Hierarchical Implicit Neural Emulators

Ruoxi Jiang Xiao Zhang Karan Jakhar Peter Y. Lu
Pedram Hassanzadeh Michael Maire Rebecca Willett

Funded by the U.S. National Science Foundation and the Simons Foundation

Problem setup

Background. Machine learning (ML)-based surrogate modeling of dynamical systems has spurred great interest in recent years due to transformative applications in climate modeling, molecular dynamics, and plasma physics.

Problem Formulation. For a nonlinear dynamical system, with f encoding the unknown governing physics:

$$\frac{\partial u}{\partial t} = f(u, x, t, \nabla_x u, \nabla_x^2 u \dots), \quad u(x,0) = u_0,$$

we consider its discretized transformation:

$$\frac{d\mathbf{u}_t}{dt} = \tilde{f}(\mathbf{u}_t, t), \quad \mathbf{u}_0 \in \mathbb{R}^m. \tag{1}$$

Here $\mathbf{u}_t$ represents the discretized state (e.g., fluid velocity on a grid) and $\tilde{f}$ represents the temporal dynamics. Given $N+1$ consecutive observations $\{\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_N\}$, *our goal* is to learn a neural emulator f_θ that approximates the underlying dynamics and predicts future states.

Autoregressive models. We train a transition operator $f_\theta : \mathbb{R}^m \to \mathbb{R}^m$ to iteratively predict next state $\hat{\mathbf{u}}_{n+1}$ from $\mathbf{u}_n$:

$$\hat{\mathbf{u}}_{n+1} = f_\theta(\mathbf{u}_n), \tag{2}$$

and chain predictions via recursive rollouts:

$$\hat{\mathbf{u}}_{n+k} = f_\theta(f_\theta(f_\theta \dots (\mathbf{u}_n))) \quad \text{for } k \text{ steps.}$$

Challenges and Inspiriation

Explicit methods & Autoregressive Error Accumulation: Classical autoregressive models are known to suffer from error accumulation, where small deviations amplify over time, leading to unphysical drift and instability in long-term rollouts.

We view *the conventional autoregressive models* as *explicit time-stepping methods*:

- For instance, the forward Euler method computes an estimate by:

$$\mathbf{u}_{n+1} \approx \mathbf{u}_n + \Delta_t \tilde{f}(\mathbf{u}_n, t_n). \tag{3}$$

- Suffers from instability for larger timesteps (Δ_t) and in stiff systems.

Inspiration: Implicit methods [1]

- Leverage both current states $\mathbf{u}_n$ and future states $\mathbf{u}_{n+1}$ to solve the implicit equation:

$$\mathbf{u}_{n+1} \approx \mathbf{u}_n + \Delta t \tilde{f}(\mathbf{u}_{n+1}, t_{n+1}) \tag{4}$$

- Can accommodate much larger time steps.
- Requires iterative root-finding to solve (4), the Newton method for instance.

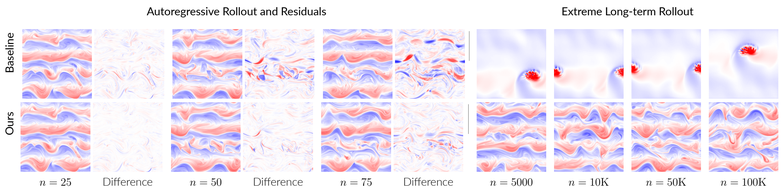


Figure 1. In a chaotic, turbulent system, our emulator achieves accurate short-to-mid-term rollout predictions (*left*). Even over extremely long sequences (up to 10^5 emulation steps, *right*), it captures the physical jet patterns, while baseline autoregressive methods quickly drift and break down.

Our approach: two-step implicit neural emulator

Our goal is to address the compounding error typically seen in autoregressive models by introducing a structure that implicitly reasons about future states. Rather than solving the implicit equation (4):

- We introduce a latent variable $\mathbf{z}_{n+1} = T(\mathbf{u}_{n+1})$, which represents an abstract encoding of the future state $\mathbf{u}_{n+1}$:

$$\hat{\mathbf{u}}_{n+1}, \hat{\mathbf{z}}_{n+2} = f_\theta(\mathbf{u}_n, \mathbf{z}_{n+1}), \tag{5}$$

  where the network is trained to simultaneously predict the next physical state $\hat{\mathbf{u}}_{n+1}$ and the latent representation $\hat{\mathbf{z}}_{n+2}$ that encodes information about a future state.

- **The choice of the transformation T:** It can be learned. For simplicity, we choose:

$$z_{n+1}^{(l)} =: \mathrm{DownSample}(u_{n+1}, r_l), \tag{6}$$

  where r_l is the downsampling factor.

The Hierarchical Implicit Neural Emulators Framework

The architecture above naturally extends to a hierarchical multi-step modeling framework in which predictions are conditioned on multiple latent representations of anticipated future states. We denote these representations as $\mathbf{z}_m^{(l)} = T^{(l)}(\mathbf{u}_m)$, where l indexes increasing levels of abstraction and we assume $\mathbf{u}_m = \mathbf{z}_m^{(0)}$. The model is then trained to predict across L hierarchical levels as follows:

$$\hat{\mathbf{u}}_{n+1}, \hat{\mathbf{z}}_{n+2}^{(1)}, \dots, \hat{\mathbf{z}}_{n+L}^{(L-1)} = f_\theta\left(\mathbf{u}_n, \mathbf{z}_{n+1}^{(1)}, \dots, \mathbf{z}_{n+L-1}^{(L-1)}\right). \tag{7}$$

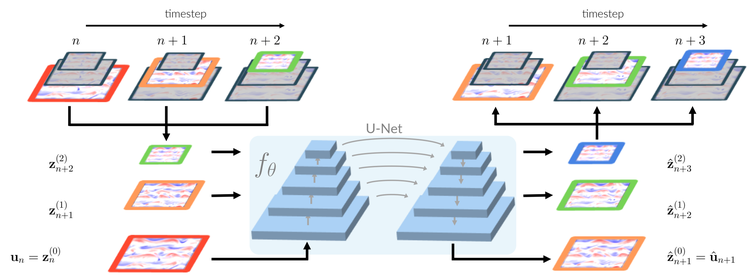


Figure 2. **Diagram of our hierarchical implicit neural emulator.** Our model conditions on both the past trajectory $\mathbf{u}_n$ and future latent variables $\mathbf{z}_{n+1}^{(1)}, \mathbf{z}_{n+2}^{(2)}$ during training, while using predictions of future states computed in the previous step of an autoregressive rollout during inference, providing richer context to effectively mitigate error accumulation for long-term predictions.

From an encoding standpoint: We process input as a hierarchial sequence.

- *Structured*: Immediate states like $\mathbf{u}_n$ retain fine-grained detail, while distant latents such as $\mathbf{z}_{n+l}^{(l)}$ provide coarse-scale insights
- *Multi-step*: Mirrors the philosophy of implicit methods like Adams–Moulton, with integrating information from multiple states.
- *Multi-scale*: the most adjacent frame includes the finest scale of information, while the most distant frame contains the most abstract information.

From a decoding standpoint: Distant states become progressively harder to predict.

- *Progressive*: Encourages a balanced learning process where both local precision and global structure are prioritized
- *Multi-step*: Can be interpreted as executing L steps of iterative refinement, distributed across temporal frames and abstraction levels without occurring additional computational cost.

Training objective. Finally, our training loss is designed to supervise both the predicted physical state and the associated abstract latents:

$$\ell(\theta) = d(\hat{\mathbf{u}}_{n+1}(\theta), \mathbf{u}_{n+1}) + \sum_{l=1}^{L-1} d(\hat{\mathbf{z}}_{n+l}^{(l)}(\theta), \mathbf{z}_{n+l}^{(l)}),$$

where $d(\cdot)$ denotes a distance metric such as $l1$ or $l2$ loss for simplicity.

Hierarchical autoregressive rollout. With a small probability p, we sample training instances in which the model receives a partially missing hierarchy of latent inputs and is tasked to reconstruct the missing parts. At the *evaluation* time, in the hierarchy $L = 2$, we first input $[\mathbf{u}_n, \mathbf{0}]$ to obtain $\hat{\mathbf{z}}_{n+1}^{(1)}$. Then, using the full spatial-temporal hierarchy states $[\mathbf{u}_n, \hat{\mathbf{z}}_{n+1}^{(1)}]$, we continue with autoregressive rollout.

Experiments

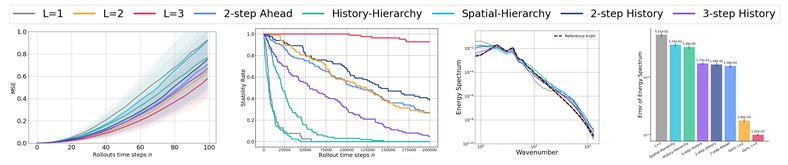
Navier-Stokes. We focus on the dimensionless vorticity-streamfunction ($\omega - \psi$) formulation of the incompressible Navier-Stokes equations in a 2D $x - y$ domain:

$$\frac{\partial \omega}{\partial t} + \mathcal{N}(\omega, \psi) = \frac{1}{Re}\nabla^2\omega - \chi\omega + f + \beta v,$$

where $\nabla^2 = -\omega$, $\boldsymbol{v} = (v_x, v_y)$ is velocity with $\omega = \nabla \times \boldsymbol{v}$. $\mathcal{N}(\omega, \psi)$ captures non-linear advection. The flow is defined by a Reynolds number $Re = 10^4$, constant forcing f, and a Rayleigh drag $\chi = 0.1$. The Coriolis parameter, $\beta = 20$, induces zonal jets characteristic of geophysical turbulence, mimicking the influence of Earth's rotation on atmospheric and oceanic flows. The domain is doubly periodic with length $L = 2\pi$.

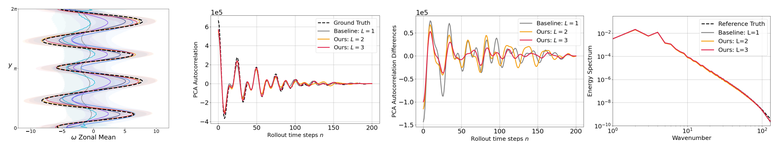
Evaluation: Stability rate. We leverage the system's conserved energy, defined as $E = \frac{1}{2}(v_x^2 + v_y^2)$. A trajectory is deemed stable if its energy remains within 5 standard deviations of this reference.

Experiments



(a) short-term estimation (b) long-term stability rate (c) Energy Spectrum (d) Energy Spectrum Error

Figure 3. **Short-term vs. long-term performance.** *Left: Short-term accuracy.* (a) MSE trend over a 100-step autoregressive rollout. *Right: long-term robustness.* (b) The stability rate over 10 times the training sequence length across 100 trials with various initial conditions for 2×10^5 steps. (c,d) Spectrum of long-term rollout for normalized data.



(a) Zonal Mean (b) PCA autocorrelations (c) Error of PCA autocorrleations (d) Short-term Energy Spectrum

Figure 4. **Further investigation of physical property.** *Left:* (a) Time-averaged zonal mean of vorticity comparing ground truth (dashed black lines) with emulator runs. *Right:* Ablation studies on the design of the hierarchy. (b,c) PCA autocorrelation. *(d)* Energy spectrum averaging over 200 rollout steps.

	1	25	50	75	100
$r^1 = 1$	9.900e-04 (2.143e-04)	5.865e-02 (2.890e-02)	2.103e-01 (8.888e-02)	4.470e-01 (1.762e-01)	7.738e-01 (2.520e-01)
$r^1 = 2$	1.747e-03 (3.942e-04)	1.146e-01 (6.328e-02)	3.774e-01 (1.797e-01)	7.082e-01 (3.088e-01)	1.023e+00 (3.352e-01)
$r^1 = 4$	7.735e-04 (1.954e-04)	5.651e-02 (3.199e-02)	1.998e-01 (1.074e-01)	4.225e-01 (1.979e-01)	7.280e-01 (2.522e-01)
$r^1 = 8$	5.248e-04 (1.148e-04)	**4.024e-02 (1.920e-02)**	**1.606e-01 (6.496e-02)**	**3.731e-01 (1.509e-01)**	**6.547e-01 (2.440e-01)**
$r^1 = 16$	**5.155e-04 (1.138e-04)**	4.898e-02 (2.011e-02)	2.039e-01 (8.009e-02)	4.416e-01 (1.789e-01)	7.399e-01 (2.658e-01)

Table 1. **Ablation study on downsampling ratio (r^1) for our $L = 2$ model on 256×256 resolution data with jet.** We evaluate model performance by computing roll-out mean squared error (MSE) for downsampling ratios $r^1 = 1, 2, 4, 8, 16$, and present the mean (standard deviation) over 100 trials with varied initial conditions.

Method	Seconds per iter.	1-step MSE	25-step MSE	50-step MSE	Zonal mean error
Baseline: $L = 1$	0.2067	5.60e−04 (1.15e−04)	8.04e−02 (3.45e−02)	3.12e−01 (1.15e−01)	4.20 (2.36)
Pushforward [2]	0.2834	1.08e−03 (2.26e−04)	9.16e−02 (4.62e−02)	3.19e−01 (1.34e−01)	1.07 (0.44)
Ours: $L = 3$	0.2204	**5.50e-04 (1.20e-04)**	**3.37e-02(1.41e-02)**	**1.40e-01 (6.16e-02)**	**0.63 (0.58)**

Table 2. **Comparison of training efficiency and multi-step forecasting errors.**

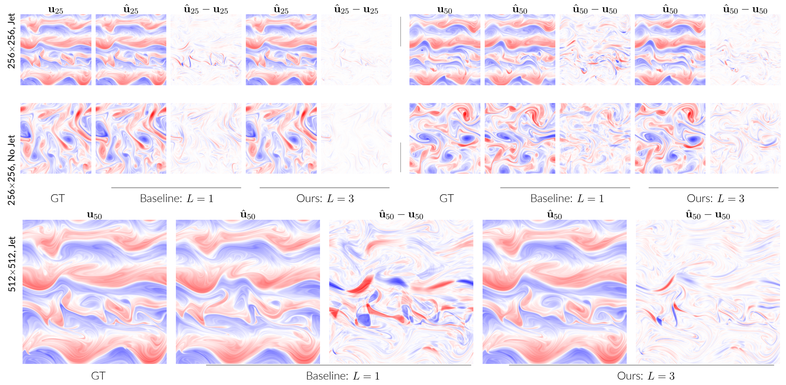


Figure 5. **Visualization of rollout estimation across multiple datasets.** We apply our approach to three different flows: (1) $Re = 10^4$, 256×256 resolution featuring zonal jets, (2) $Re = 5 \times 10^3$, 256×256 resolution without zonal jets, and (3) $Re = 10^4$, 512×512 resolution with zonal jets. Our method ($L = 3$) gives more accurate predictions with lower associated residuals in all three scenarios.

Acknowledgement and References

RJ, PL, and RW gratefully acknowledge the support of AFOSR FA9550-18-1-0166, the Eric and Wendy Schmidt AI in Science Fellowship program, and the Margot and Tom Pritzker Foundation. RJ, PL, RW, and MM gratefully acknowledge the support of the NSF-Simons AI-Institute for the Sky (SkAI) via grants NSF AST-2421845 and Simons Foundation MPS-AI-00010513. KJ and PH were supported by NSF RISE-2425898 and Schmidt Sciences, LLC.

[1] Uri M Ascher, Steven J Ruuth, and Brian TR Wetton. Implicit-explicit methods for time-dependent partial differential equations. *SIAM Journal on Numerical Analysis*, 1995.

[2] Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural pde solvers. *arXiv preprint arXiv:2202.03376*, 2022.

Correspondence to: roxie_jiang@fudan.edu.cn, zhang7@uchicago.edu, willett@uchicago.edu