جامعـة نيويورك أبـوظـبي
NYU ABU DHABI

NEURAL INFORMATION PROCESSING SYSTEMS

FlowMixer

Depth-Agnostic Neural Architecture for Interpretable Spatiotemporal Forecasting

Sometimes, One Layer is All You Need

Fares B. Mehouachi[1] | Saif Eddin Jabari[1,2]

[1]New York University Abu Dhabi, UAE

[2]NYU Tandon School of Engineering, USA

NeurIPS 2025

The Neural Depth Selection Problem in Time Series Forecasting

The Challenge:

Time series models require depth tuning per dataset

The Cost:

- Fresh search per deployment
- Wasted compute hours
- No theoretical guidance

Forecasting performance depends on depth

Which depth should we choose?

This depth sensitivity affects **all** state-of-the-art methods

Current SOTA Models:

- **TimeMixer++:** 2–16 layers (dataset dependent)
- **Chimera:** 4–12 layers
- **PatchTST:** 3–8 layers
- **TSMixer:** 2–10 layers
- **iTransformer:** 4–6 layers

Can we eliminate depth search?

Design an architecture where neural depth is **irrelevant**

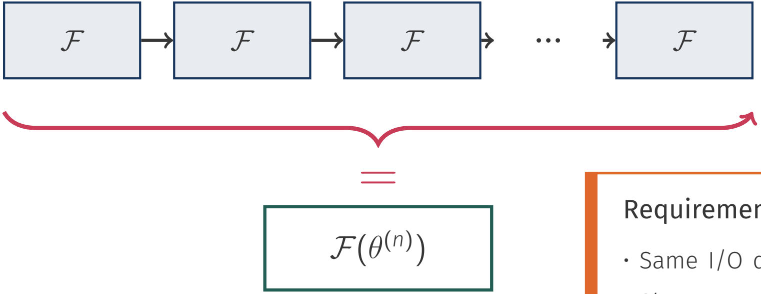
This requires a special mathematical property

The Goal: Depth Collapse Property

Can we design an architecture where **depth doesn't matter**?

Depth Collapse Property (Semi-Group Stability): $\mathcal{F}^n(X) = \mathcal{F}(X, \theta^{(n)})$

Stacking n layers = single layer with parameter $\theta^{(n)} = f(\theta^{(1)})$



Requirements:

- Same I/O dimensions
- Shape-preserving ops

FlowMixer: An architecture with depth collapse built in

$$\mathcal{F}(X) = \phi^{-1}\left(W_t \cdot \phi(X) \cdot W_f^T\right)$$

where $X \in \mathbb{R}^{n_t \times n_f}$ (time $\times$ features)

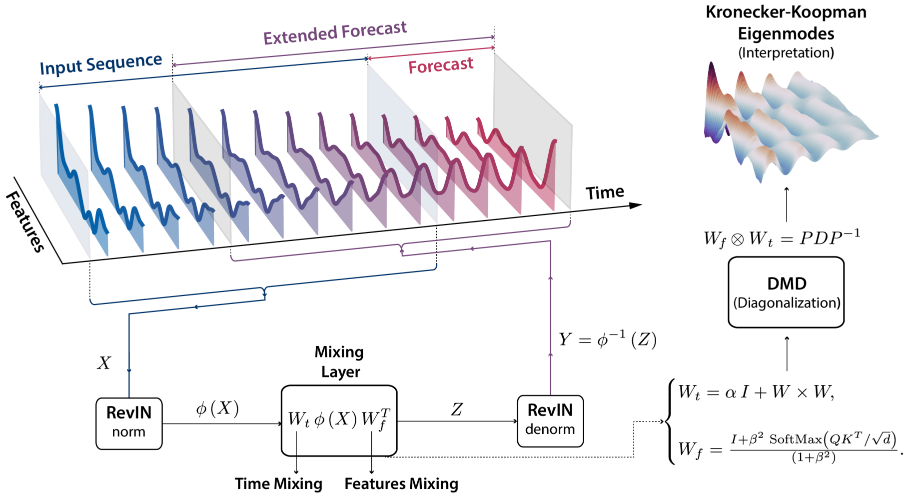
Architecture Components:

- ϕ: Reversible mapping
- W_t, W_f: Square mixing matrices
- Output = Input shape: $\mathbb{R}^{n_t \times n_f}$

Key Difference:

- Classic: $\mathbb{R}^{n_t \times n_f} \to \mathbb{R}^{h \times n_f}$ (projector)
- FlowMixer: **translation** not projection
- Forecast via: $Y' = [X[n_t - h : n_t], Y]$

FlowMixer: An architecture with depth collapse built in



Making FlowMixer Competitive: The Three Constraints

To make the **single-layer** FlowMixer **competitive**, we introduce **constraints**:

1. ϕ

RevIN

- Established in TS community
- Handles distribution shifts
- Time-dependent variant (**TD-RevIN**)

2. W_t

Time Mixing

- Non-negative: $W_t = \alpha I + W \times W$
- Correlates same-sign phenomena
- Reduces spurious correlations

3. W_f

Feature Mixing

- Stochastic: rows sum to 1
- Non-negative static attention
- Graph/Markov interpretation

Constraints Create Capabilities

Competitive Performance

Long-Horizon Time Series Forecasting (MSE)

Dataset	h	FlowMixer (2025)	Chimera (2024)	TimeMixer++ (2024)	TSMixer (2023)	PatchTST (2023)
ETTh1	96	.355	.366	**.361**	**.361**	.370
	720	.433	**.458**	.467	.463	.491
ETTm2	96	.159	.169	.170	**.163**	.166
	720	.333	**.341**	.373	.420	.380
Weather	96	.143	.146	.155	**.145**	.149
	720	**.305**	.297	.312	.320	.329
Traffic	96	.377	.366	**.376**	.392	.395
	720	.434	.443	**.441**	.444	.460

Best | **Second best**

Matching & exceeding SOTA with **zero depth tuning**

We achieved depth collapse...

...but FlowMixer has much more to offer

Started with: *"Let's eliminate depth search"* & Competitive forecasting

Discovered Four Unexpected Advantages:

1. Horizon Change w/o Retraining

Train at h, test at any h' via $W_t^{h'/h}$

2. Interpretable Eigenmodes

Kronecker-Koopman decomposition

---

3. Chaotic Systems Prediction

Lorenz, Rössler, Aizawa attractors

4. Turbulent Flow Modeling

2D Navier-Stokes, vortex shedding

Advantage 1: Algebraic Horizon Control

Horizon Change:

Traditional: retraining or rollout

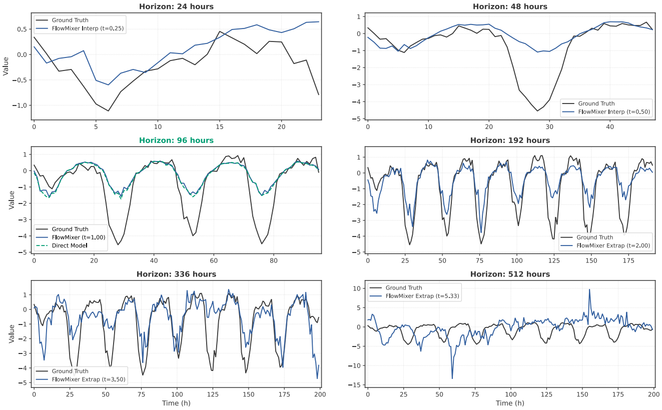
FlowMixer:

Power the mixing matrices:

$$\mathcal{F}_{h\to h'}(X) = \phi^{-1}\left(W_t^{h'/h}\,\phi(X)\,(W_f^{h'/h})^T\right)$$

Advantages:

- <1 second adjustment
- No retraining needed
- **Continuous** $h' \in \mathbb{R}^+$
- Derivatives: $\partial^\alpha/\partial t^\alpha$



Trained at $h = 96$ on ETTh1 (green line), adjusted to various horizons.
Effective range: $h'/h \in [0.5, 3]$

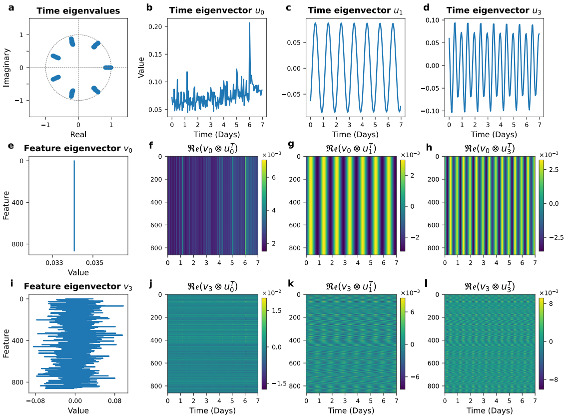
Continuous horizon control and temporal derivatives

Kronecker-Koopman Decomp.:

Via $W_t = QEQ^{-1}$, $W_f = PDP^{-1}$:

$$\phi(X) = \sum_{i,j} a_{ij}(q_i \otimes p_j)$$

- q_i: **Temporal patterns** (trends, periodicities)
- p_j: **Feature correlations** (spatial structure)
- $q_i \otimes p_j$: Coupled spatiotemporal dynamics
- **Directly extractable** (unlike Koopa, KNF)



Eigenmodes of Traffic dataset showing base trends and daily/weekly cycles. Coupling reveals propagation patterns.

Interpretability built-in, not added post-hoc

Chaotic Systems:

Exponential sensitivity to initial conditions

SOBR Enhancement:

Semi-Orthogonal Basic Reservoir

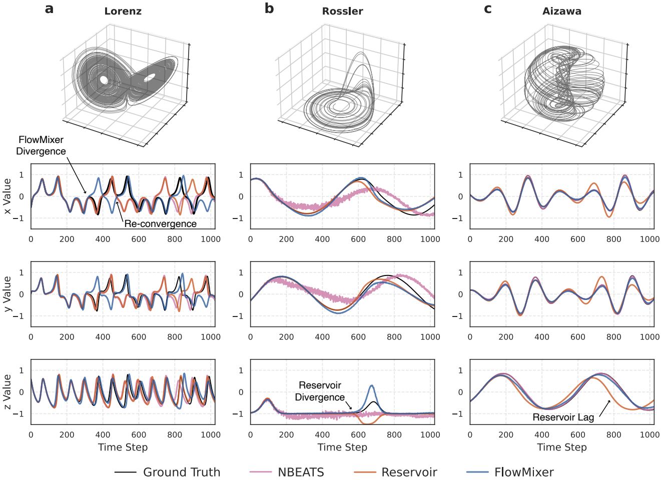
Replaces ϕ with $S \circ \phi$, lifting to higher dimensions:

$$S(X) = \sigma(U_t X U_f^T)$$

Semi-group preserved in lifted space

Results:

- 9-10 Lyapunov times
- 4-5% error
- 95.2% correlation



Lorenz, Rössler, Aizawa attractors: 1024-step predictions

Single-layer architecture, chaos-level performance

Turbulence:

Fundamentally different from time series:

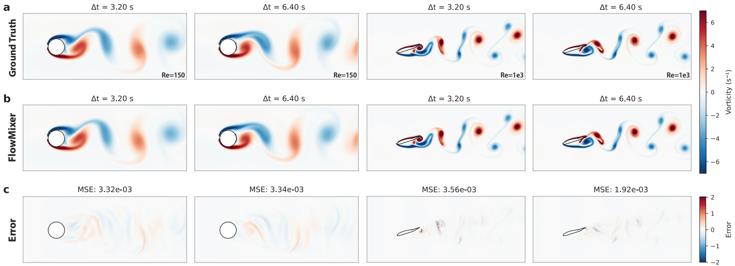
- Governed by PDEs (Navier-Stokes)
- Spatially coupled dynamics
- Multi-scale vortex interactions

Typical Approach:

- Physics-informed
- Domain knowledge

2D Incompressible Navier-Stokes:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \frac{1}{Re}\nabla^2 \mathbf{u}, \quad \nabla \cdot \mathbf{u} = 0, \quad \omega = \nabla \times \mathbf{u}$$



Vorticity ω for Cylinder (Re=150) & airfoil (Re=1000): prediction horizon 6.4 s

16× speedup vs spectral solvers | Relative error: ∼4% | Purely data-driven

From time series to turbulence prediction with the same architecture

FlowMixer: A Depth-Agnostic Forecasting Architecture

Core Achievement:

Zero depth search

Semi-group stability guarantees any depth through composition

Started with:
Eliminate depth search

Achieved:
Competitive forecasting (long horizon)

Discovered:
Four unexpected capabilities

Capabilities:

- **Algebraic horizon** control
- **Interpretable** eigenmodes
- **Chaotic systems** prediction
- **Turbulent flows** (16× speedup)

Neural Forecasting

Sometimes, One Layer is All You Need

Time series → Chaos → Turbulence

Thank You!

Questions?

Fares B. Mehouachi
New York University Abu Dhabi
fm2620@nyu.edu

Saif Eddin Jabari
NYU Tandon School of Engineering
sej7@nyu.edu

Acknowledgments: NYUAD CITIES Tamkeen Grant CG001

Code & Paper

github.com/FaresBMehouachi/FlowMixer

All code and experiments are open source

References: Key Baseline Methods

State-of-the-Art Time Series Forecasting Models:

- **TimeMixer++:** Wang et al. (2024). "TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting." *ICLR 2024*.
- **Chimera:** Bian et al. (2024). "Chimera: Effectively Modeling Multivariate Time Series with 2-Dimensional State Space Models." *arXiv:2406.04320*.
- **PatchTST:** Nie et al. (2023). "A Time Series is Worth 64 Words: Long-term Forecasting with Transformers." *ICLR 2023*.
- **TSMixer:** Chen et al. (2023). "TSMixer: An All-MLP Architecture for Time Series Forecasting." *arXiv:2303.06053*.
- **iTransformer:** Liu et al. (2024). "iTransformer: Inverted Transformers Are Effective for Time Series Forecasting." *ICLR 2024*.

Related Work on Interpretability:

- **Koopa:** Liu et al. (2023). "Koopa: Learning Non-stationary Time Series Dynamics with Koopman Predictors." *NeurIPS 2023*.
- **KNF:** Azencot et al. (2020). "Forecasting Sequential Data using Consistent Koopman Autoencoders." *ICML 2020*.

Full references available in the paper