Closure modeling through the lens of multifidelity operator learning

Shady Ahmed, Panos Stinis
Physical & Computational Sciences Directorate, PNNL

MLDL Workshop – Sandia National Laboratories – July 2023

U.S. DEPARTMENT OF ENERGY
BATTELLE
PNNL is operated by Battelle for the U.S. Department of Energy

Multiscale Phenomena

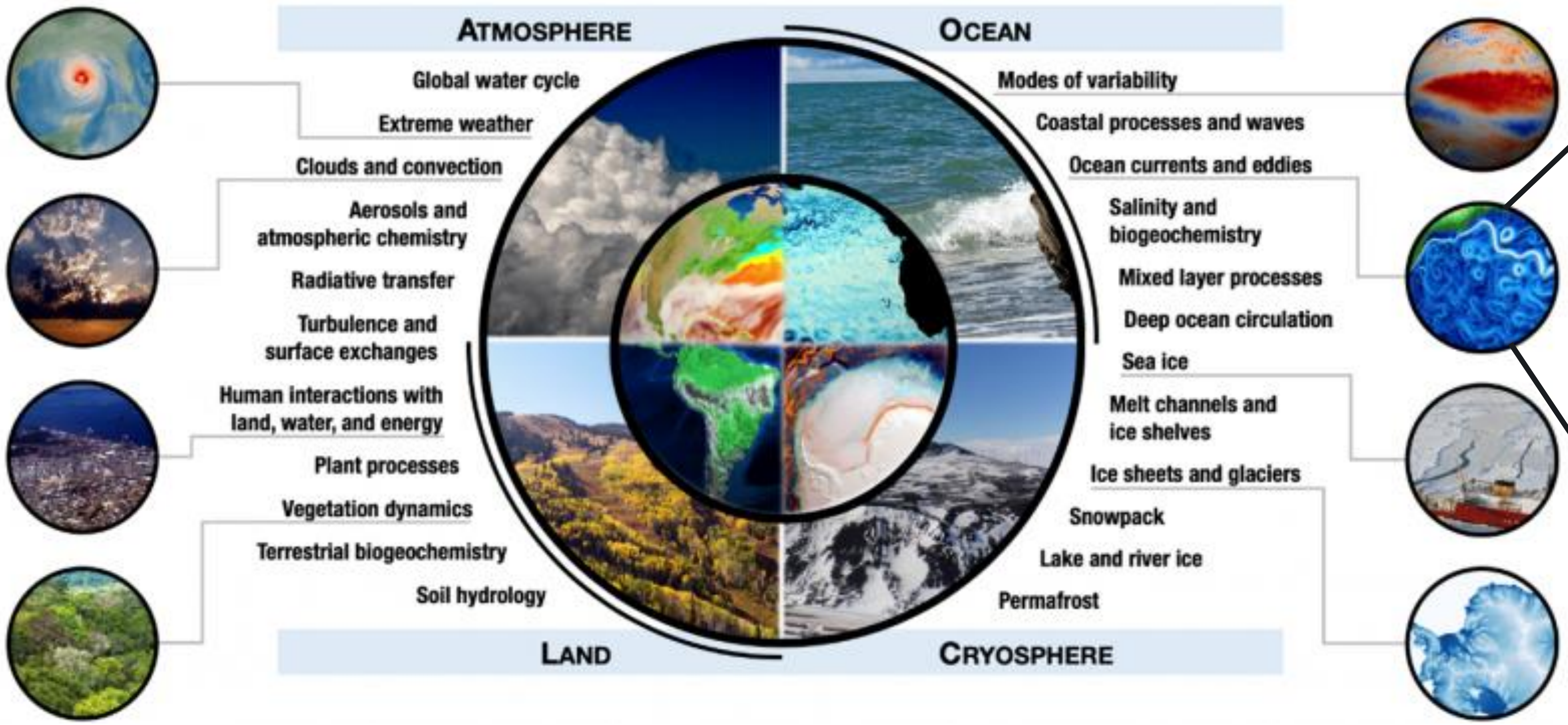
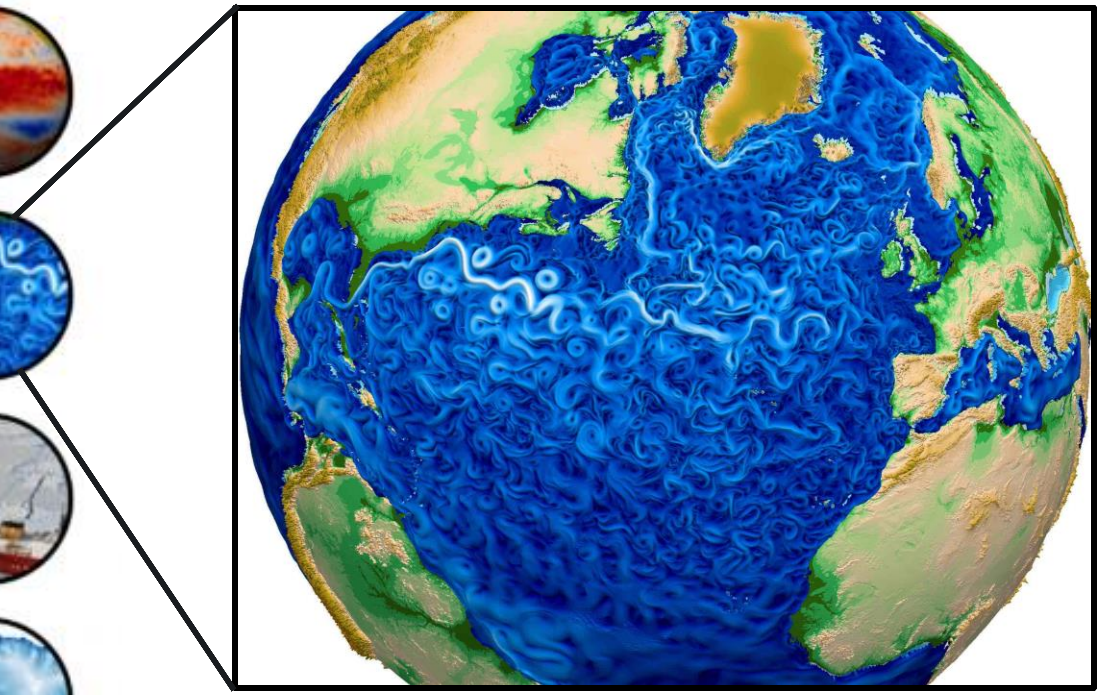


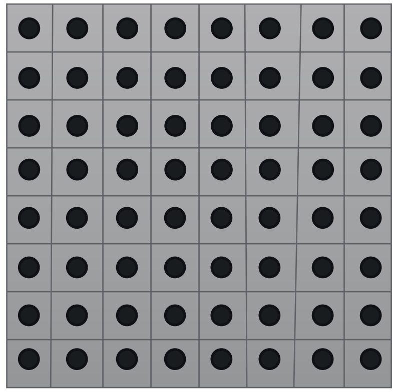
Image courtesy of Paul Ullrich, University of California, Davis
https://www.energy.gov/science/doe-explainsearth-system-and-climate-models



The Energy Exascale Earth System Model (E3SM) Project
https://e3sm.org/

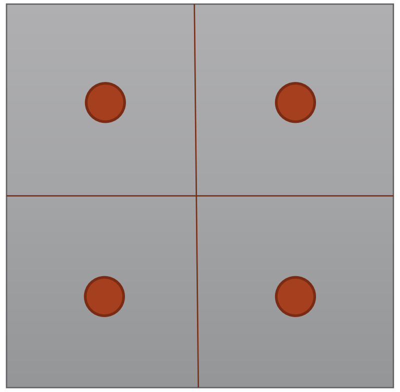
Reduced Complexity Modeling

$$\frac{\partial u}{\partial t} = \mathcal{N}\left(u, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}, \ldots; \mu\right)$$



$$\mathbf{u} \in \mathbb{R}^N$$

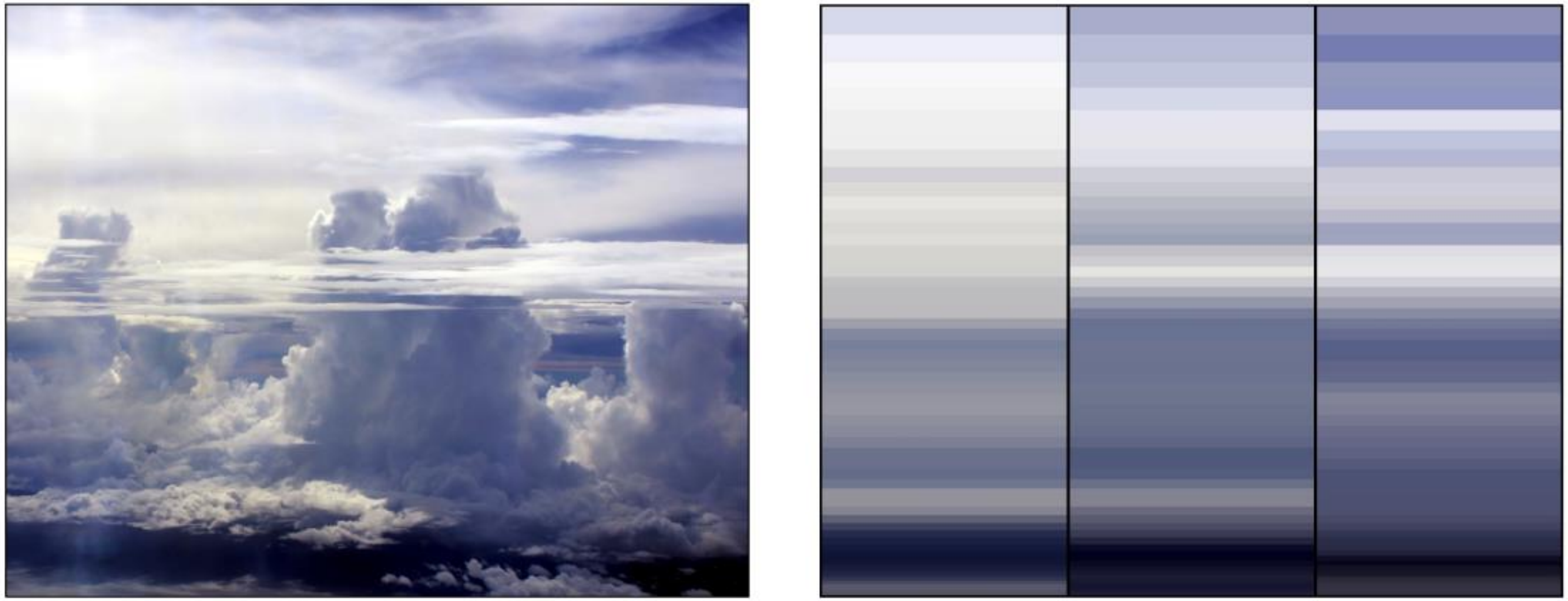
$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathcal{F}(\mathbf{u}; \mu)$$



$$\hat{\mathbf{u}} \in \mathbb{R}^n$$

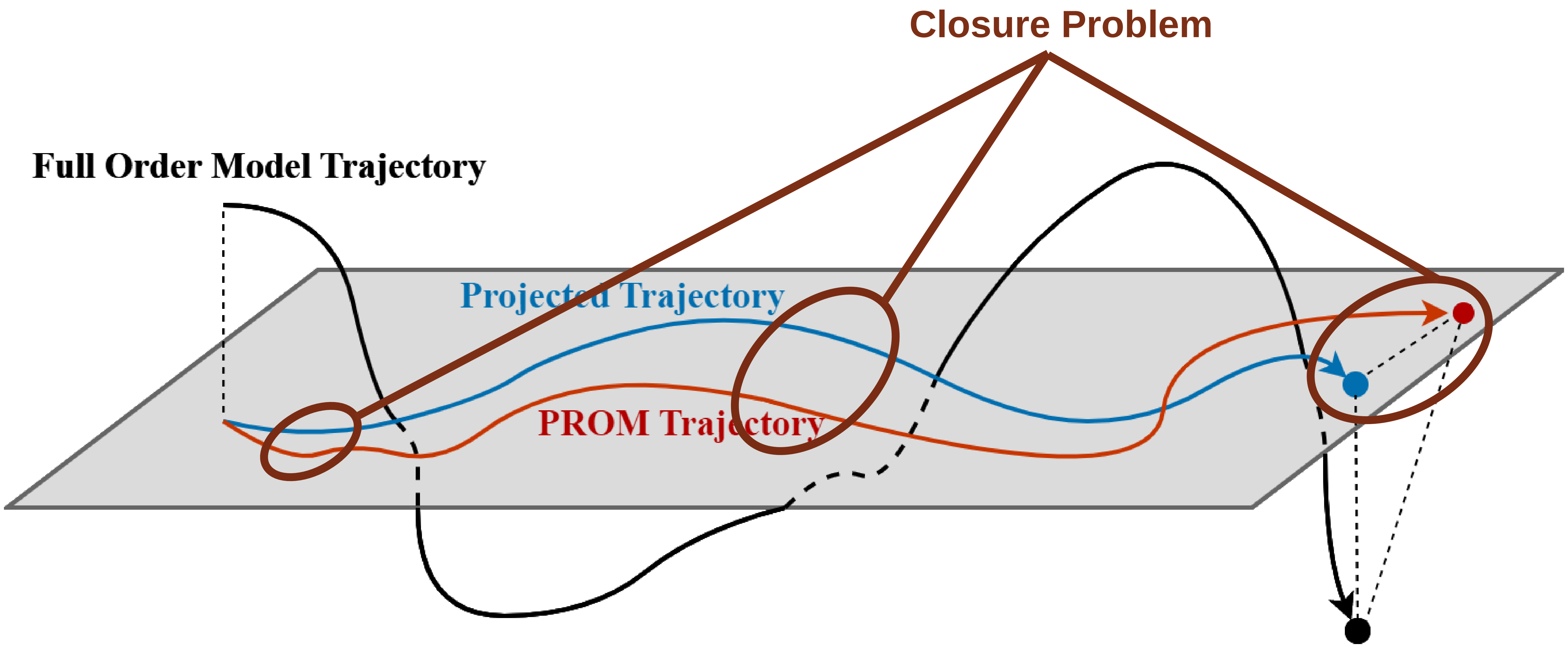
$$\frac{\mathrm{d}\hat{\mathbf{u}}}{\mathrm{d}t} = \hat{\mathcal{F}}(\hat{\mathbf{u}}; \mu)$$

Reduced Complexity Modeling



Allen Institute for AI – https://allenai.org/climate-modeling

Projection Reduced Order Modeling (PROM)



Ahmed, S. E., San, O., Rasheed, A., & Iliescu, T. (2020). A long short-term memory embedding for hybrid uplifted reduced order models. *Physica D: Nonlinear Phenomena*.

Projection Reduced Order Modeling (PROM)

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathcal{F}(\mathbf{u};\mu), \qquad \mathbf{u} \in \mathbb{R}^N$$

$$\mathbf{u}(t) \approx \sum_{i=1}^{R} a_i(t)\boldsymbol{\phi}_i = \Phi\mathbf{a}(t) \qquad R \ll N$$

Proper Orthogonal Decomposition (POD)

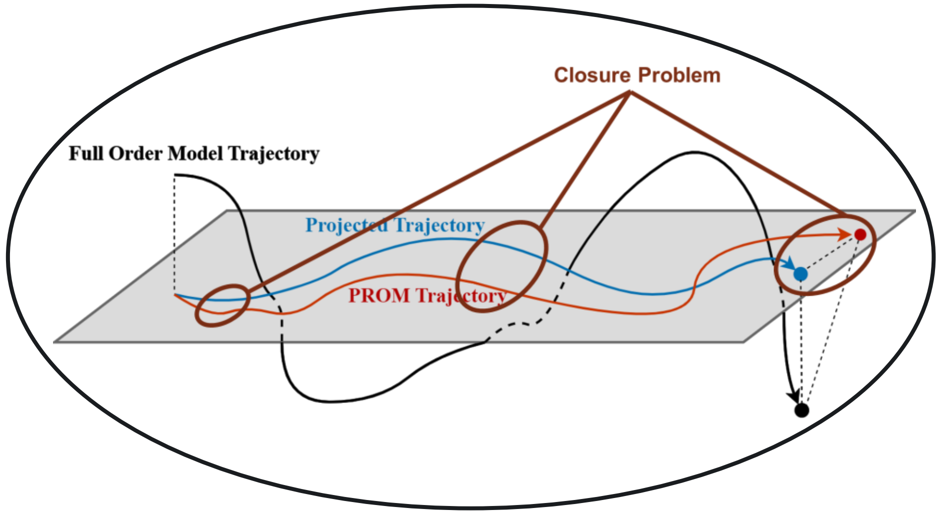
Projection Reduced Order Modeling (PROM)

$$\mathbf{u}(t) = \Phi \mathbf{a} \iff \mathbf{a}(t) = \Phi^{\mathsf{T}} \mathbf{u}(t)$$

$$\frac{\mathrm{d}\mathbf{a}}{\mathrm{d}t} = \Phi^{\mathsf{T}} \mathcal{F}(\Phi \mathbf{a}; \mu), \qquad \mathbf{a} \in \mathbb{R}^R$$

Galerkin Projection

$$\mathbf{a}(t_{n+1}) = G(\mathbf{a}(t_n); \mu) + \mathbf{c}(t_{n+1})$$



Multifidelity Learning for Closure Modeling

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathcal{F}(\mathbf{u};\mu) \quad \Longrightarrow \quad \mathbf{a}(t) = \Phi^{\mathsf{T}}\mathbf{u}(t)$$

High Fidelity Model

$$\frac{\mathrm{d}\mathbf{a}}{\mathrm{d}t} = \Phi^{\mathsf{T}}\mathcal{F}(\Phi\mathbf{a};\mu), \qquad \mathbf{a} \in \mathbb{R}^R$$

$$\mathbf{a}(t_{n+1}) = G(\mathbf{a}(t_n);\mu)$$

Low Fidelity Model

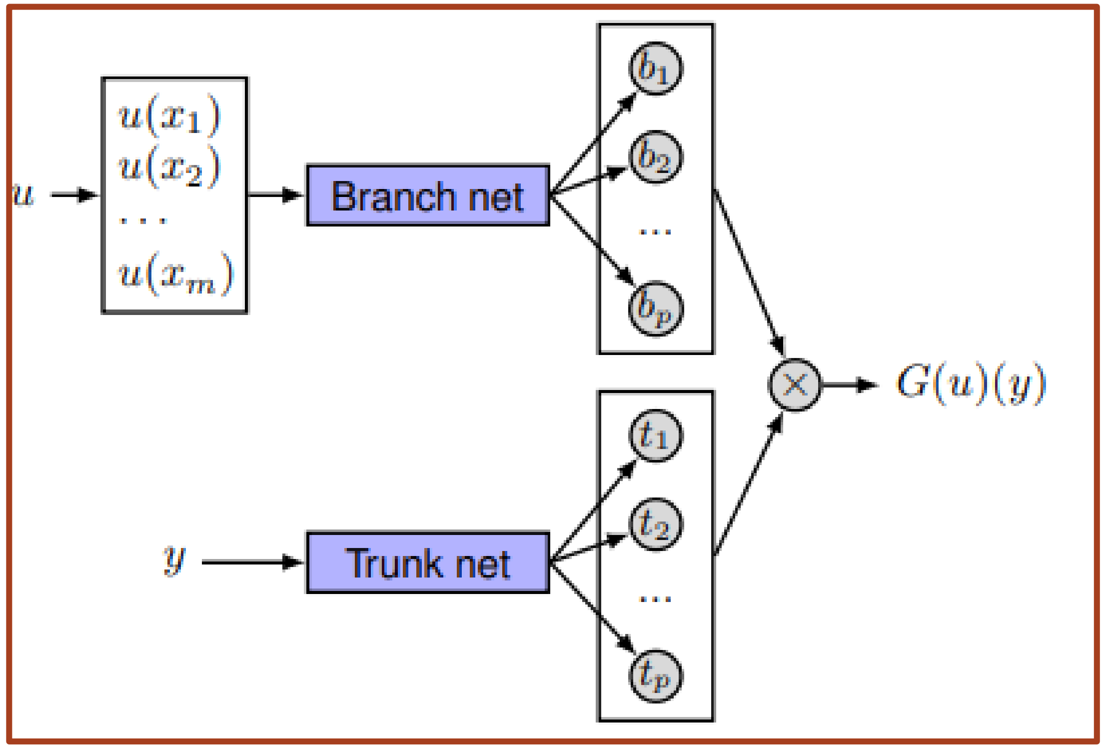
Operator Learning (Quick Detour)

Universal Approximation Theorem for ***Functions***: Suitable for similar conditions, fixed initial/boundary conditions, fixed parameters, etc.

Universal Approximation Theorem for ***Operators***: Generalizable for wide range of conditions, varying initial/boundary conditions, different parameters, etc.

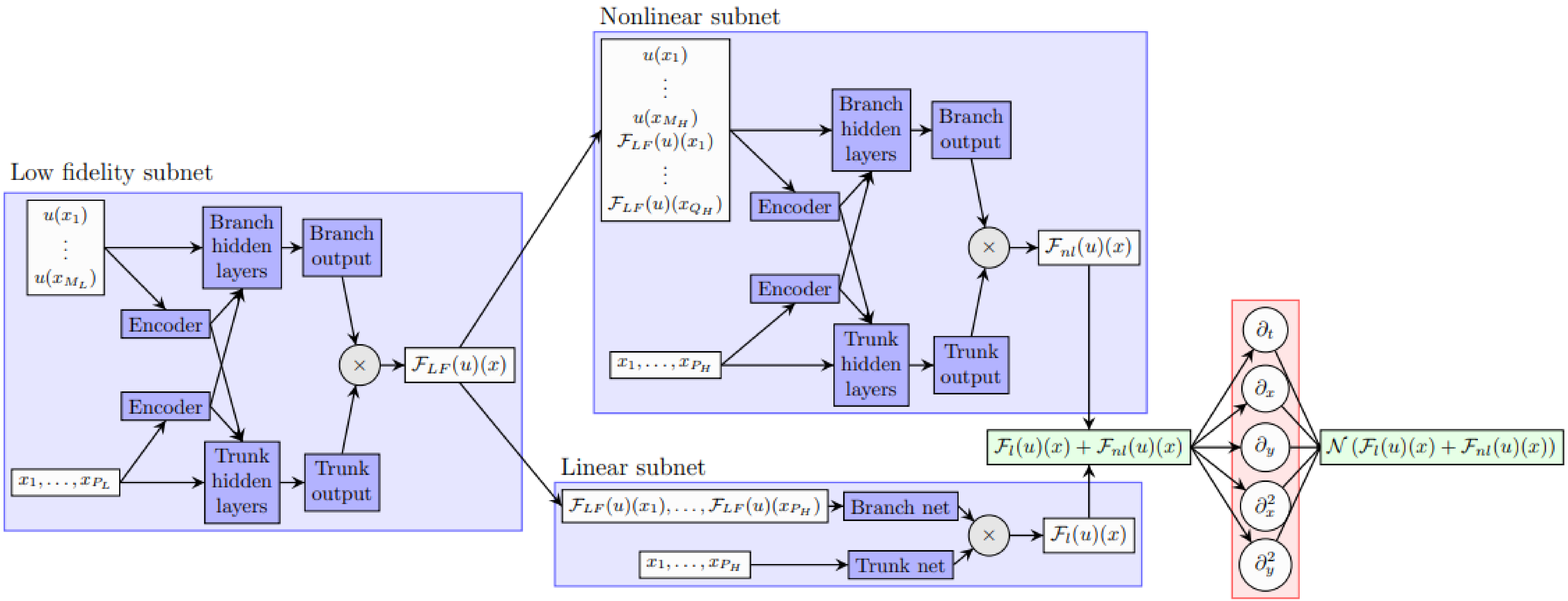
$$G(u)(y) = \sum_{k=1}^{p} b_k t_k$$

Deep Operator Network (DeepONet)



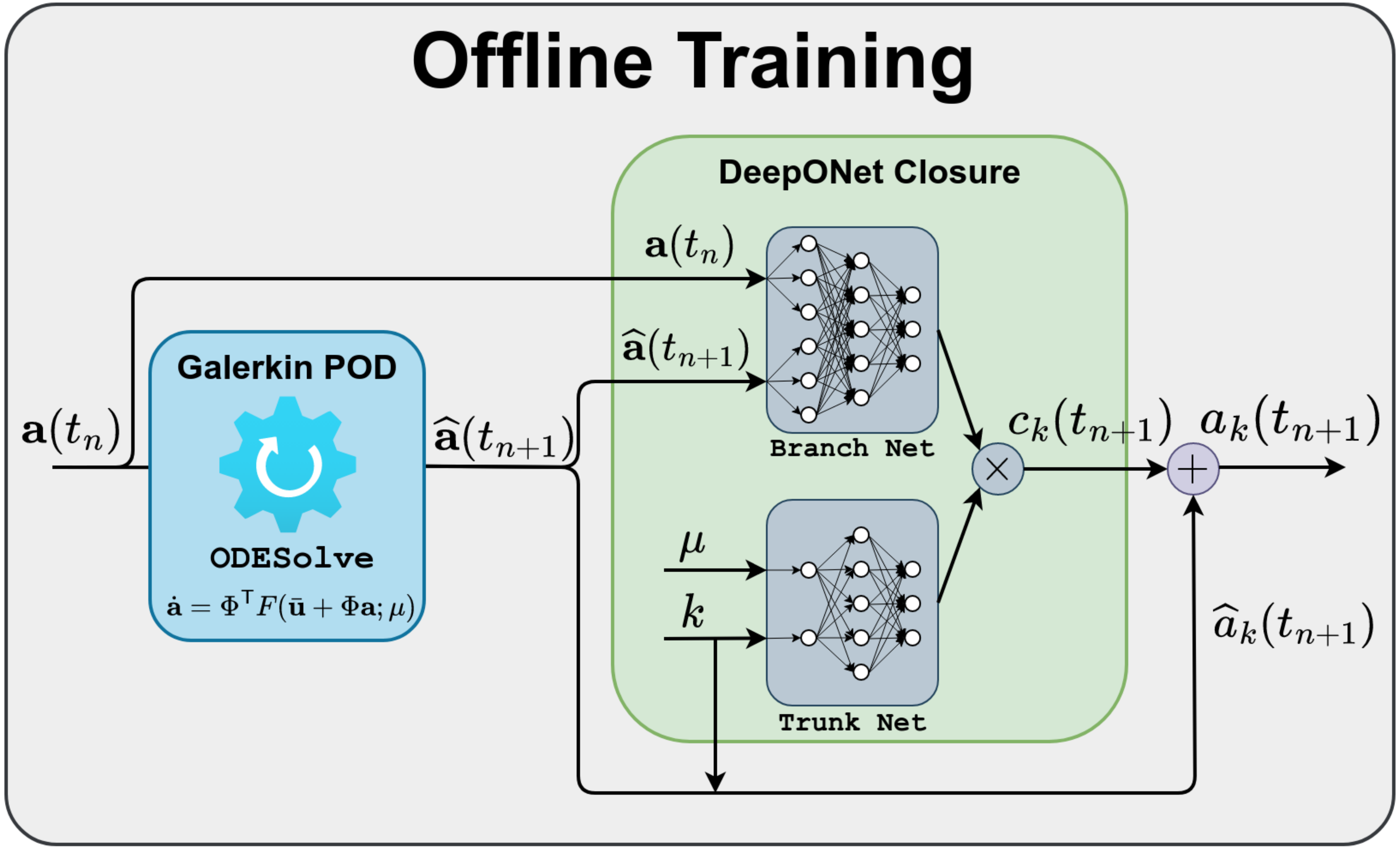
Lu, L., Jin, P., Pang, G., Zhang, Z., & Karniadakis, G. E. (2021). Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature machine intelligence*.

Multifidelity Operator Learning



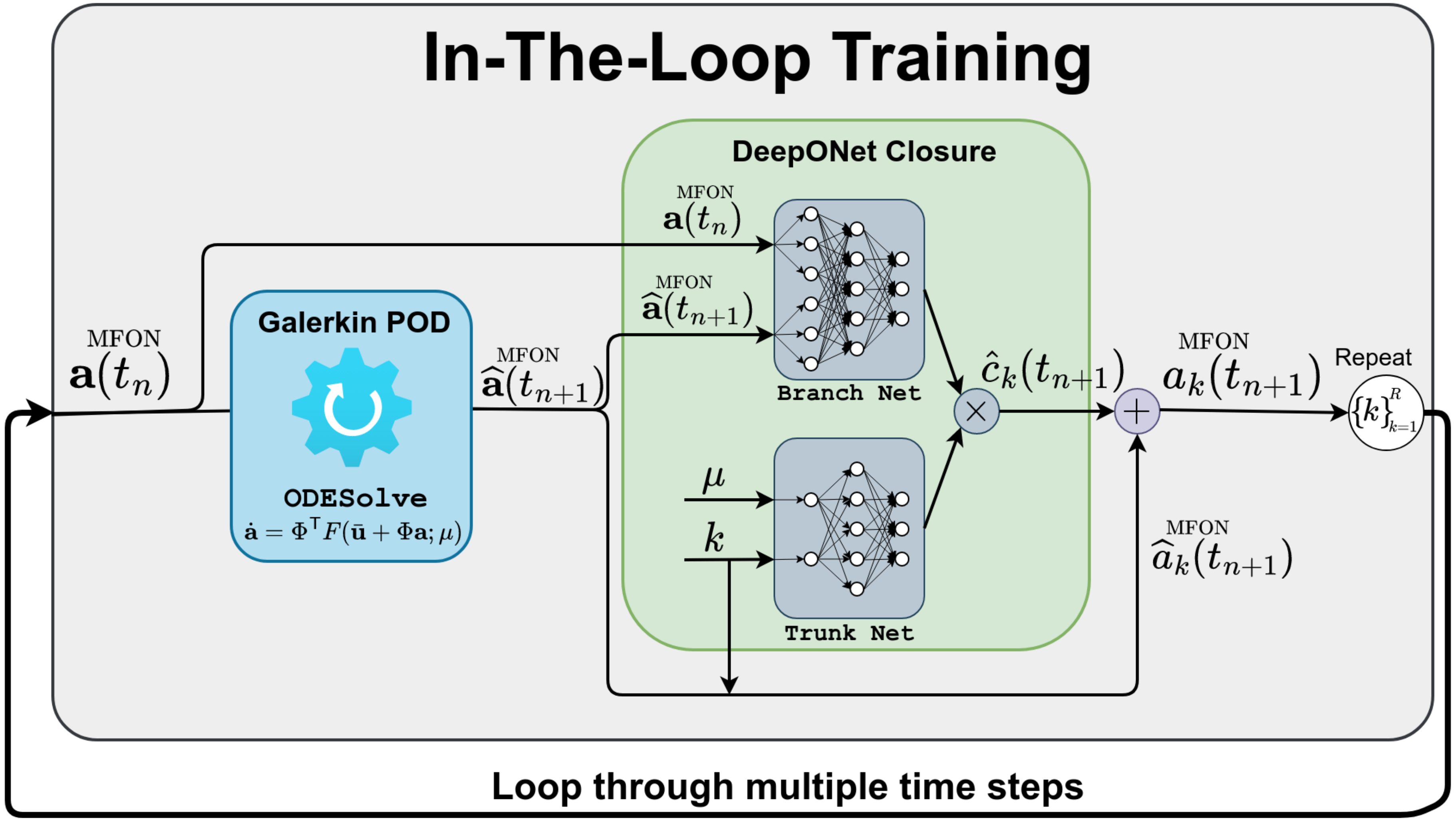
Howard, A. A., Perego, M., Karniadakis, G. E., & Stinis, P. (2022). Multifidelity deep operator networks. *arXiv preprint arXiv:2204.09157*.

Multifidelity Operator Learning for Closure Modeling



Multifidelity Operator Learning for Closure Modeling – In-The-Loop Training

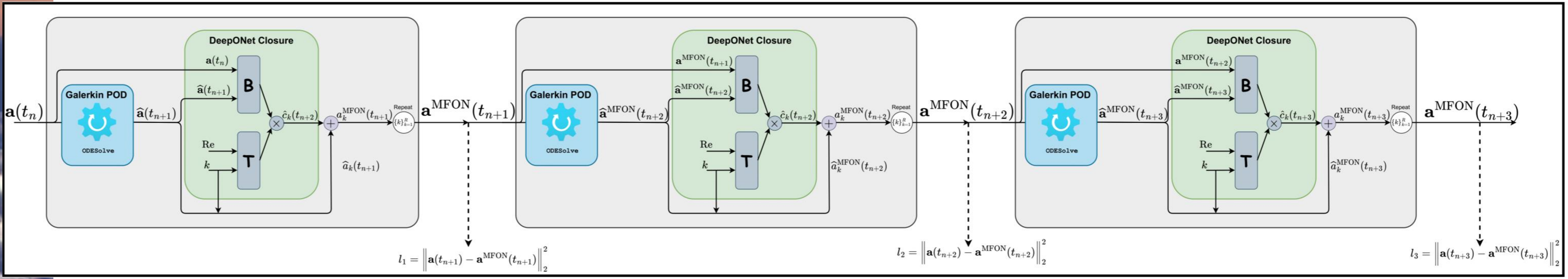
Differentiable Programming/Automatic Differentiation + Machine Learning ➔ (Coupled) In-The-Loop Training



Ahmed, S. E., & Stinis, P. (2023). A Multifidelity deep operator network approach to closure for multiscale systems. *CMAME*.

Multifidelity Operator Learning for Closure Modeling – In-The-Loop Training

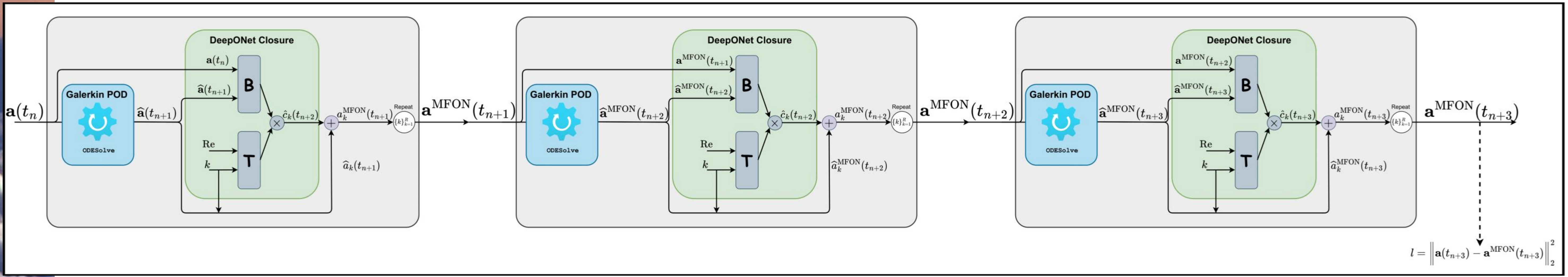
Differentiable Programming/Automatic Differentiation + Machine Learning ➔ (Coupled) In-The-Loop Training



$$l(t_n, t_{n+\tau}) = \frac{1}{\tau}\sum_{k=1}^{\tau}\left\| \mathbf{a}(t_{n+k}) - \hat{\mathbf{a}}^{\text{MFON}}(t_{n+k}) - \mathcal{Q}^{\Theta}\left(\left[\mathbf{a}^{\text{MFON}}(t_{n+k-1})^{\mathsf{T}}, \hat{\mathbf{a}}^{\text{MFON}}(t_{n+k})^{\mathsf{T}}\right]^{\mathsf{T}}\right)\left([\mu, k]^{\mathsf{T}}\right)\right\|_2^2$$

Multifidelity Operator Learning for Closure Modeling – In-The-Loop Training

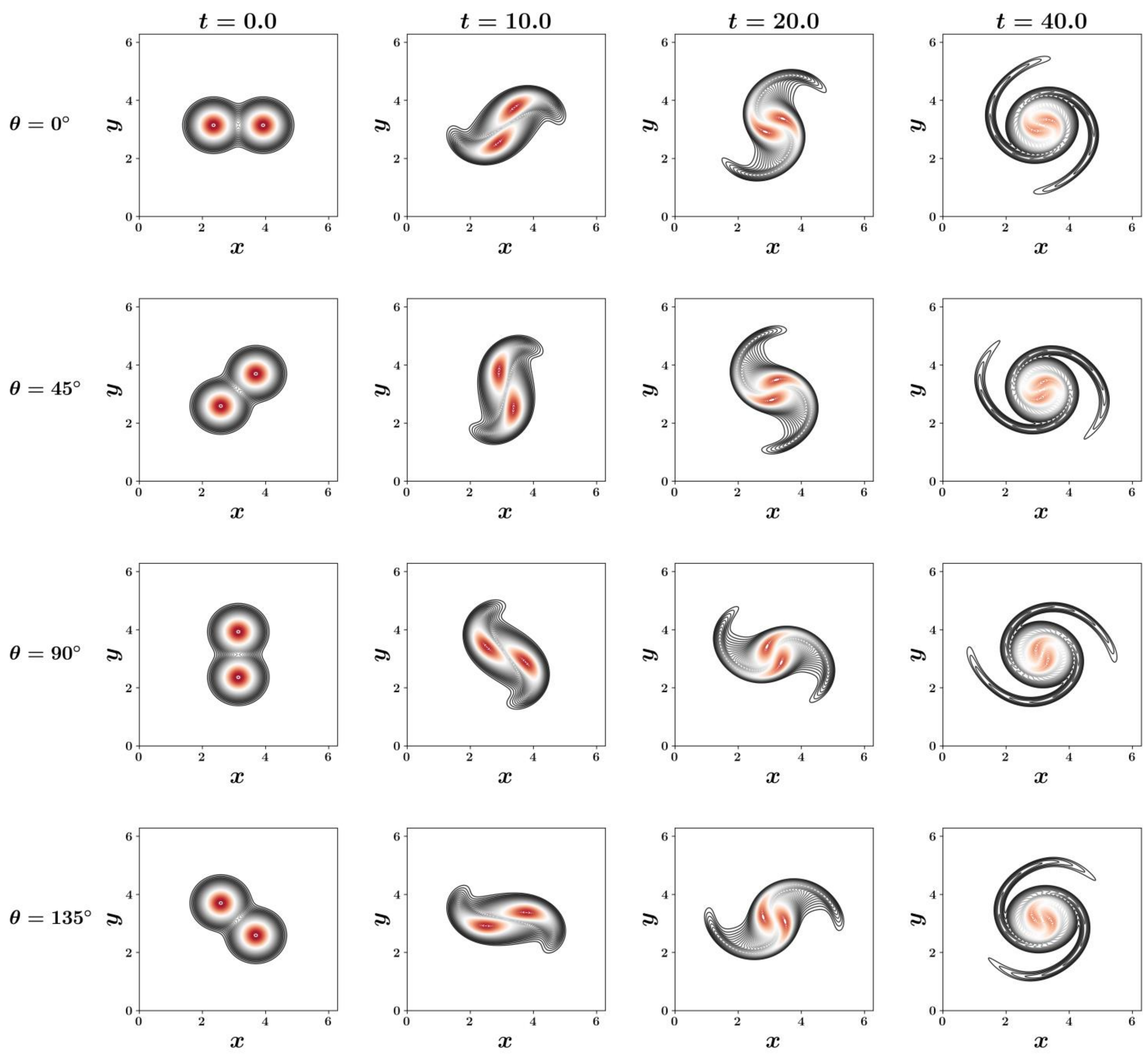
Differentiable Programming/Automatic Differentiation + Machine Learning ➔ (Coupled) In-The-Loop Training



$$l(t_n, t_{n+\tau}) = \left\| \mathbf{a}(t_{n+\tau}) - \hat{\mathbf{a}}^{\mathrm{MFON}}(t_{n+\tau}) - \mathcal{Q}^{\Theta}\left([\mathbf{a}^{\mathrm{MFON}}(t_{n+\tau-1})^{\mathsf{T}}, \hat{\mathbf{a}}^{\mathrm{MFON}}(t_{n+\tau})^{\mathsf{T}}]^{\mathsf{T}} \right)\left([\mu, k]^{\mathsf{T}} \right) \right\|_2^2$$

Numerical Tests – Vortex Merger Problem

$$\frac{\partial \omega}{\partial t} + J(\omega, \psi) = \frac{1}{\mathrm{Re}} \nabla^2 \omega$$



Full Order Model (FOM):

Finite Difference Method
256×256, $\Delta t = 10^{-4}$

Reduced Order Model (ROM):

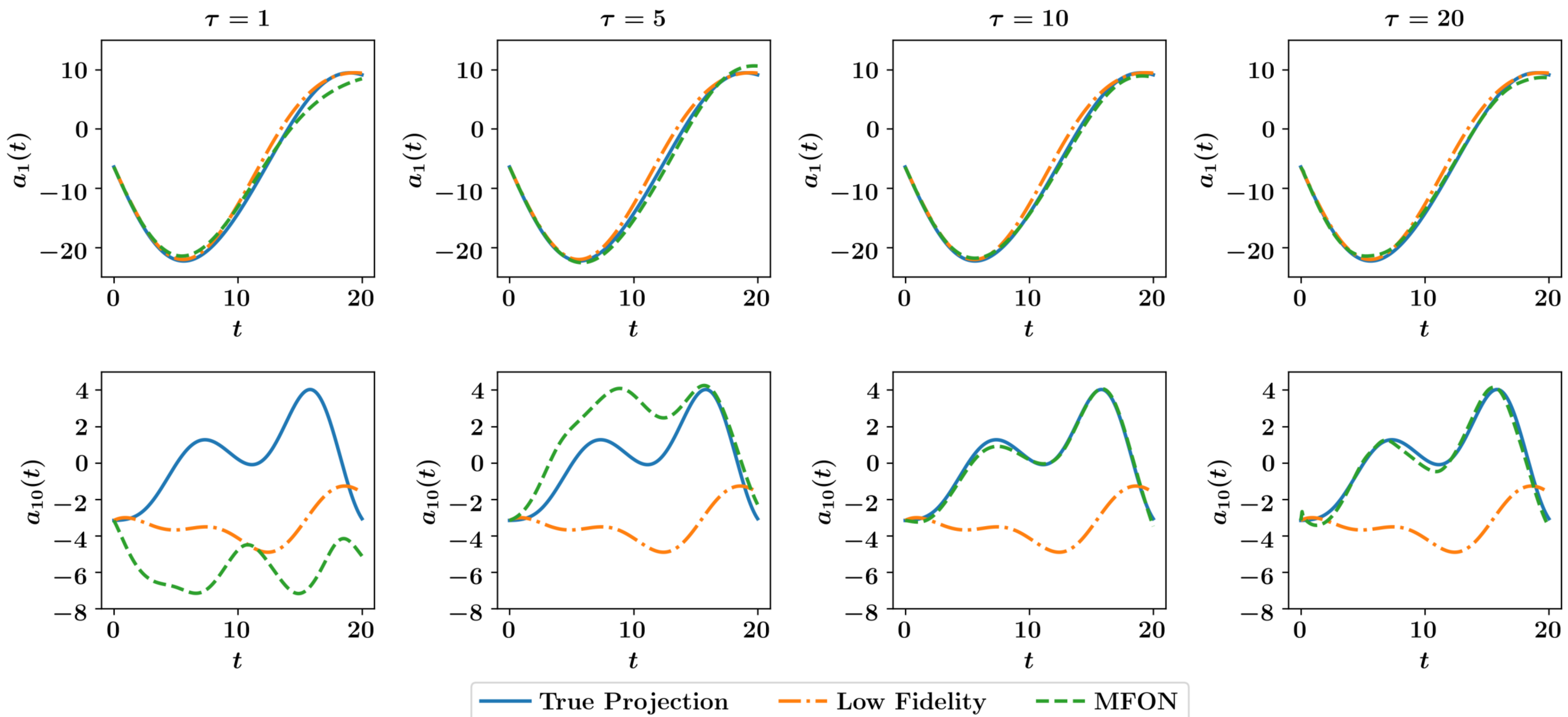
Galerkin POD
10 modes, $\Delta t = 10^{-2}$

Training:

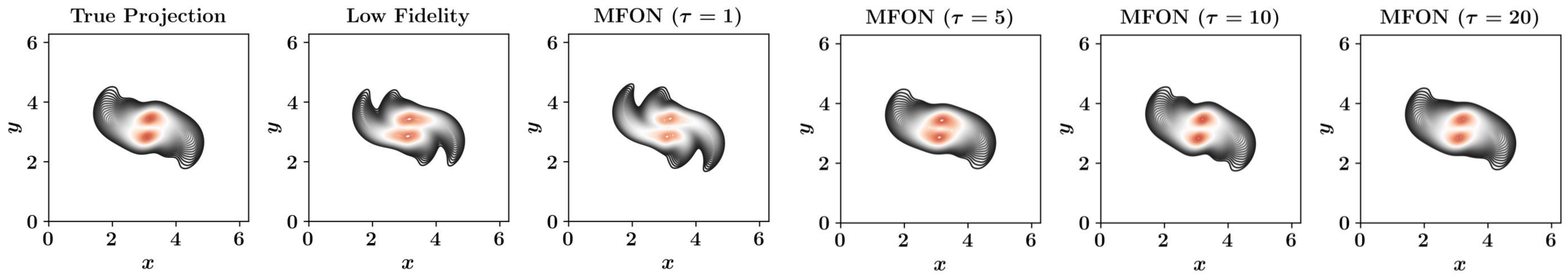
Reynolds number $\in [1000,2000]$
$\theta \in \{0,45,90,135\}$
$t \in [0,20]$

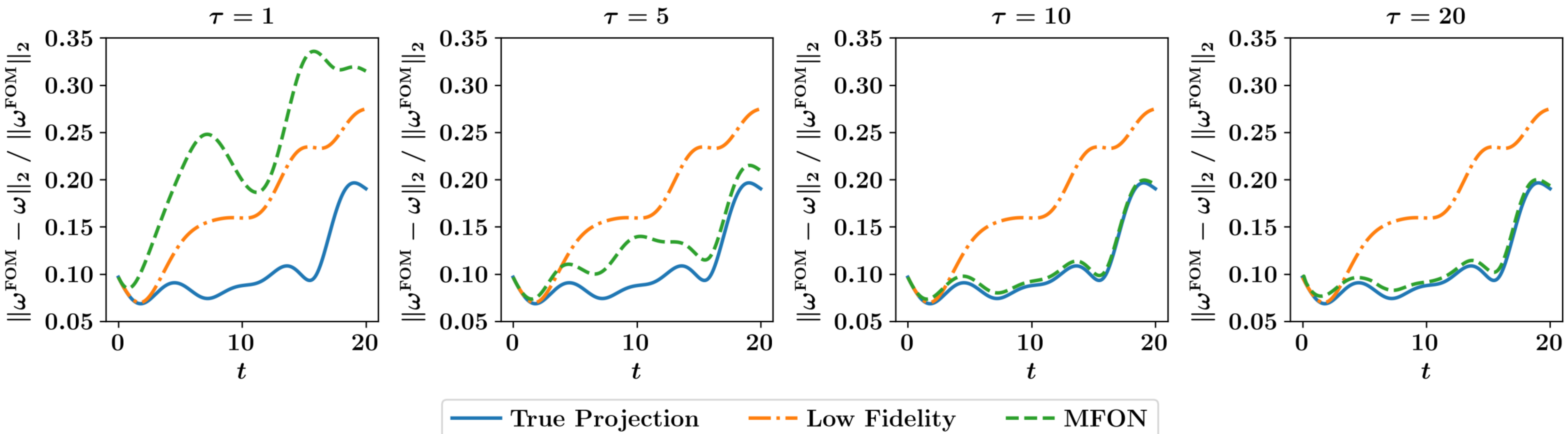
Results – Vortex Merger Problem (Interpolation)

Reynolds number $= 1500$, $\theta = 60$, $t \in [0, 20]$



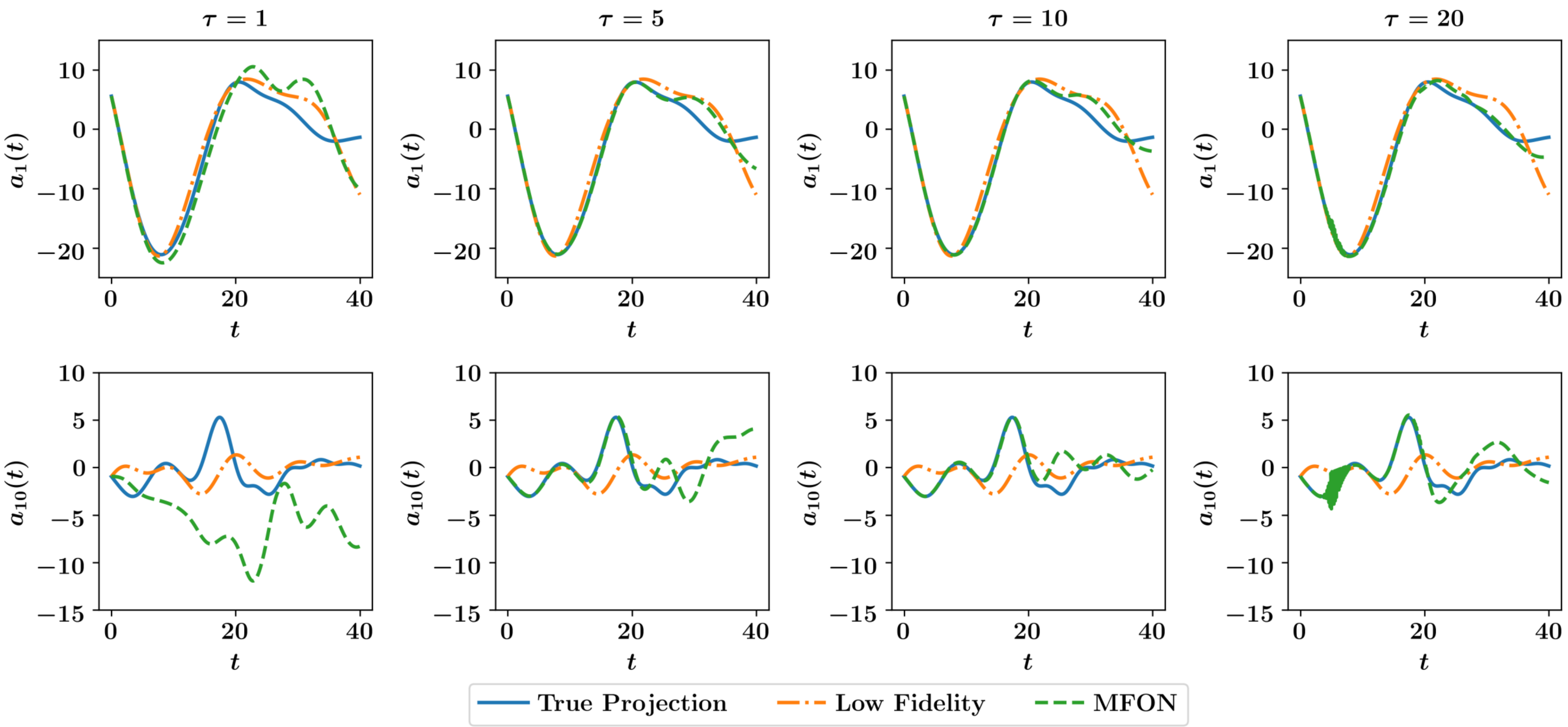
Results – Vortex Merger Problem (Interpolation)



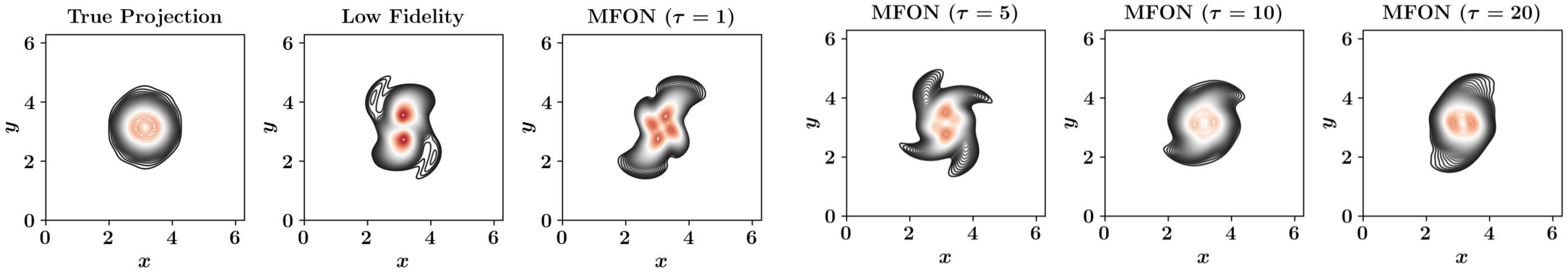


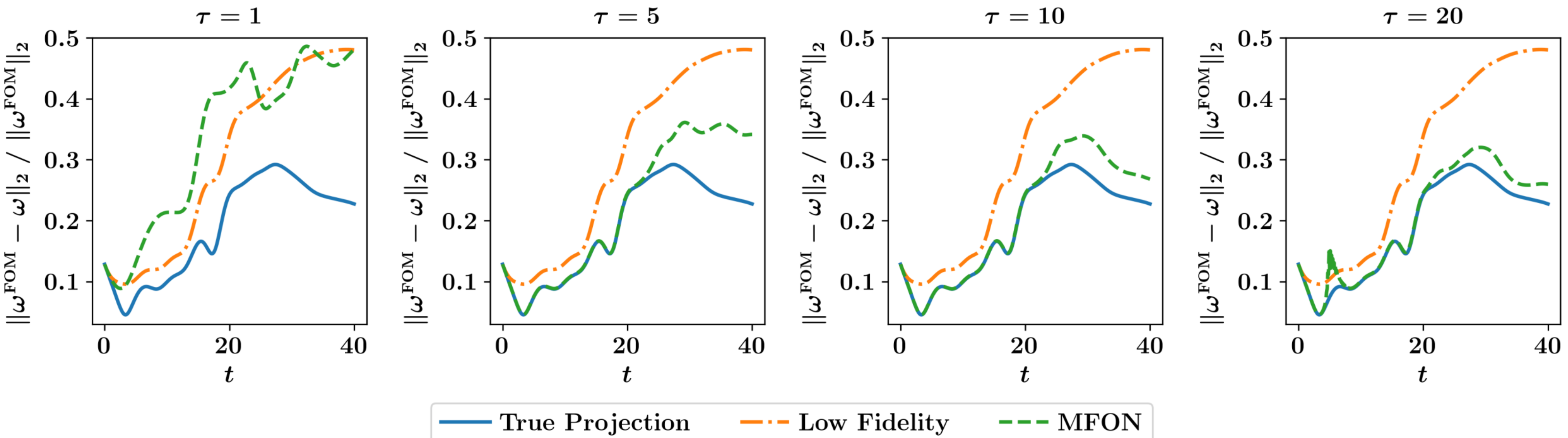
Results – Vortex Merger Problem (Extrapolation)

Reynolds number = $\mathbf{3000}$, $\boldsymbol{\theta = 45}$, $\boldsymbol{t \in [0, 40]}$



Results – Vortex Merger Problem (Extrapolation)





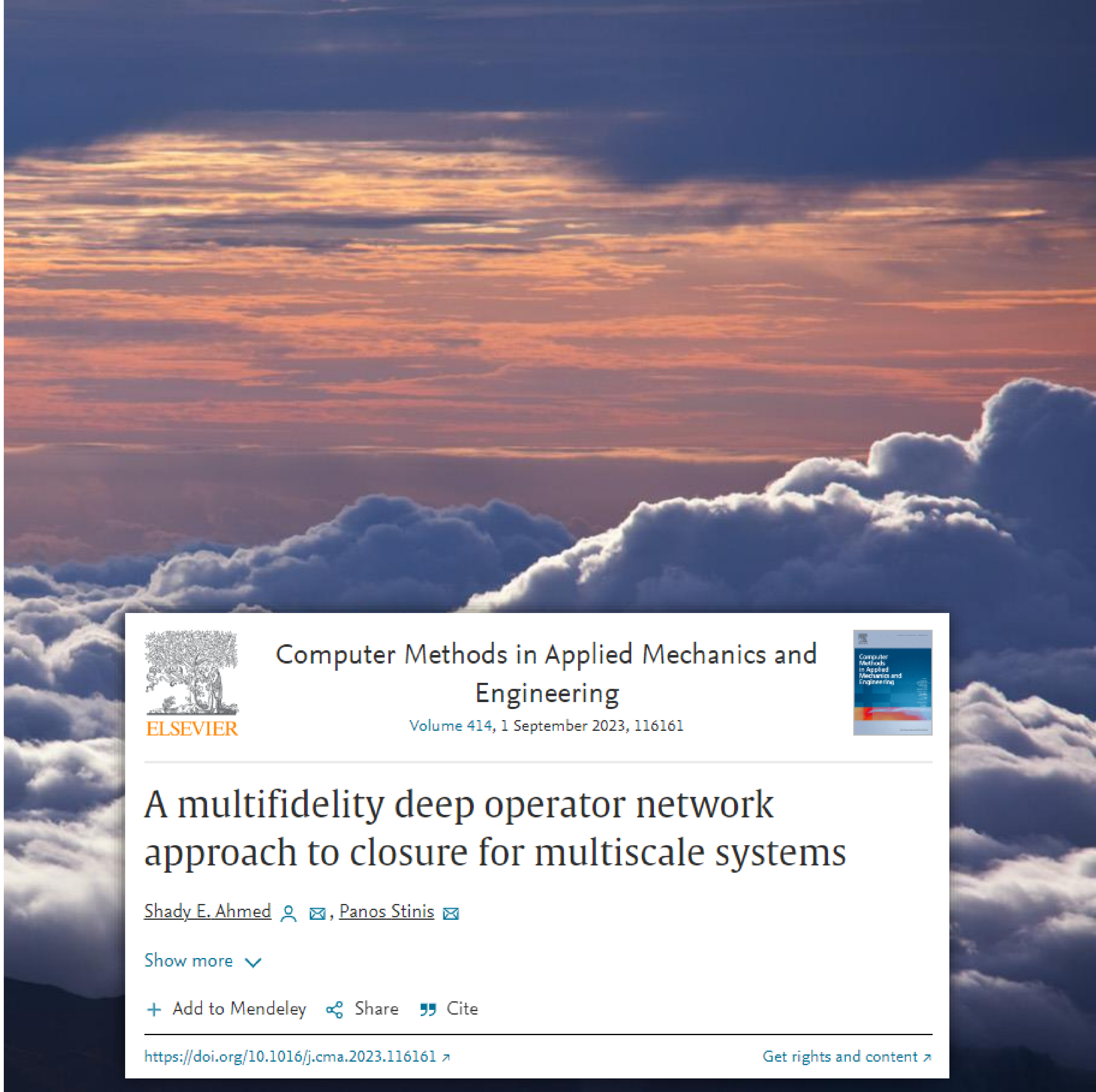
Conclusions

- Reduced order (complexity) modeling of multiscale phenomena lead to the closure problem
- Closure modeling can be viewed as a multifidelity learning problem
- Multifidelity operator network can learn closure terms for a wide range of initial conditions and/or parameters
- In-the-loop training provides a *feedback loop* so that the machine learning model ***sees the effect*** of its previous predictions ***during the training***

Ahmed, S. E., & Stinis, P. (2023). A Multifidelity deep operator network approach to closure for multiscale systems. *Computer Methods in Applied Mechanics and Engineering*.

Thank you

https://arxiv.org/abs/2303.08893


ELSEVIER
Computer Methods in Applied Mechanics and Engineering
Volume 414, 1 September 2023, 116161

A multifidelity deep operator network approach to closure for multiscale systems

Shady E. Ahmed, Panos Stinis

Show more

Add to Mendeley Share Cite

https://doi.org/10.1016/j.cma.2023.116161 Get rights and content

Supplementary

PNNL is operated by Battelle for the U.S. Department of Energy

Galerkin POD Model

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \frac{1}{\mathrm{Re}}\frac{\partial^2 u}{\partial x^2}$$

$$u(x,t) \approx \sum_{i=1}^{R} a_i(t)\phi_i(x) = \mathbf{\Phi a}(t)$$

$$\frac{\partial}{\partial t}\left(\sum_{i=1}^{R} a_i(t)\phi_i(x)\right) + \left(\sum_{i=1}^{R} a_i(t)\phi_i(x)\right)\frac{\partial}{\partial x}\left(\sum_{i=1}^{R} a_i(t)\phi_i(x)\right) = \frac{1}{\mathrm{Re}}\frac{\partial^2}{\partial x^2}\left(\sum_{i=1}^{R} a_i(t)\phi_i(x)\right)$$

$$\left(\sum_{i=1}^{R} \frac{\partial a_i(t)}{\partial t}\phi_i(x)\right) + \left(\sum_{i=1}^{R} a_i(t)\phi_i(x)\right)\left(\sum_{i=1}^{R} a_i(t)\frac{\partial \phi_i(x)}{\partial x}\right) = \frac{1}{\mathrm{Re}}\left(\sum_{i=1}^{R} a_i(t)\frac{\partial^2 \phi_i(x)}{\partial x^2}\right)$$

Galerkin POD Model

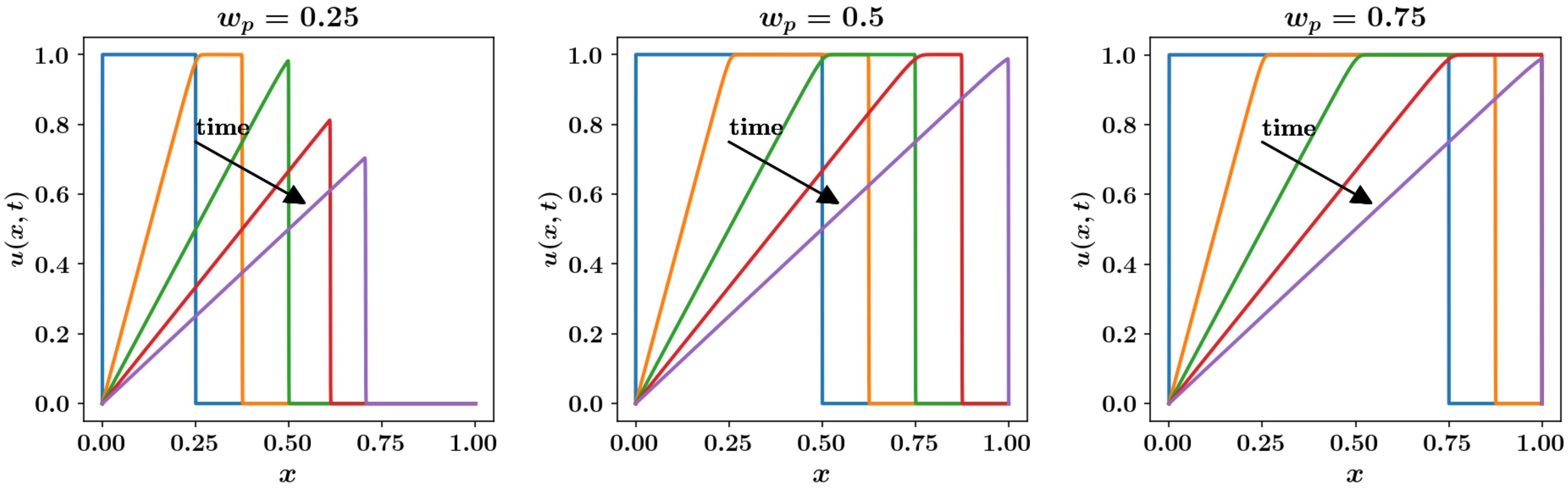
$$\left(\sum_{i=1}^{R} \frac{\partial a_i(t)}{\partial t}\phi_i(x)\right) + \underbrace{\left(\sum_{i=1}^{R} a_i(t)\phi_i(x)\right)\left(\sum_{i=1}^{R} a_i(t)\frac{\partial \phi_i(x)}{\partial x}\right)}_{\sum_{i=1}^{R}\sum_{j=1}^{R} a_i(t)a_j(t)\phi_i(x)\frac{\partial \phi_j(x)}{\partial x}} = \frac{1}{\mathrm{Re}}\left(\sum_{i=1}^{R} a_i(t)\frac{\partial^2 \phi_i(x)}{\partial x^2}\right)$$

$$\frac{\partial a_k(t)}{\partial t} + \sum_{i=1}^{R}\sum_{j=1}^{R} a_i(t)a_j(t)\left(\phi_i(x)\frac{\partial \phi_j(x)}{\partial x}; \phi_k(x)\right) = \frac{1}{\mathrm{Re}}\sum_{i=1}^{R} a_i(t)\left(\frac{\partial^2 \phi_i(x)}{\partial x^2}; \phi_k(x)\right)$$

$$\frac{d\mathbf{a}}{dt} = \mathbf{La} + \mathbf{a}^{\mathsf{T}}\mathbf{Na}$$

Numerical Tests – Burgers Problem

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \frac{1}{\mathrm{Re}}\frac{\partial^2 u}{\partial x^2}$$



Full Order Model (FOM):

Finite Difference Method
$n_x = 4096$, $\Delta t = 10^{-4}$

Reduced Order Model (ROM):

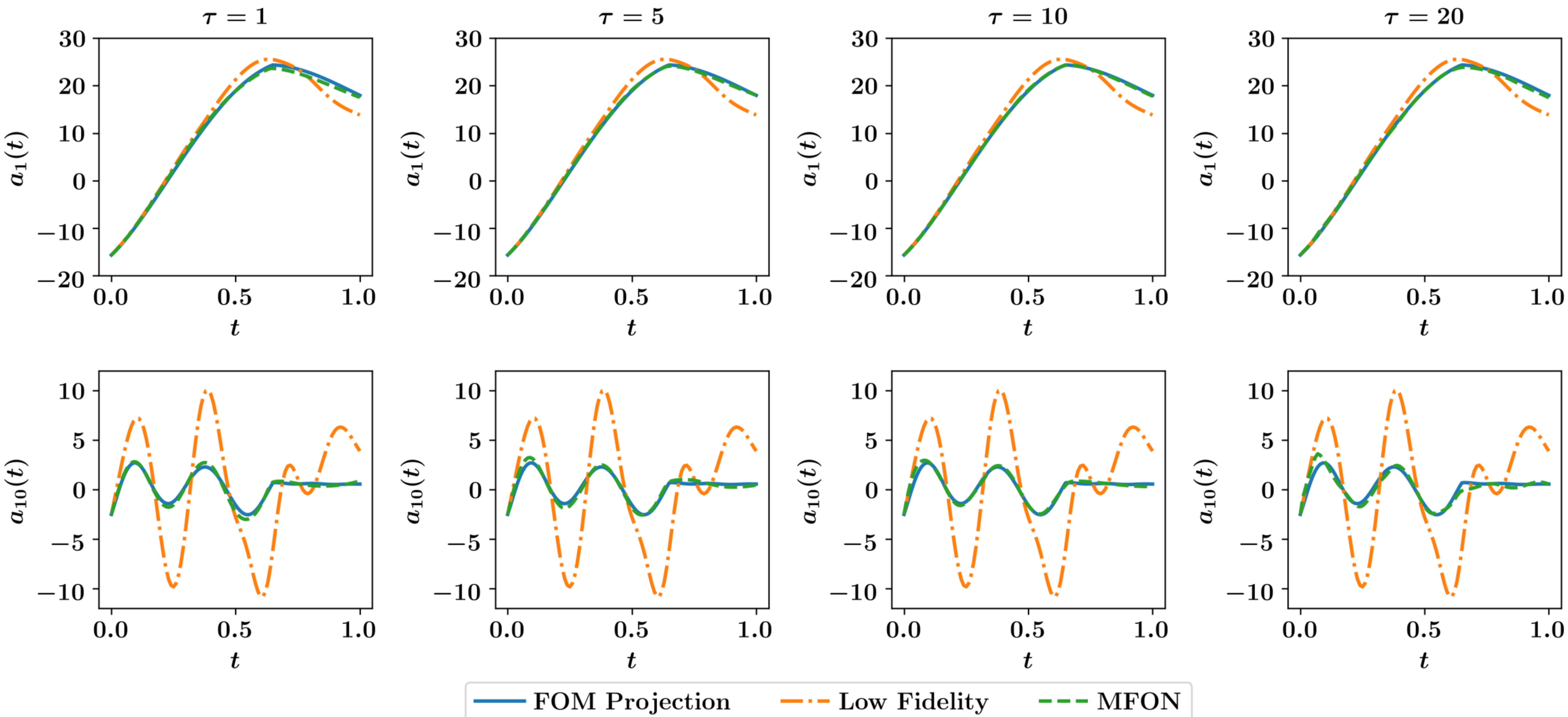
Galerkin POD
10 modes, $\Delta t = 10^{-2}$

Training:

Reynolds number $\in [2500, 10000]$
$w_p \in \{0.25, 0.50, 0.75\}$
$t \in [0,1]$

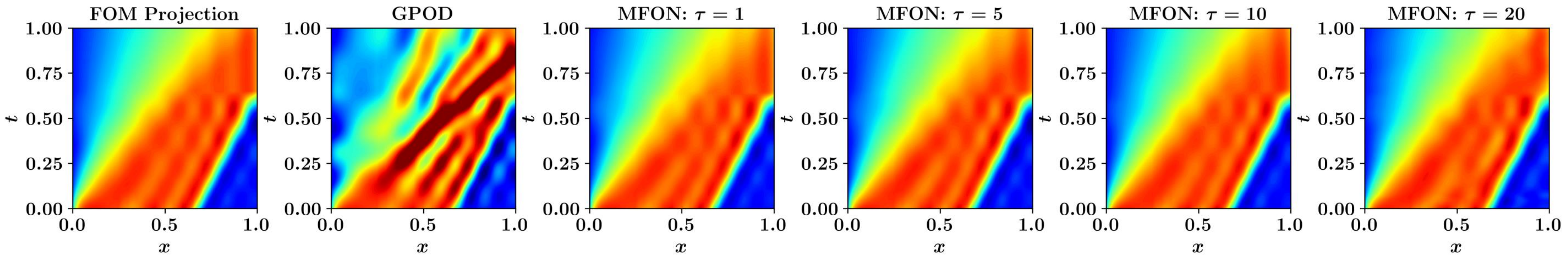
Results – Burgers Problem (Interpolation)

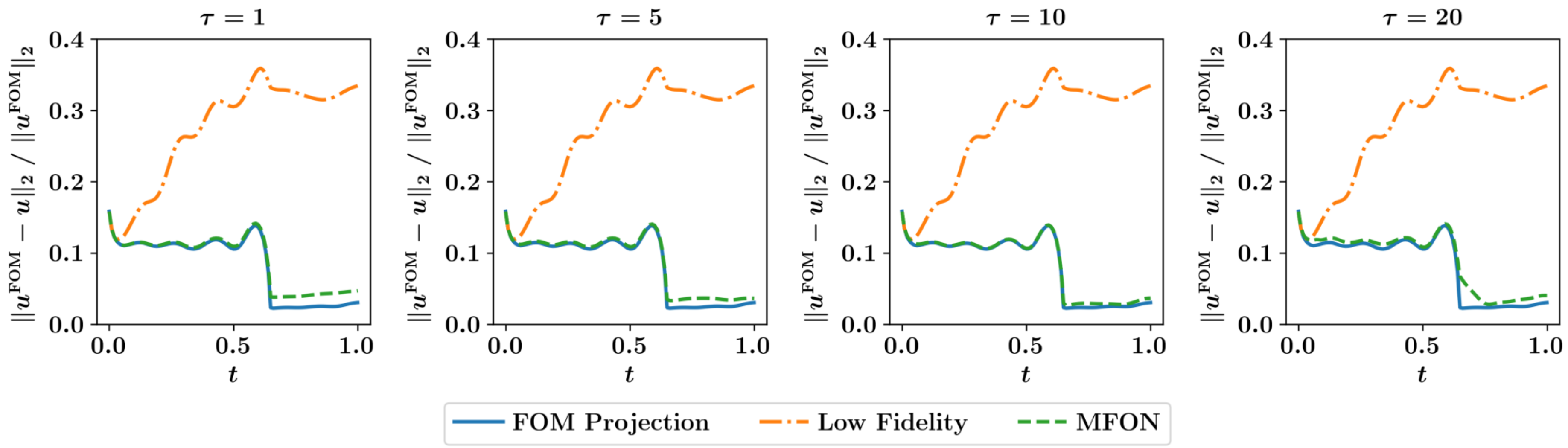
Reynolds number $= 4000,\ w_p = 0.675, t \in [0, 1]$



Results – Burgers Problem (Interpolation)

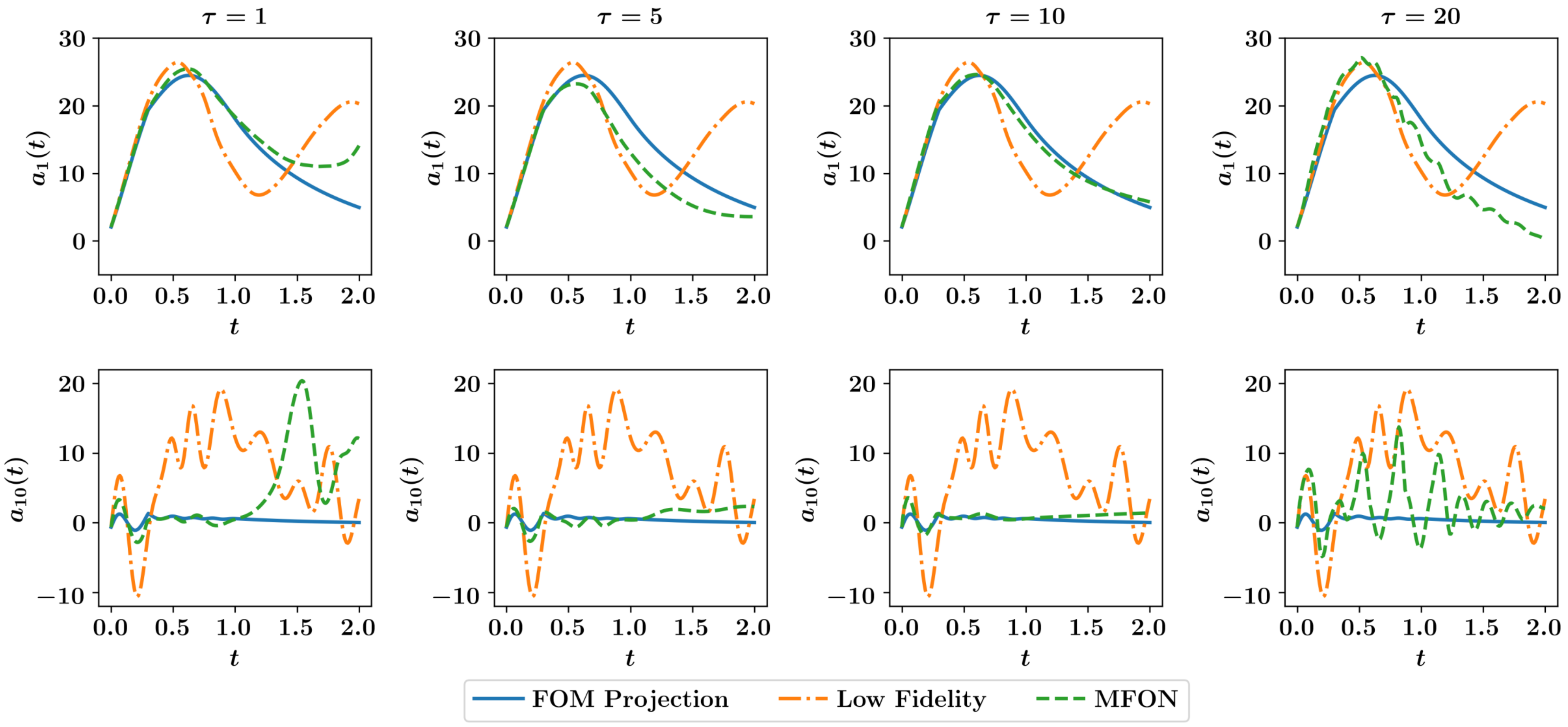
Reynolds number $= 4000,\ w_p = 0.675,\ t \in [0, 1]$





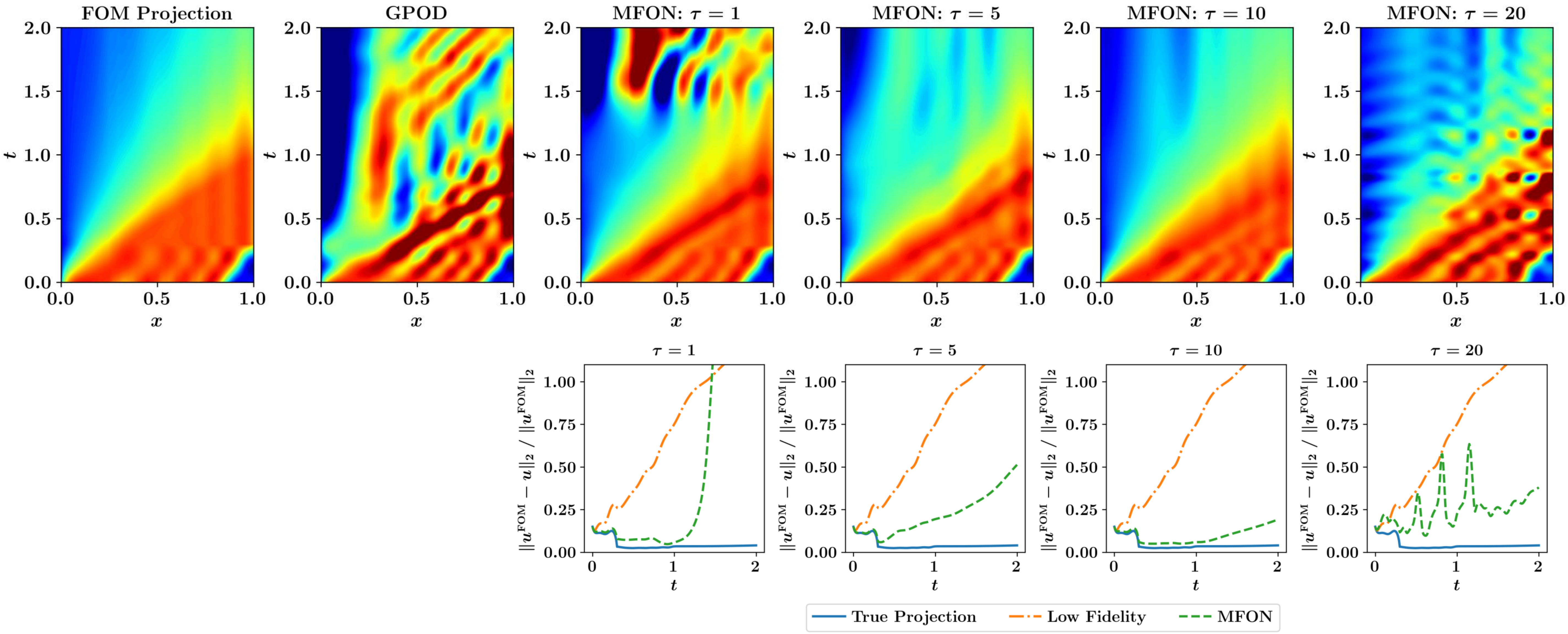
Results – Burgers Problem (Extrapolation)

Reynolds number $= 15000$, $w_p = 0.85$, $t \in [0, 2]$

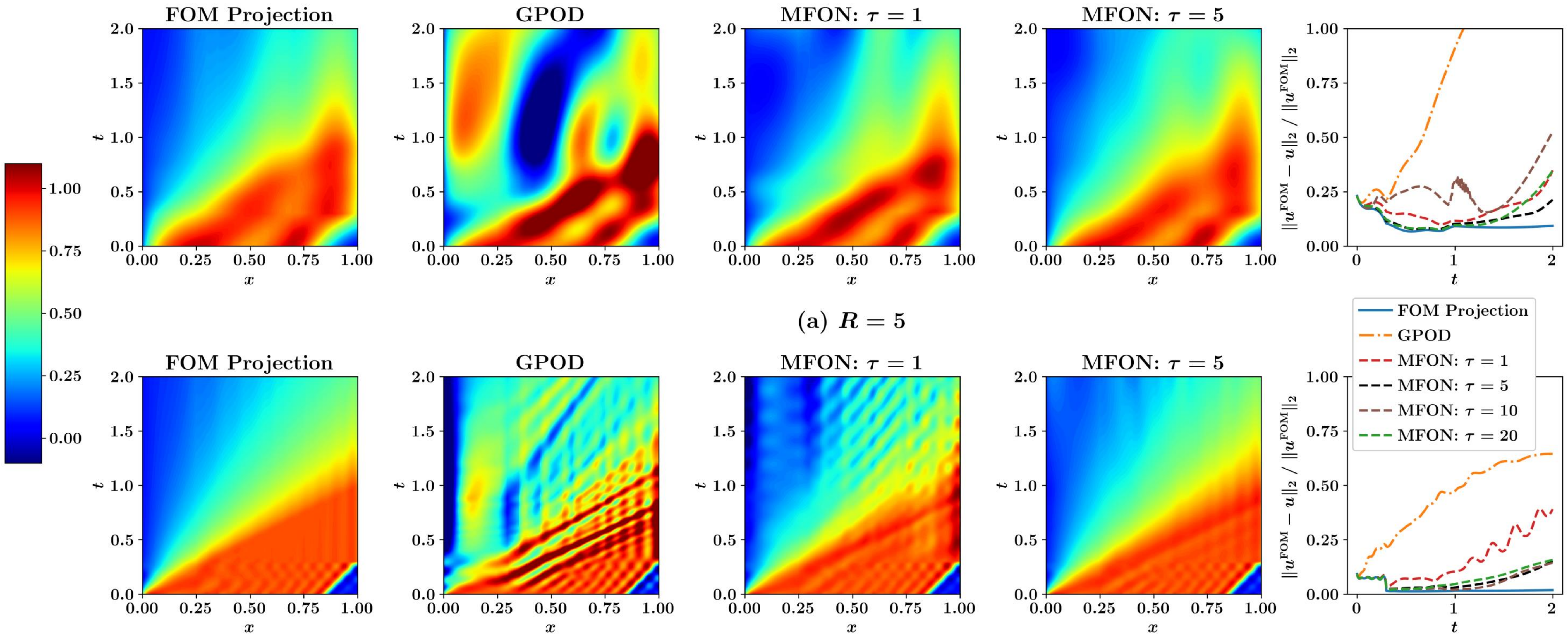


Results – Burgers Problem (Extrapolation)

Reynolds number $= 15000,\ w_p = 0.85,\ t \in [0, 2]$



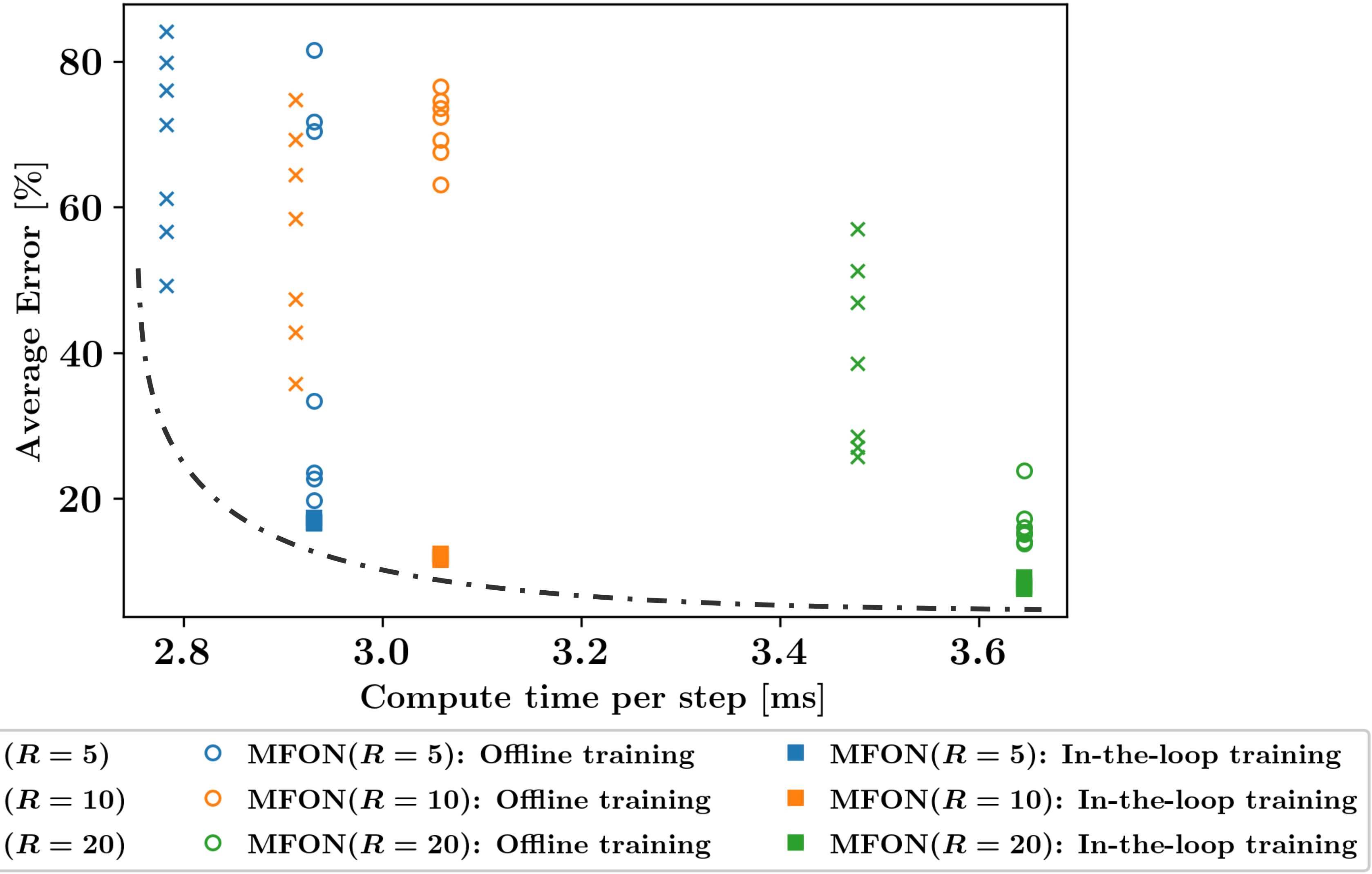
Results – Burgers Problem (Extrapolation)



(a) $R = 5$

(a) $R = 20$

Results – Burgers Problem (Accuracy vs Efficiency)



Results – Burgers Problem (Extrapolation)