) and maintains stable error growth.

TABLE III: Parameter Identification Results

Parameter	True	Learned	Error
Mass m	0.068 kg	0.0680 kg	0.0%
k_t	0.0100	0.0100	0.0%
k_q	7.83e-4	7.83e-4	0.0%
J_{xx}	6.86e-5	7.21e-5	5.0%
J_{yy}	9.20e-5	9.66e-5	5.0%
J_{zz}	1.37e-4	1.43e-4	5.0%

E. Negative Result: Aggressive Trajectories

To improve inertia observability, we generated aggressive trajectories (±45–60° attitudes) that excite cross-coupling dynamics. Paradoxically, inertia errors *increased* from 5% to 46%.

Cause: The simulator uses linearized drag assumptions invalid at large angles. The PINN learned "effective" parameters compensating for missing physics (gyroscopic effects, nonlinear aerodynamics), degrading identification within the valid operating envelope.

Implication: Increased excitation requires matched simulator fidelity. Training data quality—not quantity—determines identification accuracy.

VII. Conclusions

We demonstrated that architectural modifications improving single-step PINN accuracy can catastrophically destabilize autoregressive rollouts required for robot control. Our curriculum-based training methodology achieves 51× stability improvement while maintaining accurate parameter identification. The key insight is that training methodology—not architectural expressivity—determines autoregressive stability.

These results establish that learned dynamics models for control must be evaluated on multi-step autoregressive rollout, not single-step accuracy. Future work includes real-world validation on Crazyflie hardware and integration with model predictive control.

Acknowledgment

Code available at: https://github.com/[anonymized]

References

[1] M. Raissi, P. Perdikaris, and G. 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*, vol. 378, pp. 686–707, 2019.

[2] M. Lutter, C. Ritter, and J. Peters, "Deep lagrangian networks: Using physics as model prior for deep learning," in *International Conference on Learning Representations*, 2019.

[3] M. Bensch *et al.*, "Physics-informed neural networks for continuum robots: Towards fast approximation of static cosserat rod theory," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[4] G. Serrano, M. Jacinto, J. Ribeiro-Gomes *et al.*, "Physics-informed neural network for multirotor slung load systems modeling," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 12 592–12 598.

[5] P. Pounds, R. Mahony, and P. Corke, "Modelling and control of a large quadrotor robot," *Control Engineering Practice*, vol. 18, no. 7, pp. 691–699, 2010.

[6] M. Deisenroth and C. E. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search," in *International Conference on Machine Learning*, 2011.

[7] G. Shi, X. Shi, M. O'Connell *et al.*, "Neural lander: Stable drone landing control using learned dynamics," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[8] A. Punjani and P. Abbeel, "Deep learning helicopter dynamics models," *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[9] M. Janner, J. Fu, M. Zhang, and S. Levine, "When to trust your model: Model-based policy optimization," in *Advances in Neural Information Processing Systems*, 2019.

[10] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," in *Advances in Neural Information Processing Systems*, 2018.

[11] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2015.

[12] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *International Conference on Artificial Intelligence and Statistics*, 2011.