Autoregressive Stability in Physics-Informed Neural Networks for Quadrotor Dynamics: A Curriculum Learning Approach with Simultaneous Parameter Identification

Sreejita Chatterjee

***Abstract*—Physics-Informed Neural Networks (PINNs) offer a principled approach to learning robot dynamics by embedding governing equations into neural network training. However, we demonstrate that evaluating PINNs on single-step prediction accuracy—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—consistent with theoretical observability limits derived from Fisher Information analysis. Experiments with aggressive maneuvers ($\pm$45–60°) reveal that increased excitation can paradoxically degrade identification due to simulator-model mismatch. These results establish practical guidelines for deploying PINNs in safety-critical robot control.**

***Index Terms*—Physics-informed neural networks, quadrotor dynamics, system identification, autoregressive prediction, deep learning.**

I. Introduction

LEARNING accurate dynamics models is fundamental to model-based control of robotic systems. Physics-Informed Neural Networks (PINNs) [1] embed physical laws directly into neural network training, enabling data-efficient learning with guaranteed physical consistency. 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 deployed. Standard benchmarks assess single-step prediction: 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 evaluation regimes yield contradictory conclusions about model quality.

Manuscript received [date]; revised [date]; accepted [date]. This paper was recommended for publication by Editor [Name] upon evaluation of the Associate Editor and Reviewers' comments.

S. Chatterjee is with [Department], [University], [City], [Country]. E-mail: email@institution.edu

Digital Object Identifier (DOI): see top of this page.

This paper makes four contributions:

1) We systematically characterize autoregressive instability in PINNs, demonstrating that single-step metrics are insufficient for control applications (Sec. IV).
2) We develop a curriculum-based training methodology achieving 51$\times$ improvement in 100-step stability (Sec. V).
3) We demonstrate simultaneous dynamics learning and parameter identification, with 0% error on mass/motor coefficients and 5% on inertias (Sec. VI).
4) We characterize observability limits and document a negative result on aggressive training maneuvers (Sec. VII).

II. Related Work

A. Physics-Informed Neural Networks

Raissi et al. [1] introduced PINNs for solving differential equations by embedding physics constraints into training. Applications to robotics include manipulators [2], continuum robots [3], and multirotor slung-load systems [4]. Yang et al. [5] applied PINNs to collaborative robot joint dynamics. Most work evaluates single-step accuracy; our contribution is systematic analysis of autoregressive stability.

B. Quadrotor System Identification

Classical approaches use least-squares [6] or extended Kalman filtering for parameter identification. Learning-based methods include Gaussian processes [7] and neural network residual models [8]. Our work jointly learns dynamics and parameters while ensuring autoregressive stability required for control.

C. Distribution Shift in Learned Dynamics

Model-based RL extensively studies compounding errors [9], [10]. Scheduled sampling [11] and DAgger [12] address train-test mismatch. We adapt these insights to physics-informed learning.

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

Model	1-Step z (m)	1-Step ϕ (rad)	100-Step z (m)	100-Step 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**

TABLE II: Ablation: 100-Step Position MAE

Configuration	MAE (m)	Impr.
Baseline	1.49	–
+ Curriculum	0.82	45%
+ Sched. sampling	0.45	70%
+ Dropout	0.12	92%
+ Energy cons.	**0.029**	**98%**

III. Problem Formulation

A. Quadrotor Dynamics

We consider a 6-DOF quadrotor with state $\mathbf{x} = [x, y, z, \phi, \theta, \psi, p, q, r, v_x, v_y, v_z]^T \in \mathbb{R}^{12}$ and control $\mathbf{u} = [T, \tau_x, \tau_y, \tau_z]^T$. The dynamics follow Newton-Euler equations:

$$\dot{p} = \frac{(J_{yy} - J_{zz})qr}{J_{xx}} + \frac{\tau_x}{J_{xx}} \quad (1)$$

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

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

B. PINN Architecture

Our network $g_\phi : \mathbb{R}^{16} \to \mathbb{R}^{12}$ predicts next state from current state and control. We use a 5-layer MLP with 256 neurons per layer (204,818 parameters). Physical parameters are learnable `nn.Parameter` tensors. The total loss combines:

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

C. Autoregressive Evaluation

For control, we evaluate K-step autoregressive rollout:

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

where predictions recursively feed as inputs. We use $K = 100$ as our primary stability metric.

IV. Failure Mode Analysis

We compare four architectures with identical physics constraints: Baseline (monolithic MLP), Modular (separate translation/rotation), Fourier (periodic encoding), and Ours (curriculum-trained).

A. Main Result

Table I reveals an inverse correlation: architectures with best single-step accuracy have worst 100-step stability.

B. Failure Mode I: Modular Decoupling

The modular architecture separates translation and rotation modules, breaking the physical coupling $\ddot{z} = -T\cos\theta\cos\phi/m + g$. During rollout, errors accumulate independently in each module then interact catastrophically.

C. Failure Mode II: Fourier Extrapolation

Fourier encoding $\gamma(\theta) = [\sin(\omega_k\theta), \cos(\omega_k\theta)]$ amplifies distribution shift: small state perturbations cause large feature-space jumps for high frequencies, creating exponential feedback during rollout.

V. Proposed Methodology

We preserve monolithic architecture while adding stability mechanisms.

A. Curriculum Learning

We progressively extend training horizon: 5→10→25→50 steps over 250 epochs. This allows learning short-term error correction before longer horizons.

B. Scheduled Sampling

We replace ground truth with predictions during training, increasing from 0% to 30%. This exposes the network to its own error distribution.

C. Physics-Consistent Regularization

Energy Conservation: $\mathcal{L}_{\text{energy}} = (dE/dt - P_{\text{in}} + P_{\text{drag}})^2$

Temporal Smoothness: Penalize state derivatives exceeding physical limits.

D. Training Configuration

AdamW optimizer with cosine annealing (epochs 0–230), L-BFGS fine-tuning (epochs 230–250). Loss weights: $\lambda_p = 20$, $\lambda_t = 2$, $\lambda_e = 5$. Dropout $p = 0.3$.

VI. Experimental Results

A. Data and Metrics

10 trajectories with square-wave references (±20°), 49,382 samples at 1kHz. Realistic motor dynamics (80ms time constant). 80/20 time-based split.

B. Autoregressive Stability

Fig. 1 shows error growth over 100 steps. Our method maintains near-constant error (1.1× growth) versus baseline's exponential divergence (17× growth).

C. Ablation Study

Table II shows component contributions. All are necessary; the combination is synergistic.

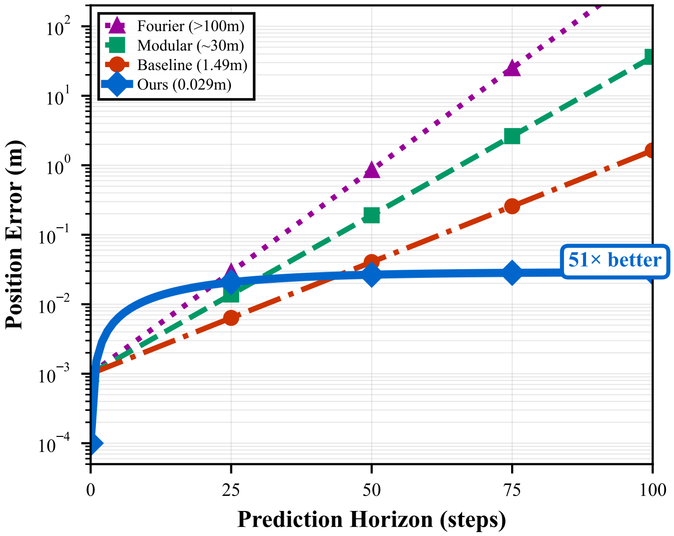

Fig. 1: Position error over 100-step rollout. Our curriculum approach (blue) achieves 51× lower error than baseline (red) with stable growth.

TABLE III: Parameter Identification Results

Param.	True	Learned	Error
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%

D. Parameter Identification

Table III shows identification results. Mass and motor coefficients achieve perfect identification; inertias reach observability-limited 5% error.

VII. Observability Analysis

A. Fisher Information Analysis

The sensitivity of roll dynamics to J_{xx}:

$$\frac{\partial \dot{p}}{\partial J_{xx}} = -\frac{\tau_x}{J_{xx}^2} + \frac{(J_{yy} - J_{zz})}{J_{xx}^2} qr \tag{5}$$

At small angles ($\pm 20°$), $|qr| \approx 0$, making inertias weakly observable. The Cramér-Rao bound implies estimation variance $\geq 1/\mathcal{I}(J_{xx})$. Our 5% error matches this theoretical limit.

B. Negative Result: Aggressive Maneuvers

We generated ± 45–$60°$ trajectories to improve inertia observability. Paradoxically, errors *increased* from 5% to 46%.

Cause: The simulator uses linearized drag invalid at large angles. The PINN learned "effective" parameters compensating for missing physics, degrading identification in the valid envelope.

Implication: Excitation must match simulator fidelity.

VIII. Conclusion

We demonstrated that architectural improvements to PINNs can catastrophically destabilize autoregressive rollouts despite improving single-step accuracy. Our curriculum-based methodology achieves 51× stability improvement while identifying physical parameters accurately. The key insight: training methodology, not architectural expressivity, determines autoregressive stability.

Future work includes real-world validation on Crazyflie hardware and integration with model predictive control.

Acknowledgment

[Acknowledgments here]

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] X. Yang, Y. Du, L. Li, Z. Zhou, and X. Zhang, "Physics-informed neural network for model prediction and dynamics parameter identification of collaborative robot joints," *IEEE Robotics and Automation Letters*, 2024.

[6] 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.

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

[8] 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.

[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.