TEXAS A&M UNIVERSITY
Engineering

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

Petrobras Workshop: Scientific Machine Learning (SciML) and Data-Driven Model Reduction for Reservoir Simulation

Eduardo Gildin
Petroleum Engineering Department at
Texas A&M University

August 11-15, 2025

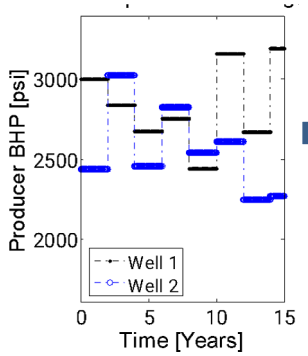
TEXAS A&M UNIVERSITY
Engineering

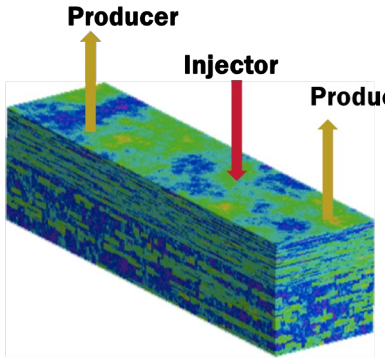
TEXAS A&M UNIVERSITY
Harold Vance Department of
Petroleum Engineering

Data-Driven Model Reduction (Classical Methods)

POD-Hybrid-based ROM

TEXAS A&M UNIVERSITY
Harold Vance Depa
Petroleum Enginee



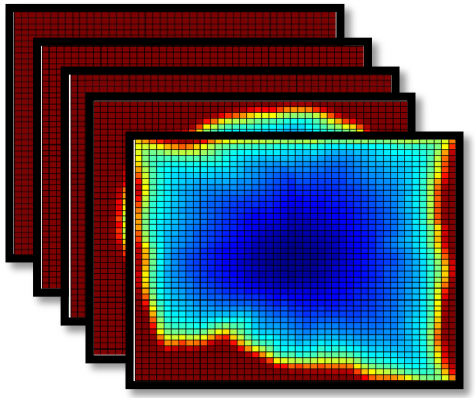


offline

$$\mathbf{R}\left(\mathbf{x}^{n+1},\mathbf{x}^{n},\mathbf{u}^{n+1}\right)=\mathbf{F}(\mathbf{x}^{n+1})+\mathbf{A}(\mathbf{x}^{n+1},\mathbf{x}^{n})+\mathbf{Q}(\mathbf{x}^{n+1},\mathbf{u}^{n+1})$$

$$\mathcal{J}=\frac{\partial \mathcal{R}}{\partial \mathbf{x}}$$

x→ state (pressures, saturations)



Form Φ

$$[U,S,V]=SVD(X)$$

$$\dot{x}=f(x)+g(x)u$$

$$r=\dot{x}_r-f(\Phi x_r)-g(\Phi x_r)u$$

$$\dot{x}_r=\Phi^T f(\Phi x_r)+\Phi^T g(\Phi x_r)u$$

Approximate

$$\mathbf{x}\approx\Phi\mathbf{x}_r$$

$$\Phi\in\mathbb{R}^{N\times r}$$

$$\Phi=\mathbf{U}[:,1:\mathbf{r}]$$

Project

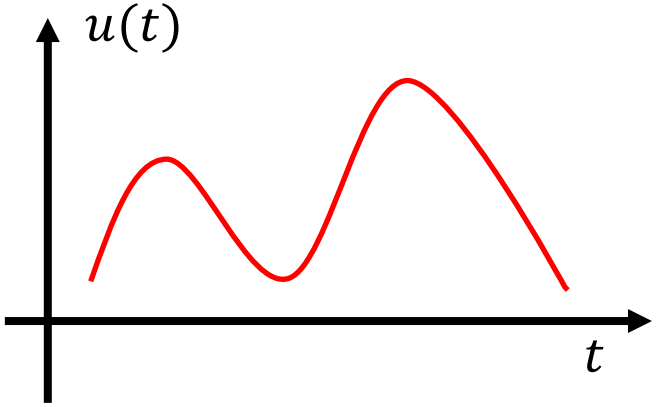
$$\Psi^{\mathrm{T}}\mathbf{r}=0$$

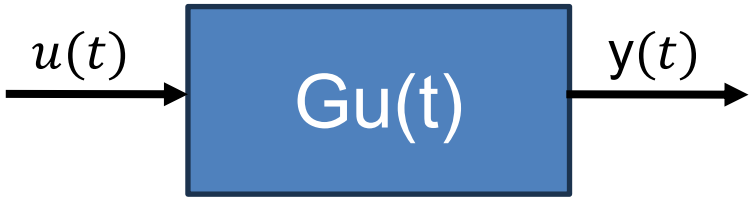
(Galerkin:$\Psi=\Phi$)

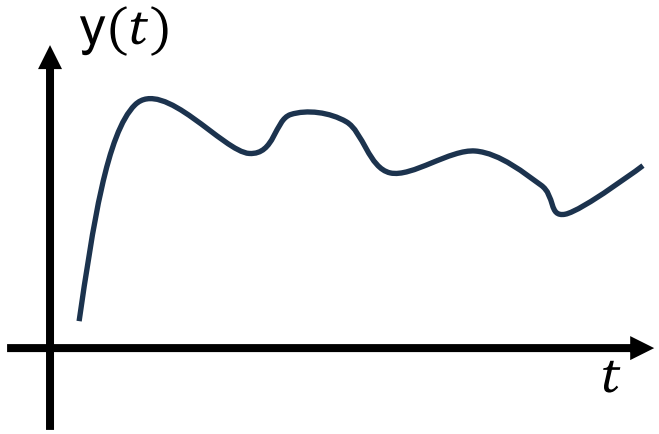
Question 1→ how to find proper projections?

Operator Inference – Dynamical Systems

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering







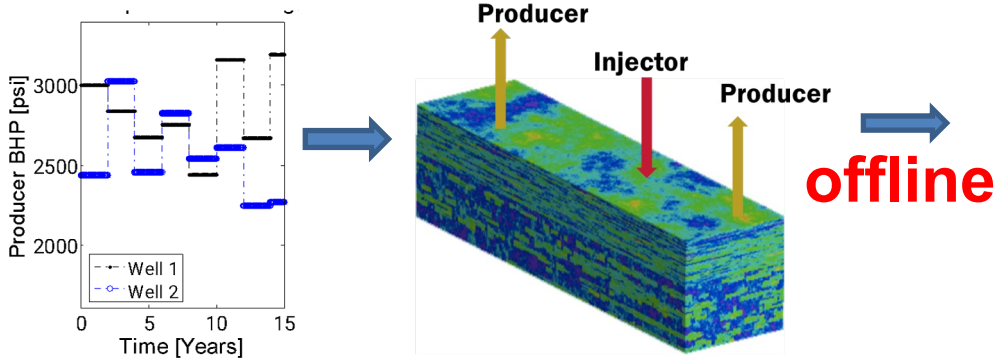
$$\mathbf{x}(t) = \mathbf{\Phi}(t, t_0)\mathbf{x}(t_0) + \int_{t_0}^{t} \mathbf{\Phi}(t, \tau)\mathbf{B}(\tau)\mathbf{u}(\tau)d\tau$$

$$Gu(t) = x(t|u(t)) = x(0) + \int_0^t f(Gu(\tau), u(\tau)d\tau$$

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \longrightarrow \quad G(s) = C(sI - A)^{-1}B + D$$

Operator Learning

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering



$$\mathbf{R}\left(\mathbf{x}^{n+1},\mathbf{x}^{n},\mathbf{u}^{n+1}\right)=\mathbf{F}(\mathbf{x}^{n+1})+\mathbf{A}(\mathbf{x}^{n+1},\mathbf{x}^{n})+\mathbf{Q}(\mathbf{x}^{n+1},\mathbf{u}^{n+1})$$

$$\mathcal{J}=\frac{\partial \mathcal{R}}{\partial \mathbf{x}}$$

x→ state (pressures, saturations)

Question 2 → can we directly identify the operator?

$$\begin{cases} z^{n+1} &= \mathbf{A}^{\mathrm{n}} z^{n}+\mathbf{B}^{\mathrm{n}} u^{n} \\ y^{n+1} &= \mathbf{C}^{\mathrm{n}} z^{n+1}+\mathbf{D}^{\mathrm{n}} u^{n} \end{cases}$$

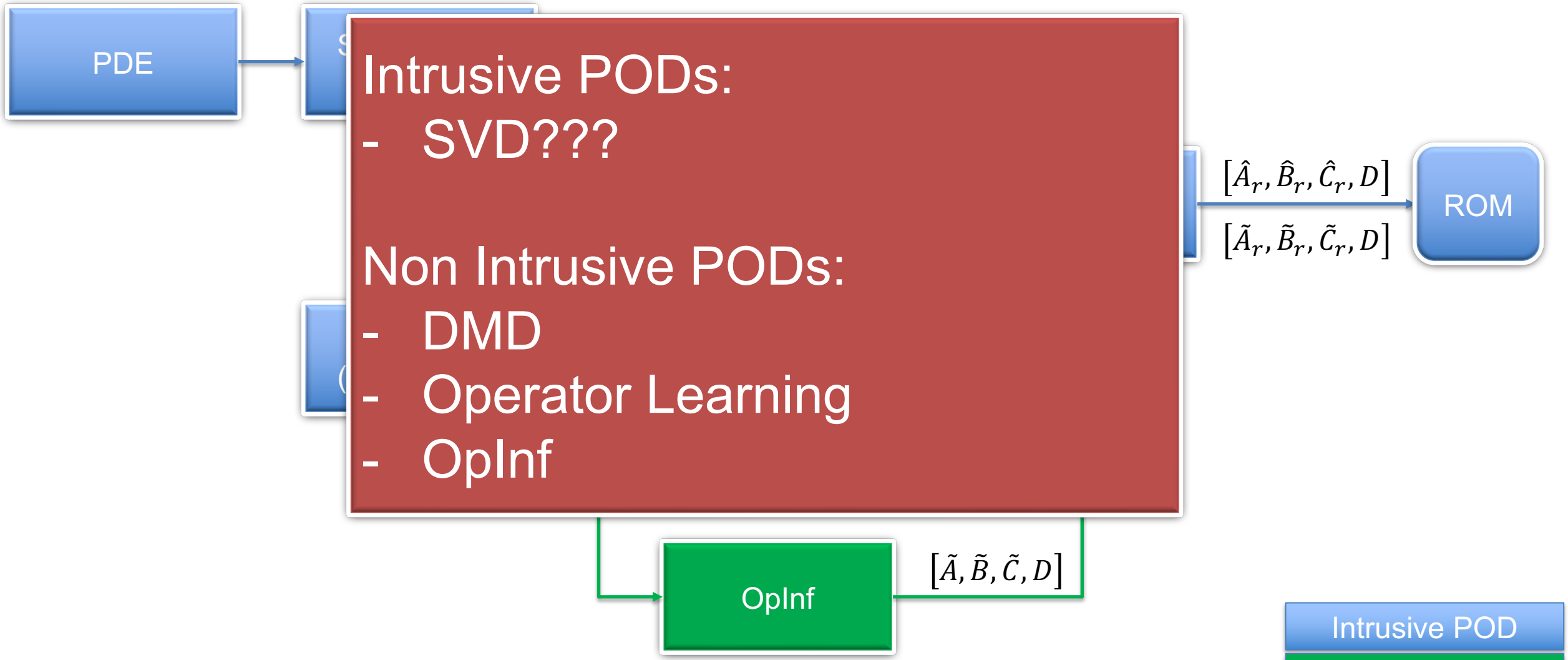
→ We aim to learn reduced models non-intrusively from training data *while maintaining some of the theoretical guarantees provided by intrusive methods.*

Solution →

- (bi)DMD
- p-DMD
- Lifting
- DMD for LPV
- Koopman
- E2CO
- FNO
- HNO

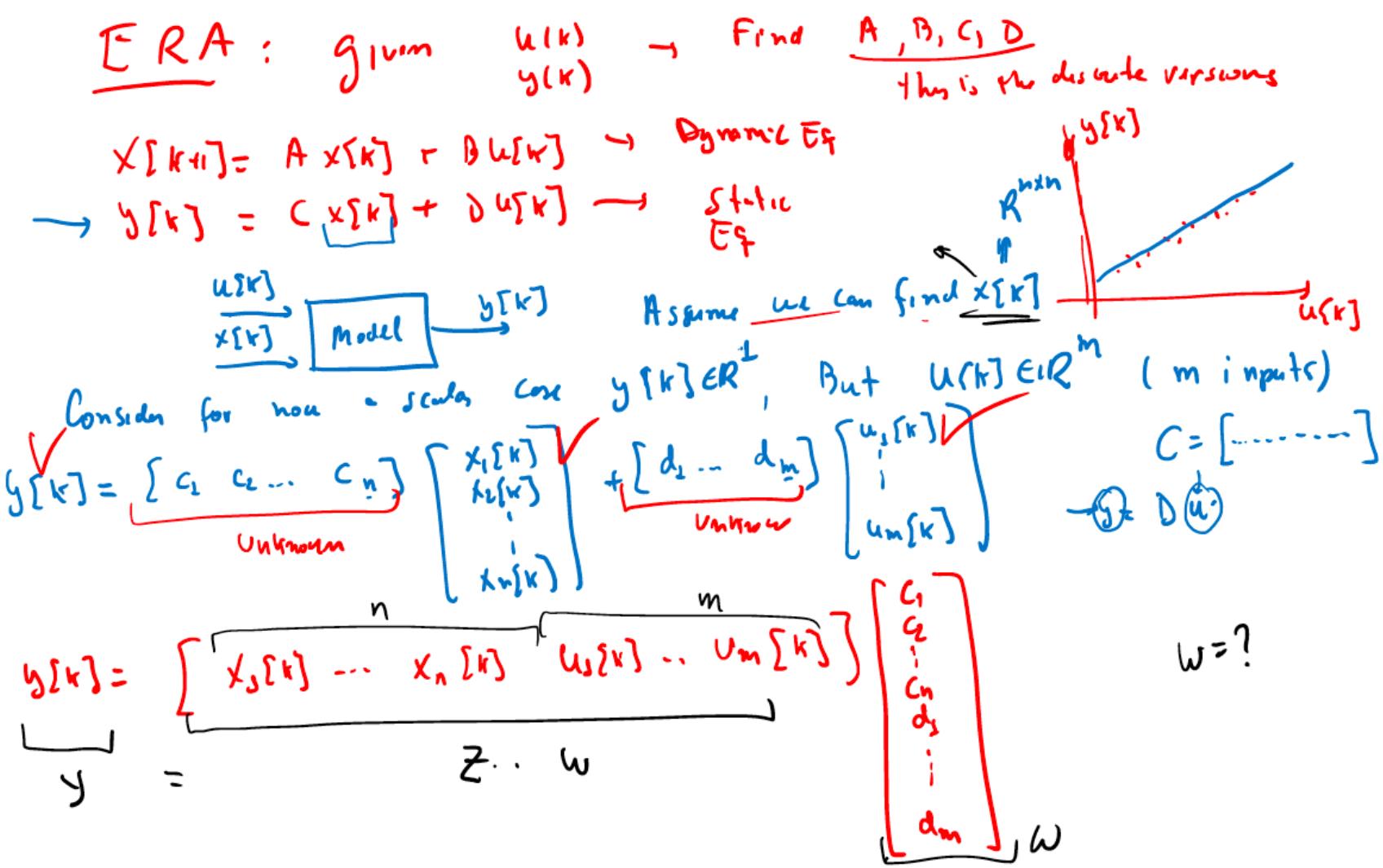
Motivation – Intrusive vs non-intrusive POD

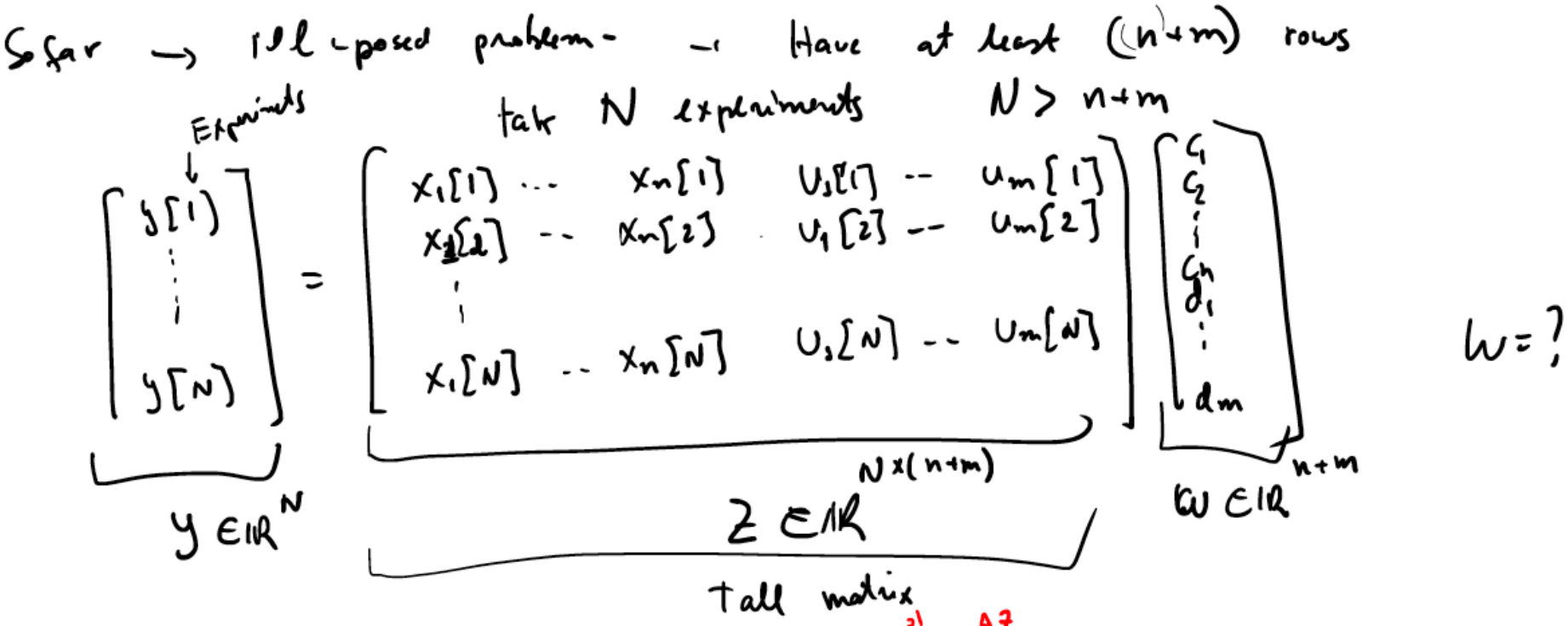
TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering



ERA – eigensystem realization

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering




So far → ill-posed problem- → Have at least (n+m) rows
take N experiments
N > n+m
Experiments
y[1]
y[N]
x1[1] ... xn[1] u1[1] -- um[1]
x1[2] -- xn[2] u1[2] -- um[2]
x1[N] -- xn[N] u1[N] -- um[N]
c1
c2
cn
d1
dm
w = ?
y ∈ IR^N
N x (n+m)
Z ∈ IR
tall matrix
w ∈ IR^(n+m)
Use Least Squares →
min || y - Z w ||_F^2
z' - Az
w* = optimal value
w* = [Z^T Z]^-1 Z^T y
Z^+ = pseudo inverse

How to Compute Pseudo-inverse without forming Z.

$U_r^T U_r = I$

$V_r^T V_r = I$

→ One can show → $Z_r \overset{svd}{=} U_r \Sigma_r V_r^T$

$$\boxed{Z^+ = V \Sigma^{-1} U^T}$$

Consider the dynamical system.

$$x[k+1] = A\,x[k] + B\,u[k]$$

Find A, B

Assume Can measure all states $x[k]$

take only one row:

$$x_i[k+1] = \begin{bmatrix} a_{i1} & \cdots & a_{in} \end{bmatrix} \begin{bmatrix} x_1[k] \\ x_2[k] \\ \vdots \\ x_n[k] \end{bmatrix} + \begin{bmatrix} b_{i1} & \cdots & b_{im} \end{bmatrix} \begin{bmatrix} u_1[k] \\ u_2[k] \\ \vdots \\ u_m[k] \end{bmatrix}$$

Concatenate:

$$X_i[k+1] = \begin{bmatrix} x^T[k] & u^T[k] \end{bmatrix} \begin{bmatrix} a_{i1} \\ \vdots \\ a_{in} \\ b_{i1} \\ \vdots \\ b_{im} \end{bmatrix}$$

$$\rightarrow y_i = Z_i w_j$$

$\longrightarrow$ Many i's to work with $\rightarrow$ independent.

$\Downarrow$

Same L.S.

$\Downarrow$

Find A, B, C, D

$$\hat{A} = T^{-1} A T$$
$$\hat{B} = T^{-1} B$$
$$\hat{C} = C T$$
$$\hat{D} = D$$

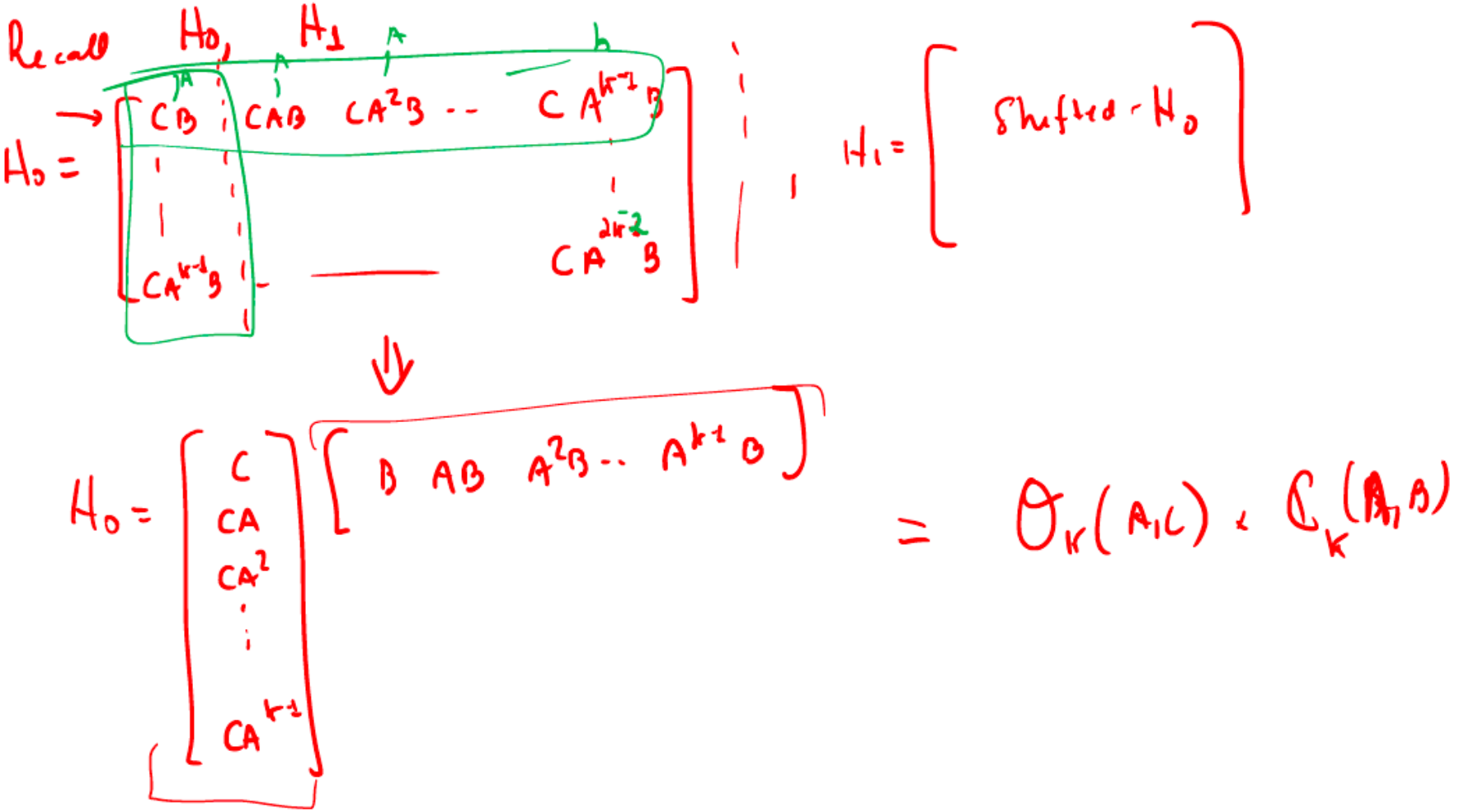
$$\dot{x}(t) = A x(t) + B u(t)$$
$$y(t) = C x(t) + D u(t)$$

$\rightarrow$

Find T, $\hat{T}$

$$\dot{\hat{x}} = \hat{T}^{-1} \hat{A} \hat{T} \hat{x} + \hat{T}^{-1} \hat{B} u$$
$$y = \hat{C} \hat{T} \hat{x} + \hat{D} u$$

$G(s) = C(sI - A)^{-1} B$


Recall
H_0
H_1
$H_0 =$
CB CAB CA^2B .. $CA^{k-1}B$
$CA^{k-1}B$ ——— $CA^{2k-2}B$
$H_1 =$ [Shifted - H_0]
$H_0 = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{k-1} \end{bmatrix} \begin{bmatrix} B & AB & A^2B & .. & A^{k-1}B \end{bmatrix}$
$= \mathcal{O}_k(A,C) \cdot \mathcal{C}_k(A,B)$

$$H_1 = \begin{bmatrix} CAB & CA^2B & \cdots & CA^kB \\ \vdots & & & \\ CA^kB & \text{---} & & CA^{2k-1}B \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{k-1} \end{bmatrix} A \begin{bmatrix} B & AB & \cdots & A^{k-1}B \end{bmatrix}$$

data

$\theta_k(A,C)$ A $C_k(A,B)$

Consequence: rank H_0 = order the minimal realization of the system.

$$\boxed{\text{rank } H_0 = r}$$

Fact:

$$\underbrace{\theta_k(A,C)}_{\text{x B}} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{k-1} \end{bmatrix} B = \begin{bmatrix} CB \\ CAB \\ CA^2B \\ \vdots \\ CA^{k-1}B \end{bmatrix} \Rightarrow \underline{H_0 e_1} = \begin{bmatrix} h[1] \\ h[2] \\ h[3] \\ \vdots \\ \end{bmatrix}$$

Similarly

$$C\,\mathcal{C}(A,B) = \begin{bmatrix} CB & CAB & \cdots & CA^{k-1}B \end{bmatrix} = \begin{bmatrix} h[1] & h[2] & \cdots & h[k] \end{bmatrix} \Rightarrow e_1^T H_0$$

take $H_b \rightarrow$ SVD

$$H_b = U \Sigma V^T = U_1 \Sigma_1 V_1^T$$

Sing H_b

$$\Sigma_1^{-1/2} U_1^T \theta_k(c,A)\, C_k(A,B)\, V_1 \Sigma_1^{-1/2} = \Sigma_1^{-1/2} U_1^T U_1 \Sigma_1 V_1^T V_1 \Sigma_1^{-1/2}$$

$$\Sigma_1^{-1/2} U_1^T \theta_k(A,c) \cdot C_k(A,B)\, V_1 \Sigma_1^{-1/2} = I_r$$

T^{-1} $\qquad$ T

$$T := C_k(A,B)\, V_1 \Sigma_1^{-1/2}$$

$$T^{-1} := \Sigma^{-1/2} U_1^T \theta_k(A,c)$$

take the transformation

$$\hat{A} = T^{-1} A T = \Sigma_1^{-1/2} U_1^T \underbrace{\theta_K(A,C)\, A\, C_K(A,B)}_{H_1} V_1 \Sigma_1^{-1/2}$$

$$\hat{A} = \Sigma_1^{-1/2} U_1^T H_1 V_1 \Sigma_1^{-1/2}$$ — only depends on data!

Similarly

$$\hat{B} = T^{-1} B = \Sigma_1^{1/2} V_1^T e_1$$

$$\hat{C} = CT = e_1^T U_1 \Sigma_1^{1/2}$$

ERA Proofs

Definition Impulse Response

$u(0) = 1$
$u(k) = 0 \quad k > 0$

If $u[k]$ is the impulse input, the output obtained is called impulse response $h(k)$, as is given by

$$h(k) = C A^{k-1} B \quad \text{for } k > 0$$

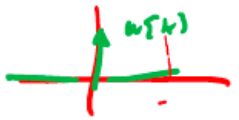
$$h(0) = 0$$

$$x[k+1] = A x[k] + B u[k]$$
$$y[k] = C x[k] + D u[k]$$

$x[0] = 0$ (0)
$x[1] = A x[0] + B u[0]$ (1)
$y[1] = CB \rightarrow h(1)$
$x[2] = A x[1] + B u[1]$ (B)
$y[2] = C x[1] + D u[1] = h(2)$
CAB

ERA Algorithm:

- given → Impulse Responses $y[k] = h(i)$ $i = 1, \ldots, 2k$
- Find $\hat{A}, \hat{B}, \hat{C}, \hat{D}$

$u[k] =$ 

$$x[k+1] = A x[k] + B u[k]$$
$$x[0] = 0$$
$$x[1] = B u[0]$$

1) Construct the Hankel matrices

$$H_0 = \begin{bmatrix} h(1) & h(2) & \cdots & h(k) \\ h(2) & h(3) & \cdots & h(k+1) \\ \vdots & & & \\ h(k) & \cdots & & h(2k-1) \end{bmatrix} \qquad H_1 = \begin{bmatrix} h(2) & h(3) & \cdots & h(k+1) \\ & & & \\ \vdots & & & \\ h(k+1) & \cdots & & h(2k) \end{bmatrix}$$

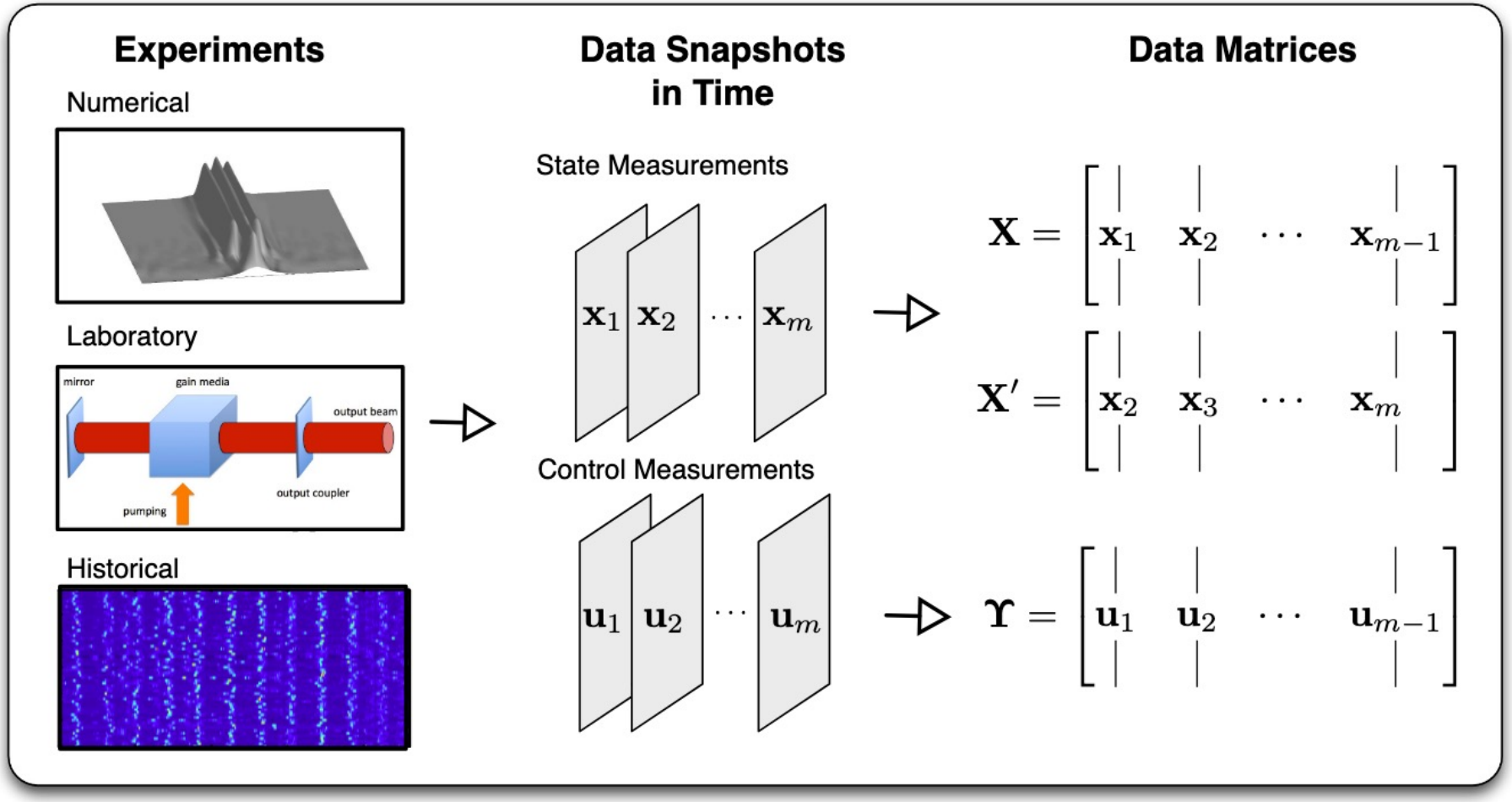
2) Compute SVD of $H_0 = [U_1 \vdots U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma_1 V_1^T$

3) Construct similarity transformation matrices

Reduced order matrices

$$\hat{A} = \Sigma_1^{-1/2} U_1^T H_1 V_1 \Sigma_1^{-1/2}$$
$$\hat{B} = \Sigma_1^{1/2} V_1^T e_1 \rightarrow \text{take the first column of } \Sigma_1^{1/2} V_1^T$$
$$\hat{C} = e_1^T U_1 \Sigma_1^{1/2} \rightarrow \text{take the first row}$$
$$\hat{D} = D$$

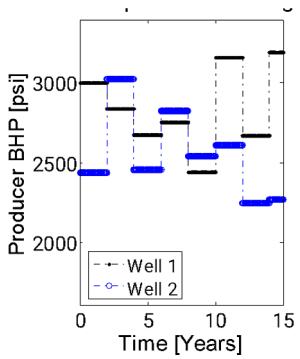
Data Collection

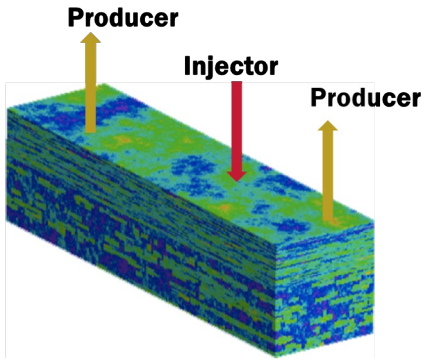


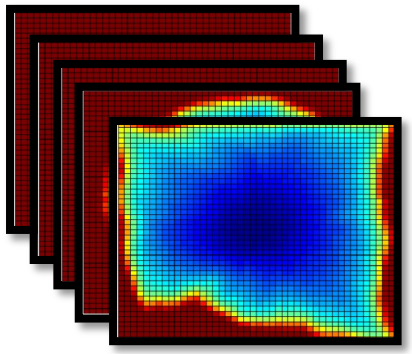
Dynamic Mode Decomposition (DMD)

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

$$\dot{x} = f(x(t), u(t), t) \longrightarrow \begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \longrightarrow \begin{cases} \dot{x}(t) = Ax(t) \\ x(0) = x_0 \end{cases}$$







$$X = \begin{bmatrix} x_1 & x_1 & \cdots & x_1 \\ x_2 & x_2 & \cdots & x_2 \\ x_3 & x_3 & \cdots & x_3 \\ \vdots & \vdots & \cdots & \vdots \\ x_n & x_n & \cdots & x_n \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ t_1 \end{matrix} \quad \begin{matrix} \\ \\ \\ \\ \\ t_2 \end{matrix} \quad \begin{matrix} \\ \\ \\ \\ \\ t_i \end{matrix} \quad \begin{matrix} \\ \\ \\ \\ \\ t_N \end{matrix}$$

$$\begin{cases} x_{k+1} = Ax_k \end{cases}$$

$$\boldsymbol{X^- = AX^+}$$

$$\boldsymbol{X^-} = [x^1 \quad x^2 \quad \cdots \quad x^{N-1}]$$

$$\boldsymbol{X^+} = [x^2 \quad x^3 \quad \cdots \quad x^N]$$

DMD "Procedure"

$$X^- = [x^1 \quad x^2 \quad \cdots \quad x^{N-1}] \qquad X^+ = [x^2 \quad x^3 \quad \cdots \quad x^N]$$

$$\{x_{k+1} = Ax_k \Rightarrow X^- = AX^+ \Rightarrow X^- \approx AX^+ \Rightarrow \min_{A \in \mathbb{R}^{n \times n}} ||X^+ - AX^-||_F$$

$$\downarrow$$

$$A = X^+(X^-)^\dagger \Leftarrow A \in \mathbb{R}^{n \times n} \text{ and DENSE}$$

DMD look for a reduced representation of A

$$X^- \approx U\Sigma V^T \Rightarrow A_r = U^T X^+ V \Sigma^{-1}$$

Model Reduction

Dynamic Mode Decomposition (DMD)

Find the dynamic properties of $\mathbf{A}$

$$\mathbf{X}' = \mathbf{A}\mathbf{X}$$

i. Find the truncated SVD of $\mathbf{X}$

$$\mathbf{X} \approx \underset{\tilde{\mathbf{U}}}{\begin{bmatrix} | & | & | \\ \mathbf{U}_1 & \mathbf{U}_2 & \mathbf{U}_r \\ | & | & | \end{bmatrix}} \underset{\tilde{\mathbf{\Sigma}}}{\begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_r \end{bmatrix}} \underset{\tilde{\mathbf{V}}^*}{\begin{bmatrix} — & \bar{\mathbf{V}}_1 & — \\ & \vdots & \\ — & \bar{\mathbf{V}}_r & — \end{bmatrix}}$$

ii. Compute reduced-order approximation $\tilde{\mathbf{A}}$

$$\tilde{\mathbf{A}} = \tilde{\mathbf{U}}\mathbf{X}'\tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}$$

iii. Investigate the dynamic properties of $\tilde{\mathbf{A}}$

$$\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\mathbf{\Lambda}$$

iv. Solve for the dynamic modes of $\mathbf{A}$

$$\mathbf{\Phi} = \mathbf{X}'\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{W}$$

DMD with Control (DMDc)

Find the dynamic properties of $\mathbf{A}$ and $\mathbf{B}$

$$\mathbf{X}' = \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{\Upsilon}$$

i. Construct the input data matrix

$$\mathbf{\Omega} = \begin{bmatrix} \mathbf{X} \\ \mathbf{\Upsilon} \end{bmatrix}$$

ii. Find the truncated SVD of input matrix $\mathbf{\Omega}$

$$\mathbf{\Omega} \approx \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^* = \begin{bmatrix} \tilde{\mathbf{U}}_1 \\ \tilde{\mathbf{U}}_2 \end{bmatrix} \tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^*$$

iii. Find the truncated SVD of output matrix $\mathbf{X}'$

$$\mathbf{X}' \approx \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^*$$

iv. Compute reduced-order approximation of $\mathbf{A}$

$$\tilde{\mathbf{A}} = \hat{\mathbf{U}}^*\mathbf{X}'\tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{U}}_1^*\hat{\mathbf{U}}$$

v. Investigate the dynamic properties of

$$\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\mathbf{\Lambda}$$

vi. Solve for the dynamic modes of

$$\mathbf{\Phi} = \mathbf{X}'\mathbf{V}\mathbf{\Sigma}^{-1}\tilde{\mathbf{U}}_1^*\hat{\mathbf{U}}\mathbf{W}$$

Algorithm: DMD

1) Given $\dot{x} = f(t, x, u)$ → non linear function (Full order Simulator)
2) Perform simulations to get state-snapshots (N snapshots) → collect data
3) Form these matrices

$$Z = [x_1 \; x_2 \; \cdots \; x_{N-1}]$$

$$Z' = [x_2 \; \cdots \; x_N]$$

4) Compute low rank approx of Z (SVD of Z) → r rank

$$Z_r = U_r \Sigma_r V_r^*$$

$$Z' = A \, U_r \Sigma_r V_r^*$$

5) Perform POD on A

$$\tilde{A} = U_r^* A U_r \longrightarrow \boxed{\tilde{A} = U^* Z' V \Sigma^{-1}}$$

6) Compute $eig(\tilde{A})$ → eigenvalues of (A)

$$\tilde{A}\,W = W\,\Lambda$$

(W: eigenvectors; Λ: eigenvals)

7) Form Projecton (DMD)

$$\Phi = Z' V \Sigma^{-1} W$$

→ eigenvectors of A

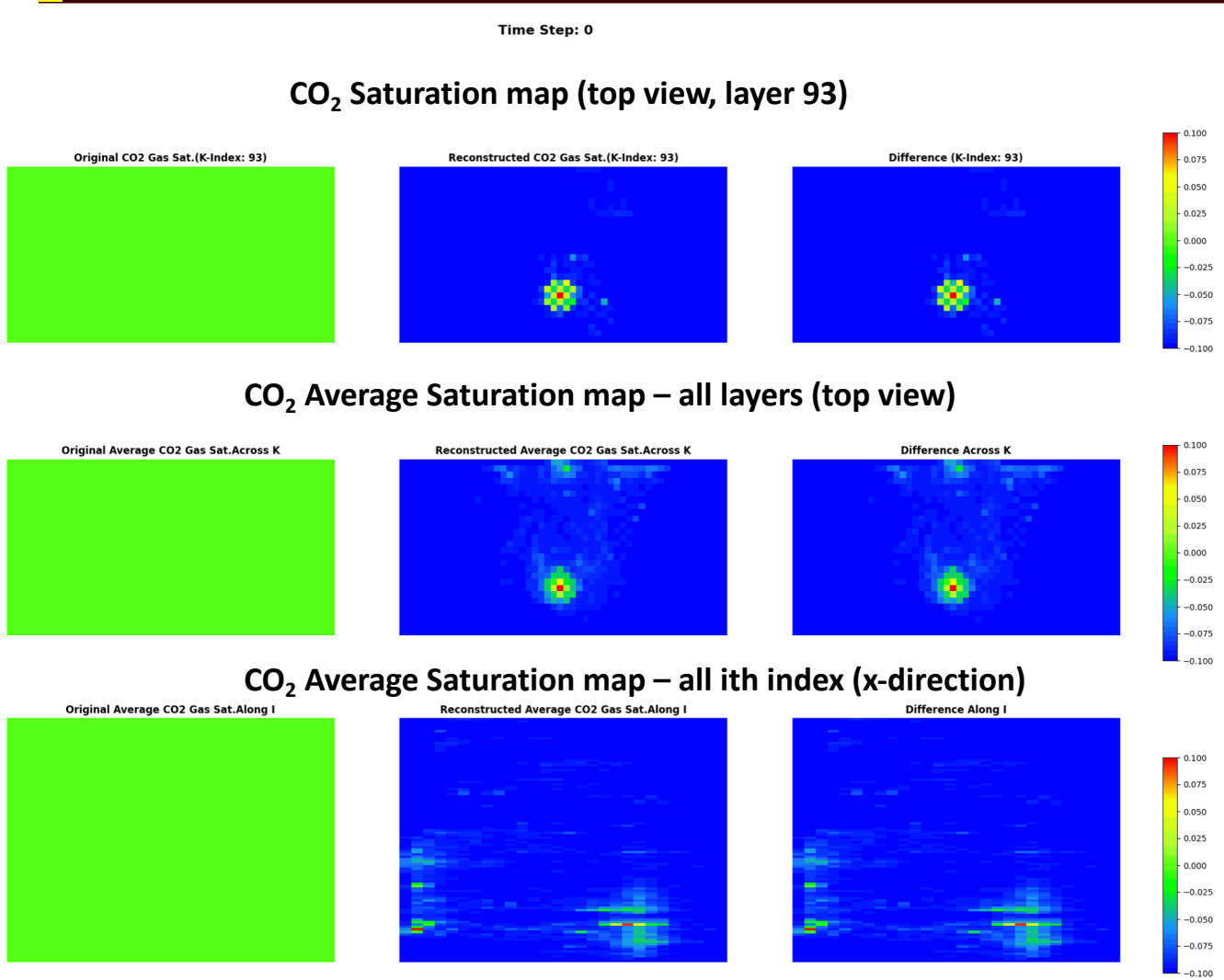
8) Evolve in time:

$$\tilde{x}(k\Delta t) = \Phi\,\Lambda^{t}\,b_0$$

(b_0: I.C)

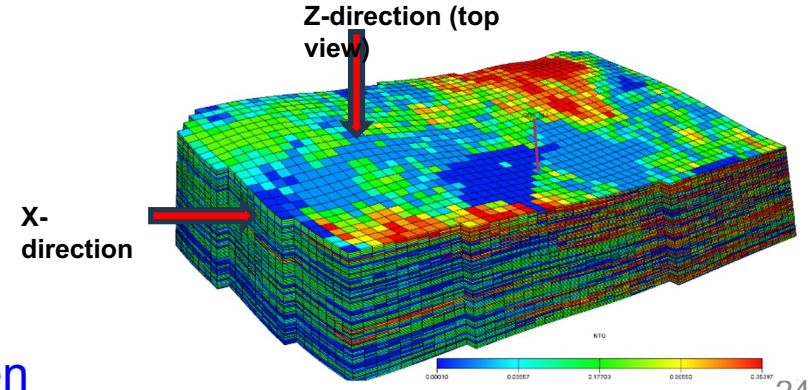
Sparse DMD Method for CCS

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering



Results : CO_2 plume evolution

	Numerical model	Sp-DMD
Run time (seconds)	6000 (approx. 2hrs)	123 secs



[Rapid Geological CO2 Storage Forecast and Optimization: A Data-driven Dynamic Mode Decomposition Model Order Reduction Approach](#)
D Voulanas, E Gildin - arXiv preprint arXiv:2407.20541, 2024

Why not only DMD?

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

DMD

- *best-fits linear operators to state trajectories*
- *Methods based on Koopman operators have been developed to extend DMD to nonlinear*
- *challenge is selecting observables such that the dynamics become close to linear.*

OpInf

- *Operator Inference fits operators of reduced models to data;*
- *however, it allows for nonlinear terms and thus can capture nonlinear dynamics.*
- *Operator Inference explicitly embeds the underlying physics through the structured form of the reduced model it learns.*

Data-driven operator inference for nonintrusive projection-based model reduction
B. Peherstorfer and K. Willcox, Computer Methods in Applied Mechanics and Engineering, 2016

OpInf – Non-linear systems

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

Step 1 : Data collection + Lift to higher dimensions

$$X = \begin{matrix} t_1 & t_2 & t_i & t_N \end{matrix} \begin{bmatrix} x_1 & x_1 & \cdots & x_1 \\ x_2 & x_2 & \cdots & x_2 \\ x_3 & x_3 & \cdots & x_3 \\ \vdots & \vdots & \cdots & \vdots \\ x_n & x_n & \cdots & x_n \end{bmatrix}$$

$$\dot{x}(t) = ax(t) + e^{-x(t)}.$$

$$\text{Set } x_1 = x \text{ and take } x_2 = e^{-x_1}$$

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} a & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -a & -1 \end{bmatrix} \left(\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \otimes \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \right)$$

$$\boldsymbol{f}(\boldsymbol{x}(t;\boldsymbol{\mu}), \boldsymbol{u}(t);\boldsymbol{\mu}) = \boldsymbol{A}_1(\boldsymbol{\mu})\boldsymbol{x}(t;\boldsymbol{\mu}) + \boldsymbol{A}_2\boldsymbol{x}^2(t;\boldsymbol{\mu}) + \boldsymbol{B}(\boldsymbol{\mu})\boldsymbol{u}(t)\,.$$

Learning nonlinear reduced models from data with operator inference
B. Kramer, B. Peherstorfer, and K. Willcox Annual Review of Fluid Mechanics, 2024

OpInf – Non-linear systems

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

Step 2 : Construct low-dimensional (latent space)

$$[U, S, V] = SVD(X) \xrightarrow[\text{POD}]{\text{Projection}} x \approx V\hat{x}$$

Step 3 : reduced model construction by OpInf

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t;\boldsymbol{\mu}) = \boldsymbol{f}(\boldsymbol{x}(t;\boldsymbol{\mu}), \boldsymbol{u}(t);\boldsymbol{\mu}) = \sum_{i=1}^{\ell} \boldsymbol{A}_i(\boldsymbol{\mu})\boldsymbol{x}^i(t;\boldsymbol{\mu}) + \boldsymbol{B}(\boldsymbol{\mu})\boldsymbol{u}(t),$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\boldsymbol{x}}(t;\boldsymbol{\mu}) = \sum_{i=1}^{\ell} \hat{\boldsymbol{A}}_i(\boldsymbol{\mu})\hat{\boldsymbol{x}}^i(t;\boldsymbol{\mu}) + \hat{\boldsymbol{B}}(\boldsymbol{\mu})\boldsymbol{u}(t),$$

Least square solutions

$$J_j(\hat{\boldsymbol{A}}_1(\boldsymbol{\mu}_j),\ldots,\hat{\boldsymbol{A}}_\ell(\boldsymbol{\mu}_j),\hat{\boldsymbol{B}}(\boldsymbol{\mu}_j)) = \sum_{k=1}^{K}\left\|\sum_{i=1}^{\ell}\hat{\boldsymbol{A}}_i(\boldsymbol{\mu}_j)\breve{\boldsymbol{x}}^i(t_k;\boldsymbol{\mu}_j)+\hat{\boldsymbol{B}}(\boldsymbol{\mu}_j)\boldsymbol{u}(t_k;\boldsymbol{\mu}_j)-\breve{\boldsymbol{x}}'(t_k;\boldsymbol{\mu}_j)\right\|_2^2,$$

For Linear Systems (first order)

$$\dot{\hat{\mathbf{X}}}_r = \hat{\mathbf{A}}_r\hat{\mathbf{X}}_r + \hat{\mathbf{B}}_r\mathbf{U} \Longrightarrow \dot{\hat{\mathbf{X}}}_r = \boldsymbol{O}\boldsymbol{\Gamma}$$

$$\boldsymbol{\Gamma}_{r+P\times K} = \begin{bmatrix} \hat{\mathbf{X}}_r^T & \mathbf{U}^T \end{bmatrix}^T$$

$$\mathbf{O}_{r\times r+P} = \begin{bmatrix} \hat{\mathbf{A}}_r & \hat{\mathbf{B}}_r \end{bmatrix}$$

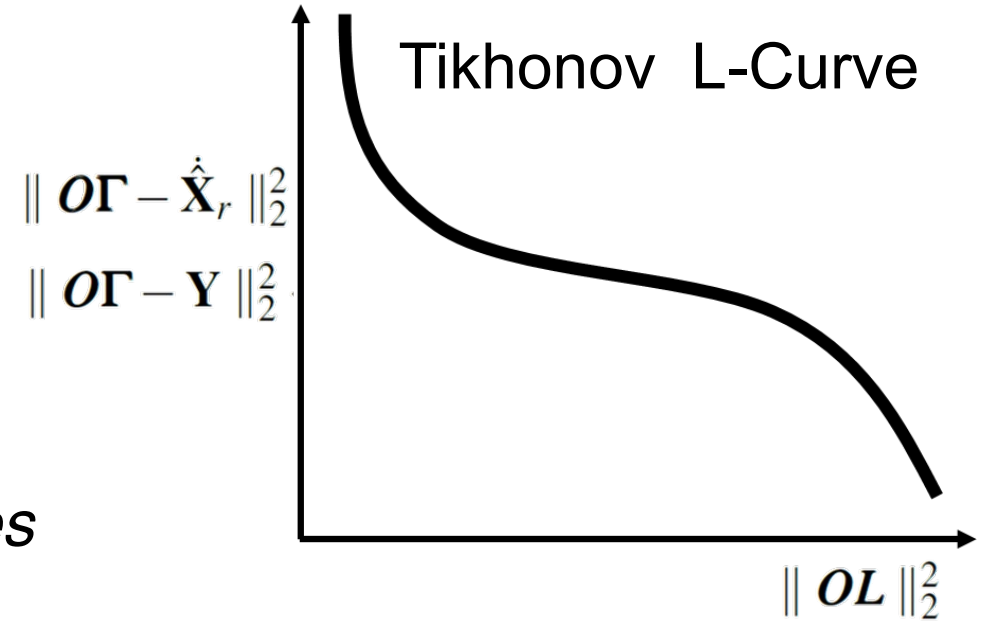
$$\hat{\boldsymbol{O}} = \underset{\boldsymbol{O}}{arg\,min} \parallel \dot{\hat{\mathbf{X}}}_r - \boldsymbol{O}\boldsymbol{\Gamma} \parallel^2$$

Operator Inference - OpInf

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

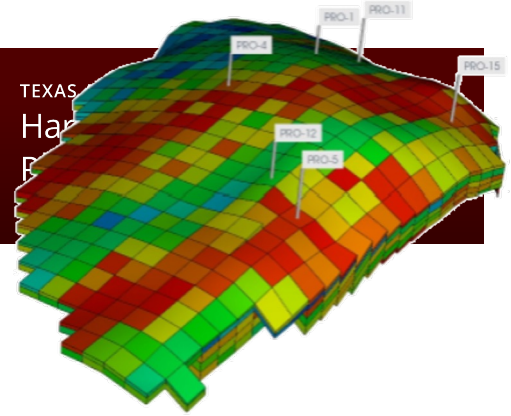
Tikhonov regularization has been proposed to enhance Operator Inference by preventing overfitting

$$\hat{\boldsymbol{O}} = \min_{\boldsymbol{O}} \left[\| \boldsymbol{O}\boldsymbol{\Gamma} - \dot{\hat{\mathbf{X}}}_r \|_2^2 + \lambda \| \boldsymbol{O}\boldsymbol{L} \|_2^2 \right]$$



Regularization imposes a bias on the learning process to guide Operator Inference toward meaningful models that have predictive capabilities and that generalize well to unseen parameters, inputs, and initial conditions.

OpInf – Example 1

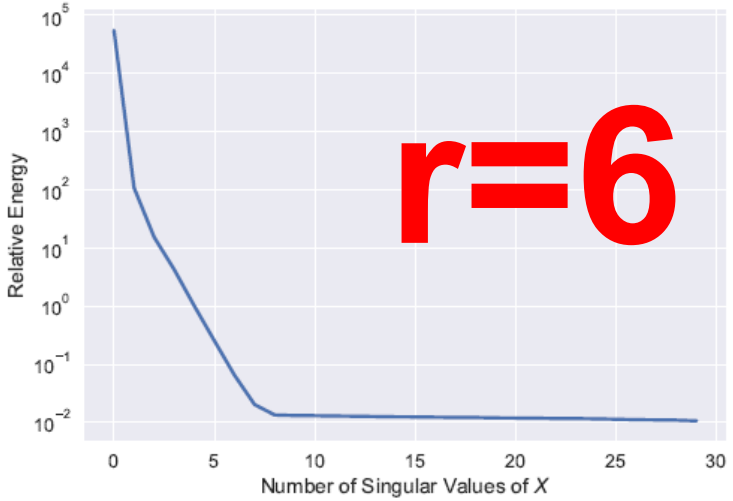


PUNQ MODEL (Single-Phase)

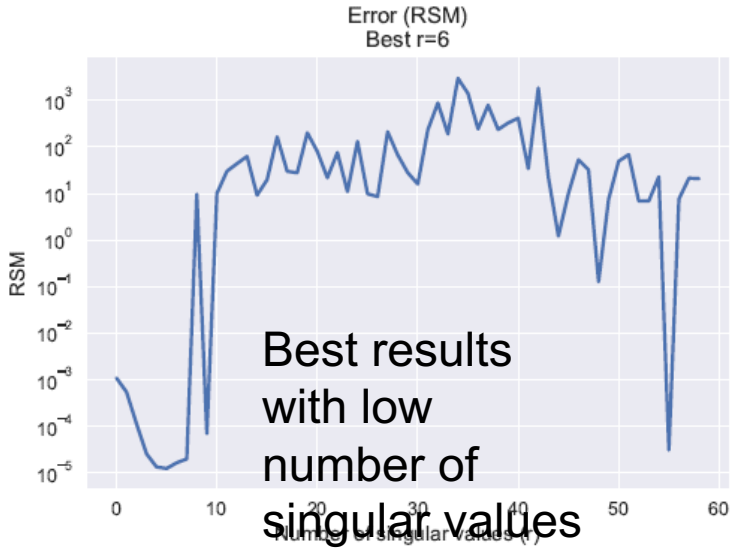
19 × 28 × 5
= 2660 cells

$$\mathbf{X}_{N\times K} = \begin{bmatrix} p_1^{(1)} & p_1^{(2)} & p_1^{(3)} & \cdots & p_1^{(K)} \\ p_2^{(1)} & p_2^{(2)} & p_2^{(3)} & \cdots & p_2^{(K)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_N^{(1)} & p_N^{(2)} & p_N^{(3)} & \cdots & p_N^{(K)} \end{bmatrix}$$

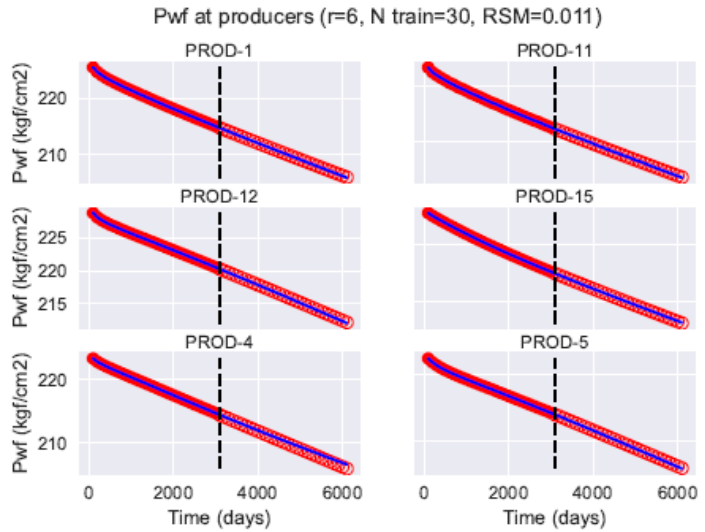
$$\boldsymbol{Y}(t) = \begin{bmatrix} \boldsymbol{p}_{wf}^{(1)} & \boldsymbol{p}_{wf}^{(2)} & \boldsymbol{p}_{wf}^{(3)} & \cdots & \boldsymbol{p}_{wf}^{(K)} \\ \boldsymbol{BSW}^{(1)} & \boldsymbol{BSW}^{(2)} & \boldsymbol{BSW}^{(3)} & \cdots & \boldsymbol{BSW}^{(K)} \end{bmatrix}$$



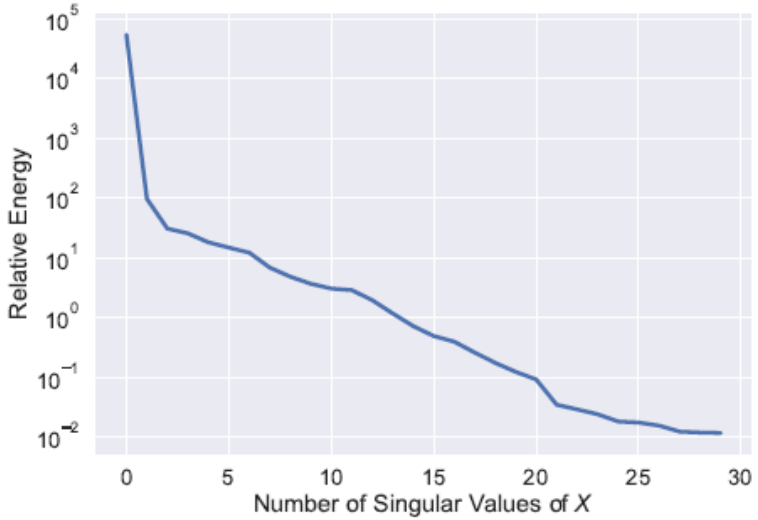
r=6

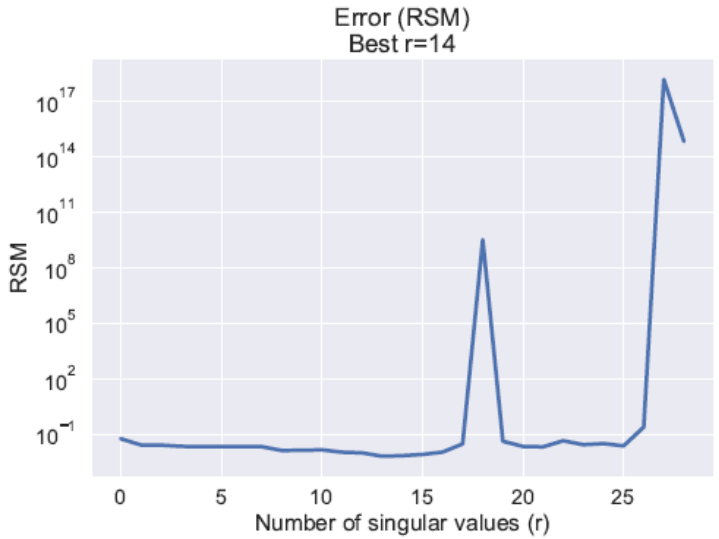


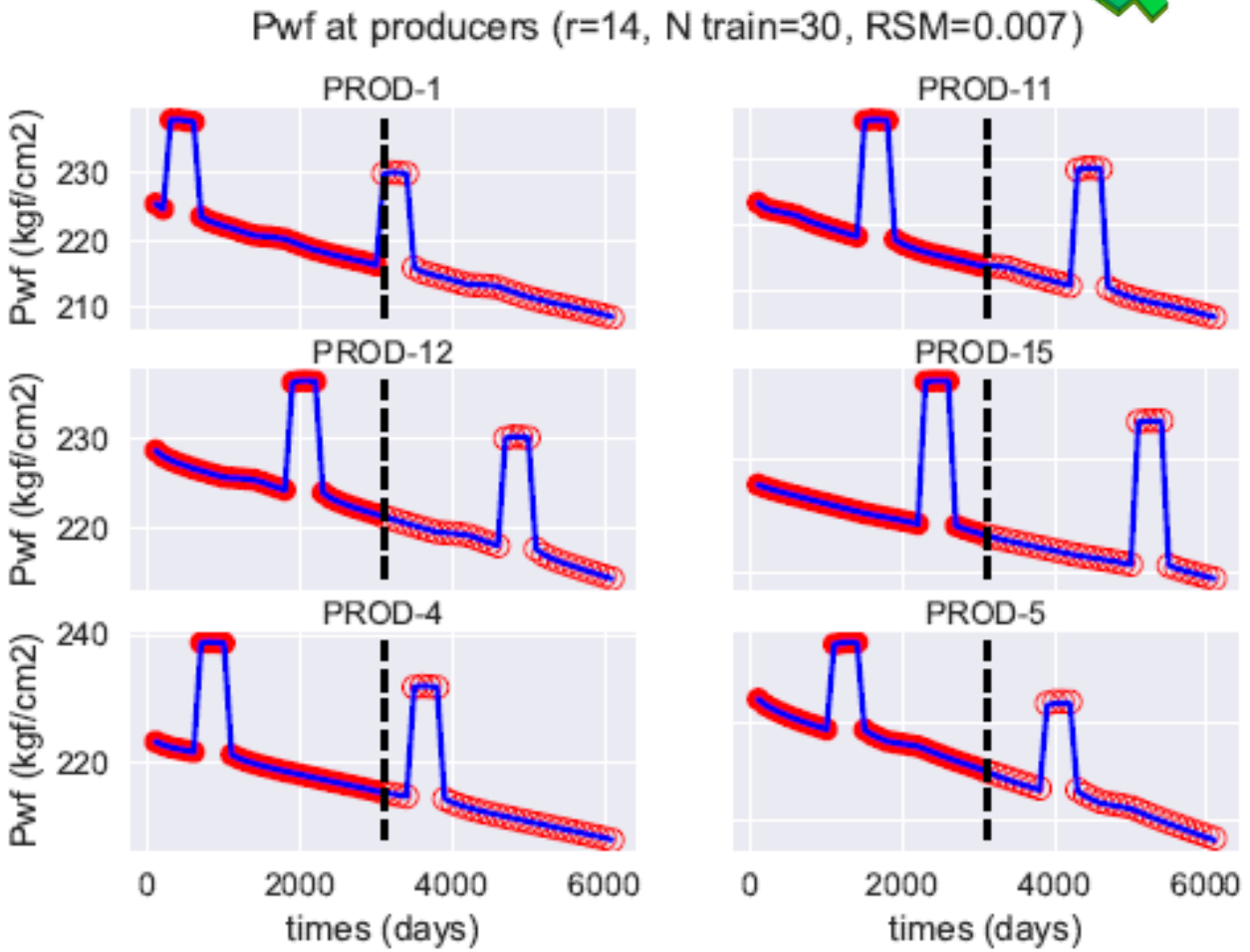
Best results with low number of singular values



OpInf – Example 1



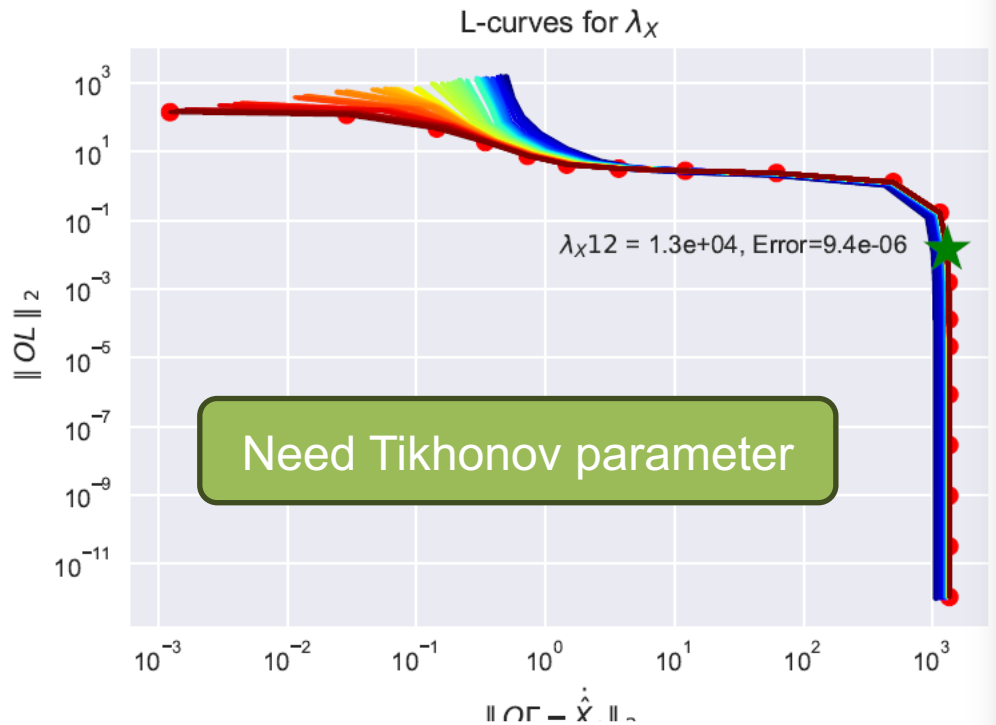


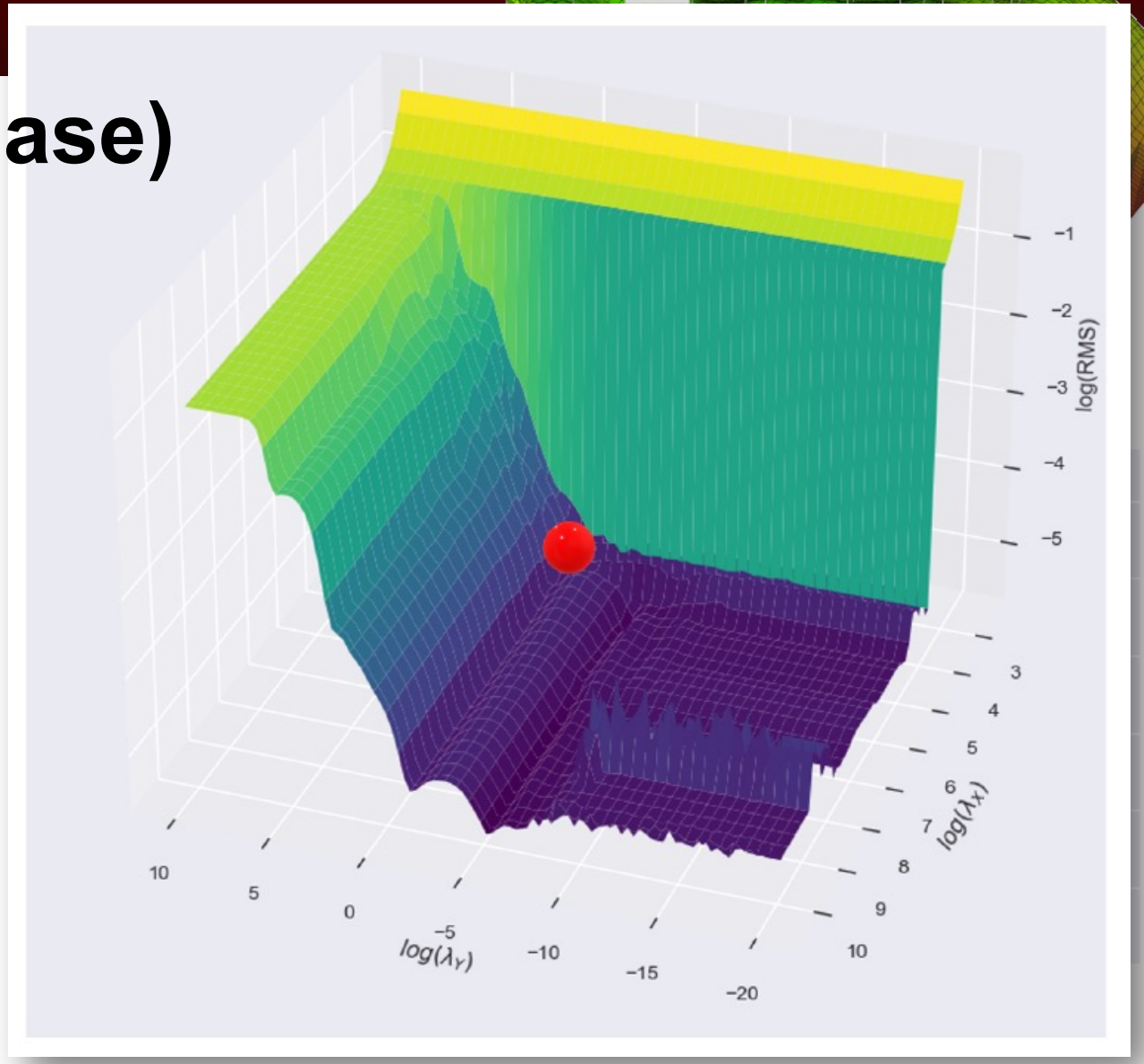


Example 2

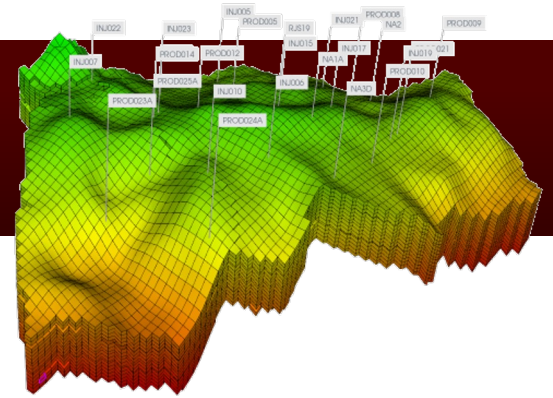
UNISIM –I –M (Two-Phase)

81 × 58 × 20 = 93, 960 cells

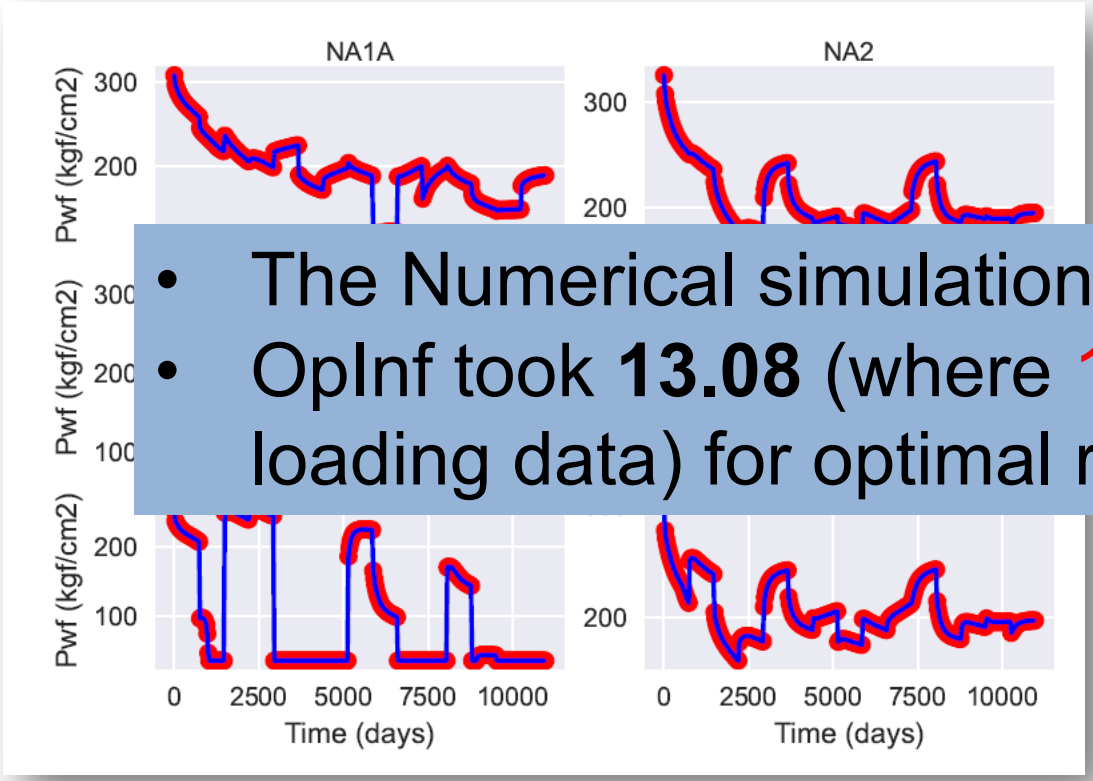


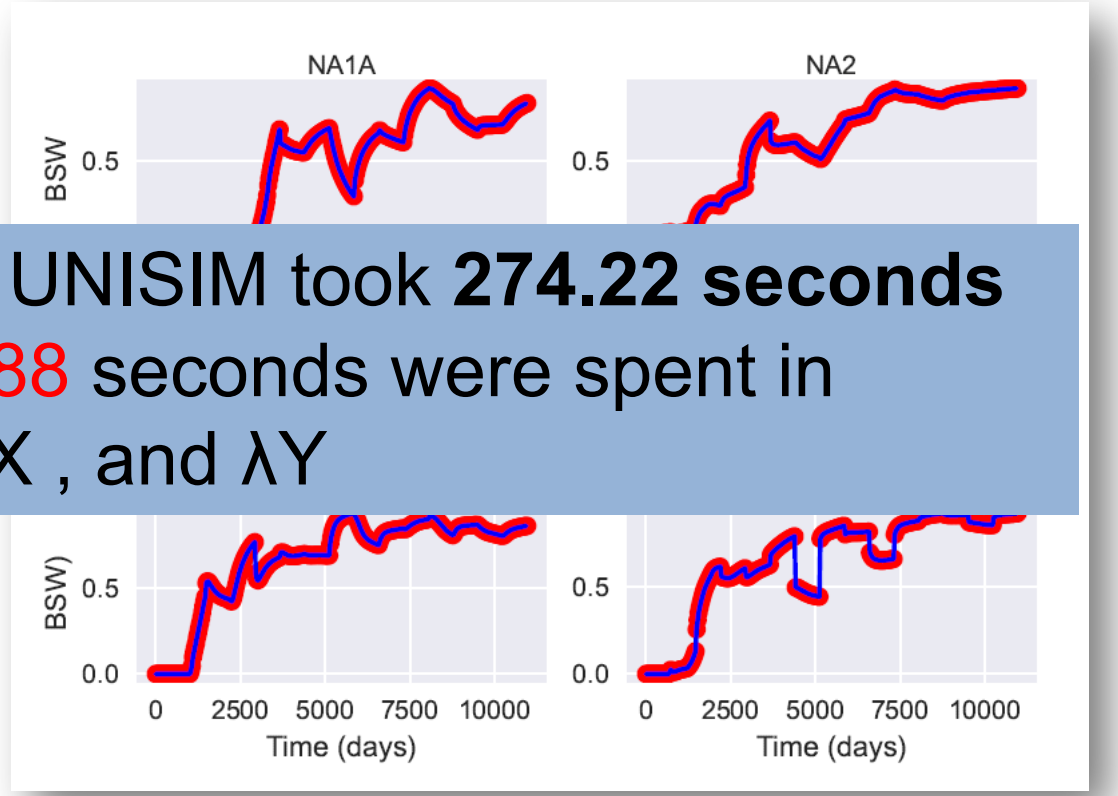


Two Phase Model: Unisim



r=25





- The Numerical simulation of UNISIM took **274.22 seconds**
- OpInf took **13.08** (where 10.88 seconds were spent in loading data) for optimal r, λX , and λY

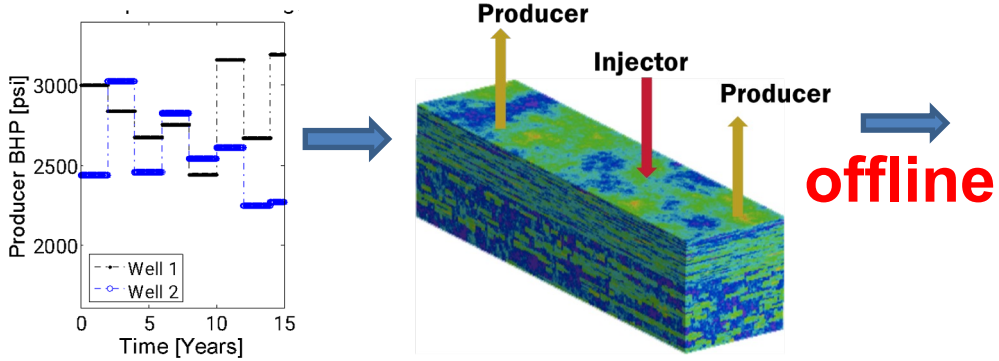
TEXAS A&M UNIVERSITY
Engineering

TEXAS A&M UNIVERSITY
Harold Vance Department of
Petroleum Engineering

Data-Driven MOR (Based on Deep Learning)

Problem Setting - Operator Learning

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering



$$\mathbf{R}\left(\mathbf{x}^{n+1},\mathbf{x}^{n},\mathbf{u}^{n+1}\right)=\mathbf{F}(\mathbf{x}^{n+1})+\mathbf{A}(\mathbf{x}^{n+1},\mathbf{x}^{n})+\mathbf{Q}(\mathbf{x}^{n+1},\mathbf{u}^{n+1})$$

$$\mathcal{J}=\frac{\partial \mathcal{R}}{\partial \mathbf{x}}$$

x→ state (pressures, saturations)

Question → can we directly identify the Operator (or dynamical system)?

$$F:\mathcal{A}\times\Theta\rightarrow u$$

$$\begin{cases} z^{n+1} &= \mathbf{A}^{\mathrm{n}}z^{n}+\mathbf{B}^{\mathrm{n}}u^{n} \\ y^{n+1} &= \mathbf{C}^{\mathrm{n}}z^{n+1}+\mathbf{D}^{\mathrm{n}}u^{n} \end{cases}$$

→ We aim to learn reduced models non-intrusively from training data *while maintaining some of the theoretical guarantees provided by intrusive methods.*

Solution →

(bi)DMD
p-DMD
Lifting

DMD for LPV

Koopman

E2CO

FNO
HNO

Reservoir Simulation and Model Reduction

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

Residual form of the reservoir simulator nonlinear system is (after assumptions and fully implicit discretization):

$$g(x^{n+1}, u^{n+1}) = \mathbf{F}(x^{n+1}) + \mathbf{Acc}(x^{n+1}, x^n) + \mathbf{Q}(x^{n+1}, u^{n+1}) = 0$$

solved by Newton's method, Jacobian matrix: $\mathbf{J} = \frac{\partial g}{\partial x}$

Can be written as a (reduced) linearized system → See POD, POD-TPWL, POD-DEIM, POD-DL, etc

MOR → $\boldsymbol{z = \Phi x}$ →

$$\begin{cases} z^{n+1} = \mathbf{A}^{\mathrm{n}} z^n + \mathbf{B}^{\mathrm{n}} u^n \\ y^{n+1} = \mathbf{C}^{\mathrm{n}} z^{n+1} + \mathbf{D}^{\mathrm{n}} u^n \end{cases}$$

Hypothesis

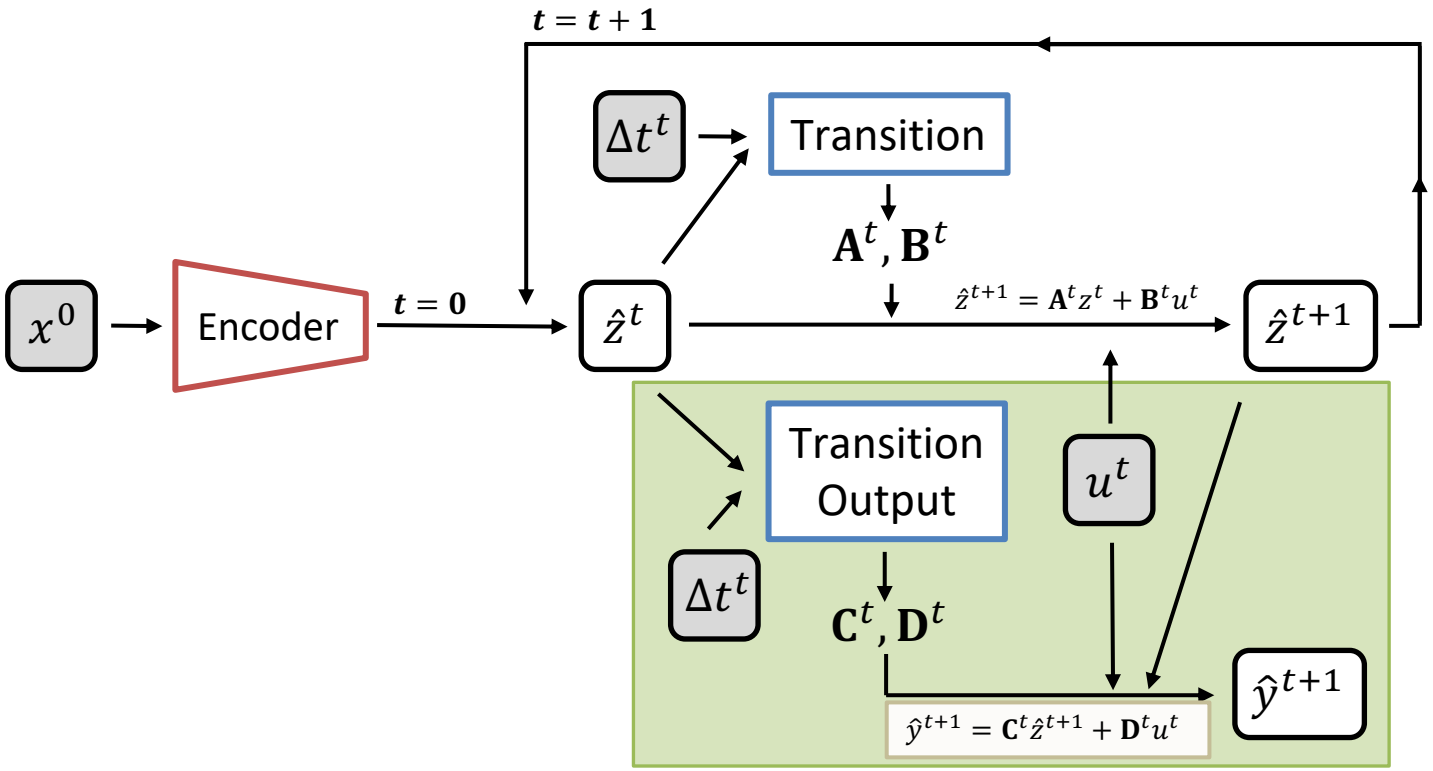
It is possible to estimate matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and $\mathbf{D}$ based on the dynamical evolution of the states?

(c.f., DMD/DMDio)

E2CO – Embed to Control and Observe*

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

For reservoir simulation we are interested on the **model outputs** (e.g. well flow rates), not only on the states



Manuel Watter, Jost Tobias Springenberg, Joschka Boedecker and Martin Riedmiller. , Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images. 2015. www.arxiv.org/abs/1506.07365.

Z.L. Jin, Y. Liu and L.J. Durlofsky, Deep-learning-based surrogate model for reservoir simulation with time-varying well controls. Journal of Petroleum Science and Engineering (2020), doi: https://doi.org/10.1016/j.petrol.2020.107273. and

Do not account for output

$$\begin{cases} z^{n+1} = \mathbf{A}^n z^n + \mathbf{B}^n u^n \\ y^{n+1} = \mathbf{C}^n z^{n+1} + \mathbf{D}^n u^n \end{cases}$$

E2CO New architecture with Physical loss functions

* Coutinho EJR, Dall'Aqua M and Gildin E (2021) Physics-Aware Deep-Learning-Based Proxy Reservoir Simulation Model Equipped With State and Well Output Prediction. Front. Appl. Math. Stat. 7:651178. doi: 10.3389/fams.2021.651178

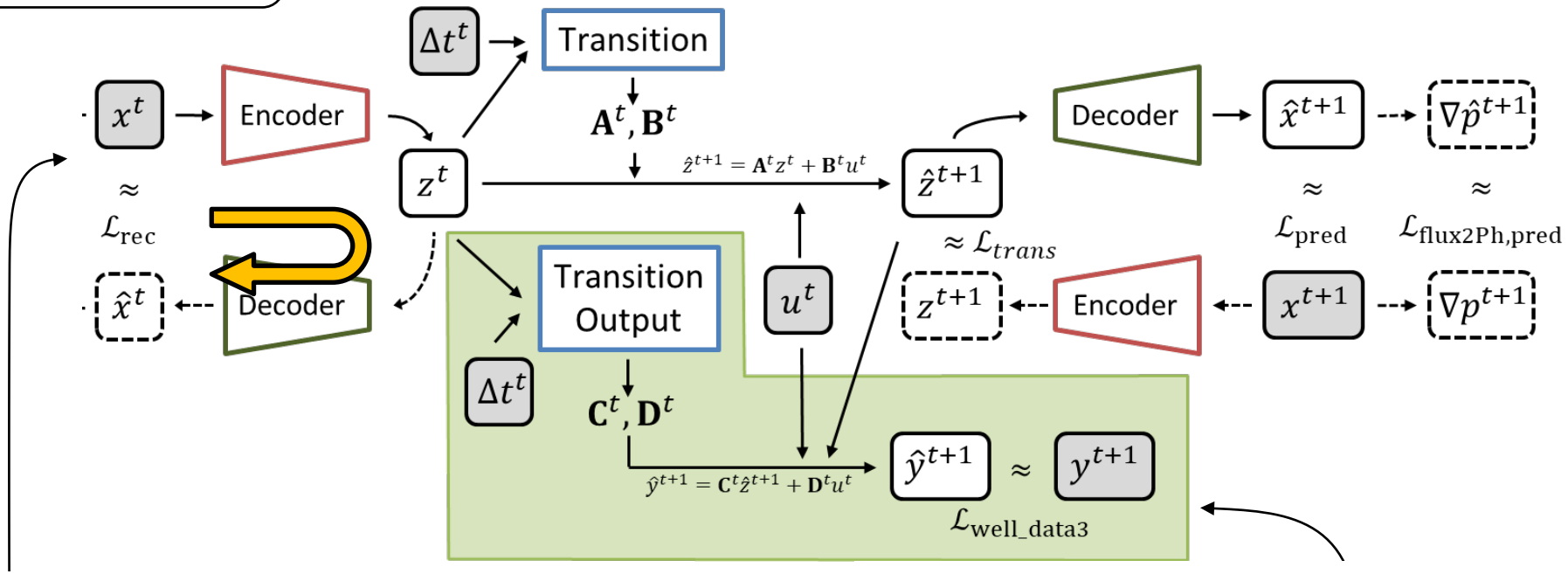
E2CO – How to Train?

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

Loss Function

$$\mathcal{L}_{\text{rec}} = \|x^t - \hat{x}^t\|$$

$$(\mathcal{L}_{\text{flux2Ph}})_i = (\mathcal{L}_{\text{flux2Ph,rec}})_i + (\mathcal{L}_{\text{flux2Ph,pred}})_i,$$



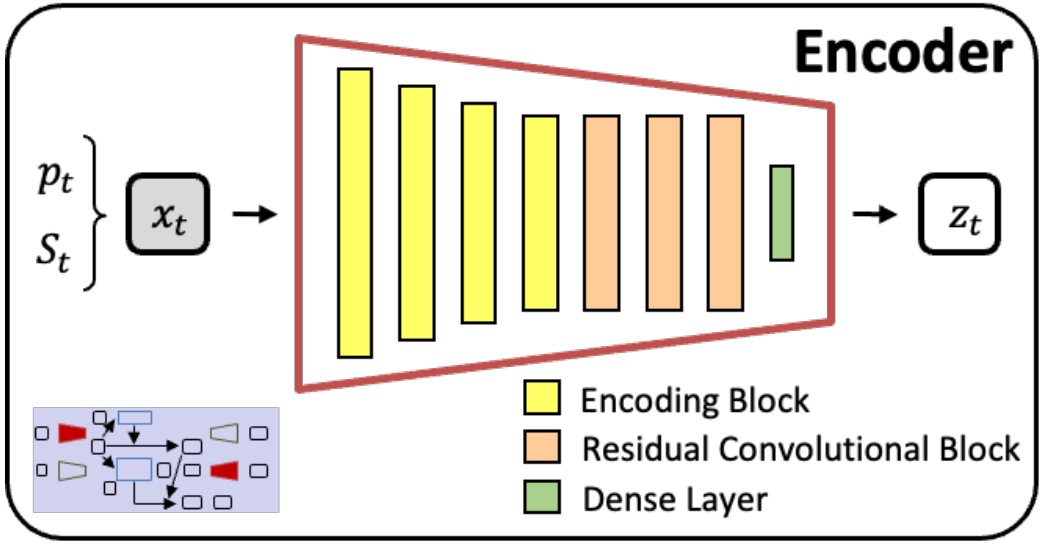
$$(\mathcal{L}_{\text{flux2Ph,rec}})_i = k\left\{\left\|k_{r,o}(S_w^t)\nabla p^t - k_{r,o}(\hat{S}_w^t)\nabla\hat{p}^t\right\|_2^2 + \left\|k_{r,w}(S_w^t)\nabla p^t - k_{r,w}(\hat{S}_w^t)\nabla\hat{p}^t\right\|_2^2\right\}_i,$$

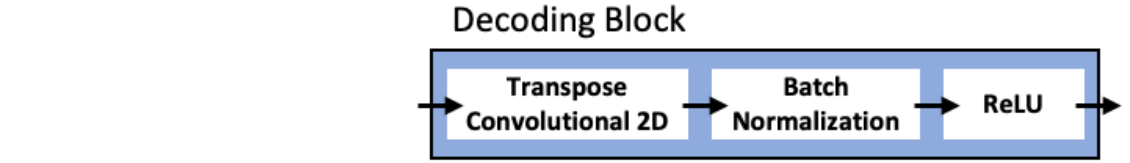
$$(\mathcal{L}_{\text{flux2Ph,pred}})_i = k\left\{\left\|k_{r,o}(S_w^{t+1})\nabla p^{t+1} - k_{r,o}(\hat{S}_w^{t+1})\nabla\hat{p}^{t+1}\right\|_2^2 + \left\|k_{r,w}(S_w^{t+1})\nabla p^{t+1} - k_{r,w}(\hat{S}_w^{t+1})\nabla\hat{p}^{t+1}\right\|_2^2\right\}_i,$$

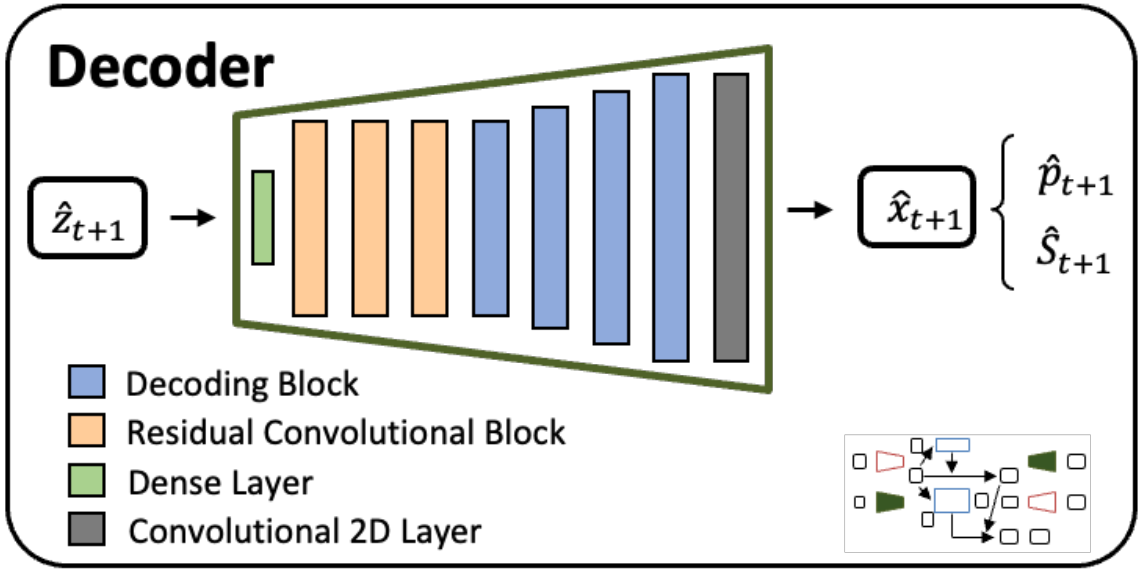
** Coutinho EJR, Dall'Aqua M and Gildin E (2021) Physics-Aware Deep-Learning-Based Proxy Reservoir Simulation Model Equipped With State and Well Output Prediction. Front. Appl. Math. Stat. 7:651178. doi: 10.3389/fams.2021.651178

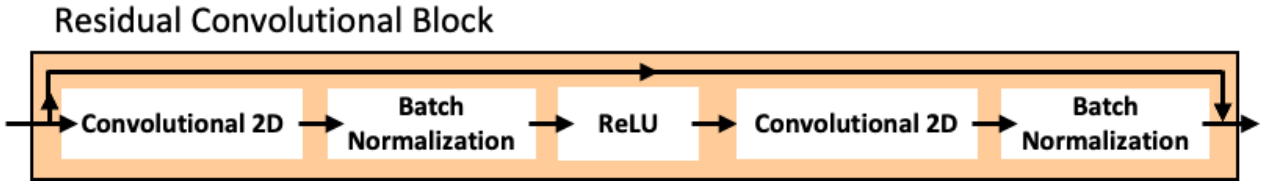
NN Structure - Autoencoder

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering



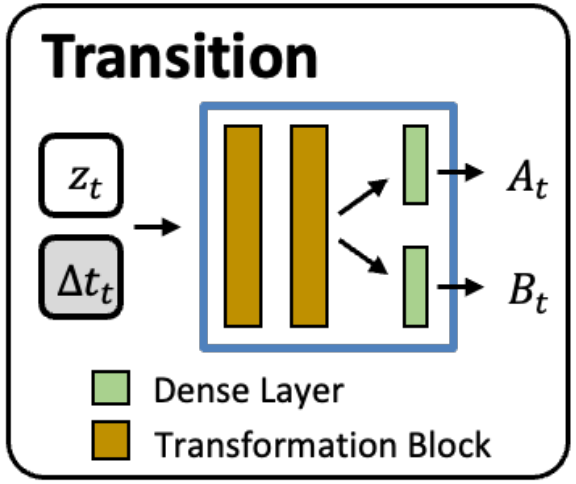






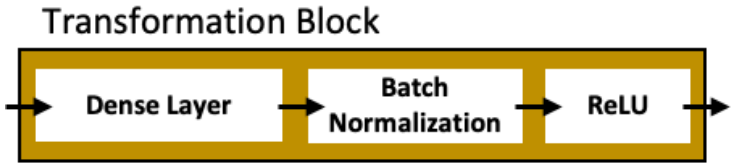
NN Structure – Transitions NN

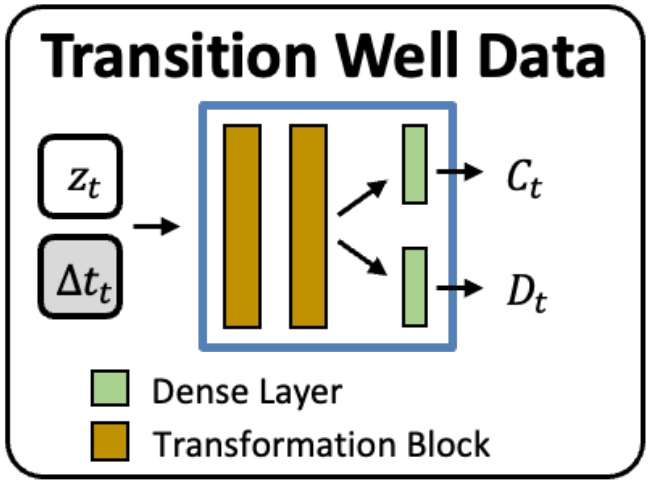
TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering



Adapted from Jin, et. al 2020

$$\hat{z}_{t+1} = A_t z_t + B_t u_t$$





$$\hat{y}_{t+1} = C_t \hat{z}_{t+1} + D_t u_t$$

What is (Fourier) Neural Operators?

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

Problem setting

$$-\nabla \cdot \left(K(x)\nabla \mathrm{u}(\mathrm{x})\right) = f(x), \text{ with proper BC on } \partial D$$

Solution → Green's function

G → Green's function or Kernel Function

$$u(x) = \int_D G_K(x,y) f(y) dy + non\ zero\ BC$$

Neural Operators → Approximate the Kernel (or convolution) by a NN

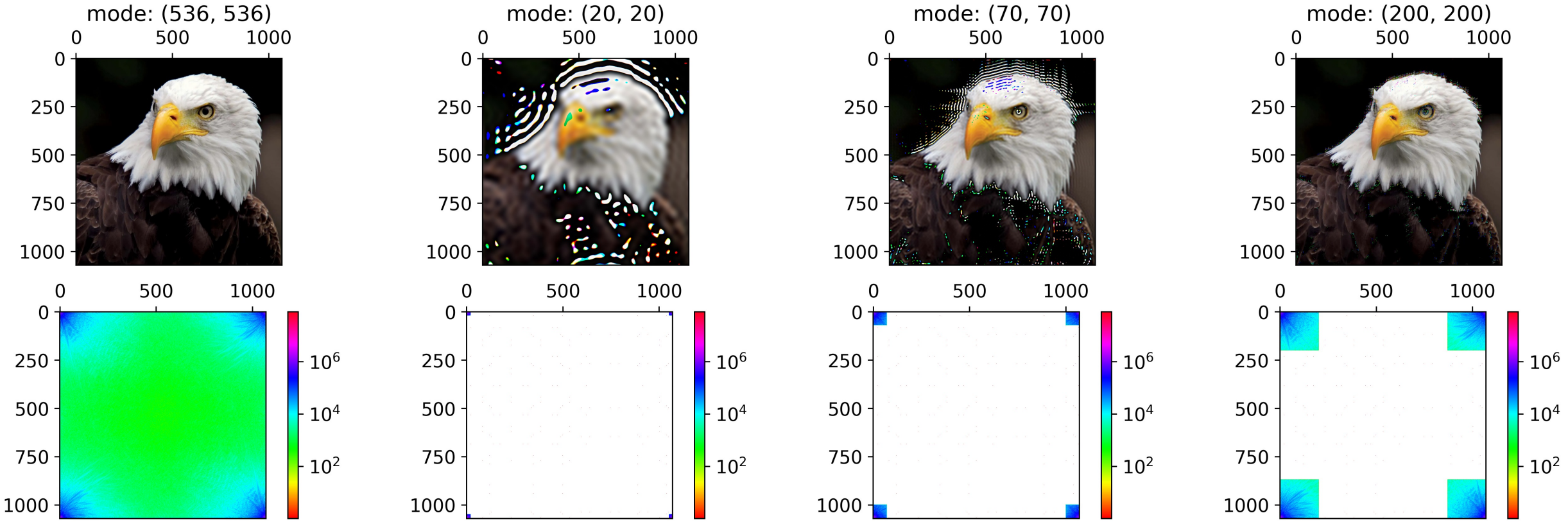
Architecture

$$v_{l+1}(s) = \sigma\left(W_l v_l(s) + \int_D \kappa_l(s,z) v_l(z)\, dz + b_l(s)\right)$$

Image recovery considering Fourier high modes

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

- High modes are positioned at the corners of the image in Fourier space
- Inverting back high models to the real space recovers the image to a great accuracy.
- Gibbs phenomena at sharp edges.

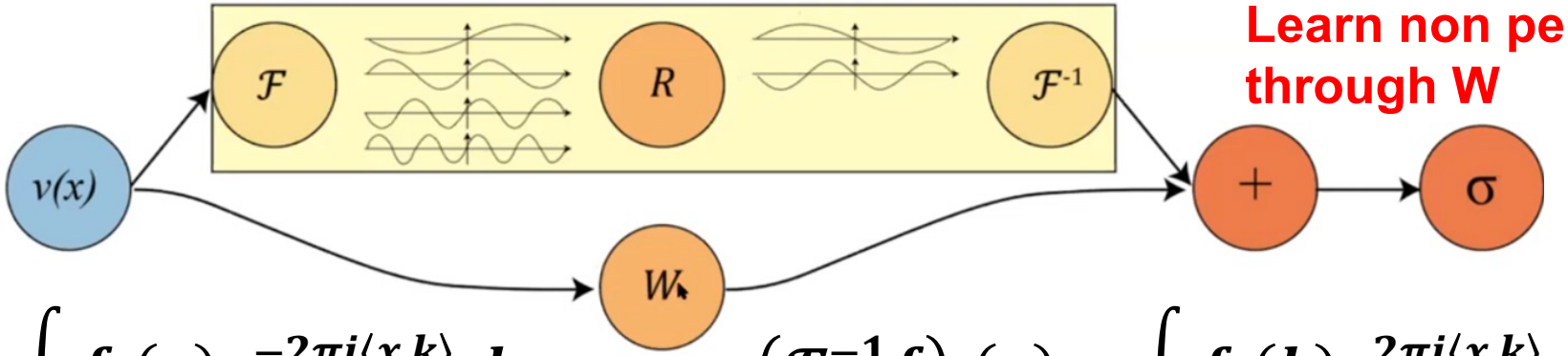


Source: Google images

Fourier Neural Operator (FNO)

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

FNO leverages neural networks with Fourier transforms



Learn non periodic inputs through W

$$(\mathcal{F}f)_j(k) = \int_D f_j(x) e^{-2\pi i \langle x,k \rangle} dx ; \qquad \left(\mathcal{F}^{-1}f\right)_j(x) = \int_D f_i(k) e^{2\pi i \langle x,k \rangle} dk$$

$$(\mathcal{K}(a;\phi)v_t)(x) = \mathcal{F}^{-1}\left(\mathcal{F}\left(\kappa_\phi\right) \cdot \mathcal{F}(v_t)\right)(x)$$

$$\boldsymbol{(\mathcal{K}(\phi)v_t)(x) = \mathcal{F}^{-1}\left(R_\phi \cdot \mathcal{F}(v_t)\right)(x)}$$

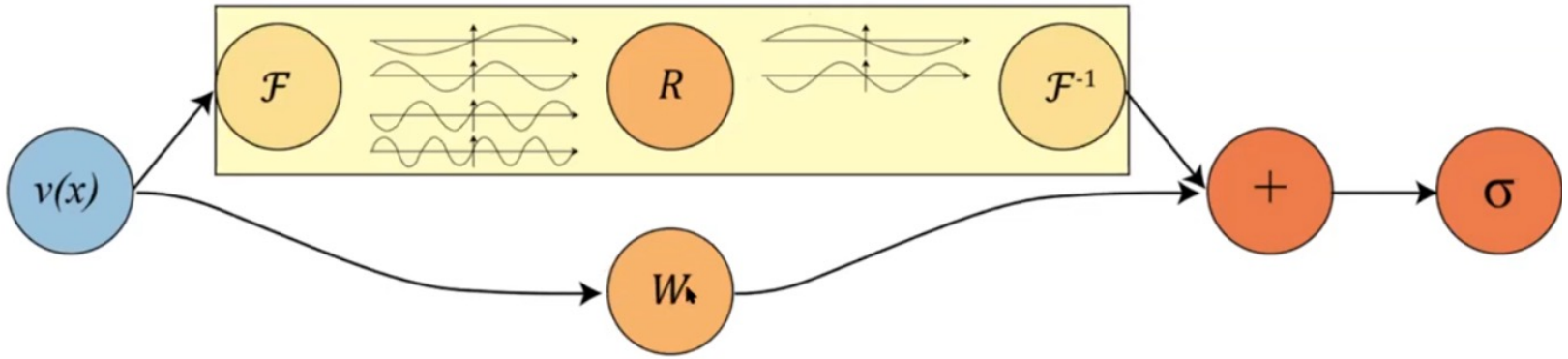
R_ϕ is the Fourier transform of a periodic function parameterized by ϕ.

Kovachki et al, Neural Operator: Learning Maps Between Function Spaces With Applications to PDEs Journal of Machine Learning Research 24 (2023)

Fourier Neural Operator (FNO)

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

FNO leverages neural networks with Fourier transforms to map its input function to the target function in the frequency domain, showcasing notably superior performance compared to traditional Convolutional Neural Networks (CNNs)



Learn non periodic inputs through W

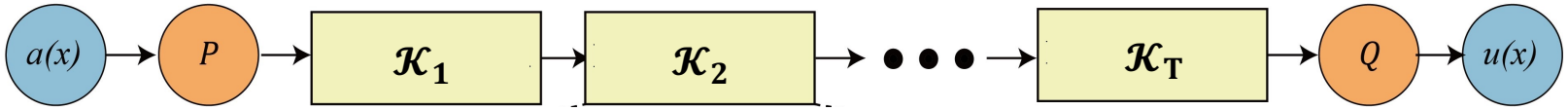
$$v_{l+1}(s) = \sigma\left(W_l v_l(s) + \int_D \kappa_l(s,z) v_l(z)\, dz + b_l(s)\right), \quad l = 0, \ldots, L-1$$

Kovachki et al, Neural Operator: Learning Maps Between Function Spaces With Applications to PDEs Journal of Machine Learning Research 24 (2023)

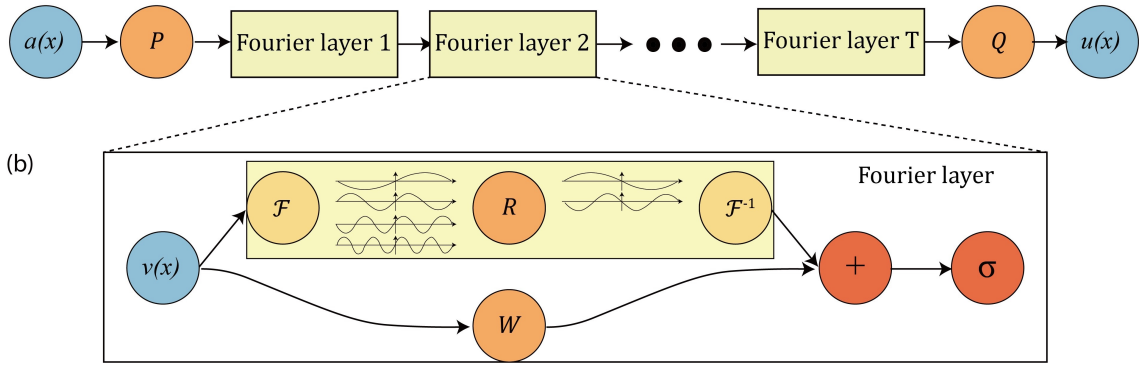
How about non-linear PDE's

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

- Embeddings → Lift to higher dimensions (P, Q)
- Compose several layers for non-linear pde's



- The R tensor is a multi-dimensional tensor of learnable weights that learns the high Fourier modes.

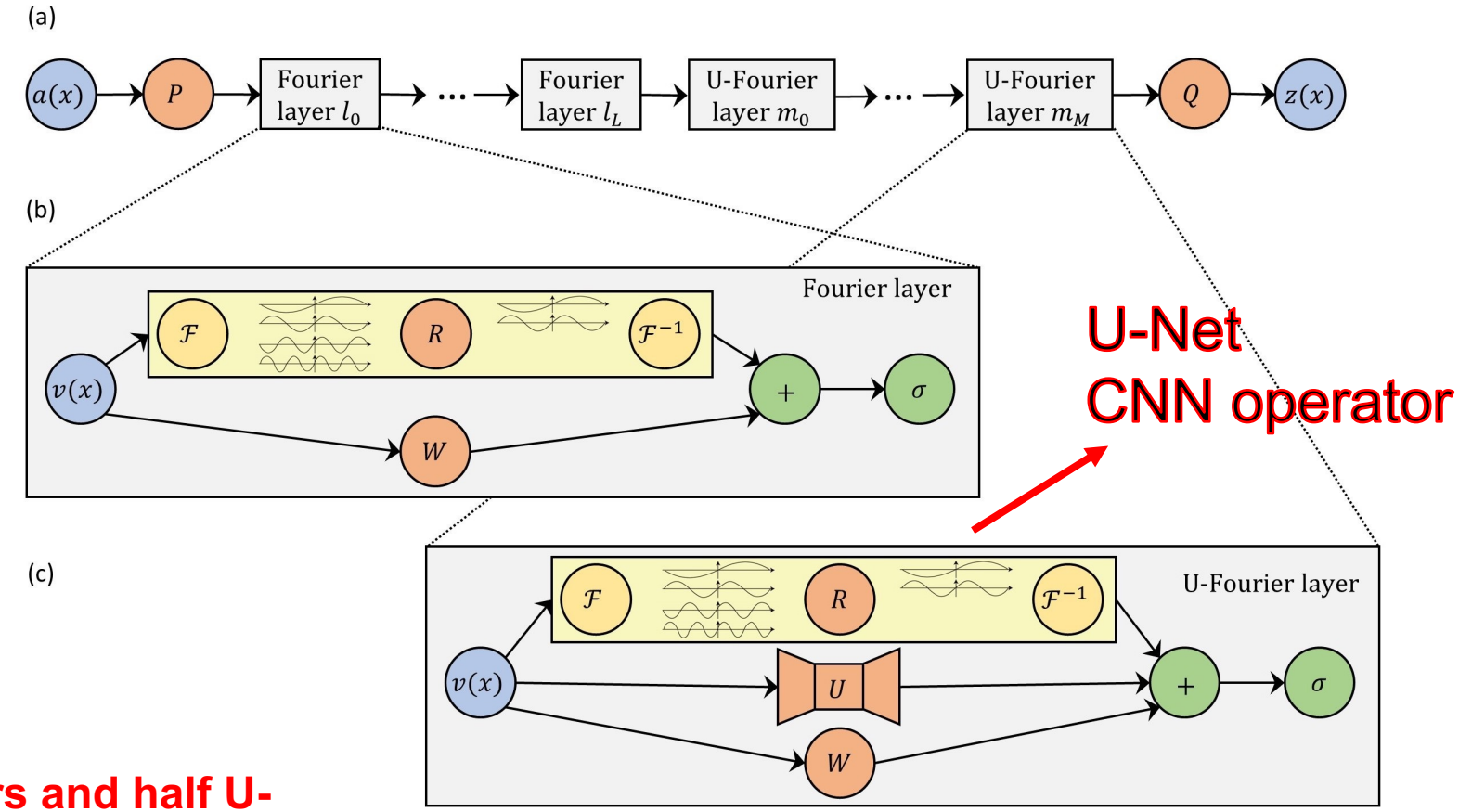


Kovachki et al, Neural Operator: Learning Maps Between Function Spaces With Applications to PDEs Journal of Machine Learning Research 24 (2023)

U-FNO for better accuracy

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

The U-Net processes local convolution to enrich the representation power of the U-FNO in higher frequencies information



U-Net
CNN operator

For the multi-phase flow problems → the architecture with half Fourier layers and half U-Fourier layers performs better

Wen et al, U-FNO - an enhanced Fourier neural operator-based deep-learning model for multiphase flow Advances in Water Resources Volume 163, May 2022, 104180

Fourier neural operator (FNO) applied to Reservoir Simulation

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

Goal:

Train a FNO that enables pressure and saturation predictions on:

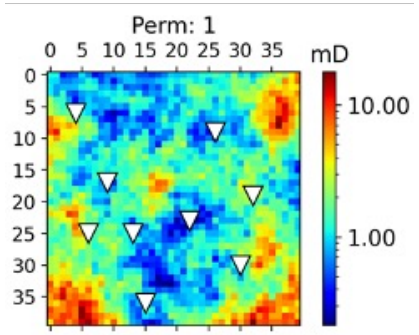
- New/unseen permeability fields
- New/unseen well locations.
- New/unseen well schedules (controls).
- New number of wells.

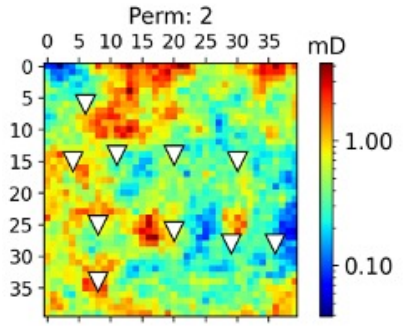
Input design:

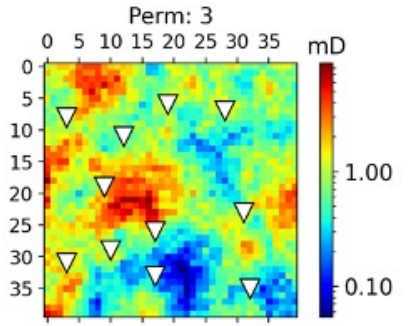
The input tensor $a(x) = (N_s, N_c, N_x, N_y, N_t)$

where:

N_s: number of samples
N_c: number of channels
N_x: number of x nodes
N_y: number of y nodes
N_t: number of t nodes







U – Fourier neural operator: Training

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

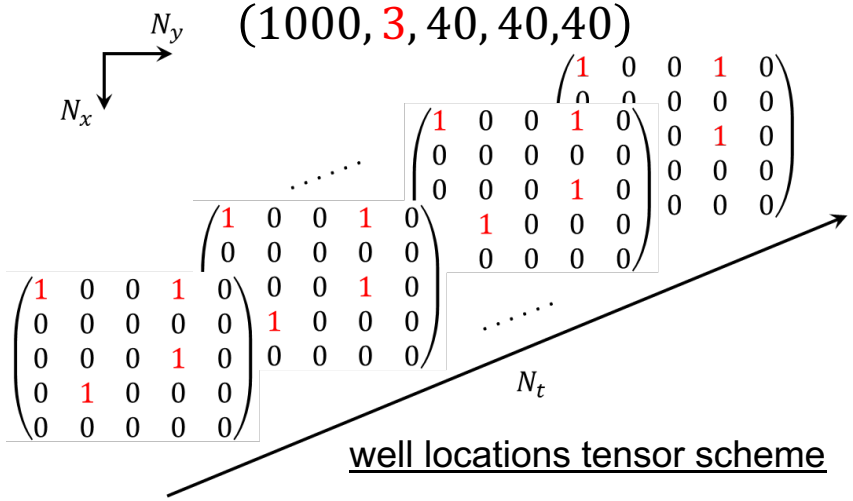
Training and testing:

- Generate 4200 simulations for 40x40 different permeability fields with 6 wells (4 prod, 2 inj) in each field randomly distributed in space. The simulation runs from 0 to 2400 days, with results cache every 10 days.

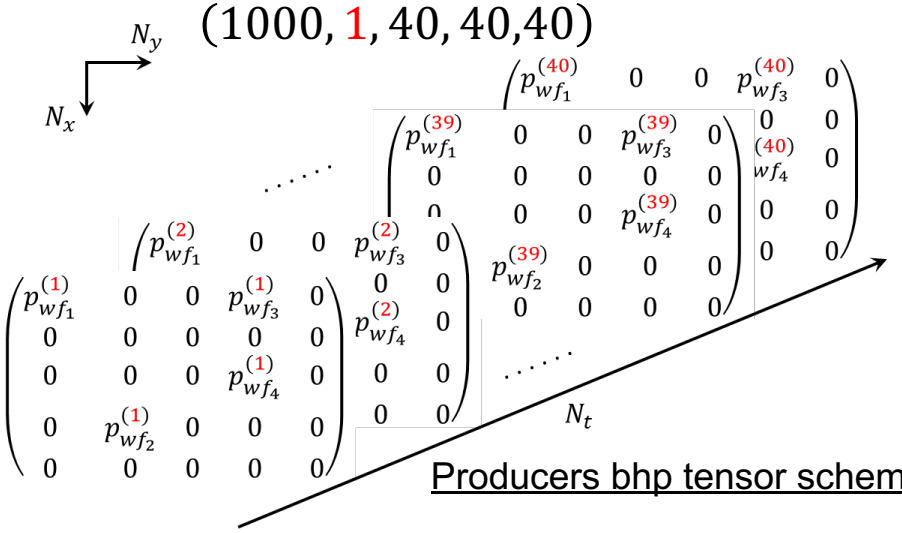
- 3500 for training, and 500 for validation, 200 for testing.
- Testing samples include from 3 up to 12 wells, whereas training and validation samples are restricted to 6 wells only.

- The training input tensor shape is: $(1000, 6, 40,40,40)$. The 7 channels are: (permeability, prod_bhps, inj_rates, well_locs, p_incon, sw_incon)

- The output tensor shape is: $(1000, 2, 40, 40,40)$



well locations tensor scheme

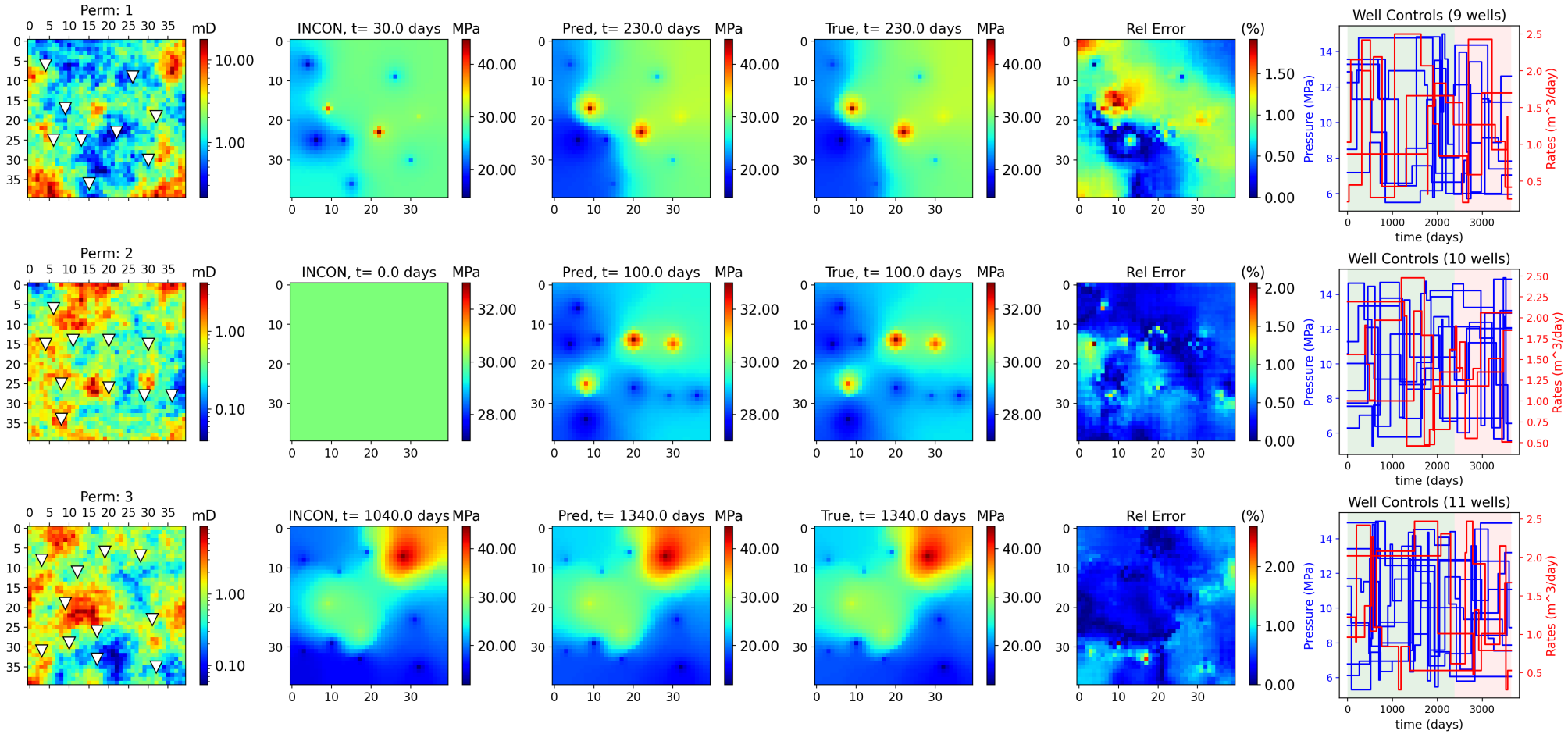


Producers bhp tensor scheme

U-FNO – PRESSURE PREDICTIONS

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

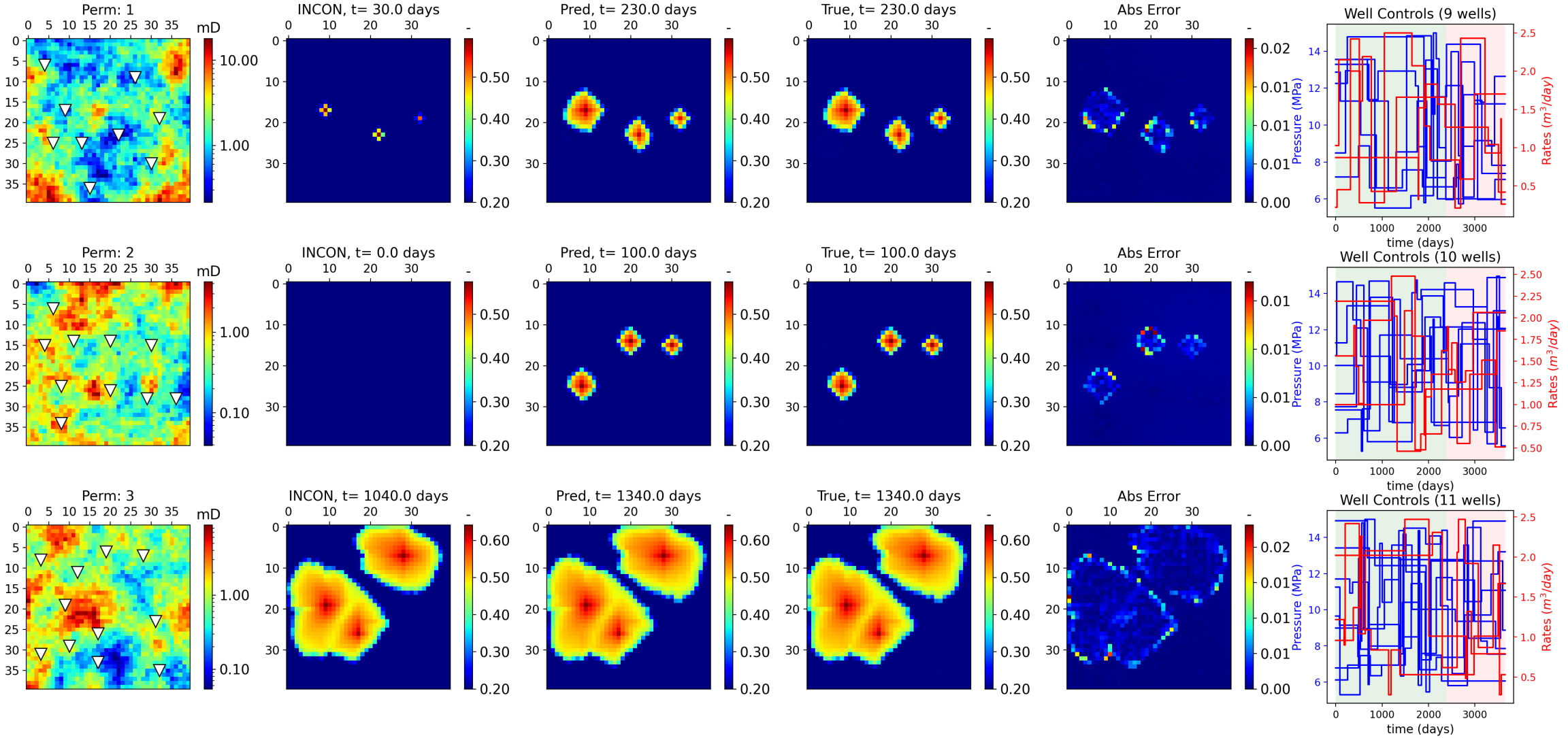
- 3500 training samples, 500 validation , 200 testing
- Training and validation samples have 6 wells (4 producers, 2 injectors)
- **Testing samples could have from 3 up to 12 wells.**



~ 3-4 hrs training

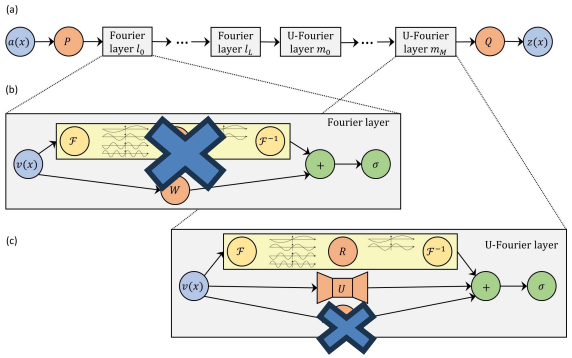
U-FNO – SATURATRION PREDICTIONS

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering



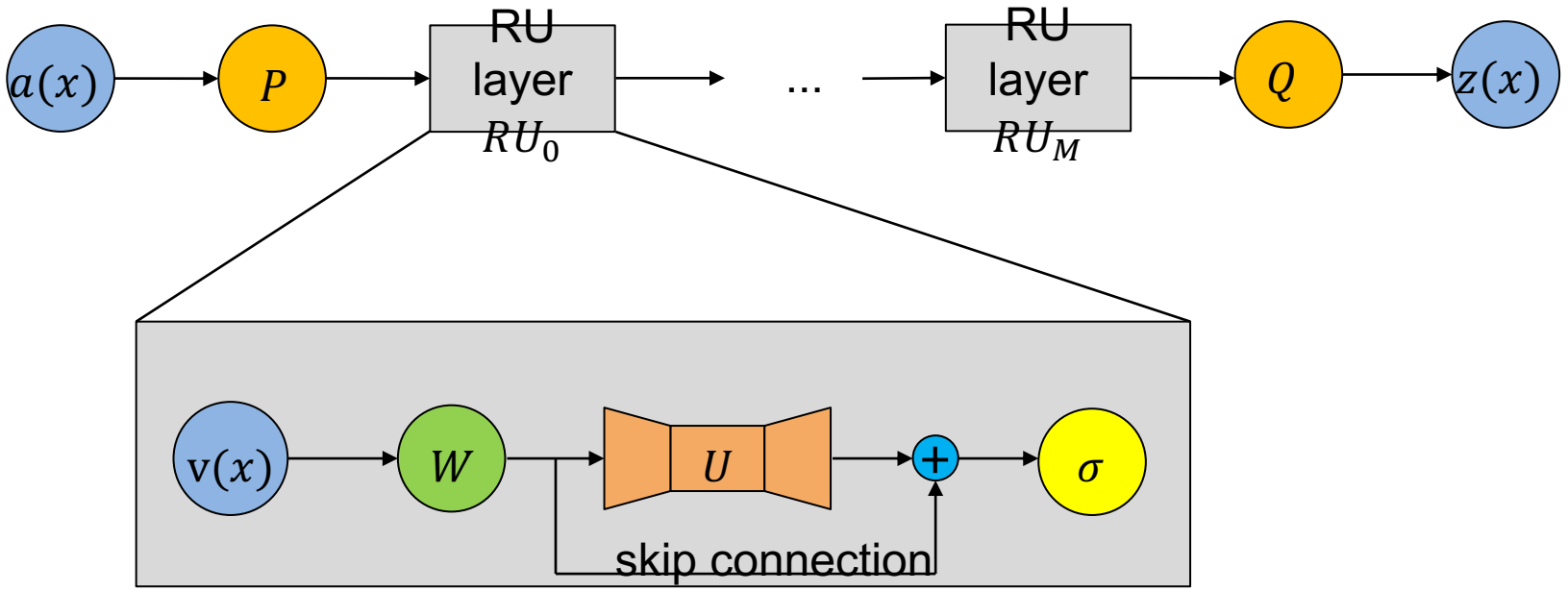
OUR NEW ARCHITECTURE – RESIDUAL U-NET

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering



OUR model is 40% faster to train than U-FNO

- The residual connection replaces the Fourier block.
- In contrast to Fourier, there is no lost information when considering the high modes only, because the skip connection will retain all information.
- 3RU-NET includes 3 RU layers
- 5RU-NET includes 5 RU layers



Added skip connection

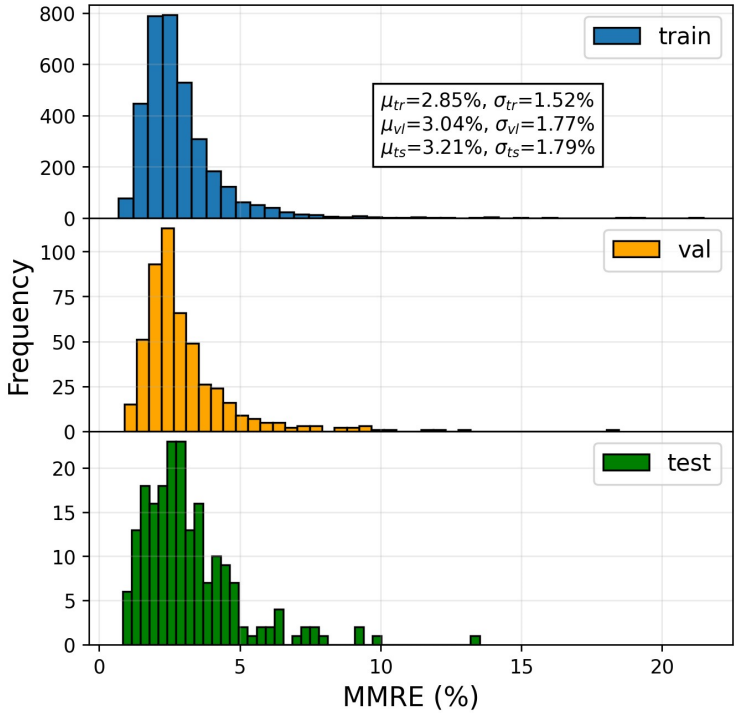
OURS vs U-FNO – ERRORS

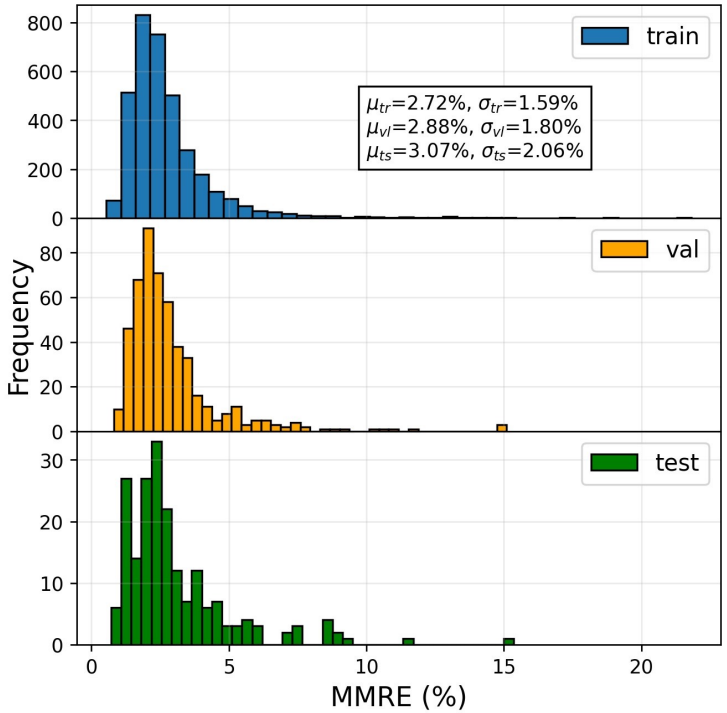
TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

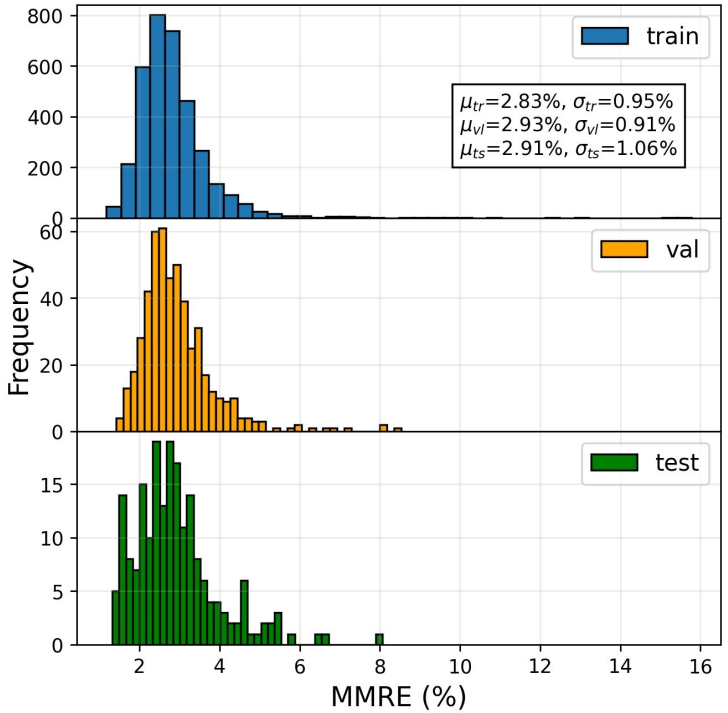
- 3500 training samples, 500 validation , 200 testing
- Training and validation samples have 6 wells (4 producers, 2 injectors)
- Testing samples have from 3 up to 12 wells.

OUR model is 40% faster to train than

U-FNO (3 U-Fourier blocks): 221 sec/epoch OURS_3RU-NETS: 132 sec/epoch OURS_ 5RU-NETS : 220 sec/epoch



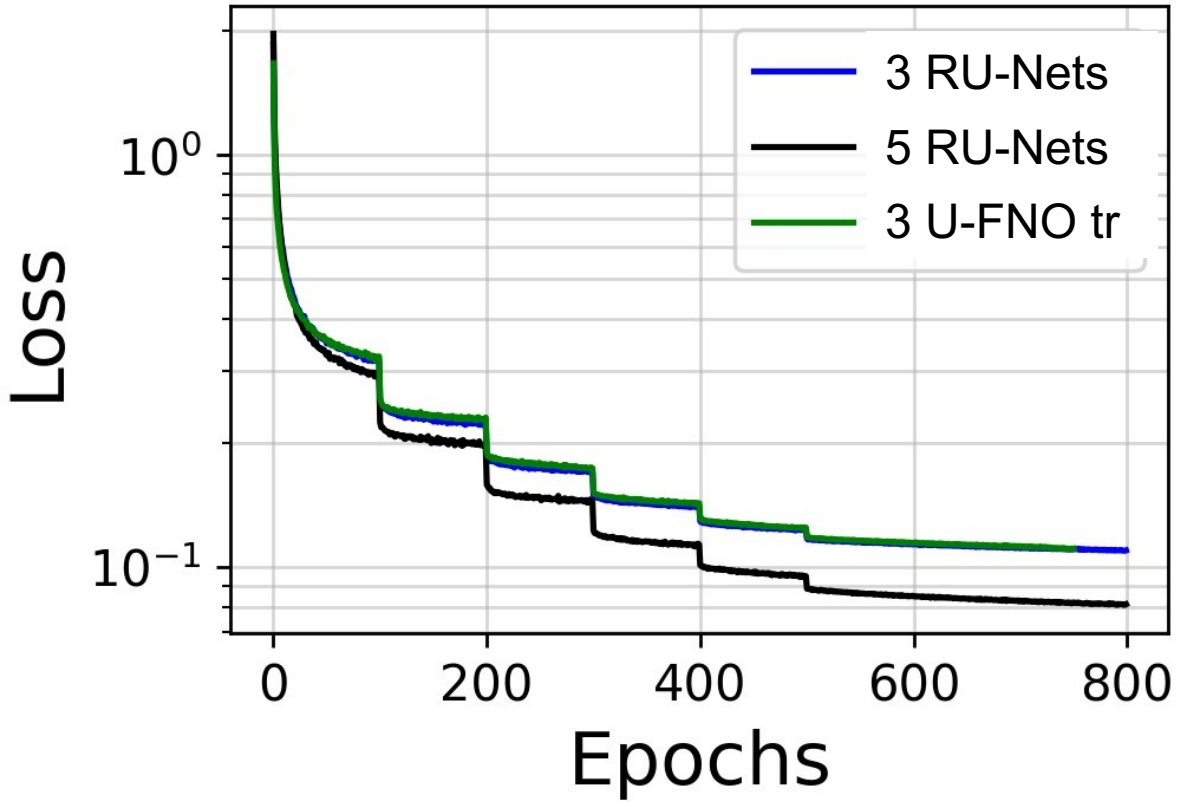


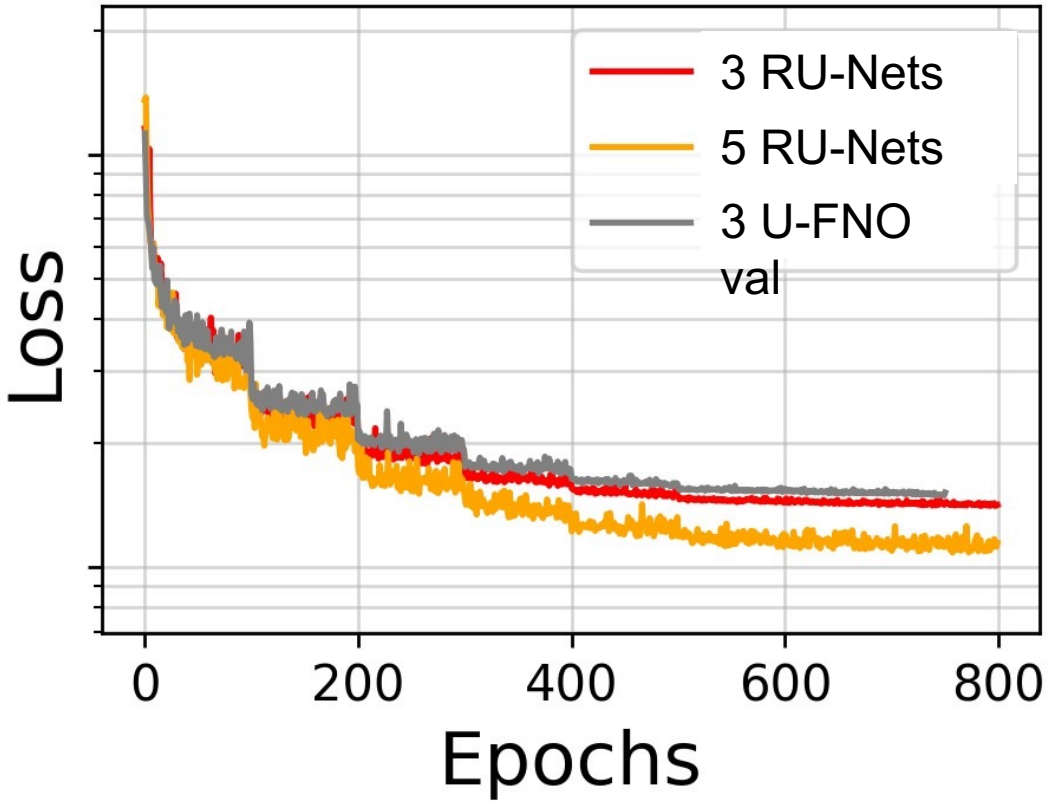


OURS vs U-FNO – Training and Validation Losses

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

- 3500 training samples, 500 validation , 200 testing
- Training and validation samples have 6 wells (4 producers, 2 injectors)
- Testing samples have from 3 up to 12 wells.

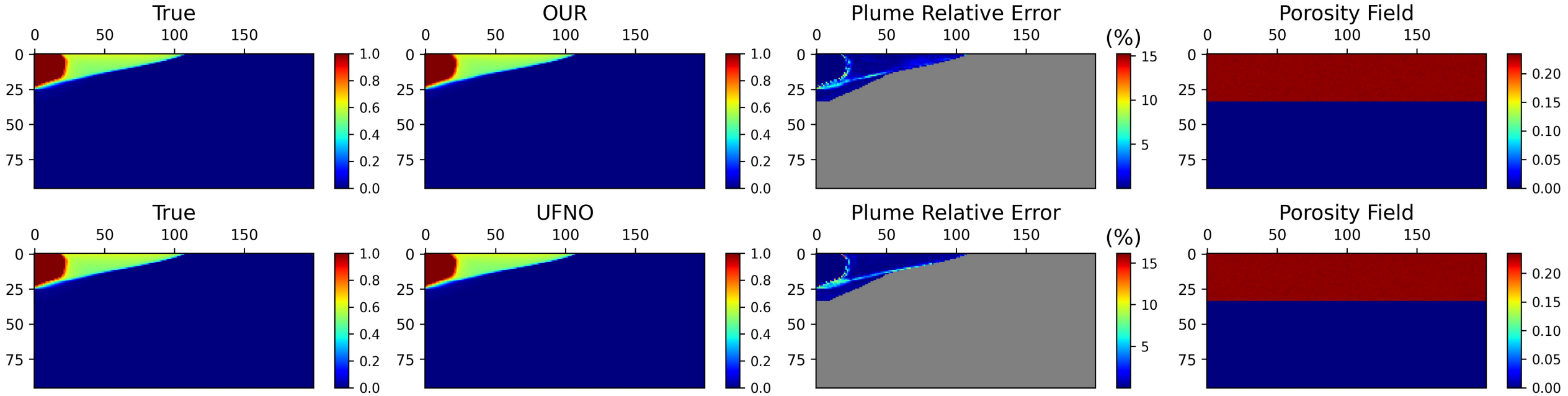




OURS vs U-FNO BENCHMARKING – SATURATION PREDICTION

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

Benchmarking our model against U-FNO →
with the same dataset used in the U-FNO paper (CCS case)

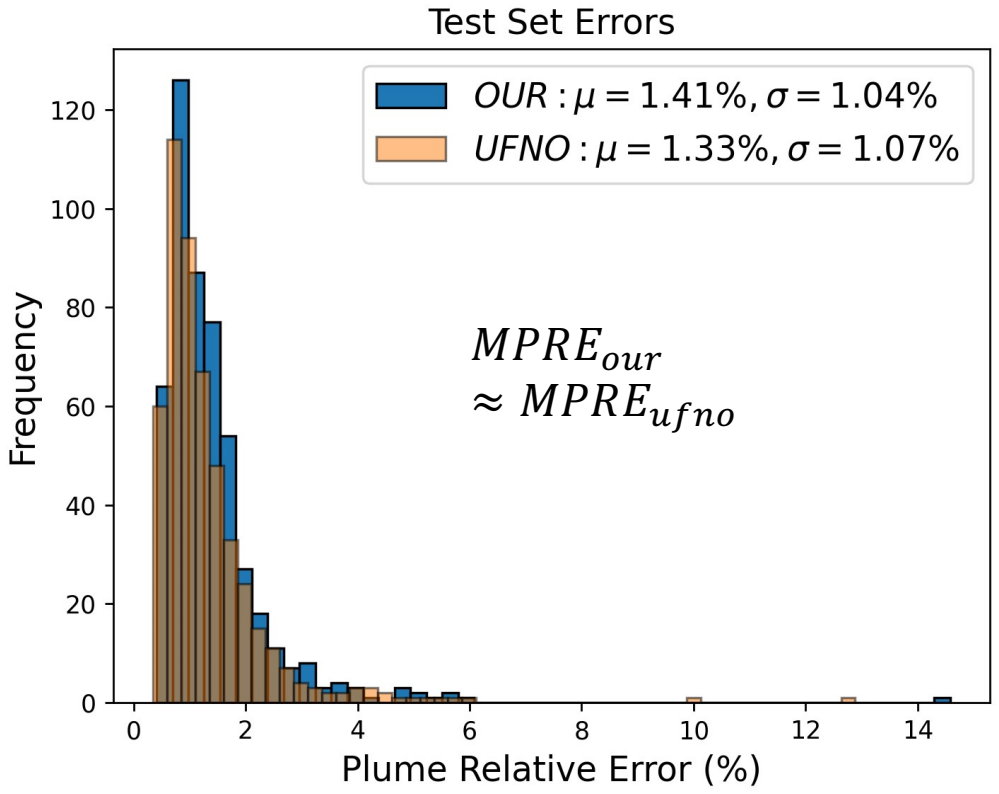


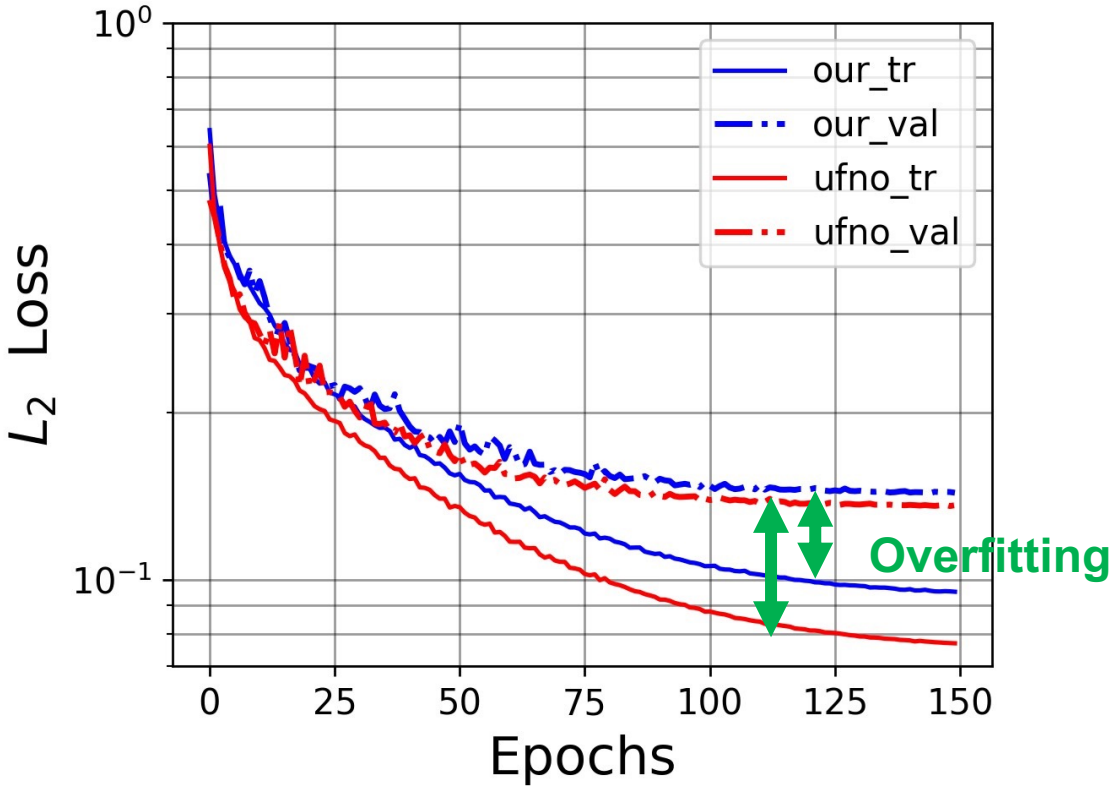
Wen et al, U-FNO - an enhanced Fourier neural operator-based deep-learning model for multiphase flow Advances in Water Resources Volume 163, May 2022, 104180

OURS vs U-FNO BENCHMARKING ERRORS

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

Less overfitting in our model – the distance between training loss and validation loss denotes overfitting.





Why this works? Perhaps an explanation…

NIKOLA B. KOVACHKI et al, OPERATOR LEARNING: ALGORITHMS AND ANALYSIS, arXiv:2402.15715, Feb 2024

Special case of an averaging neural operator

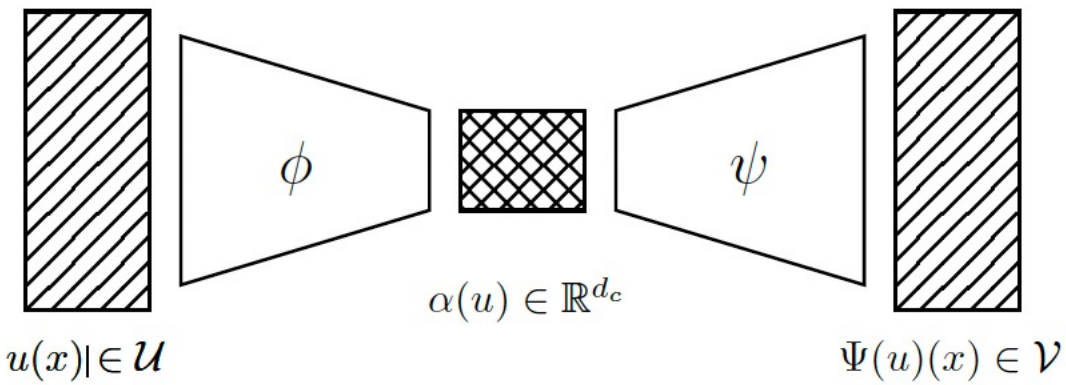


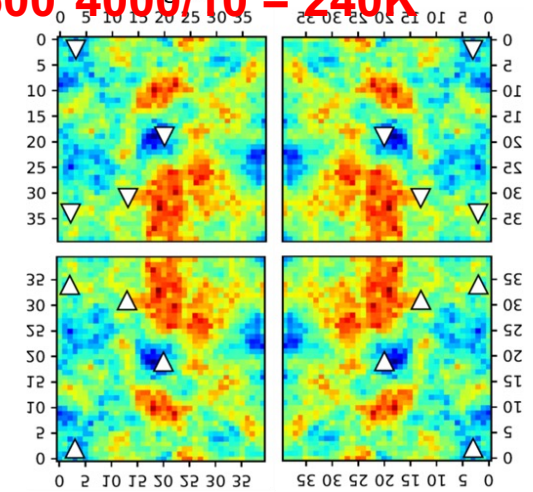
FIGURE 5. Special case of an averaging neural operator, illustrated as a nonlinear encoder-decoder architecture; with encoder $u \mapsto \alpha = \int_{\mathfrak{D}} \phi(u(y), y)\, dy$, and decoder $\alpha \mapsto \psi(\alpha, \cdot)$.

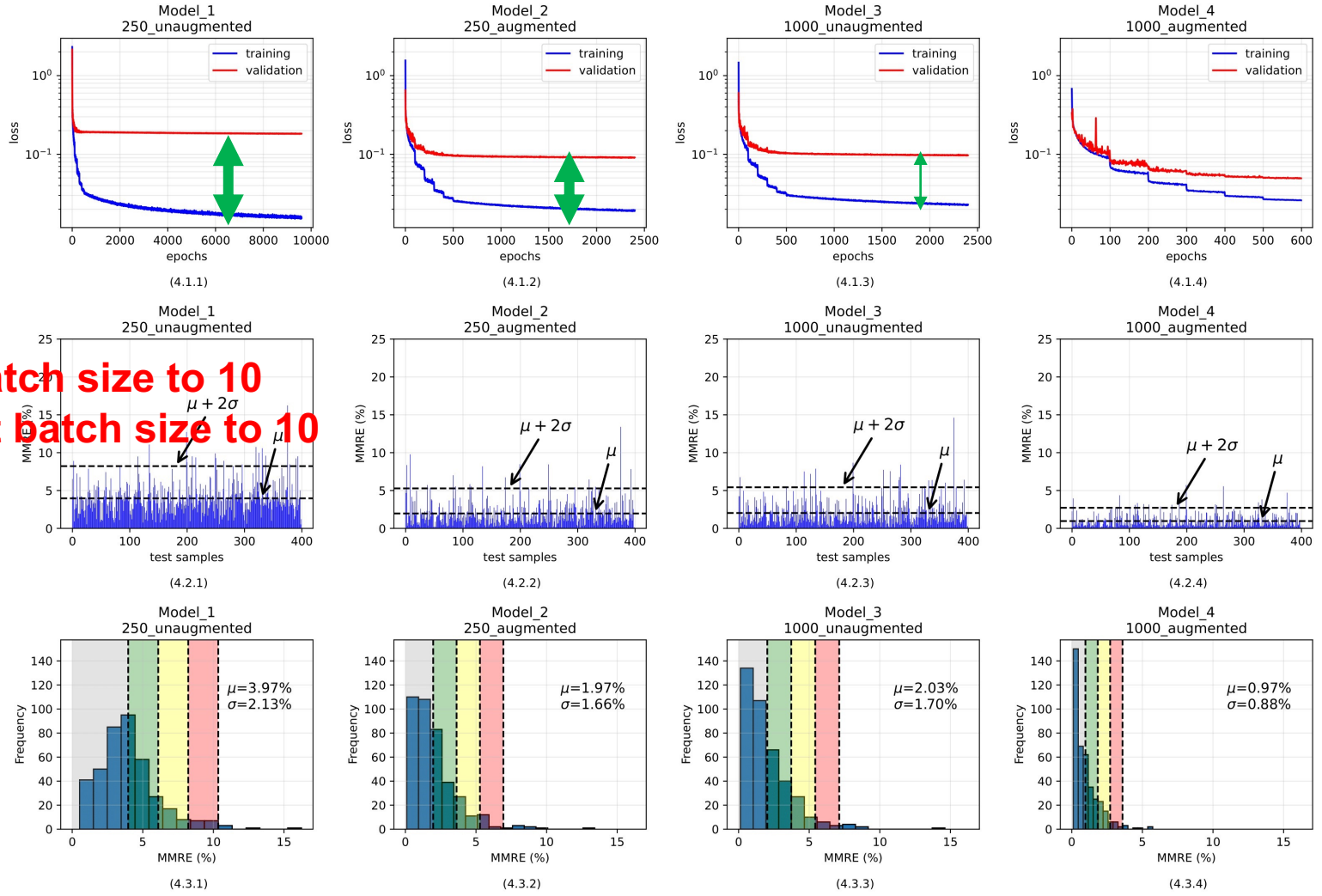
DATA AUGMENTATION REDUCES OVERFITTIN

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

	# samples	Epochs
Model 1	250 base	9600
Model 2	250 base + aug	2400
Model 3	1000 base	2400
Model 4	1000 base + aug	600

Model 1 → 9600*250/10 = 240K → Kept batch size to 10
Model 2/3 → 2400*1000/10 = 240K → Kept batch size to 10
Model 4 → 600*4000/10 = 240K



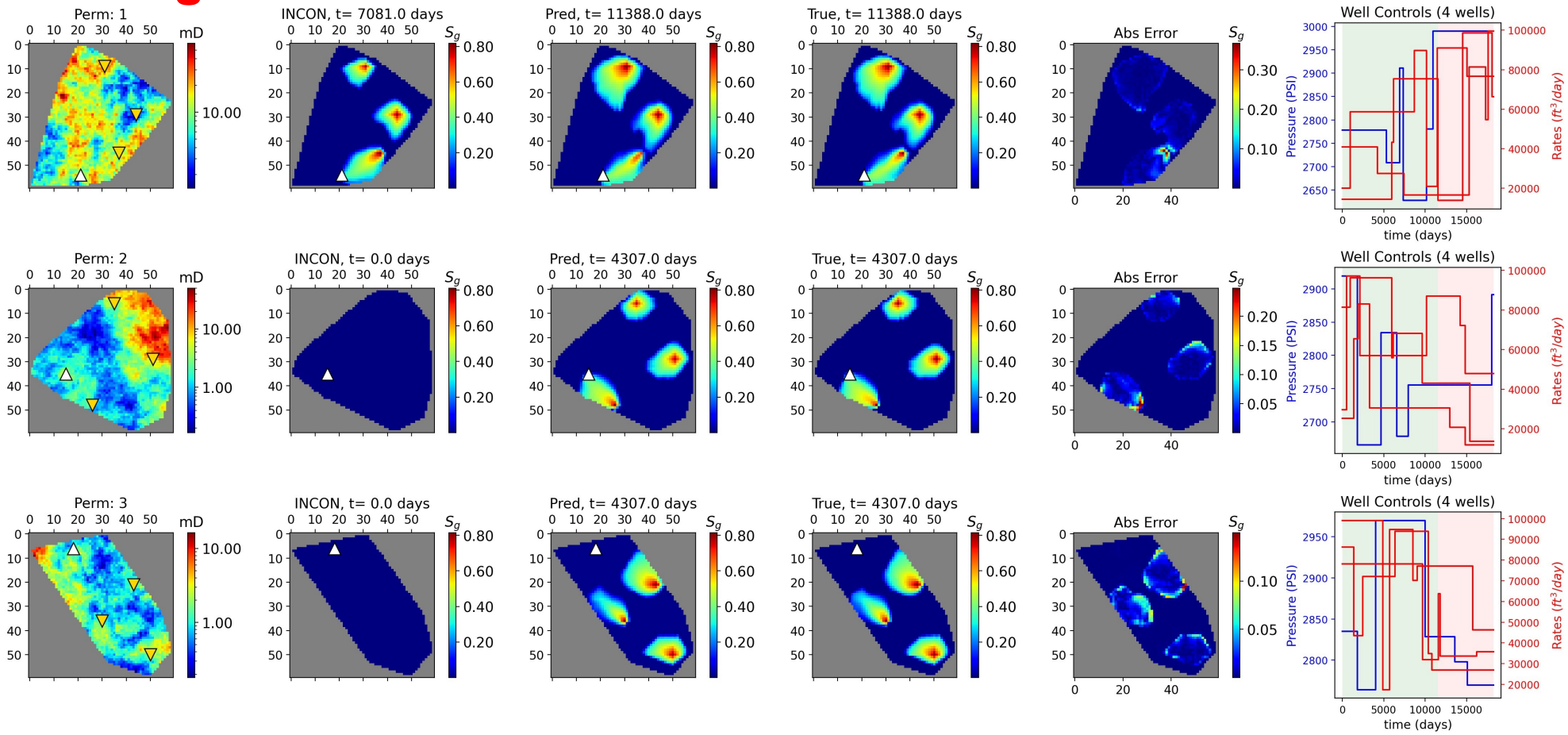


OUR MODEL FOR CCS ON IRREGULAR GEOMETRIES SATURATION PREDICTIONS – ONGOING WORK

TEXAS A&M UNIVERSITY
Harold Vance Department of Petroleum Engineering

Changing the geometry of the reservoir
60X60 → 1500 ft each grid blcok

3 CO2 injectors, 1 producer



Predicting on irregular geometries allows using all depleted reservoirs simulation data to train a global reservoir digital twin that is suitable for all shapes and geologies of different reservoirs.