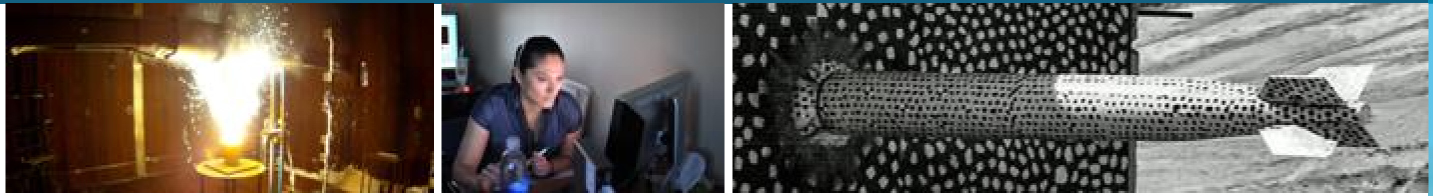
SAND2022-15745C

Reduced order modeling with boosting Barlow Twins self-supervised learning for **3-Dimensional** subsurface physics



T. Kadeethum and H. Yoon

Geomechanics Department, Sandia National Laboratories

Albuquerque, NM, USA

Collaborators: V. L. S. Silva (ICL), P. Salinas (OpenGoSim), C. Pain (ICL)

3rd International Conference on Coupled Processes in Fractured Geological Media: Observation, Modeling, and Application

This work was supported by DOE Office of Fossil Energy and Carbon Management project -Science-informed Machine Learning to Accelerate Real Time (SMART) Decisions in Subsurface Applications-Carbon Storage.

SAND2022-xxxx

Why reduced order model?

Full order model (FOM) is computationally demanding.

This would take 1-2 hours[1,2].

Imagine if you do 100,000 times of this type of simulation.

FOM is computationally very expensive for high fidelity simulations, uncertainty quantification, optimization, or inverse modeling

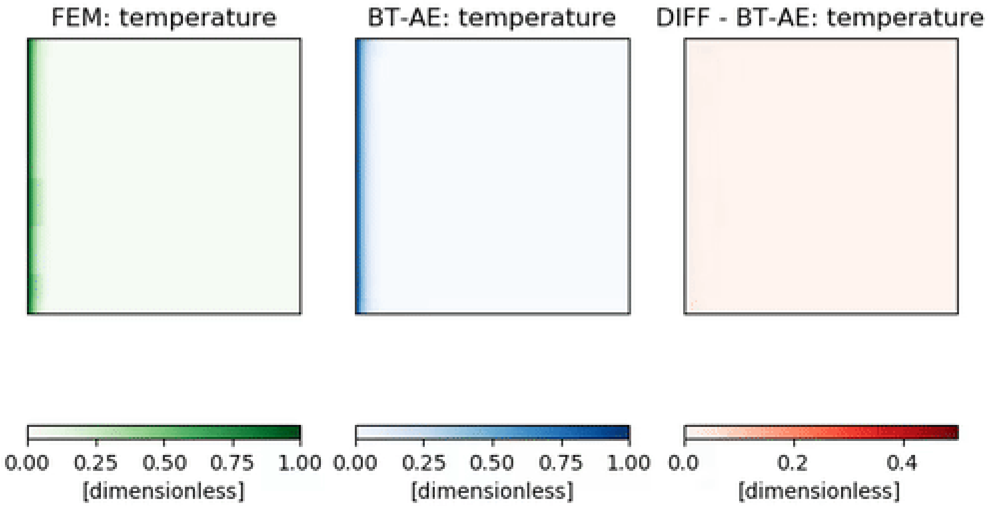
---

[1] Kadeethum et al., 2022, Advances in Water Resources
[2] Kadeethum et al., 2021, Computers & Geosciences

Motivation

1. A unified framework suitable for problems that lie within both **linear** and **nonlinear** manifolds

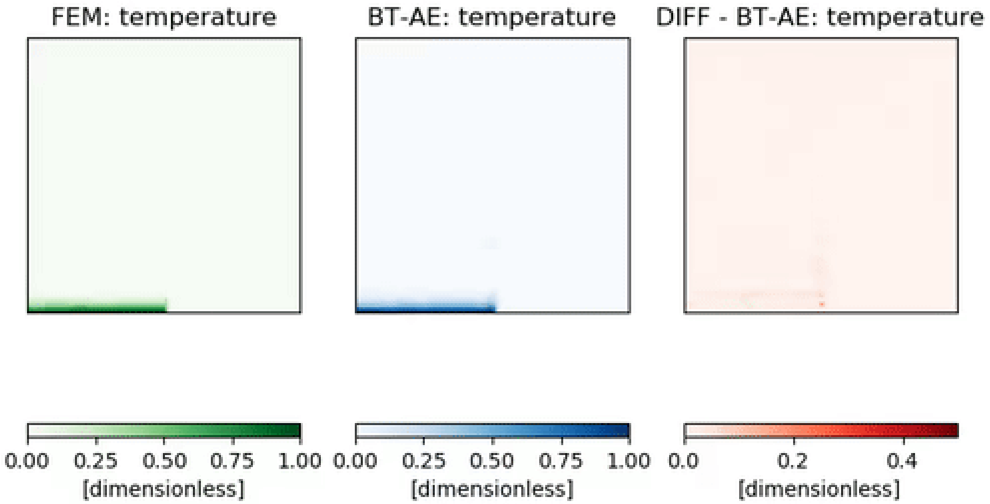
(proper orthogonal decomposition (POD) yields optimal data compression for linear manifolds) [1]



POD performs better

One ring to rule them all
(The Lord of the Rings)

AE-based performs better



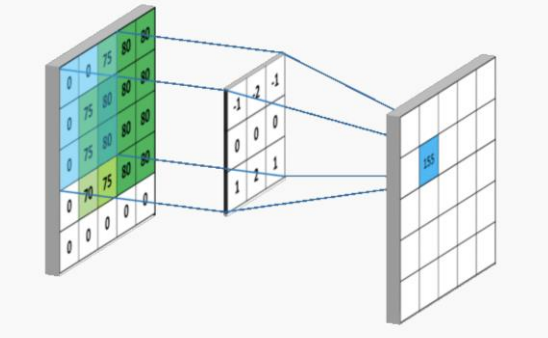
---

[1] Kadeethum et al. (2022, Advances in Water Resources)

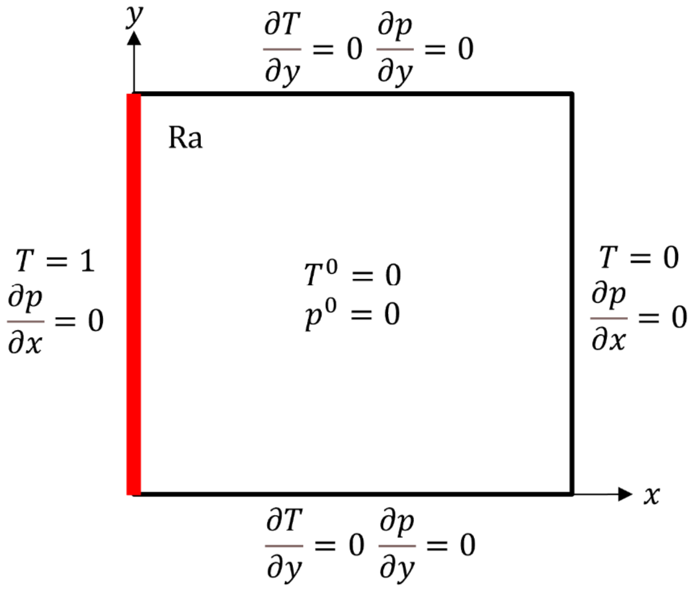
Motivation - continued

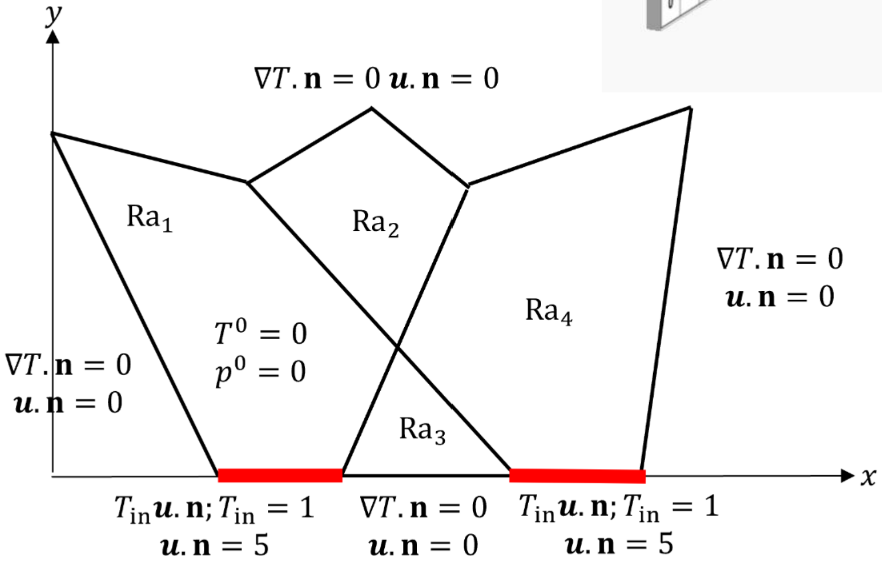
1. A unified framework suitable for problems that lie within both **linear** and **nonlinear** manifolds (proper orthogonal decomposition (POD) yields optimal data compression for linear manifolds) [1]

2. A framework that does not rely on ‘convolutional layers,’ which makes our framework applicable to both **structured** and **unstructured** meshes [1, 2]



[3]





[1] Kadeethum et al. (2022, Advances in Water Resources)

[2] Kadeethum et al. (2021, Nature Computational Science)

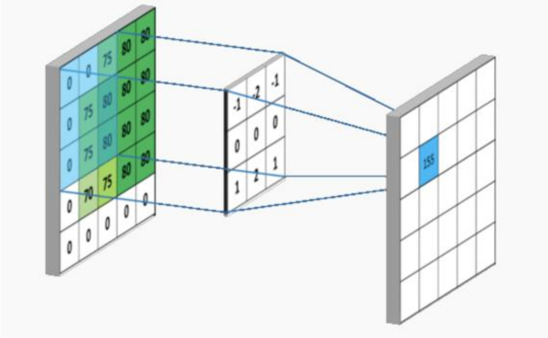
[3] https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac

Motivation - continued

1. A unified framework suitable for problems that lie within both **linear** and **nonlinear** manifolds

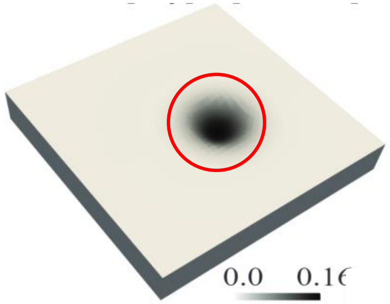
(proper orthogonal decomposition (POD) yields optimal data compression for linear manifolds) [1]

2. A framework that does not rely on 'convolutional layers,' which makes our framework applicable to both **structured** and **unstructured** meshes [1, 2]



[3]

3. Applying machine learning techniques for the physics-based problems with point source (or Dirac delta distribution) such as contact problems or subsurface flow with wells ➔ how to deal with imbalanced training data?



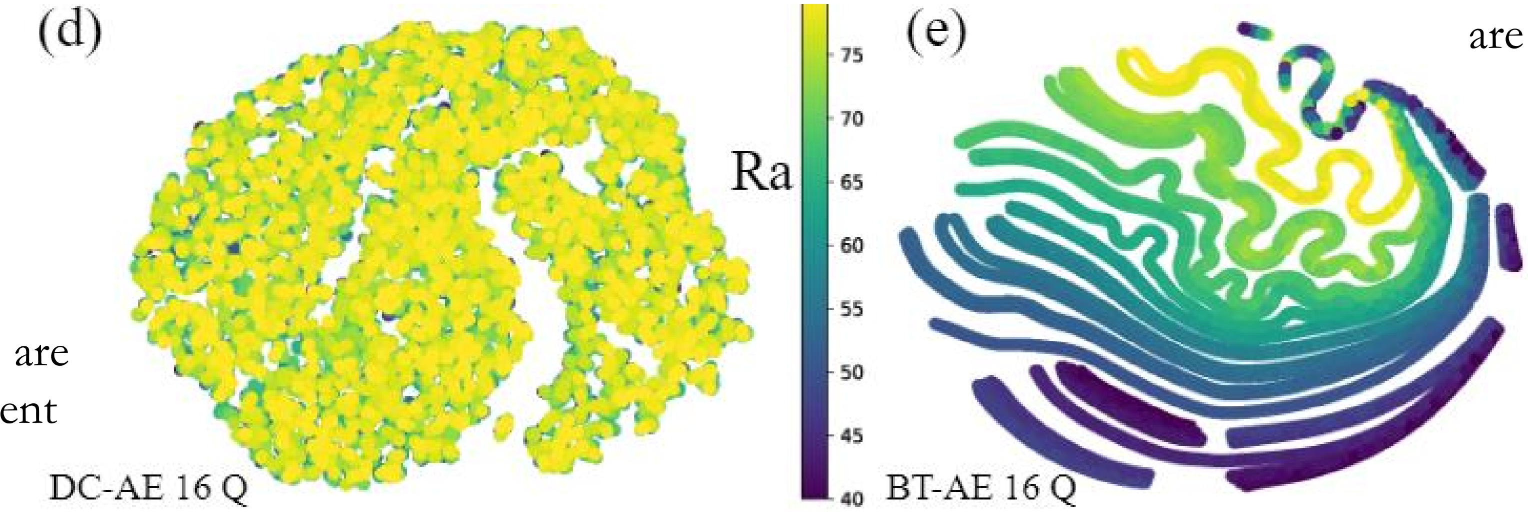
---

[1] Kadeethum et al. (2022, Advances in Water Resources)
[2] Kadeethum et al. (2021, Nature Computational Science)
[3] https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac

Idea

1.A key to develop a good ROM is to produce **better reduced manifolds** [1].

2. We apply Barlow Twins (BT) self-supervised learning [1,2], where BT maximizes the information content of the embedding with the latent space through a **joint embedding architecture**



The nonlinear manifolds are well structure

The nonlinear manifolds are not well structured in latent space

---

[1] Kadeethum et al. (2022, Scientific Report , accepted)

[2] Zbontar et al. (2021, arXiv:2103.03230)

Idea - continued

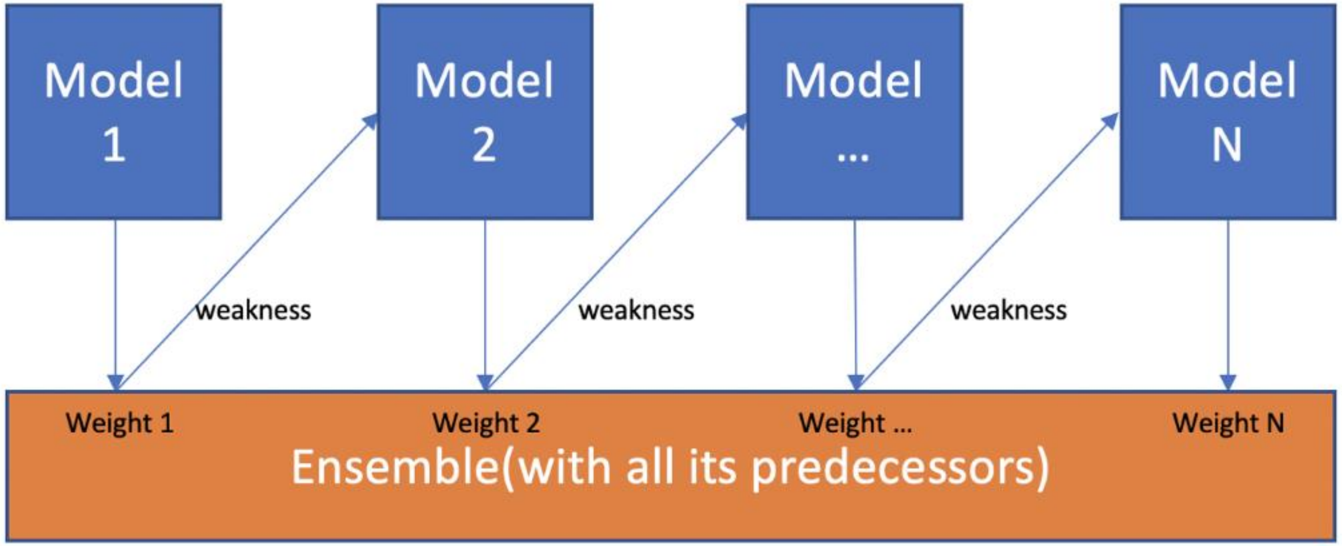
3. We apply a boosting concept for our previous BT-ROM [1]

4. Each model (in general sense) is trained **sequentially** using **subsample** from the training set with **weights**

6. The weights are calculated based on the current model's performance (i.e., **more error more weights**)

5. This way, the $\mathbf{model}^{\mathbf{n+1}}$ is forced to learn the samples that $\mathbf{model}^{\mathbf{n}}$ fails to mimic

[2]



[1] Kadeethum et al. (2022, Scientific Report, accepted)

[2] https://towardsdatascience.com/boosting-algorithms-explained-d38f56ef3f30

Methodology

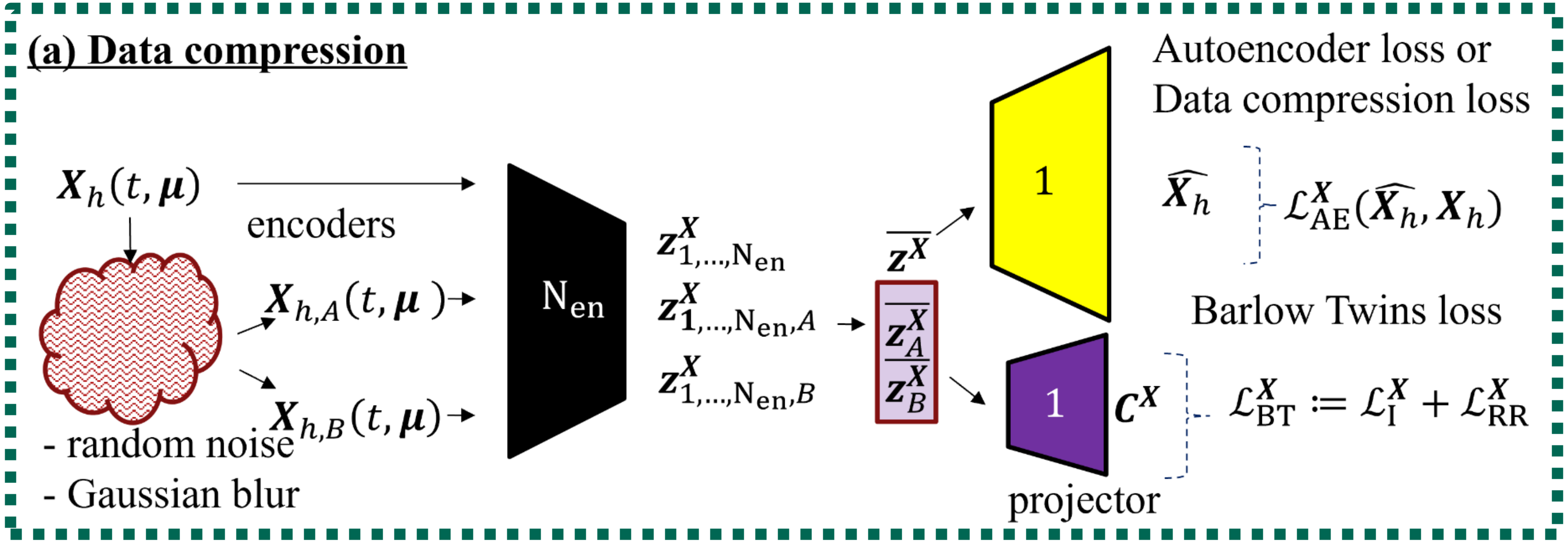
Data compression: training BBT-AE model

The machine learning model has **many** encoder, **one** decoder, and **one** projector.

The main goal is to maximizes the information content of the embedding with the latent space through a joint embedding architecture.

Resulting in a **better reduced manifolds**

<u>If we have 1 encoder, our model is BT-ROM</u>
<u>If we have more than 1 encoders, our model is BBT-ROM</u>



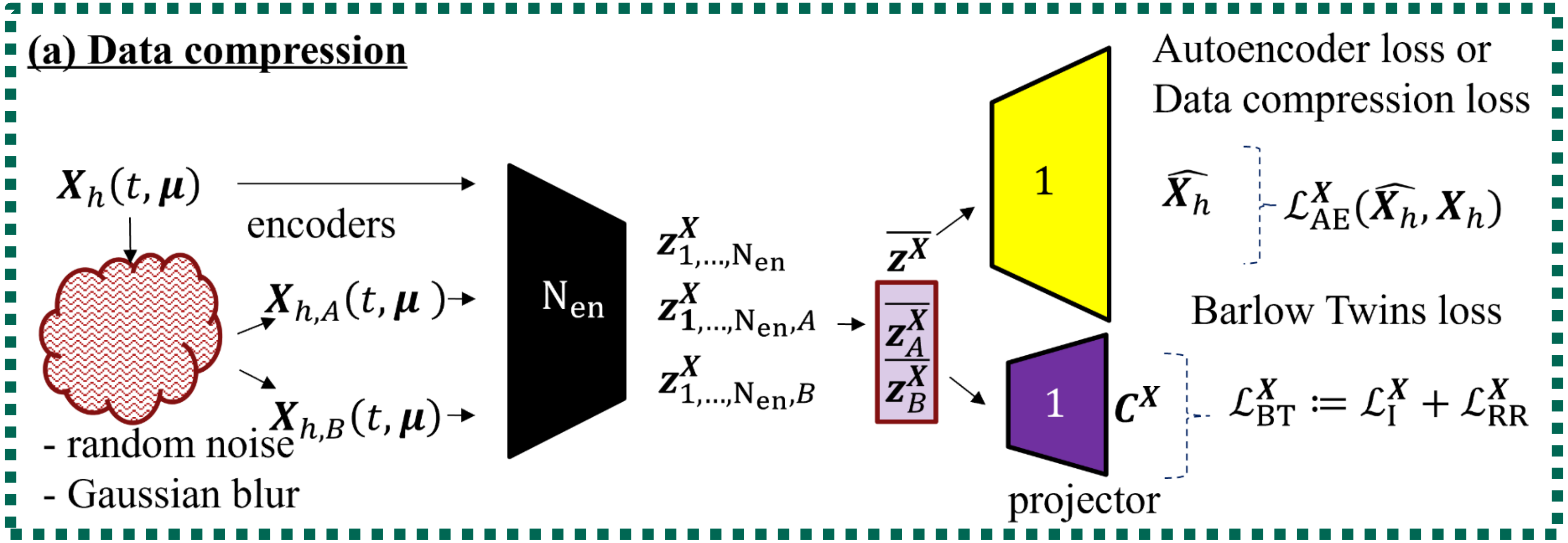
Methodology - continued

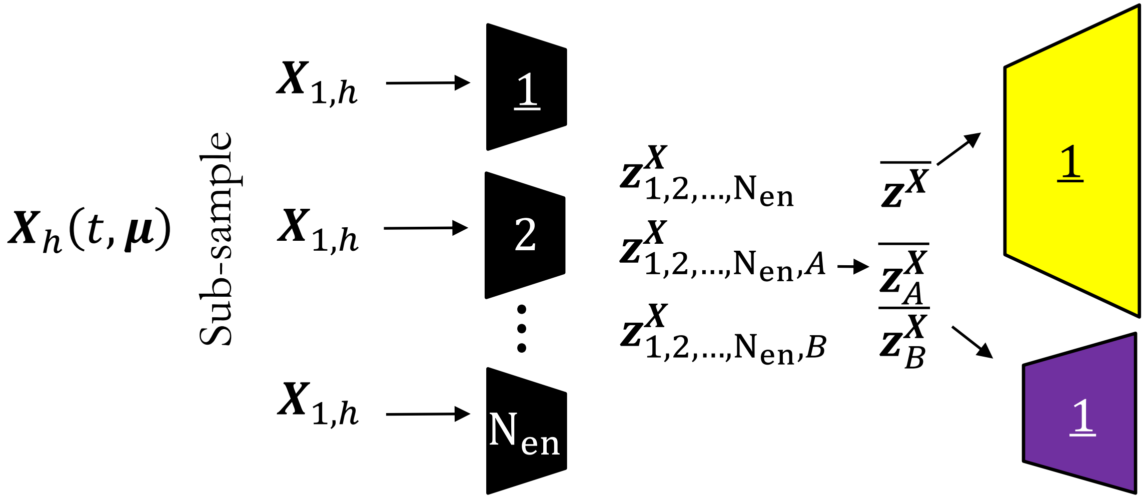
Boosting in-a-nut-shell:

We train each encoder sequentially

Encoder 1

- all other encoders are frozen
- $\boldsymbol{X}_{1,h}$ is sub-sampled from $\boldsymbol{X}_h(t,\boldsymbol{\mu})$ with equal weights
- Encoder 1, Decoder 1, and Projector 1 are trained



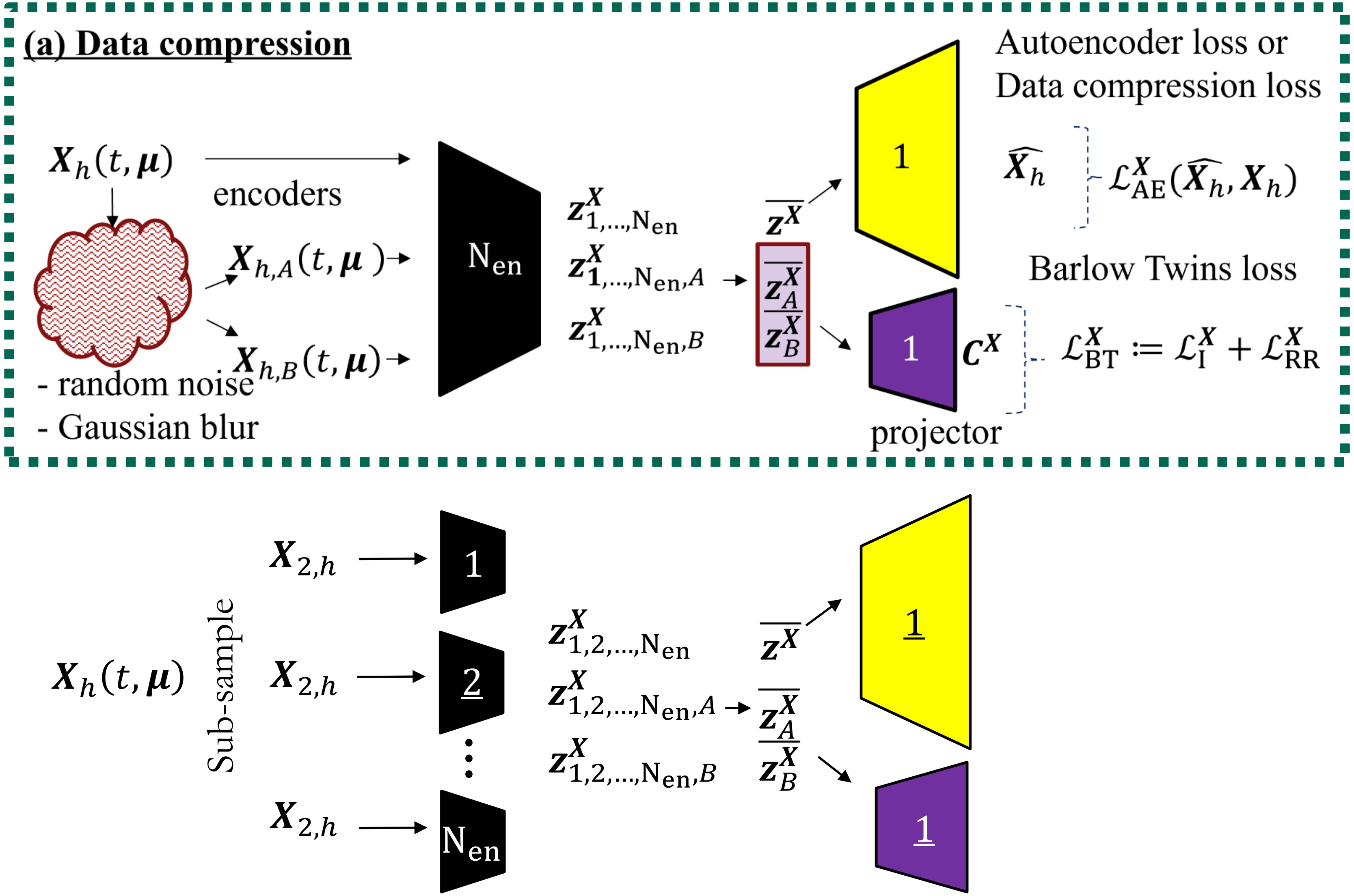


Methodology - continued

Boosting in-a-nut-shell:

We train each encoder <u>sequentially</u>

Encoder 2
- all other encoders are frozen
- $\boldsymbol{X}_{2,h}$ is sub-sampled from $\boldsymbol{X}_h(t,\boldsymbol{\mu})$ with <u>more weights toward samples that Encoder 1 fails to capture, i.e., higher loss values</u>
- Encoder 2, Decoder 1, and Projector 1 are trained

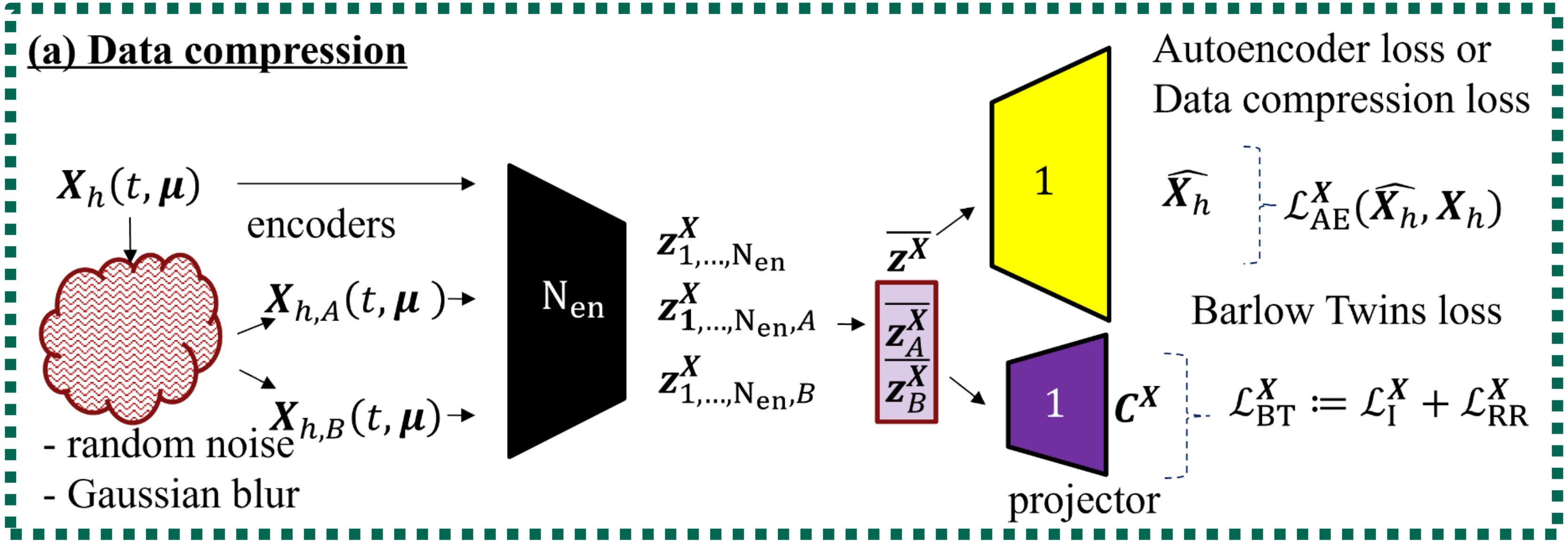


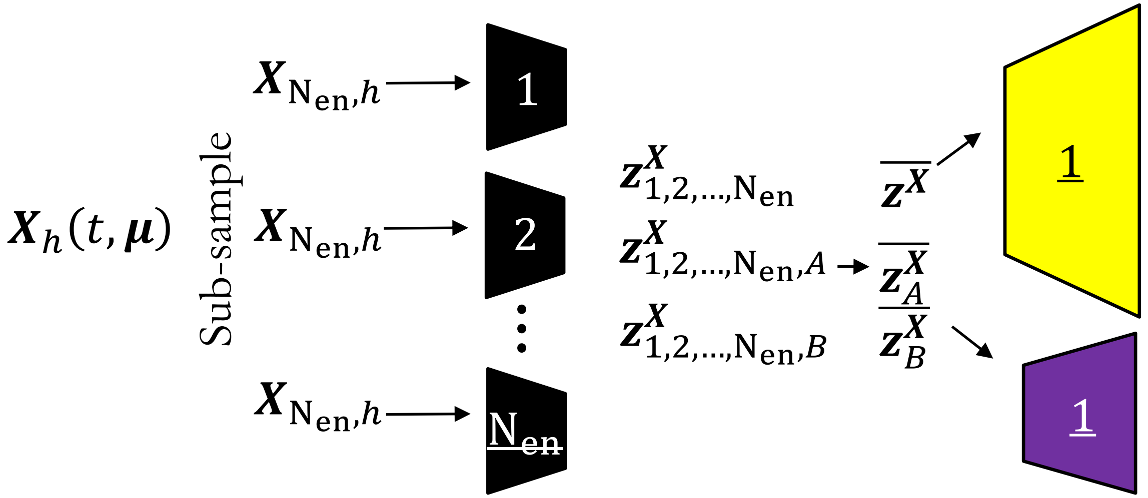
Methodology - continued

Boosting in-a-nut-shell:

We train each encoder sequentially

Encoder N_{en}
- all other encoders are frozen
- $\boldsymbol{X}_{2,h}$ is sub-sampled from $\boldsymbol{X}_h(t,\boldsymbol{\mu})$ with more weights toward samples that Encoder **1, …, $\mathbf{N}_{\mathrm{en}}$** fails to capture, i.e., higher loss values
- Encoder N_{en}, Decoder 1, and Projector 1 are trained

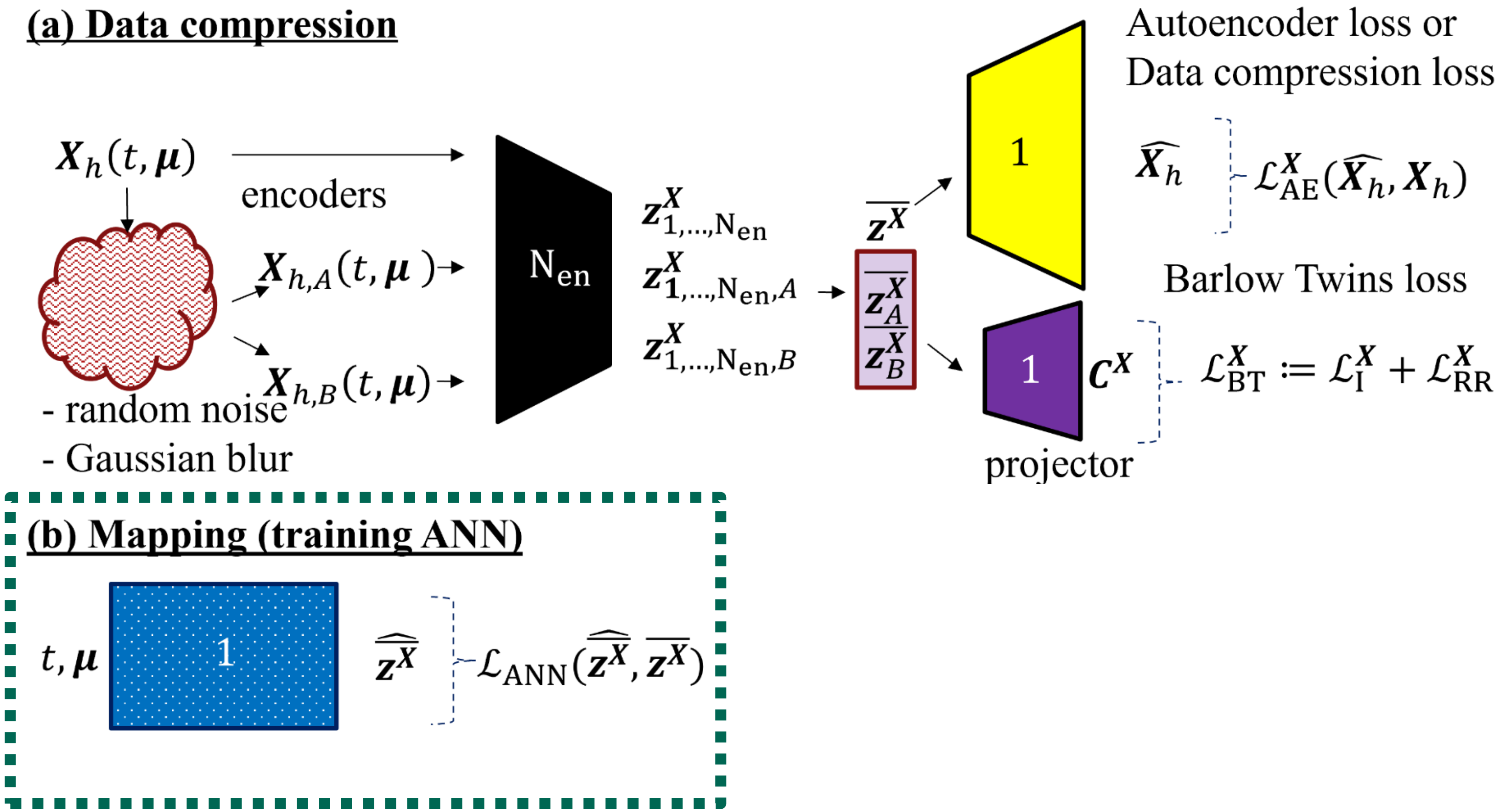




Methodology - continued

We then map our parameters to reduced manifolds using ANN.

*We note that we could use other regressors such as GP or RBF.



(a) Data compression

$\boldsymbol{X}_h(t,\boldsymbol{\mu})$

encoders

$\boldsymbol{X}_{h,A}(t,\boldsymbol{\mu})$

$\boldsymbol{X}_{h,B}(t,\boldsymbol{\mu})$

- random noise
- Gaussian blur

$\mathrm{N_{en}}$

$\boldsymbol{z}^{\boldsymbol{X}}_{1,\ldots,\mathrm{N_{en}}}$

$\boldsymbol{z}^{\boldsymbol{X}}_{1,\ldots,\mathrm{N_{en}},A}$

$\boldsymbol{z}^{\boldsymbol{X}}_{1,\ldots,\mathrm{N_{en}},B}$

$\overline{\boldsymbol{z}^{\boldsymbol{X}}}$

$\overline{\boldsymbol{z}^{\boldsymbol{X}}_A}$

$\overline{\boldsymbol{z}^{\boldsymbol{X}}_B}$

1

Autoencoder loss or Data compression loss

$\widehat{\boldsymbol{X}_h}$ $\mathcal{L}^{\boldsymbol{X}}_{\mathrm{AE}}(\widehat{\boldsymbol{X}_h}, \boldsymbol{X}_h)$

Barlow Twins loss

1

$\boldsymbol{C}^{\boldsymbol{X}}$ $\mathcal{L}^{\boldsymbol{X}}_{\mathrm{BT}} := \mathcal{L}^{\boldsymbol{X}}_{\mathrm{I}} + \mathcal{L}^{\boldsymbol{X}}_{\mathrm{RR}}$

projector

(b) Mapping (training ANN)

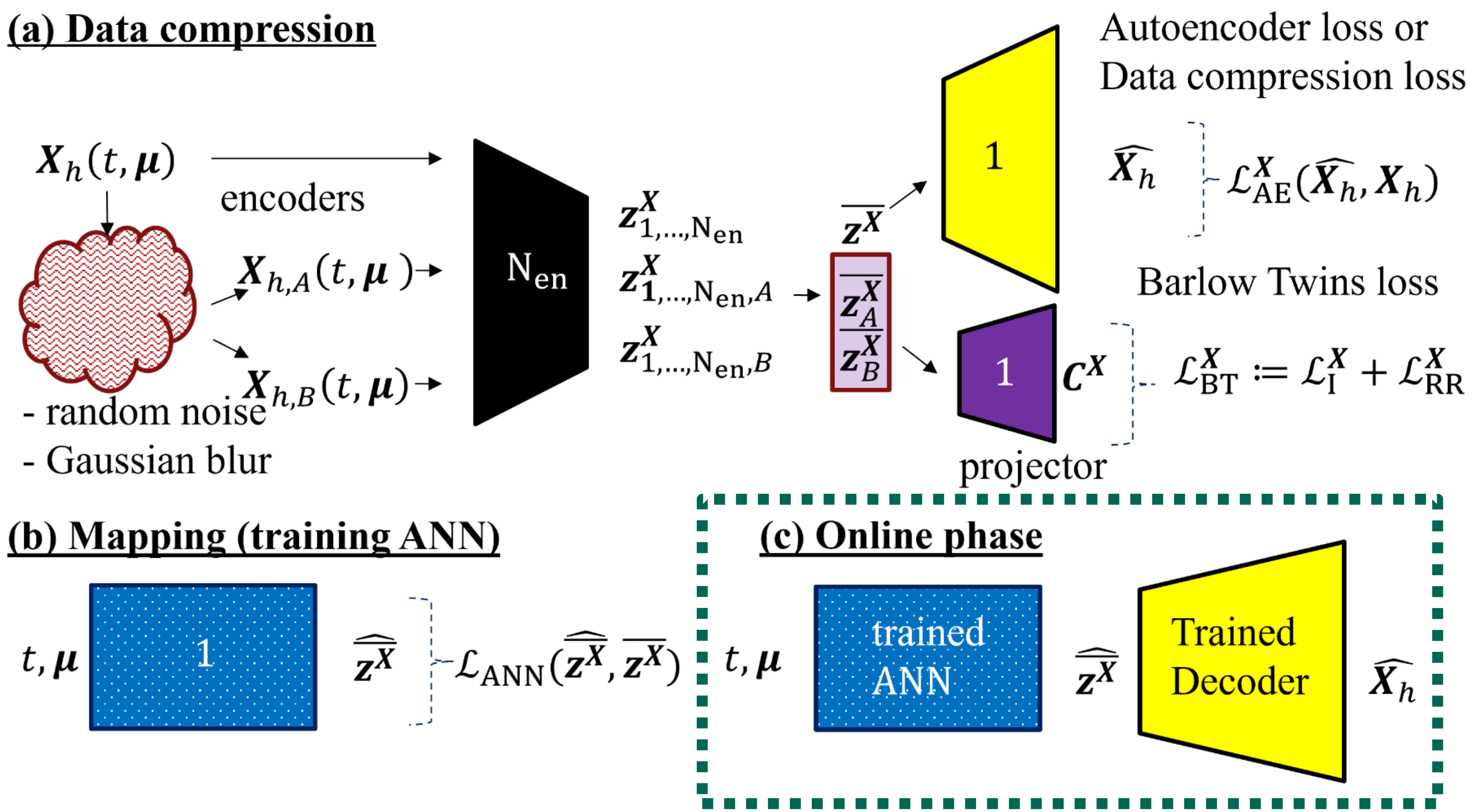
$t, \boldsymbol{\mu}$

1

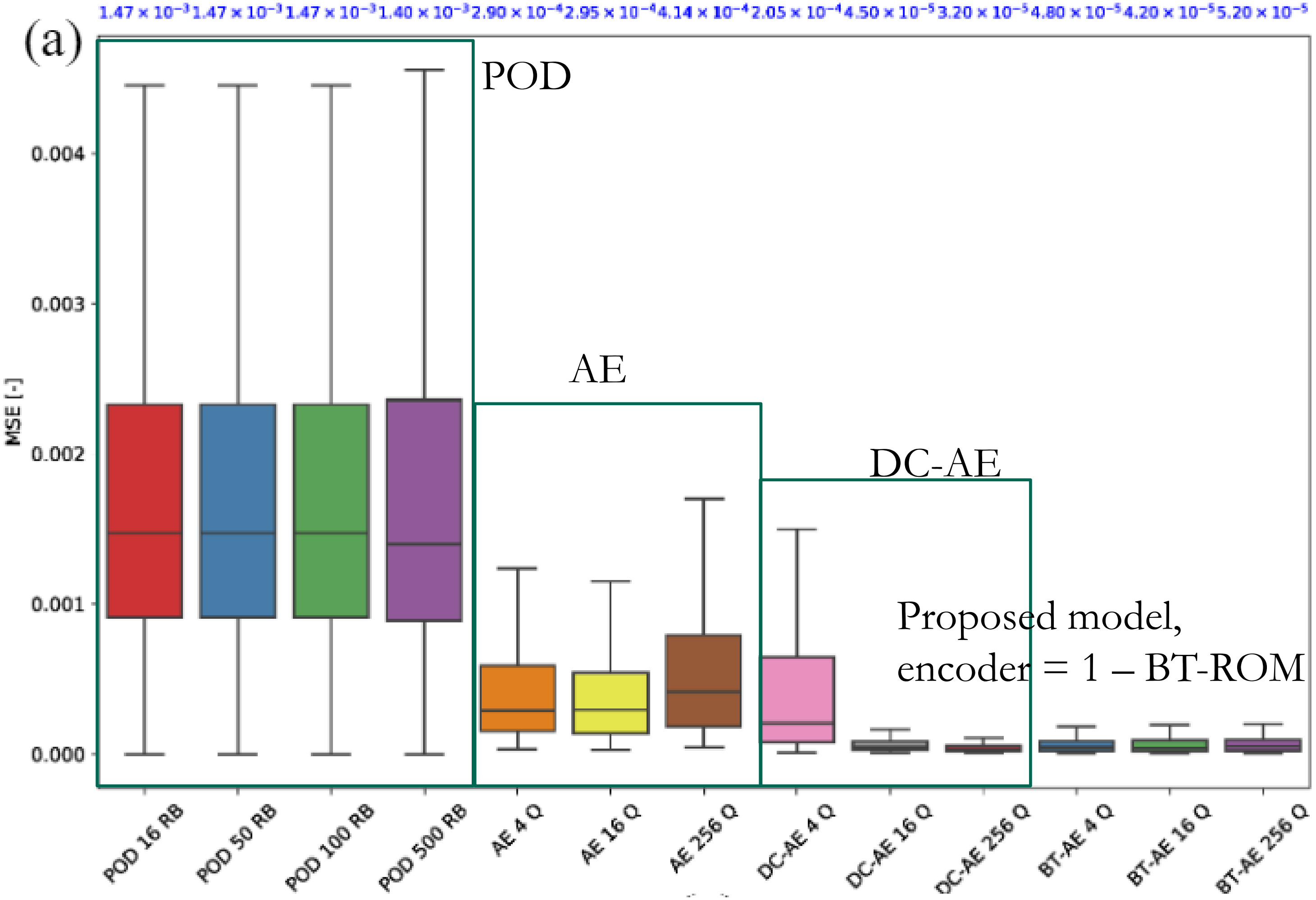
$\widehat{\boldsymbol{z}^{\boldsymbol{X}}}$ $\mathcal{L}_{\mathrm{ANN}}(\widehat{\boldsymbol{z}^{\boldsymbol{X}}}, \overline{\boldsymbol{z}^{\boldsymbol{X}}})$

Methodology - continued

During the online or prediction phase, we approximate our quantities of interest through the **trained ANN** and **trained decoder**.



Do we achieve better reduced manifolds?

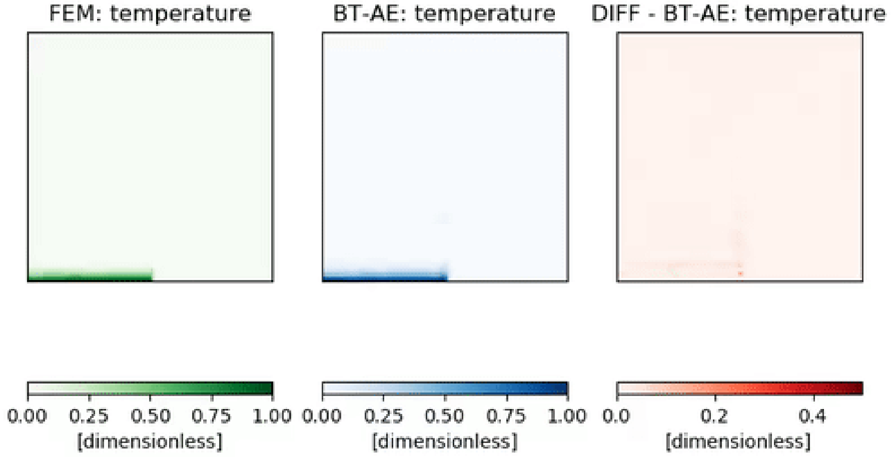


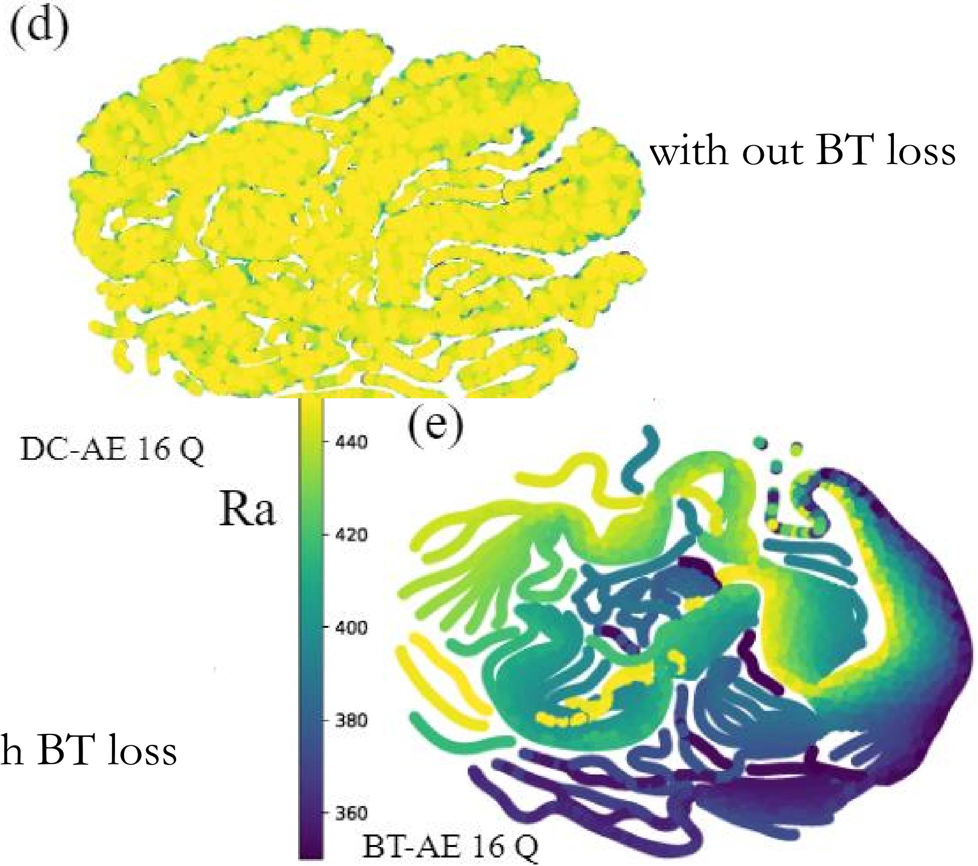
POD

AE

DC-AE

Proposed model,
encoder = 1 – BT-ROM





with out BT loss

DC-AE 16 Q

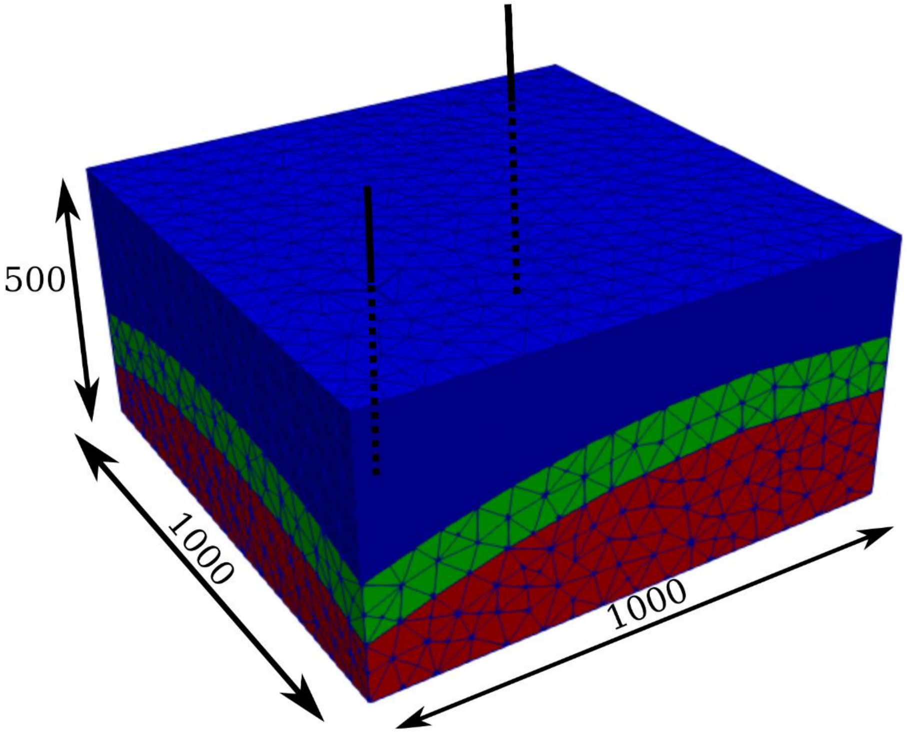
with BT loss

BT-AE 16 Q

Kadeethum et al. (2022, Advances in Water Resources)
Kadeethum et al. (2022, Scientific Report , accepted)

Results: multiphase flow in porous media

- Multiphase flow in porous media
- One producer, one injector



Permeability of the last layer is the parameters

$$k = [1.0 \times 10^{-14}, 1.0 \times 10^{-12}]$$

Training: 160
Validation: 5% of Training, i.e., Training set is less than 160
Testing: 40
Degrees of freedom: 133036

FOM: Each case takes 55 mins through 128 AMD EPYC 7452

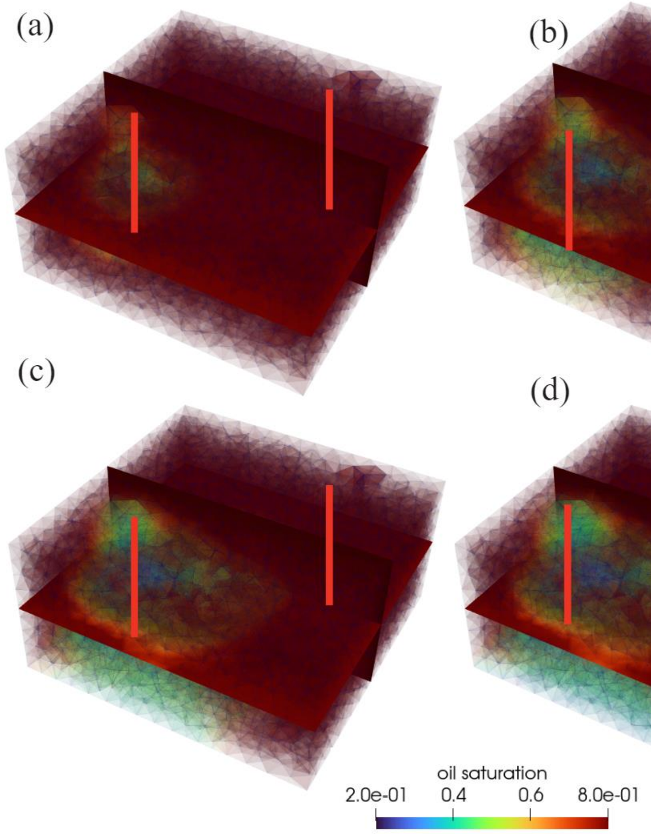
ROM: Each prediction takes 0.001 secs, training takes 250 mins through 1 NVIDIA Quadro RTX 8000

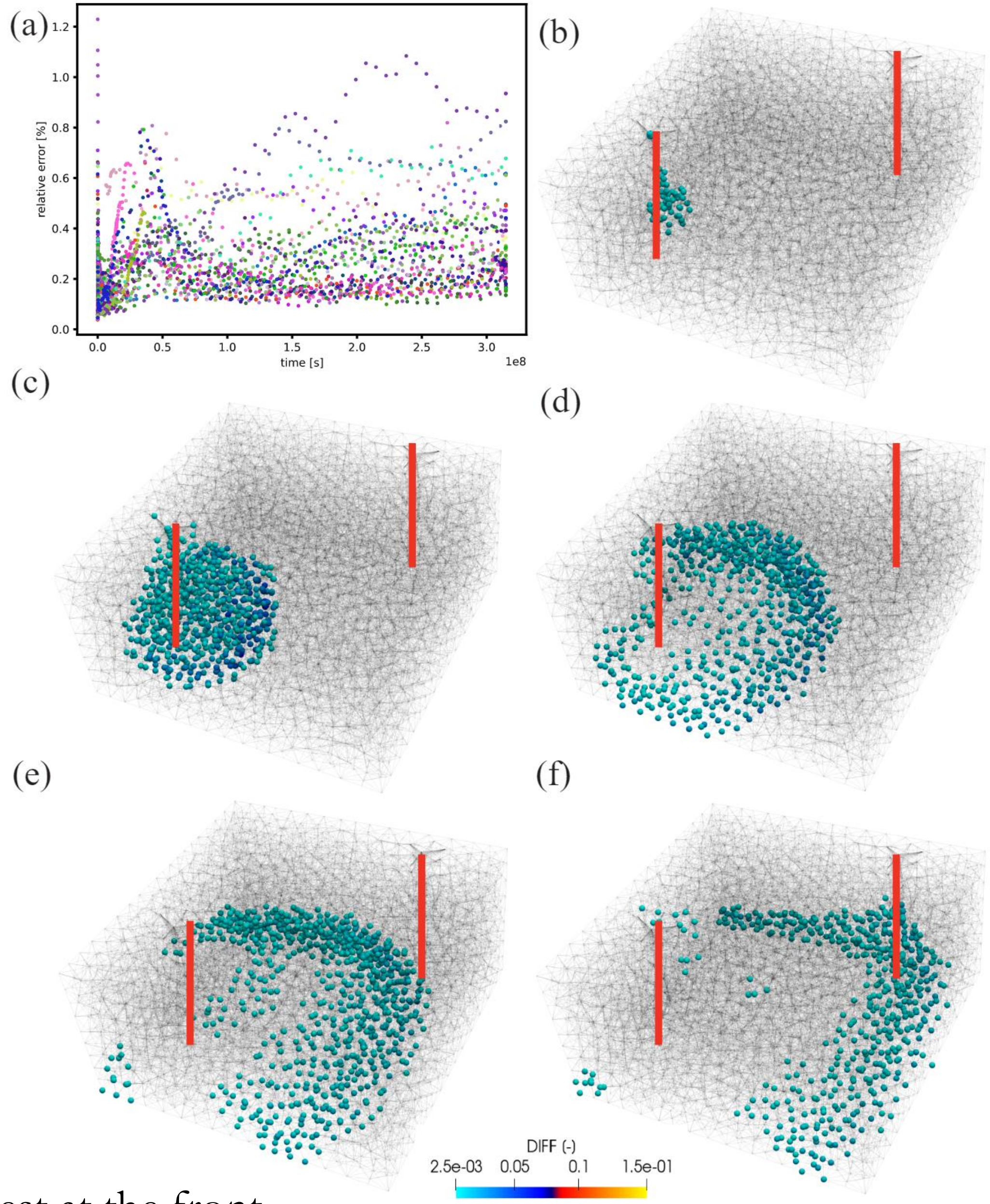
*** ROM requires an overhead of data generation + training

Results: multiphase flow in porous media - continued

Maximum relative error is 1.2 %
Relative error = r.m.s.e./||FOM||

(a) $t = 4.1 \times 10^7$, (b) $t = 1.3 \times 10^8$,
(c) $t = 2.4 \times 10^8$, (d) $t = 3.2 \times 10^8$ secs

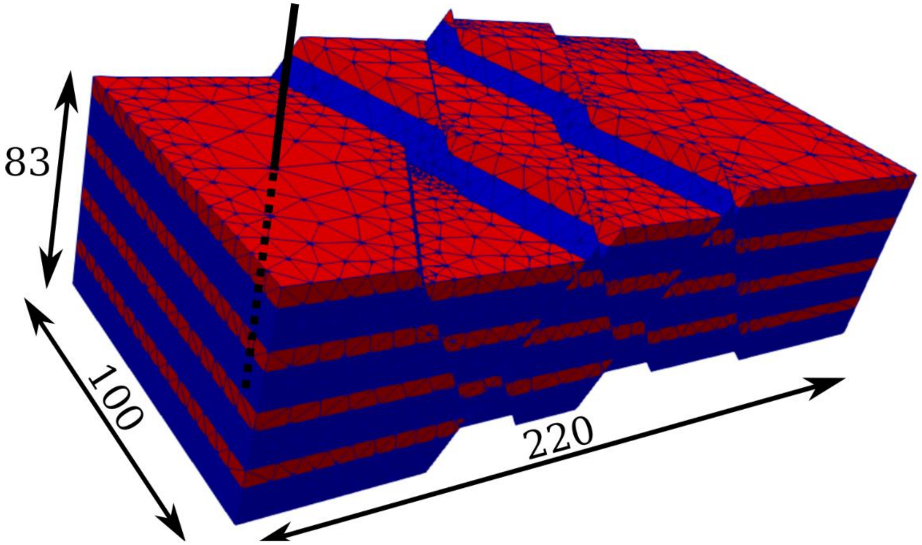




Errors are at the highest at the front

Results: multiphase flow in porous media - continued

- Multiphase flow in porous media
- One injector
- Many faults



Permeabilities of each layer are the parameters

$$k_1 = [1.0 \times 10^{-14}, 1.0 \times 10^{-12}]$$
$$k_2 = [1.0 \times 10^{-14}, 1.0 \times 10^{-12}]$$

Training: 400
Validation: 5% of Training, i.e., Training set is less than 400
Testing: 20
Degrees of freedom: 88164

FOM: Each case takes 36 mins through 128 AMD EPYC 7452

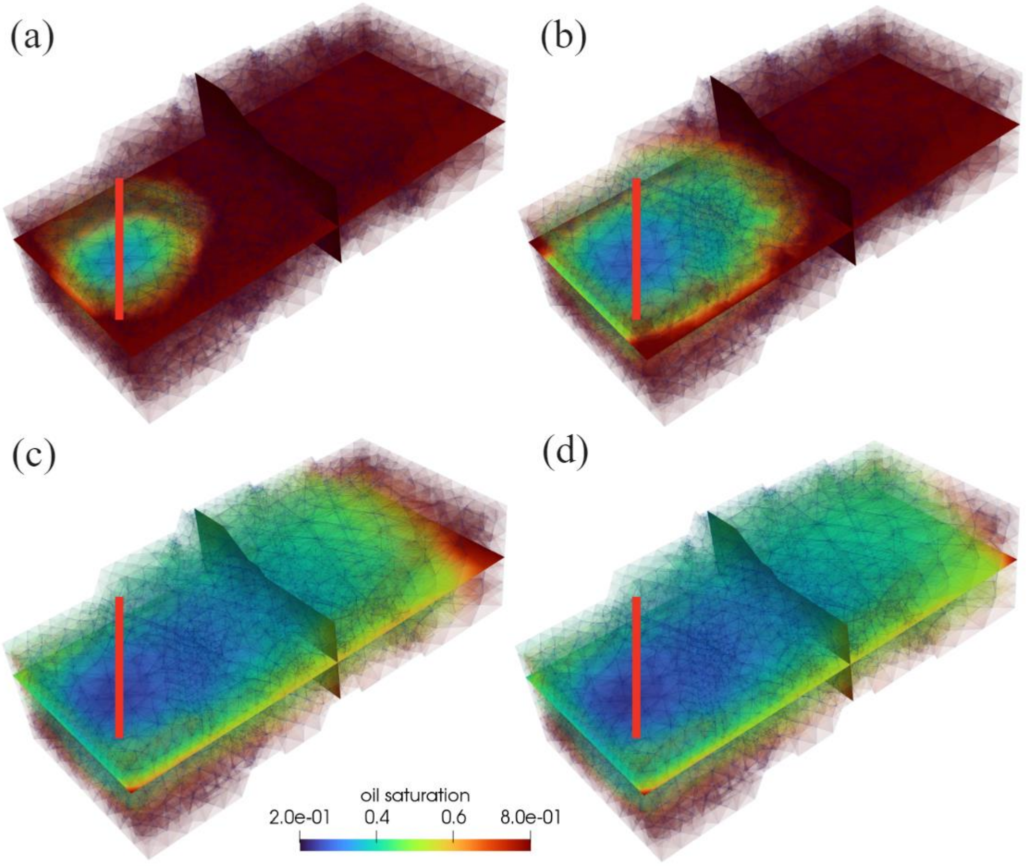
ROM: Each prediction takes 0.001 secs, training takes 220 mins through 1 NVIDIA Quadro RTX 8000

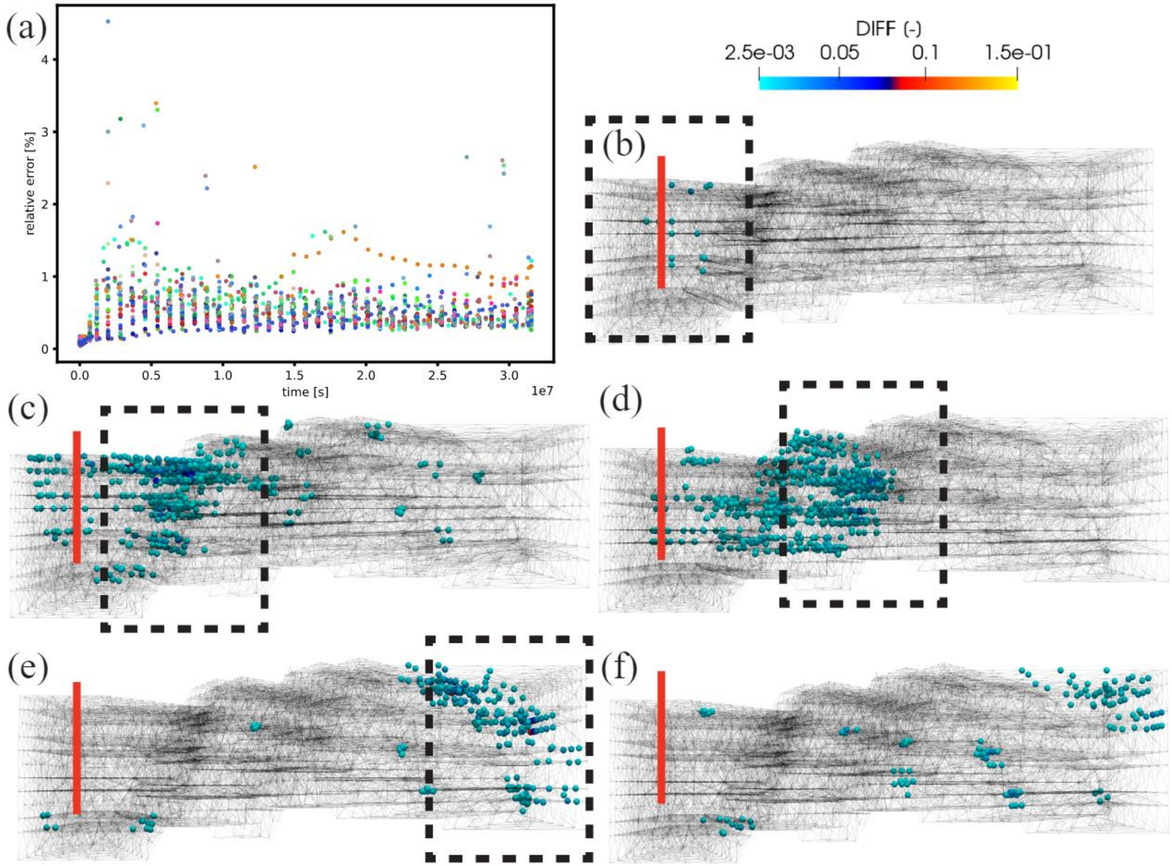
*** ROM requires an overhead of data generation + training

Results: multiphase flow in porous media - continued

(a) $t = 2.0 \times 10^6$, (b) $t = 6.3 \times 10^6$,
(c) $t = 2.5 \times 10^7$, (d) $t = 3.1 \times 10^7$ secs

Maximum relative error is 4.4 %
Relative error = r.m.s.e./||FOM||





Errors are at the highest at the front

Results: Geologic carbon storage (GCS)

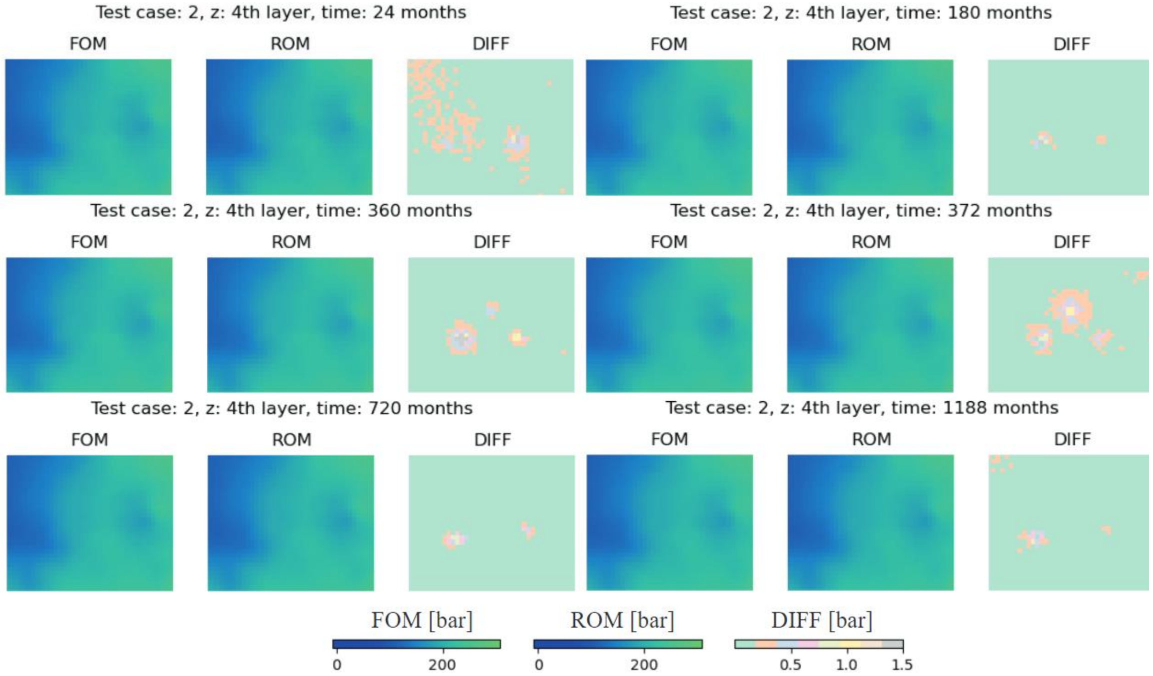
- CO_2 injection into two injection wells
- One production well
- 15 layered 3-D system with homogeneous properties per each layer

ML Parameters

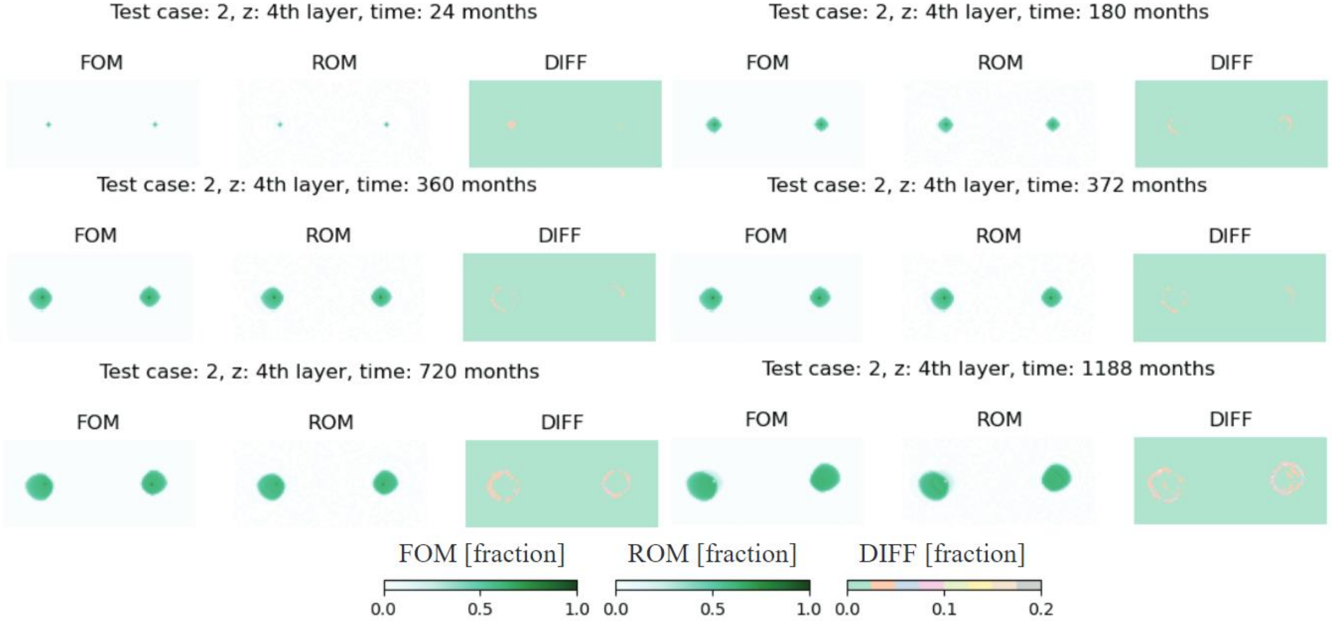
Training: 36 cases
Testing: 4 cases

Parameters	Range
characteristic (P)	[10, 50, 90]
injection ratio (i)	[(0.4, 0.6), (0.5, 0.5), (0.6, 0.4)]
timestamp (t)	[0, 1, 2, …, 1187, 1188, 1189]

Pressure

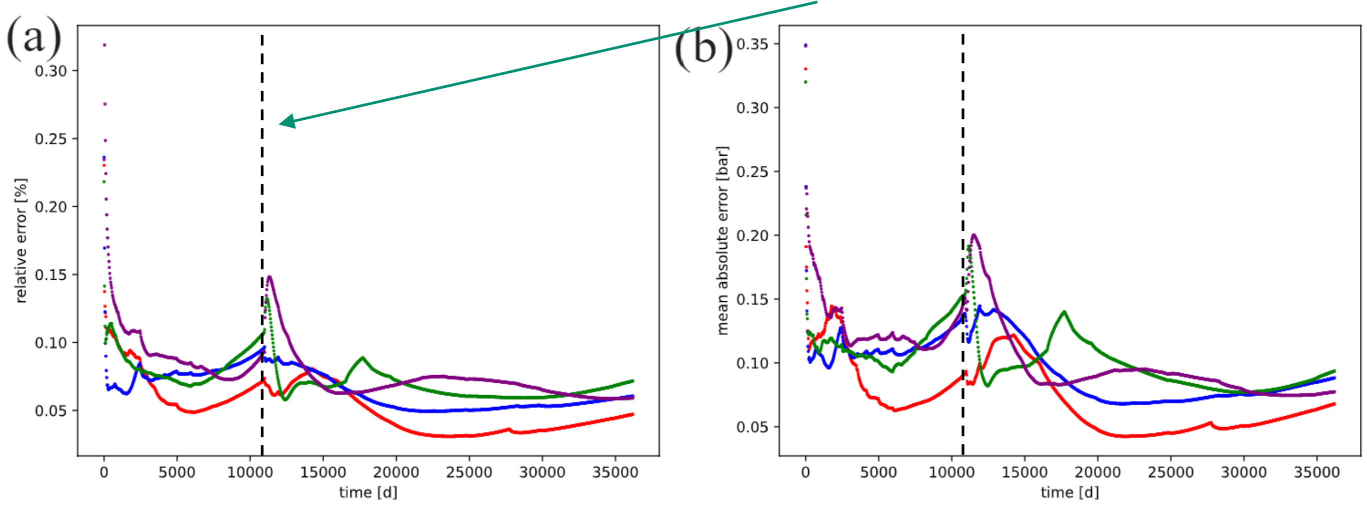


CO_2 Saturation

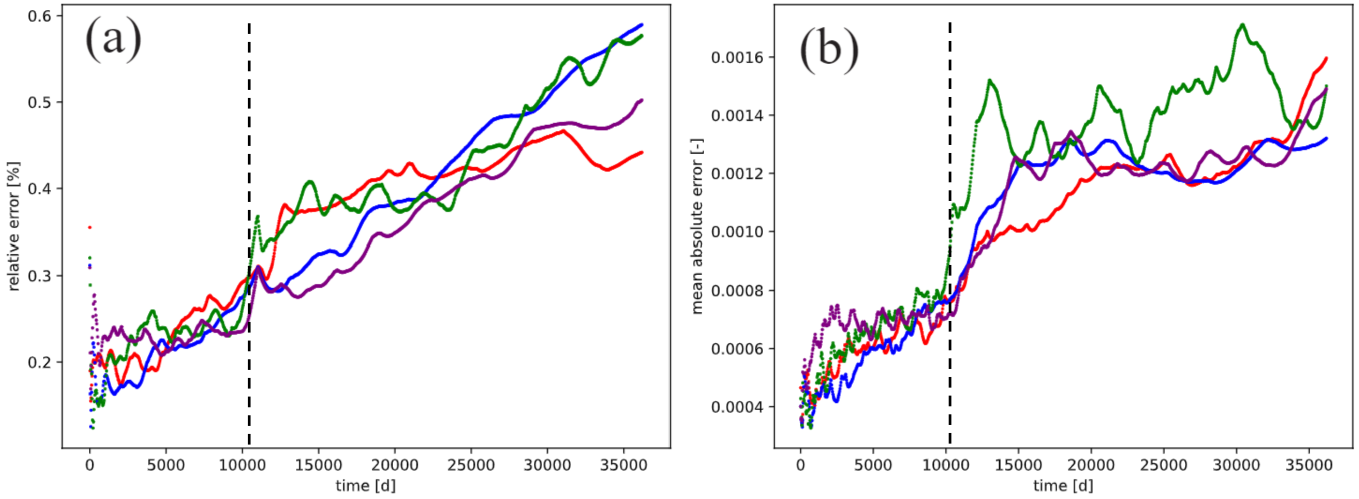


Results: Geologic carbon storage (GCS) - continue

Pressure: Errors are high when CO_2 injection has stopped



CO_2 Saturation: Errors propagate through time



Since we subsampled our FOM, we would not compare the wall time for this.

ROM: Prediction at each timestamp takes ~0.001 secs, training takes ~120 mins through 1 NVIDIA Quadro RTX 8000

*** ROM requires an overhead of data generation + training

Relative error = RMSE / ||FOM||

Conclusions

1. A ROM framework that works in an optimal way for both **linear** and **nonlinear** manifolds

2. A ROM framework that can be applied for both **structured** and **unstructured** meshes

3. A ROM framework that can handle **data imbalanced** problems

4. An uncertainty-aware BT-ROM is in progress to achieve uncertainty quantification (IEEE Access 2022, in review)