Physics-informed Reduced Order Modeling of Time-dependent PDEs via Differentiable Solvers

Nima Hosseini Dashtbayaz[1,*], Hesam Salehipour[2], Adrian Butscher[2], Nigel Morris[2]

[1]Western University, [2]Autodesk Research

*Research conducted during internship at Autodesk

TL;DR: Physics-informed Reduced Order Model

Φ-ROM is a physics-informed reduced order model (ROM) for time-dependent PDEs that learns the temporal dynamics from **differentiable solvers**.

Compared to Data-Driven ROMs, **Φ**-ROM:

- ✓ Generalizes better to unseen initial conditions and parameters,
- ✓ Forecasts beyond the training time-horizon,
- ✓ Is robust to irregular and sparse data,
- ✓ Is more data efficient.

Φ-ROM is also more robust to various physical phenomena compared to other physics-informed approaches (e.g. PINNs).

Overview

For PDEs of the following form,

$$\dot{u} = \mathcal{N}(u;\beta), \quad u(t,x): \mathcal{T}\times\Omega \to \mathbb{R}^m,$$

parameterized by β, a neural ROM generally:

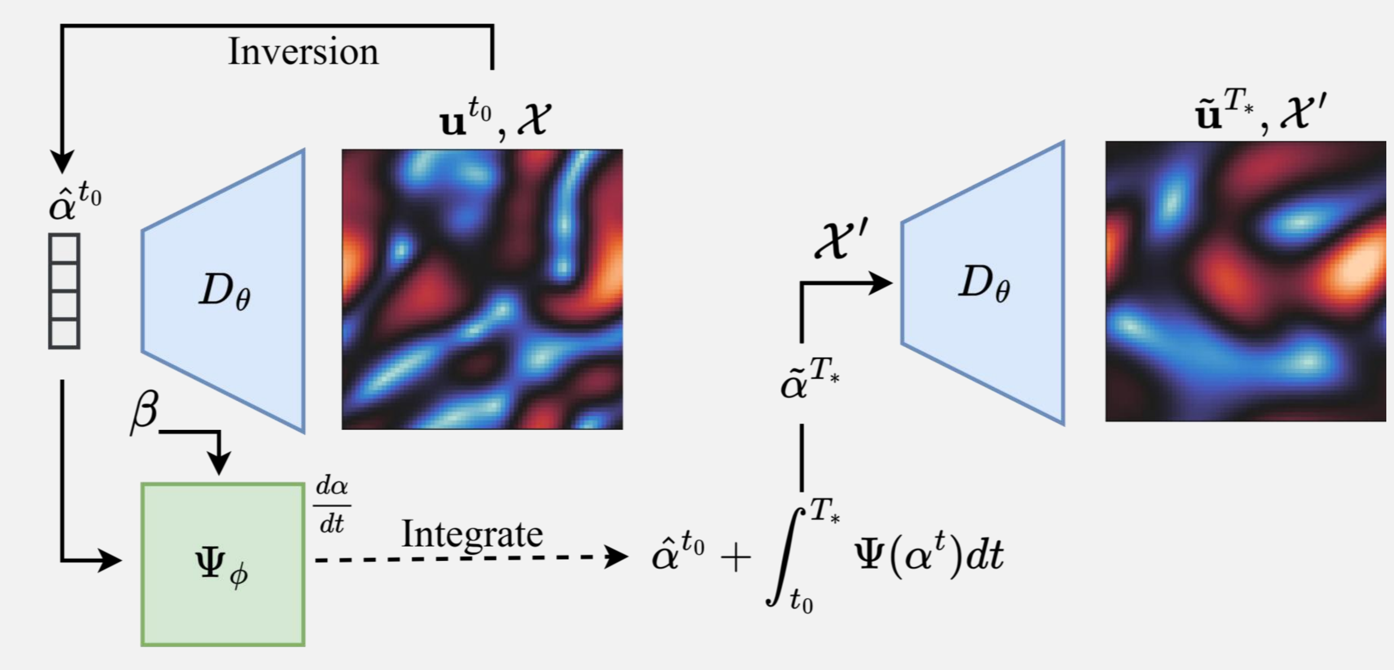
1. **Encodes:** $u_{t_0} \xrightarrow{D^\dagger} \alpha_{t_0}$
2. **Forecasts:** $\alpha_{t_0} \xrightarrow{\Psi} \alpha_T$
3. **Decodes:** $\alpha_T \xrightarrow{D} u_T$

➤ Fast and efficient simulation within a reduced manifold of solutions.

→ Data-driven ROMs fail to accurately model the latent dynamics consistent with the true physics.

Φ-ROM learns the latent dynamics directly from a differentiable solver so that they are consistent with the physics.
➤ Ψ is consistent with physics.

Φ-ROM Framework

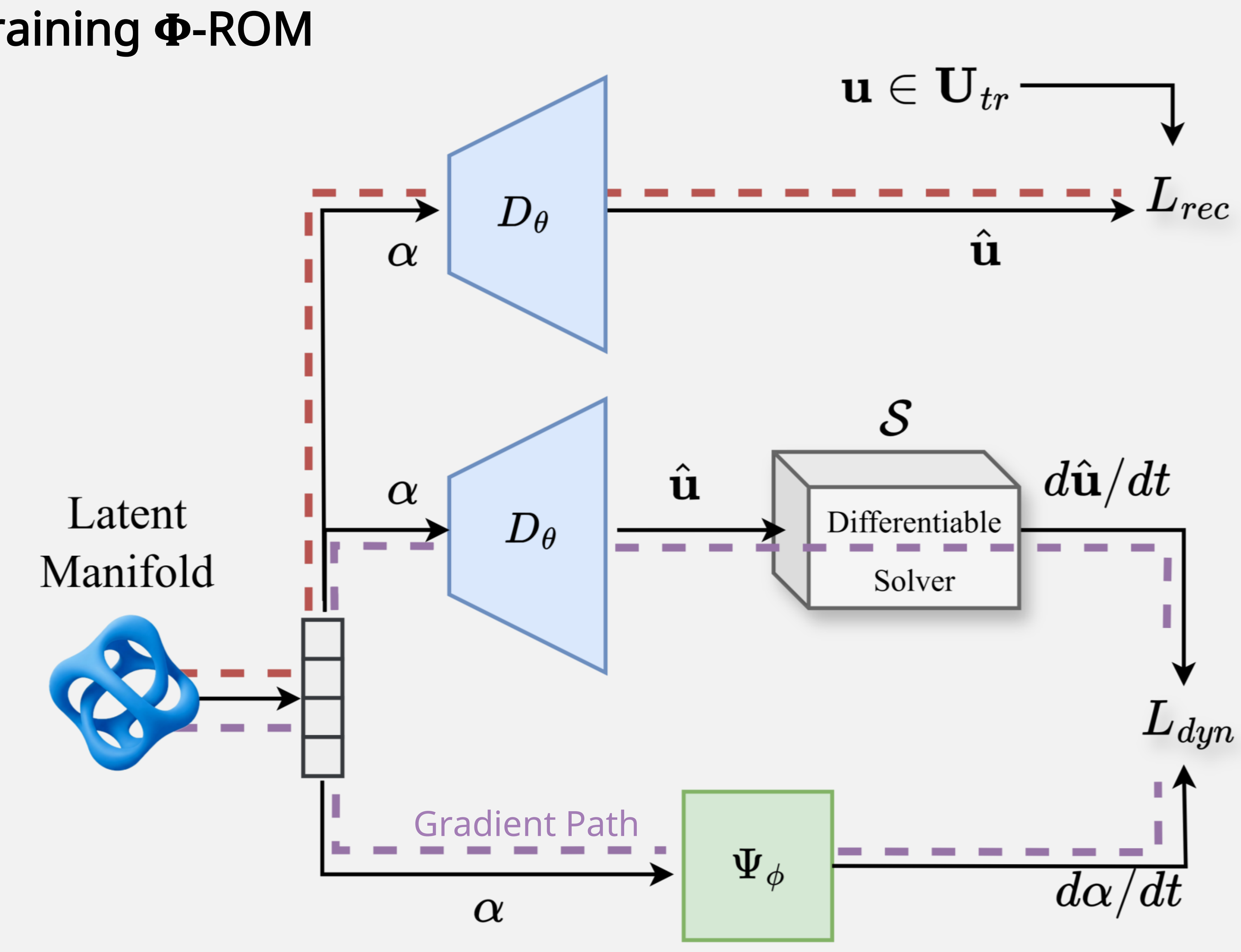


Φ-ROM has the same architecture as DINO:

1. **INR Decoder:** $D(\alpha_t, \mathcal{X}) = \hat{\mathbf{u}}_t$
2. **Dynamics network:** $\Psi(\alpha) = d\alpha/dt = \dot{\alpha}$

→ ✓ Mesh-free ✓ Continuous in space ✓ Continuous in time

Training Φ-ROM



For a latent coordinate α corresponding to snapshot $\mathbf{u}$ in the dataset:

1. Reconstruct: $D(\alpha) = \hat{\mathbf{u}}$
2. Get solver dynamics: $d\hat{\mathbf{u}}/dt = \mathcal{S}[\hat{\mathbf{u}}]$
3. Get latent derivatives: $d\alpha/dt = \Psi(\alpha)$
4. Take decoder Jacobians: $J_D(\alpha)\dot{\alpha} = d\hat{\mathbf{u}}/dt$

$$\rightarrow \quad \mathrm{L}_{dyn}(\alpha) = (\Psi(\alpha) - J_D^\dagger(\alpha)\mathcal{S}[\hat{\mathbf{u}}])^2$$

Minimize the dynamics loss along with a data reconstruction loss.

➤ Since $\mathcal{S}$ is differentiable, errors backpropagate through the decoder and regularize the latent solution manifold, while Ψ learns directly from $\mathcal{S}$.

Results: Generalization, Forecasting, and Sparse Data

PDEs: **(1)** Diffusion, **(2)** Burgers', **(3)** N-S, **(4)** KdV, **(5)** LBM (bluff-body)
Solvers: **(1)** FD, **(2)** FD, **(3)** FVM, **(4)** Spectral, **(5)** Lattice Boltzmann
Baselines: DINo [1], CROM [2], PINN-ROM [3]

N-S & KdV: Forecasting for unseen initial conditions

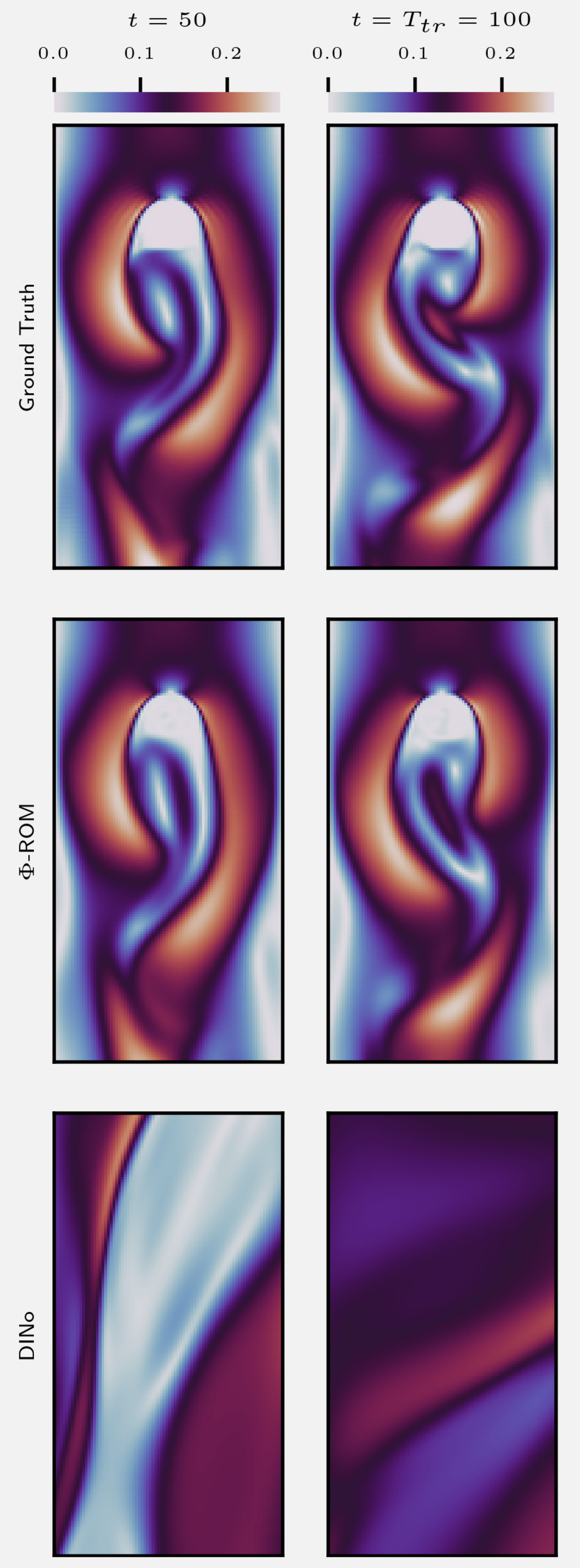
PDE / Time	NS $[0,T_{tr}]$	NS $[T_{tr},T_{te}]$	KdV $[0,T_{tr}]$	KdV $[T_{tr},T_{te}]$
Full Training: $\mathcal{X}_{tr} = \mathcal{X}_\mathcal{S} = \mathcal{X}_{te}$				
Φ-ROM	**0.170**	**0.373**	**0.233**	**0.486**
DINo	0.580	1.543	0.459	0.728
Sparse Training: $\lvert\mathcal{X}_{tr}\rvert = 2\%\lvert\mathcal{X}_\mathcal{S}\rvert$, $\mathcal{X}_{te} = \mathcal{X}_\mathcal{S}$				
Φ-ROM	**0.189**	**0.394**	**0.280**	**0.567**
DINo	0.594	1.517	0.902	1.396

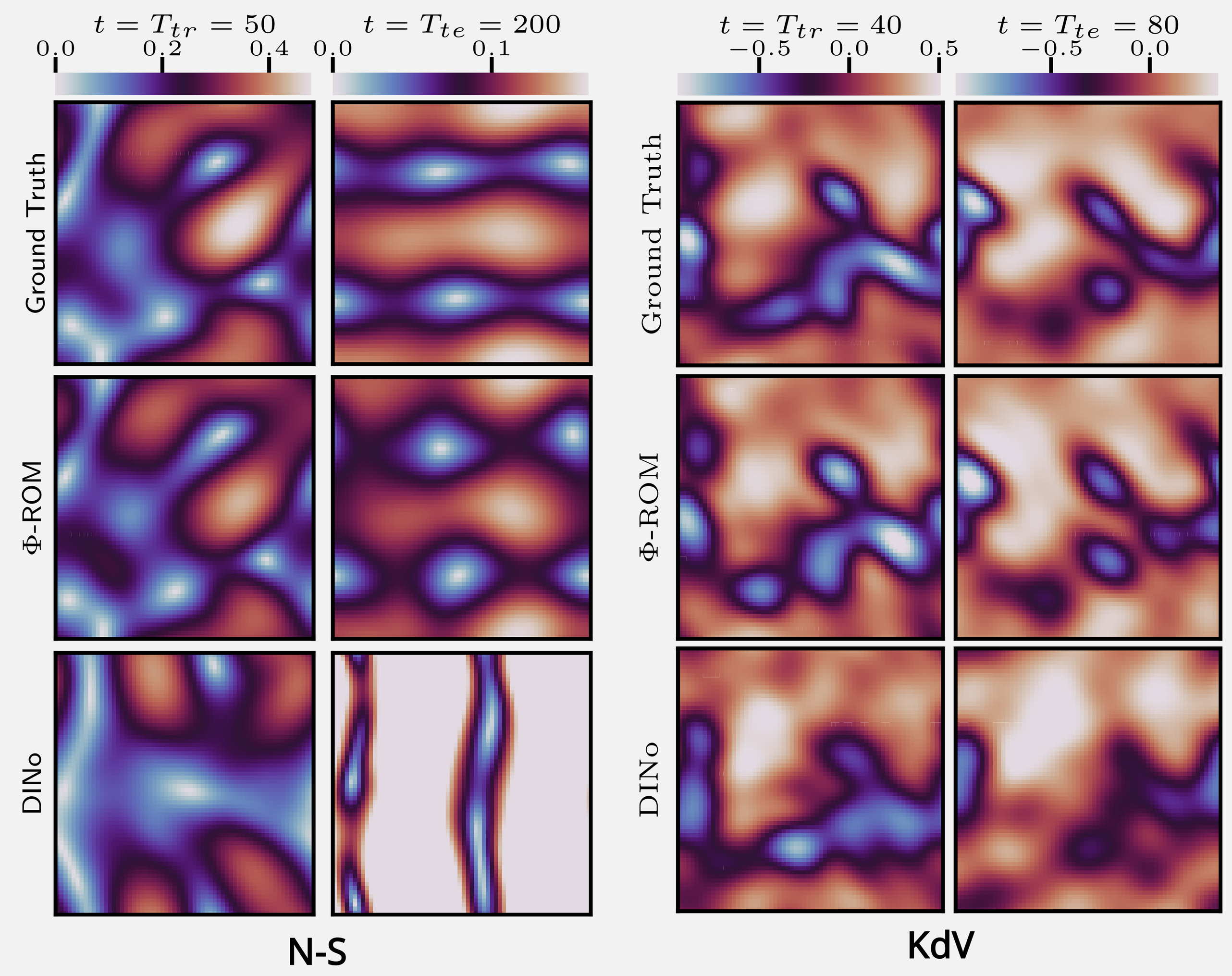
Compared to data-driven DINo:

- ✓ **Φ**-ROM is more stable in time and extrapolates beyond the training time horizon.
- ✓ **Φ**-ROM generalizes better to unseen initial conditions and out-of-dist. parameters.
- ✓ **Φ**-ROM learns and forecast the dynamics from sparse observations (as low as 2% of the data).
- ✓ **Φ**-ROM is robust to the underlying numerical method of the solver.

LBM: Generalization to OOD Reynolds numbers

β	$\beta \in \mathbf{Re}_{tr} = [100, 200]$	$\beta \in \mathbf{Re}_{te} = [200, 300]$
Full Training: $\mathcal{X}_{tr} = \mathcal{X}_\mathcal{S} = \mathcal{X}_{te}$		
Φ-ROM	0.049	**0.115**
DINo	**0.011**	0.457
Sparse Forecasting: $\mathcal{X}_{tr} = \mathcal{X}_\mathcal{S}$, $\lvert\mathcal{X}_{te}\rvert = 2\%\lvert\mathcal{X}_\mathcal{S}\rvert$		
Φ-ROM	0.049	**0.182**
DINo	**0.011**	0.400
Sparse Training: $\lvert\mathcal{X}_{tr}\rvert = 2\%\lvert\mathcal{X}_\mathcal{S}\rvert$, $\mathcal{X}_{te} = \mathcal{X}_\mathcal{S}$		
Φ-ROM	**0.065**	**0.188**
DINo	0.369	0.412





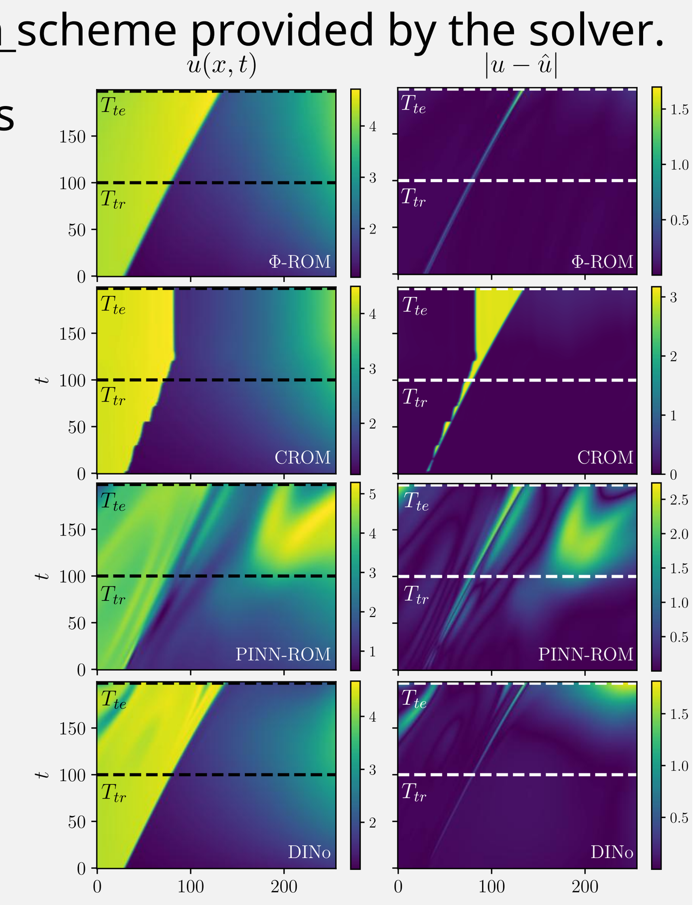
Results: Discretized Solver vs Continuous Physics, Data Efficiency

PINN-ROM and CROM use continuous physics along with auto-diff, while **Φ**-ROM incorporates the discretized physics embedded in a numerical solver.

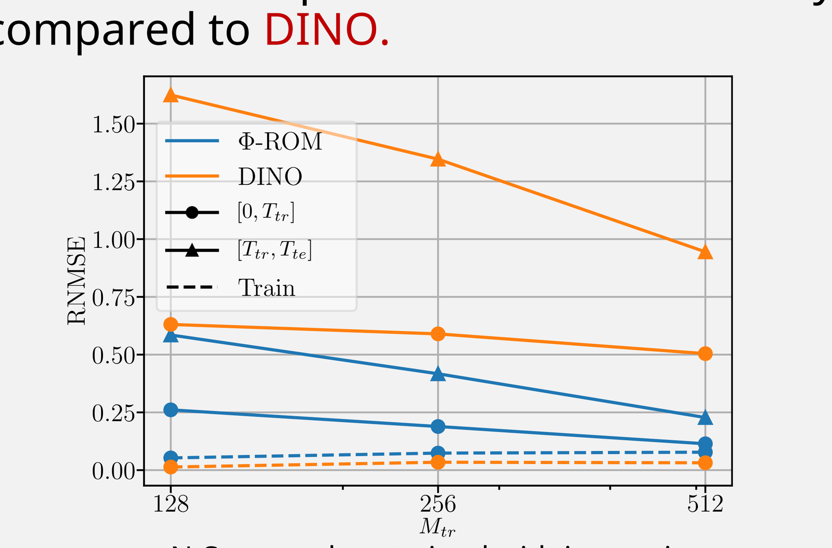
➤ **Φ**-ROM does not suffer from auto-diff inaccuracies.

➤ **Φ**-ROM follows a convergent discretization scheme provided by the solver.

As a result, **Φ**-ROM is robust to various physical phenomena, such as shock waves.



Φ-ROM also improves the data-efficiency compared to DINO.



N-S error when trained with increasing number of data trajectories

References:
[1] Yin, Yuan, et al. "Continuous pde dynamics forecasting with implicit neural representations."
[2] Chen, Peter Yichen, et al. "CROM: Continuous reduced-order modeling of PDEs using implicit neural representations."
[3] Kim, Minji, et al. "Physics-informed reduced order model with conditional neural fields."

Code and Paper

nhosse5@uwo.ca