



Sandia
National
Laboratories

Test Cases for Combustion Chemistry in a Isobaric Reactor

Simone Venturi, Tiernan Casey

Extreme-Scale Data Science & Analytics (8739)

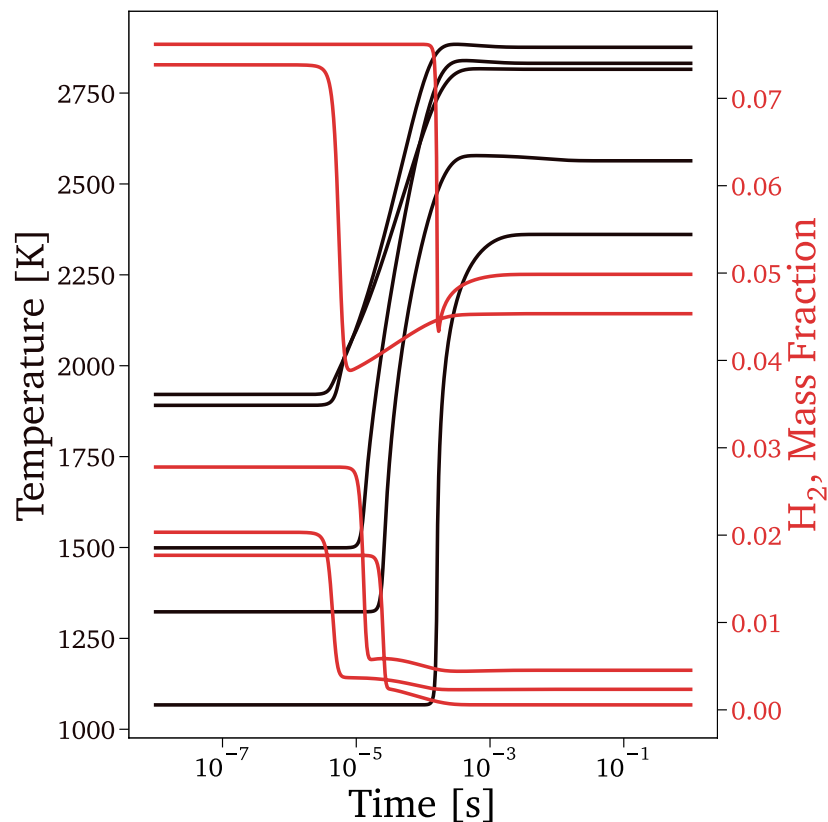


Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

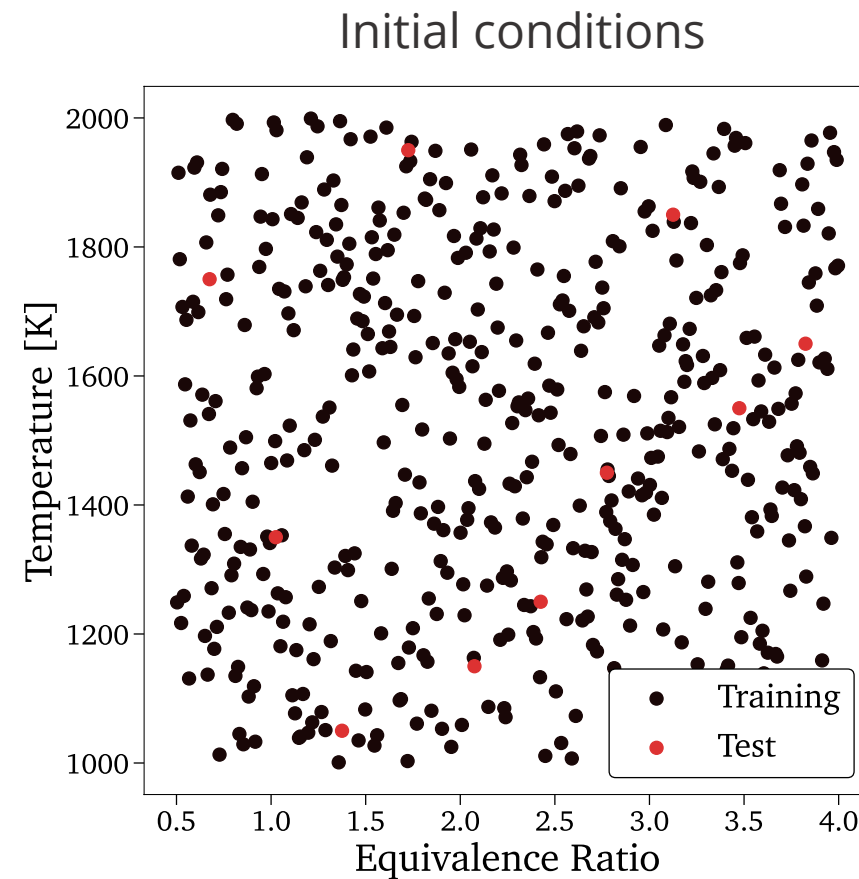
A Combustion Chemistry in Isobaric Reactor Test Case



Isobaric 0-D Reactor (Hydrogen-air), 20 state variables
(i.e., temperature and 19 species mass fractions)



Some training scenarios

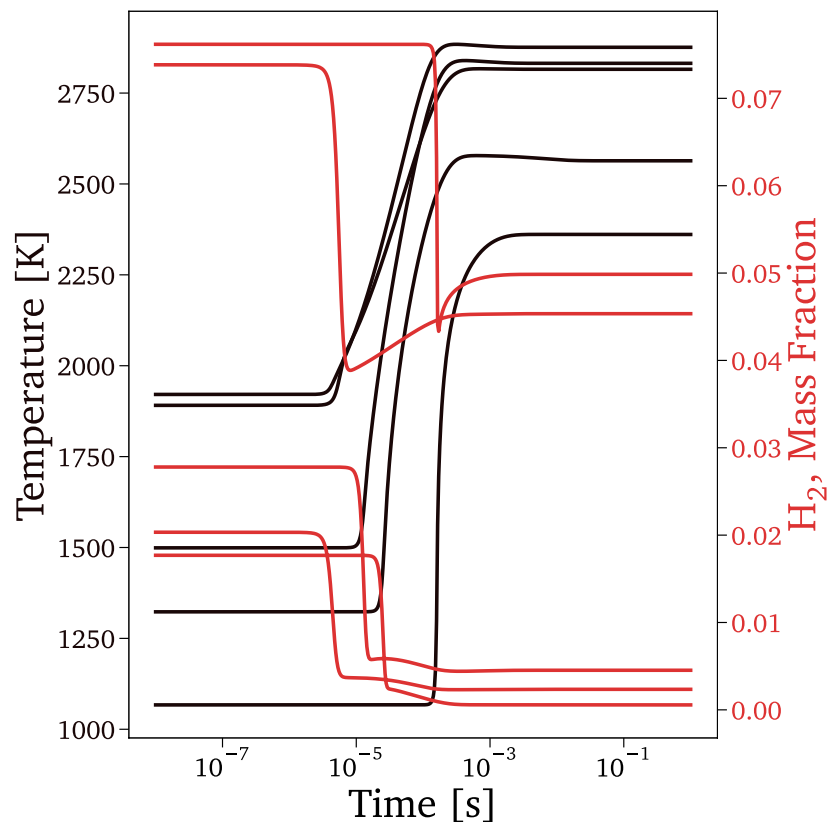


The physical system is implemented in
`$WORKSPACE_PATH/ROMNet/romnet/romnet/pinn/system/0DReact.py`

A Combustion Chemistry in Isobaric Reactor Test Case

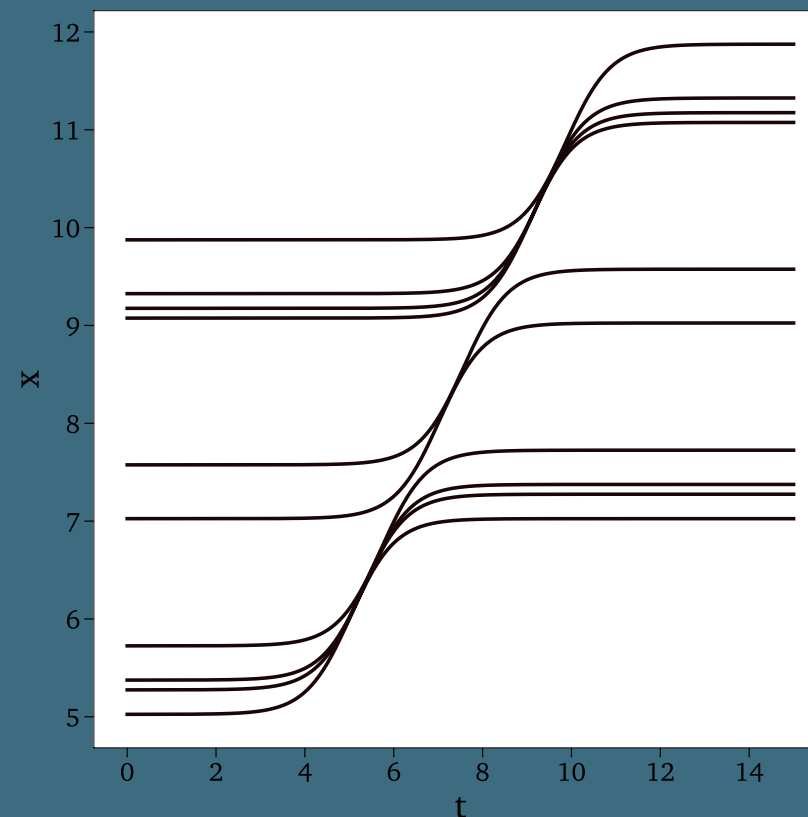


Isobaric 0-D Reactor (Hydrogen-air), 20 state variables
(i.e., temperature and 19 species mass fractions)



Some training scenarios

Note:
Similitudes with
translating hyperbolic tangent



The physical system is implemented in
\$WORKSPACE_PATH/ROMNet/romnet/romnet/pinn/system/0DReact.py

A Combustion Chemistry in Isobaric Reactor Test Case



Run python scrip:

`$WORKSPACE_PATH/ROMNet/romnet/scripts/generating_data/0DReactor/Generate_Data_1_Isobaric.py`
for generating simulation data

Note: The script needs to be run twice, the second time for generating test data

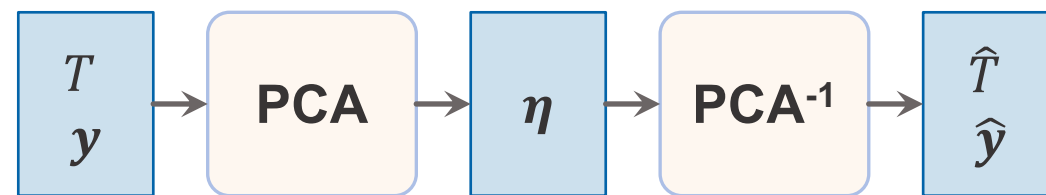
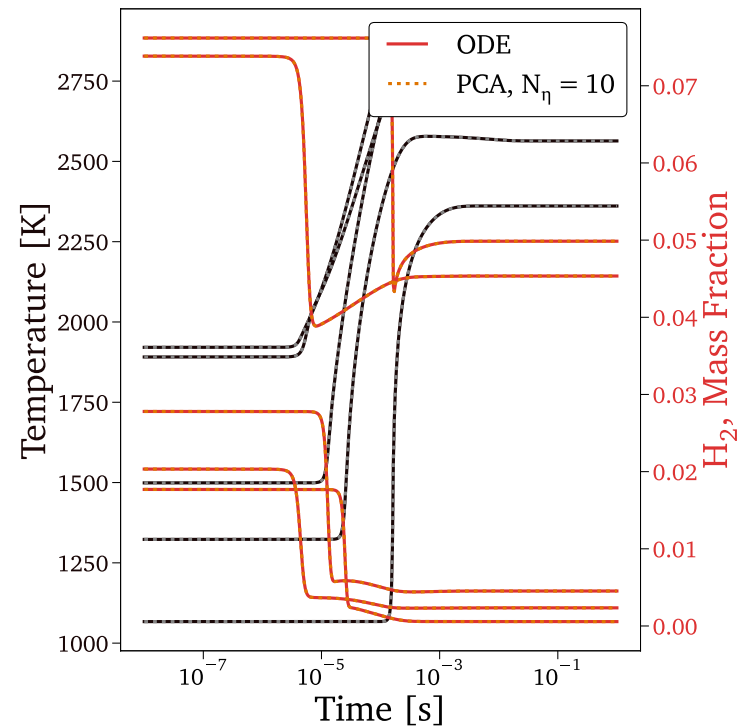
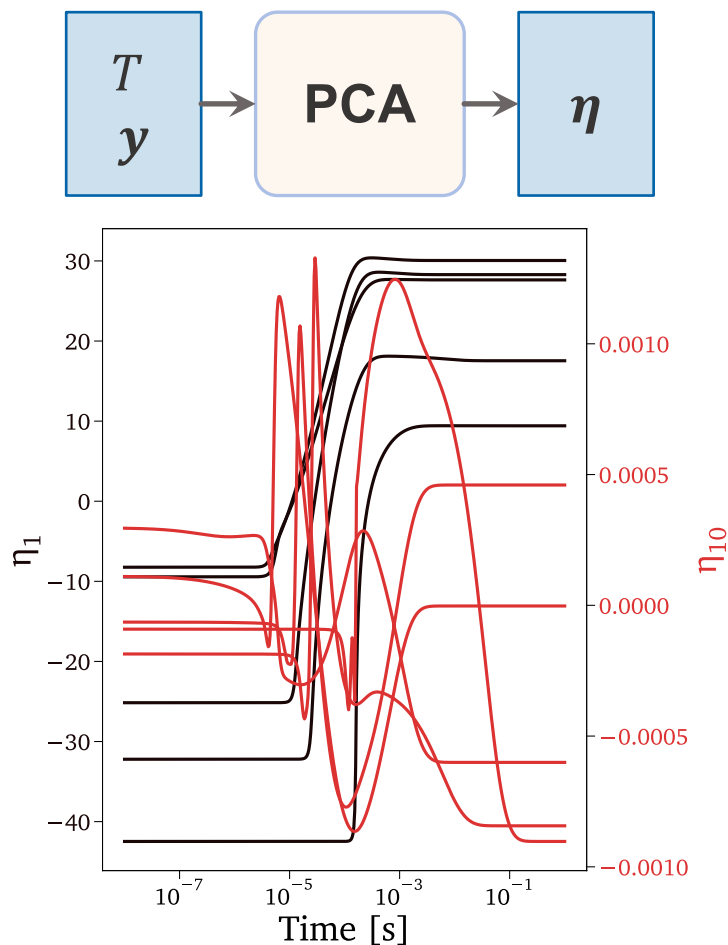
Run Jupyter Notebook:

`$WORKSPACE_PATH/ROMNet/romnet/scripts/generating_data/0DReactor/Generate_Data_3_Isobaric.ipynb`
for generating training and test data

A Combustion Chemistry in Isobaric Reactor Test Case



Employed PCA for reducing the dimensionality of the state space



10 principal components (η) are sufficient for good accuracy

A Combustion Chemistry in Isobaric Reactor Test Case



Run python scrip:

`$WORKSPACE_PATH/ROMNet/romnet/scripts/generating_data/0DReactor/Generate_Data_2_Isobaric.py`
for generating PCA simulation data

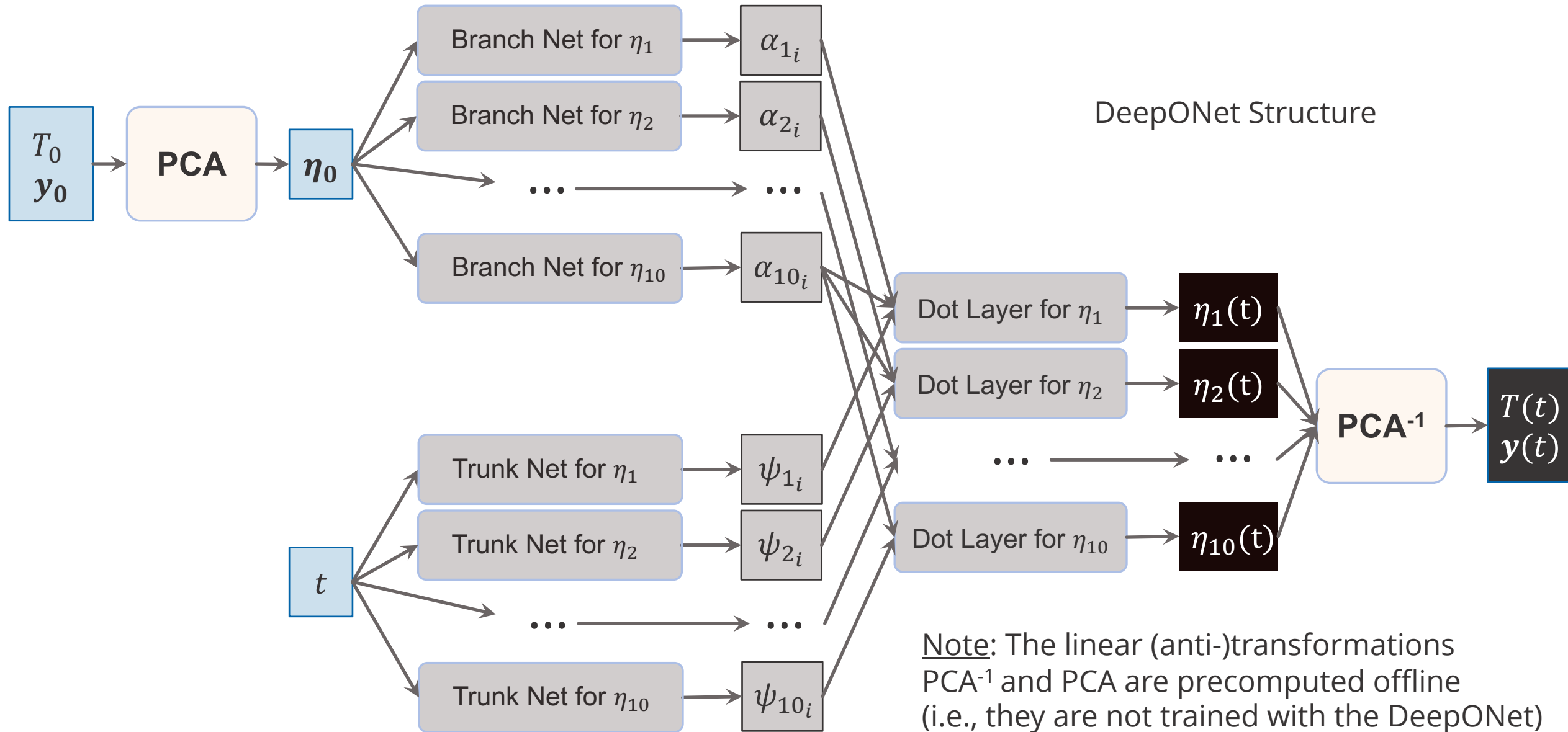
Note: The script needs to be run twice, the second time for generating test data

Run Jupyter Notebook:

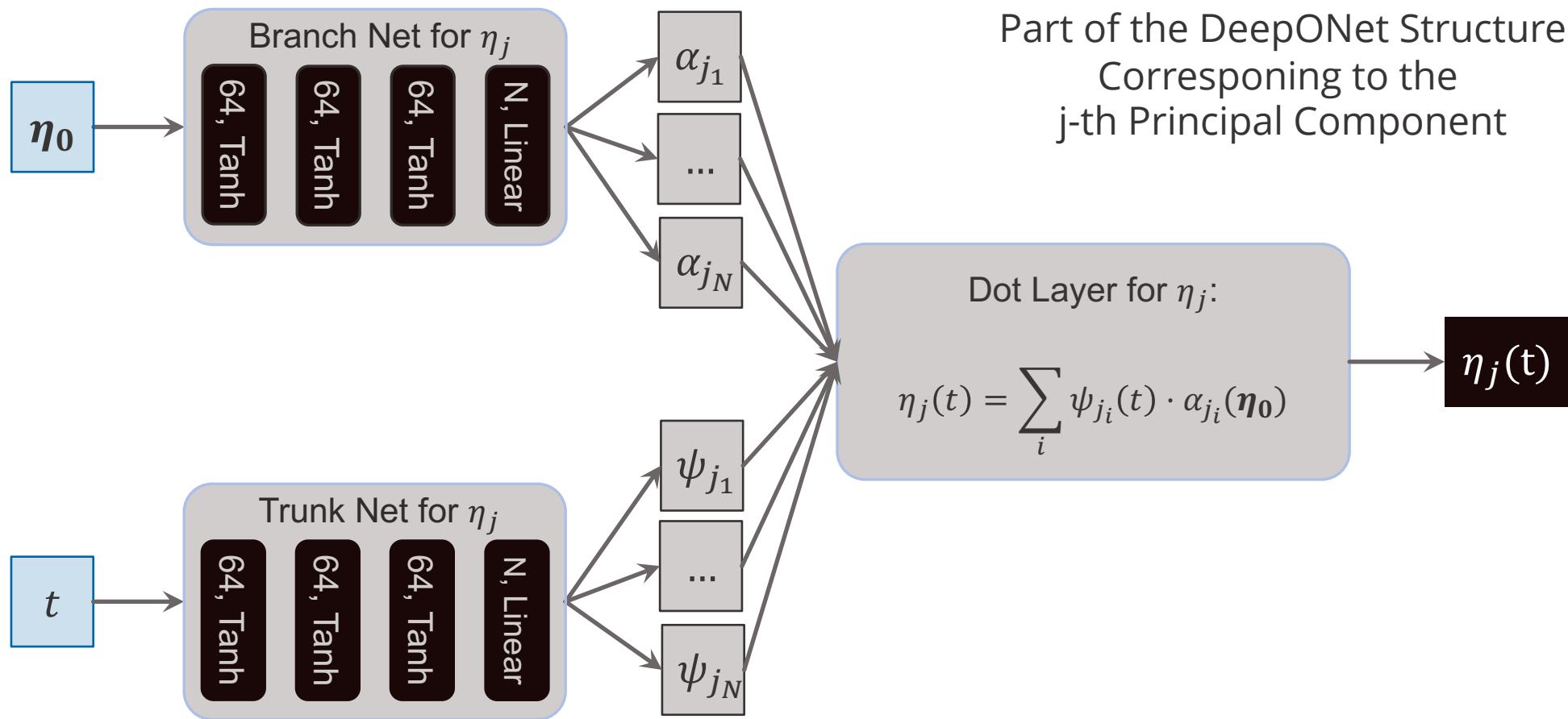
`$WORKSPACE_PATH/ROMNet/romnet/scripts/generating_data/0DReactor/Generate_Data_4_Isobaric.ipynb`
for generating PCA training and test data

Test Case 9

A Combustion Chemistry in Isobaric Reactor Test Case



A Combustion Chemistry in Isobaric Reactor Test Case



After being trained (even with large number of data and large number of neurons, N),
the DeepONet generates highly oscillatory predictions

A Combustion Chemistry in Isobaric Reactor Test Case



Test Case 9: Data-driven deep operator network (DeepONet) for predicting Principal Components

9.1. Copy \$WORKSPACE_PATH/ROMNet/romnet/input/ODReact/DeepONet/TestCase9/ROMNet_Input.py
to \$WORKSPACE_PATH/ROMNet/romnet/input/ROMNet_Input.py

9.2. In \$WORKSPACE_PATH/ROMNet/romnet/input/ROMNet_Input.py, change:
9.2.1. "self.WORKSPACE_PATH = ..."

9.3. Move to \$WORKSPACE_PATH/ROMNet/romnet/app/

9.4. Run: "python3 ROMNet.py ../input/"

9.5. Postprocess results via: \$WORKSPACE_PATH/ROMNet/romnet/scripts/postprocessing/ODReact/DeepONet/Predict_DeepONet.ipynb

A Combustion Chemistry in Isobaric Reactor Test Case



Investigating the issue: a principal component analysis

Aggregation of training scenarios for $\eta_j(t)$, where j represents the scenario index:

$$\mathbf{H}_j = \begin{bmatrix} | & | & \cdots & | & | \\ \eta_{j1} & \eta_{j2} & \cdots & \eta_{j499} & \eta_{j500} \\ | & | & & | & | \end{bmatrix}$$

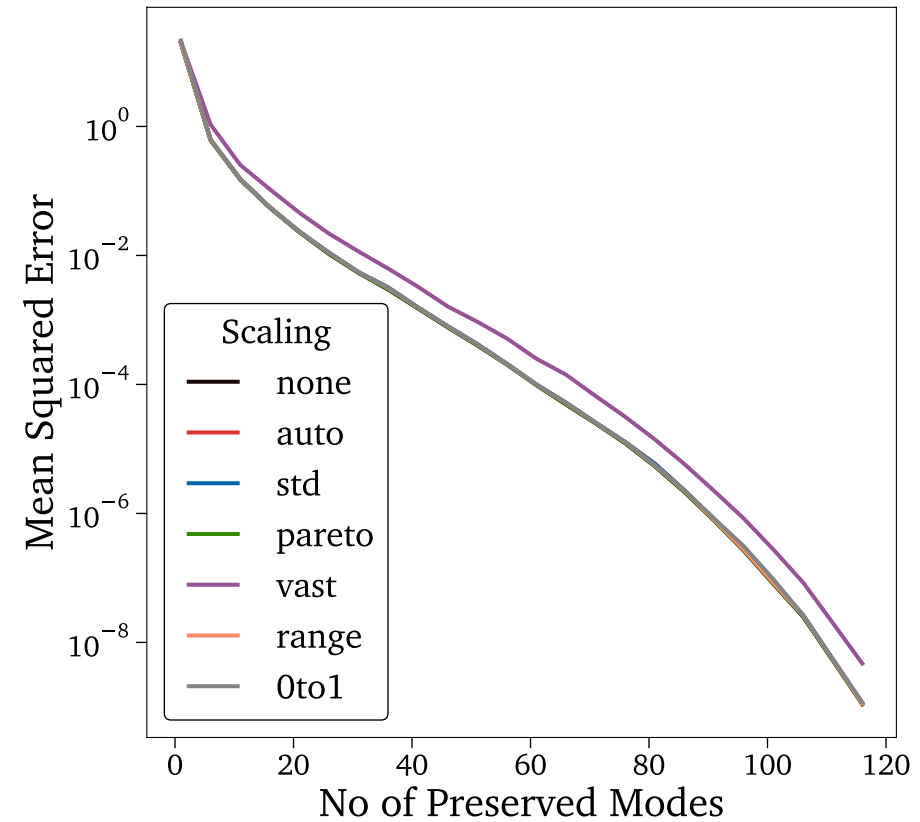
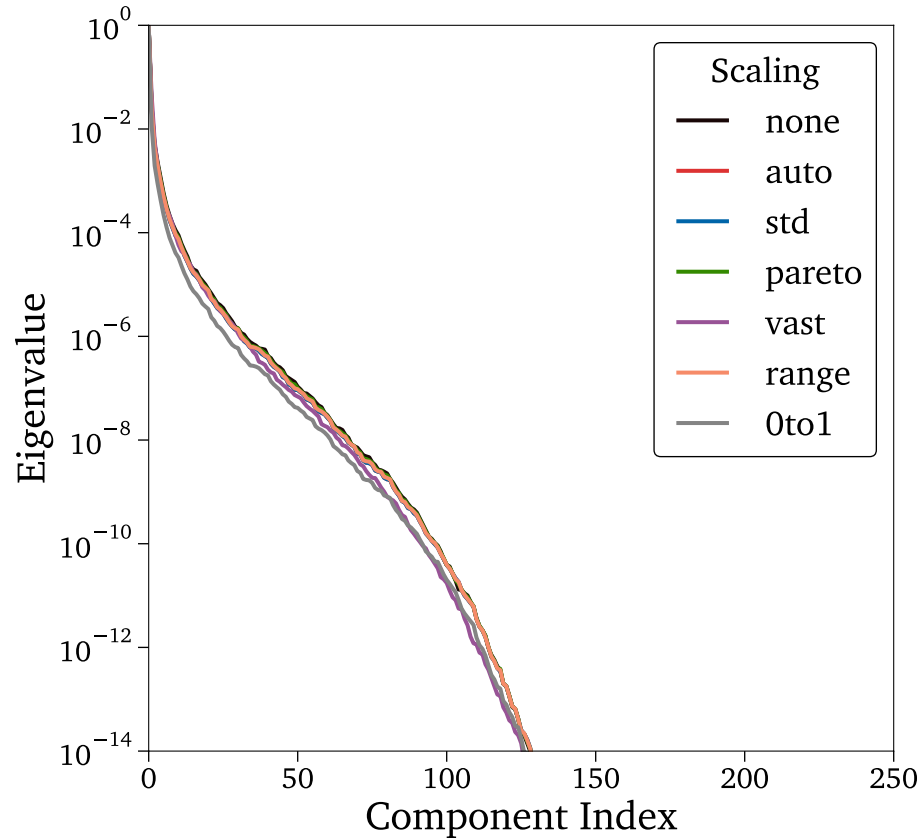
$$\dim(\mathbf{H}_j) = N_t \times N_s$$

No. time
instants No. of
scenarios

A Combustion Chemistry in Isobaric Reactor Test Case



Eigendecomposition of R_{H_j} : $\Psi_j = \frac{H_j - C_j}{D_j} A_j$ (Note: results are shown for $j = 1$ (i.e., for η_1))

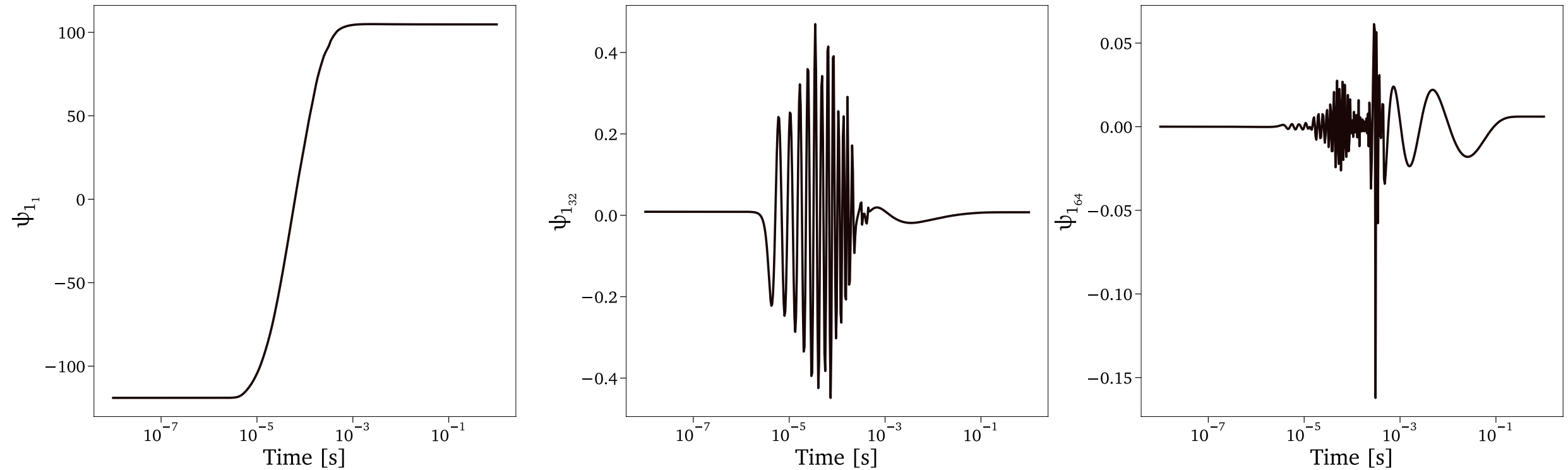


A relatively large number of modes needs to be preserved in order to predict η_j with good accuracy

A Combustion Chemistry in Isobaric Reactor Test Case



Low energy modes are highly oscillatory and hard to be learnt by the DeepONet's trunk nets



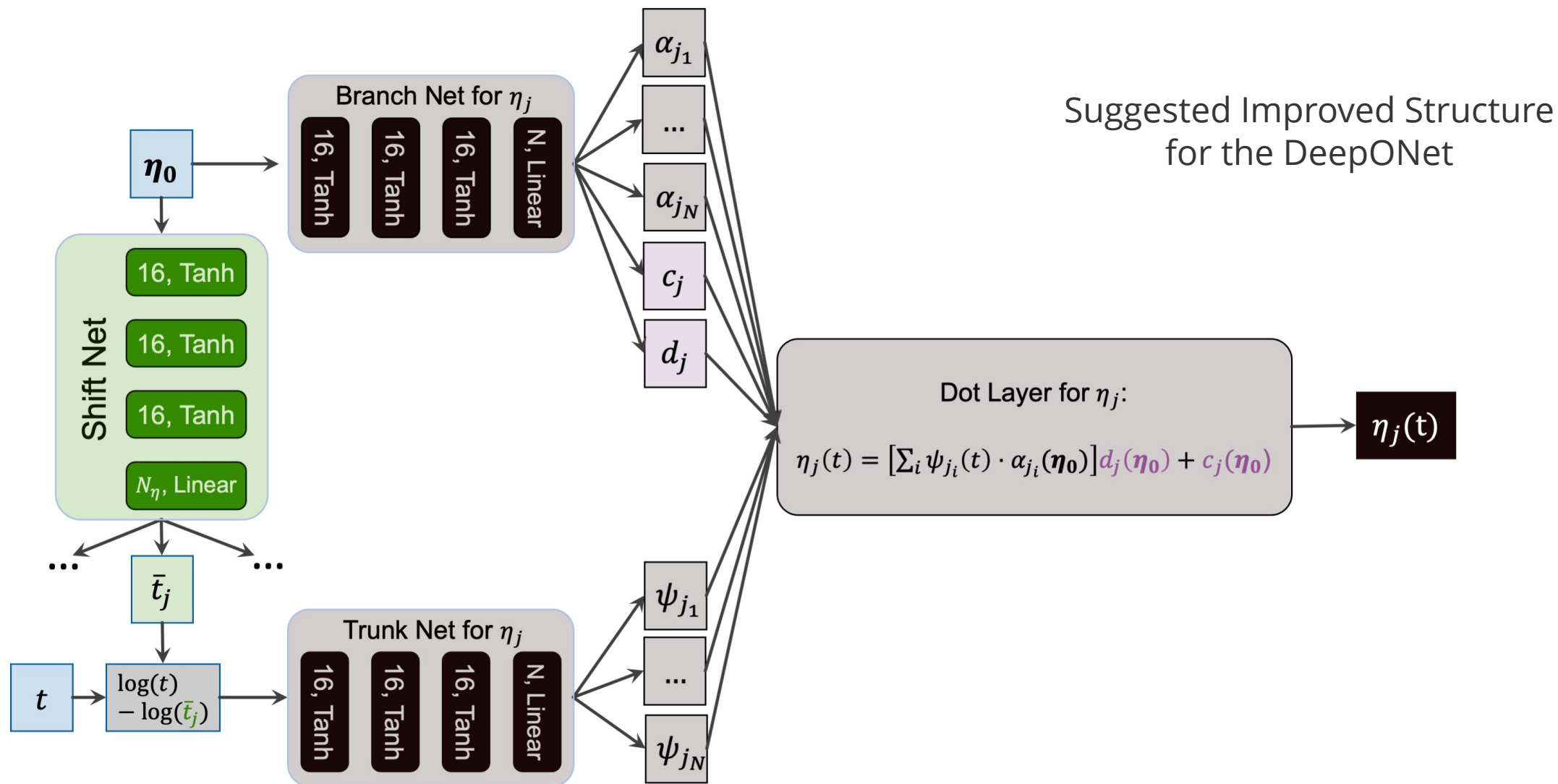
A Combustion Chemistry in Isobaric Reactor Test Case



Run Jupyter Notebook:
\$WORKSPACE_PATH/ROMNet/romnet/scripts/generating_data/0DReactor/Generate_Data_5.ipynb
for generating PCA data

Test Case 10

A Combustion Chemistry in Isobaric Reactor Test Case

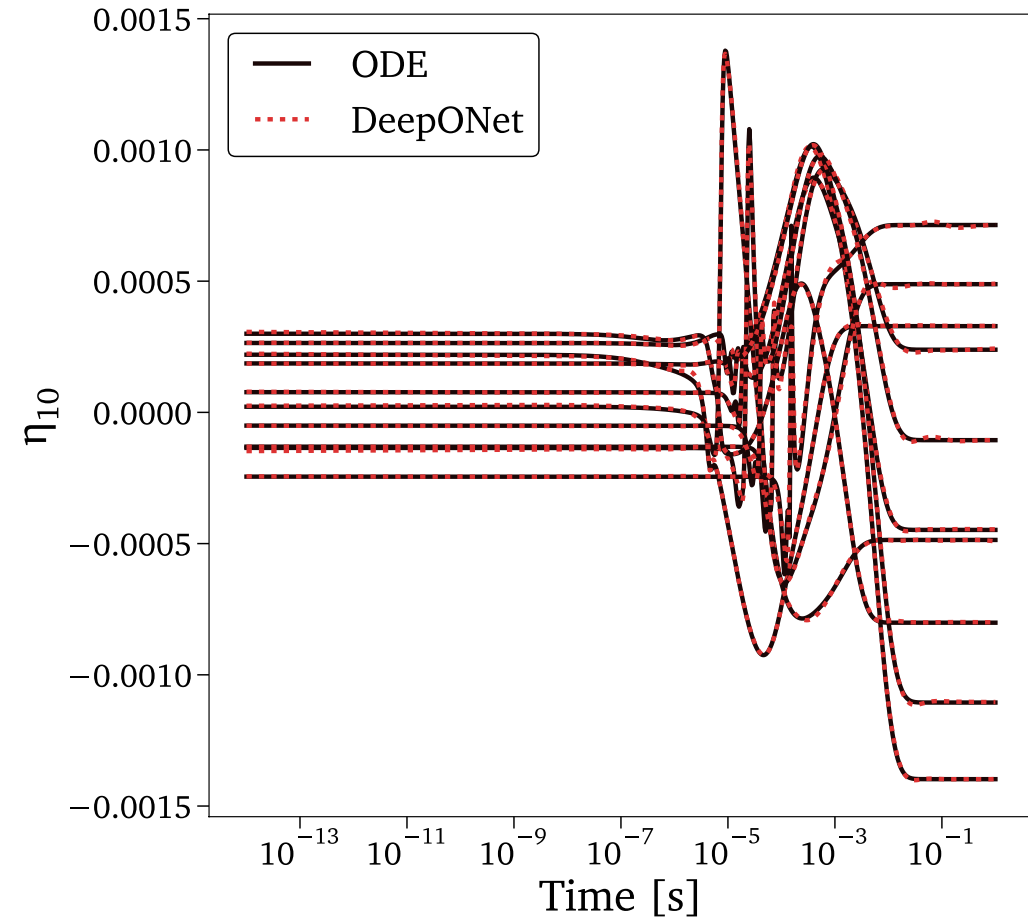
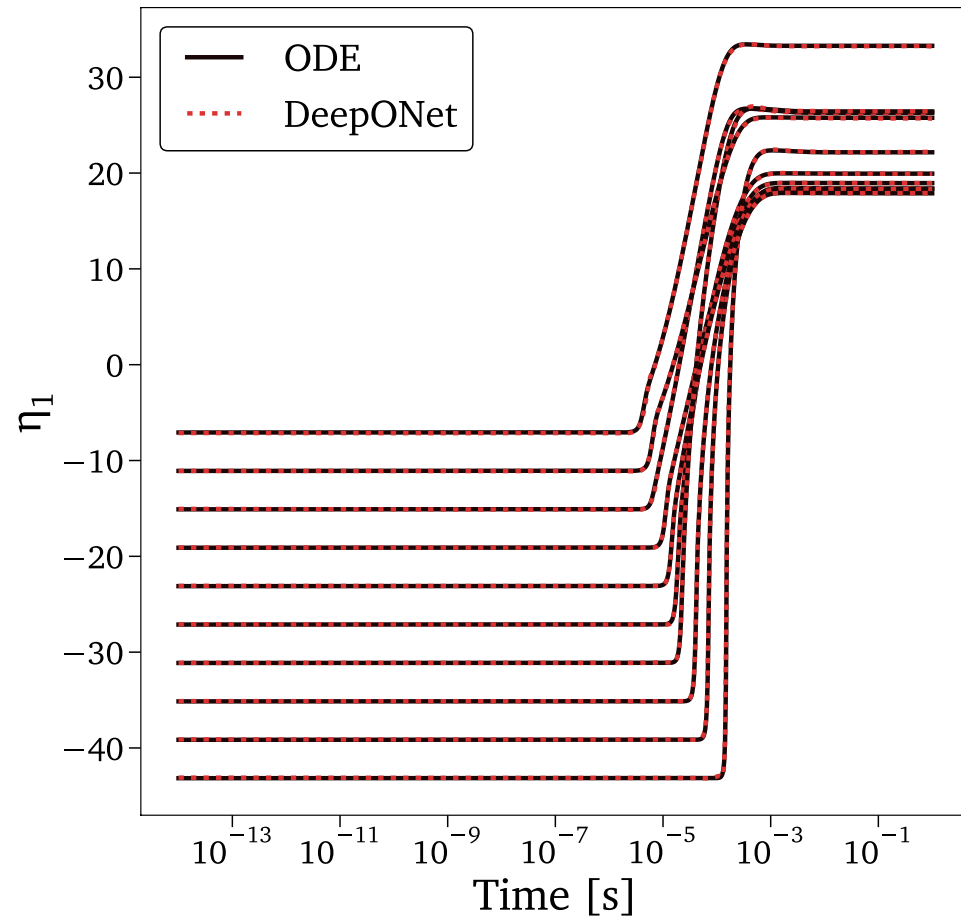


A Combustion Chemistry in Isobaric Reactor Test Case



Results from the improved structure

Predicted time-dependent principal components for test scenarios

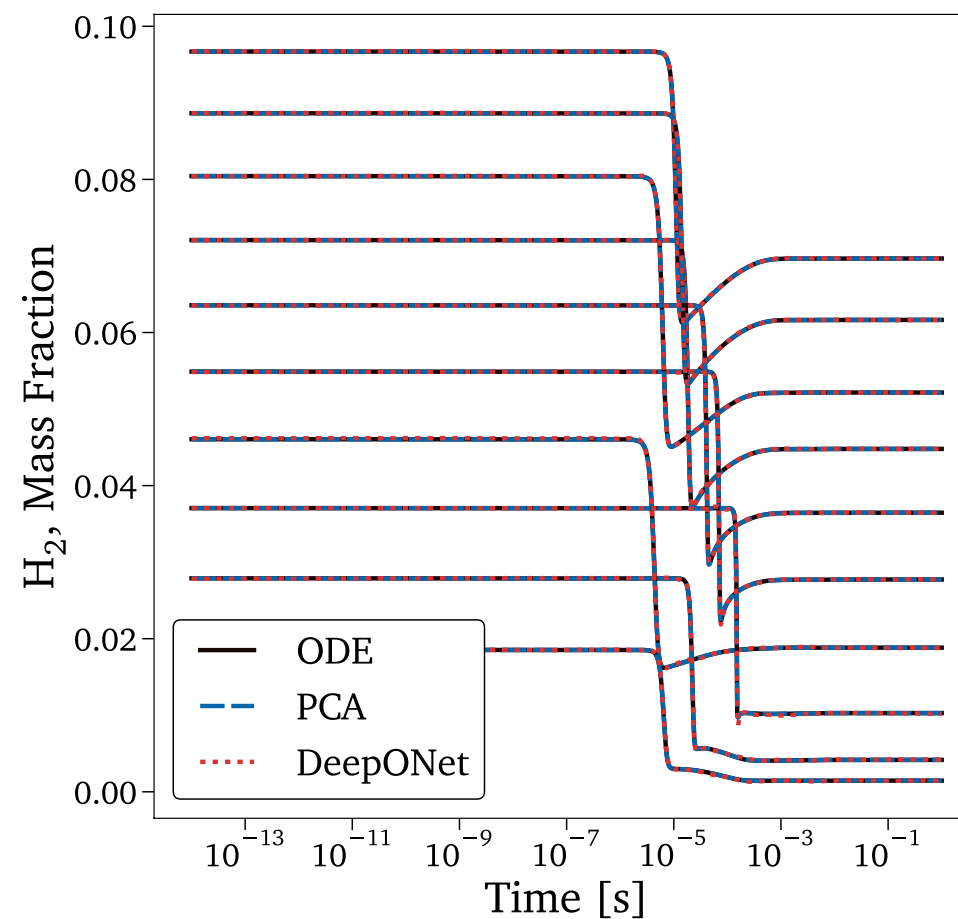
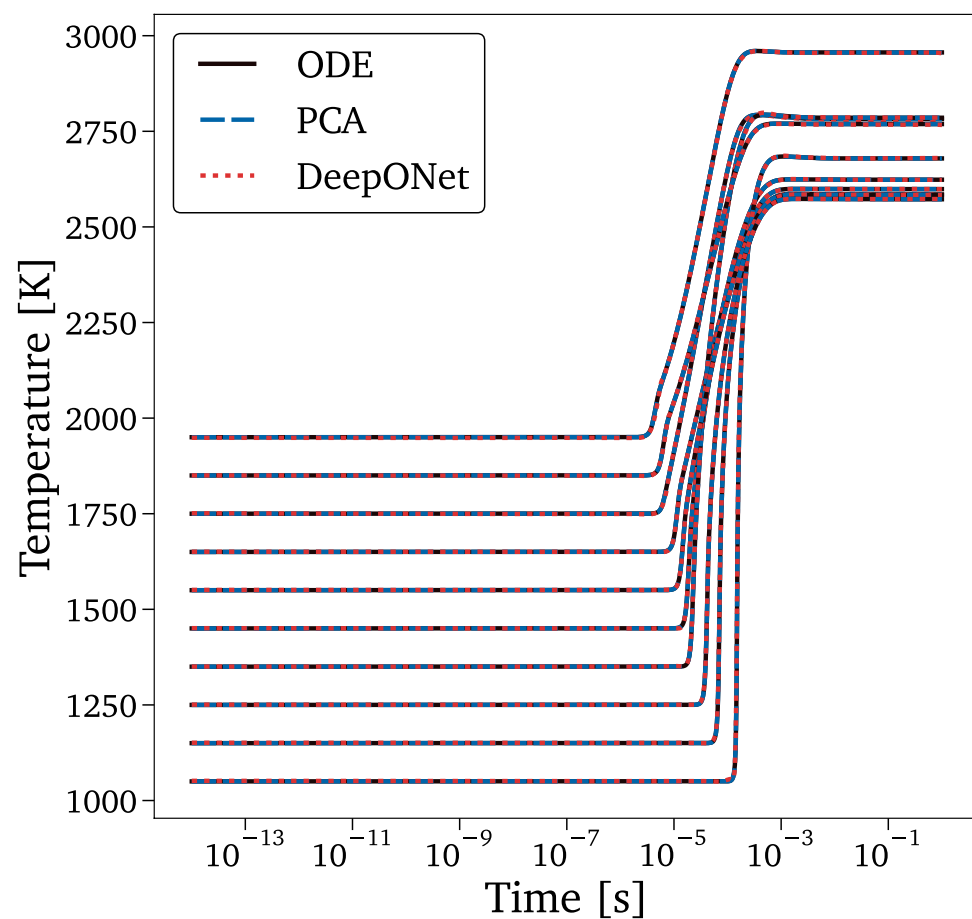


A Combustion Chemistry in Isobaric Reactor Test Case



Results from the improved structure

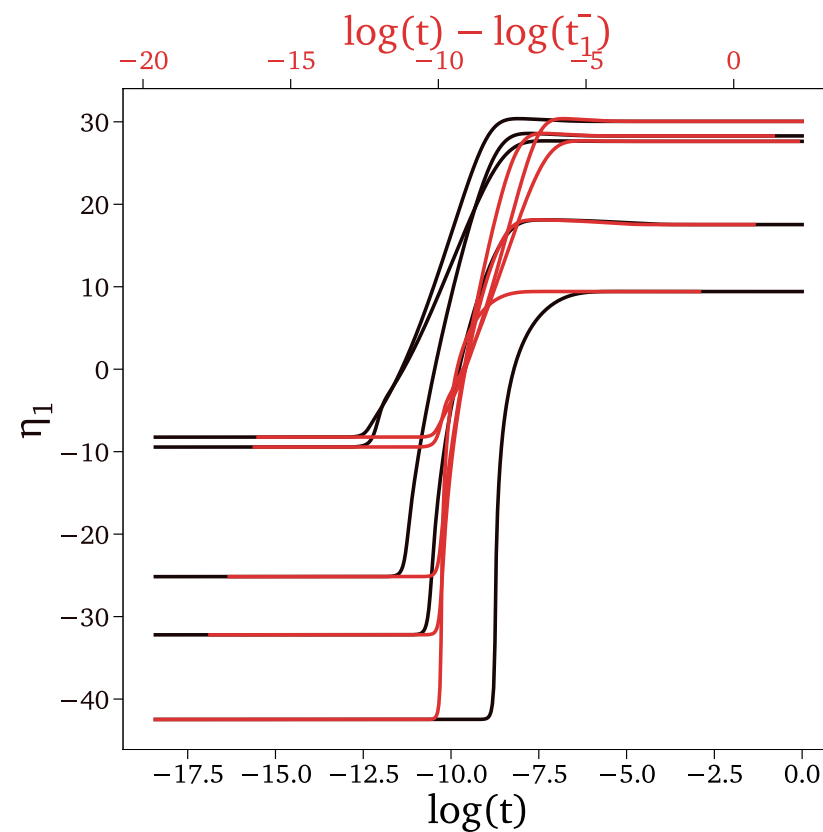
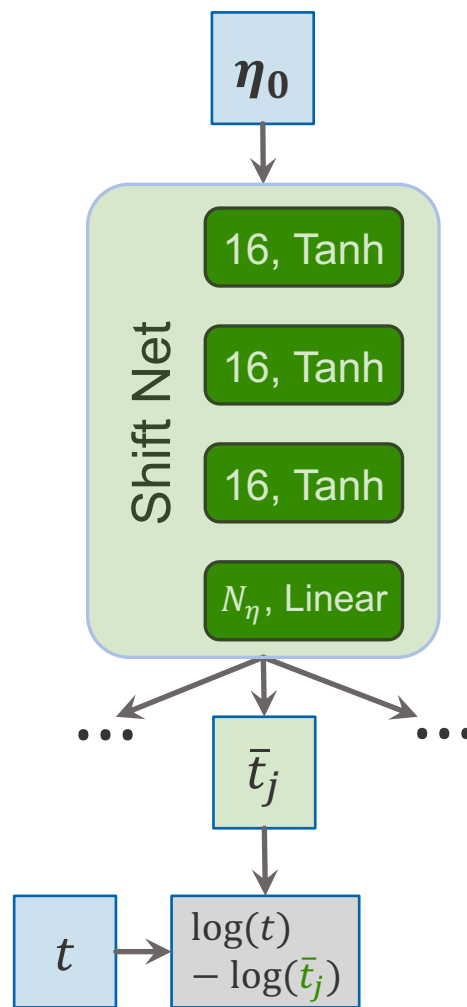
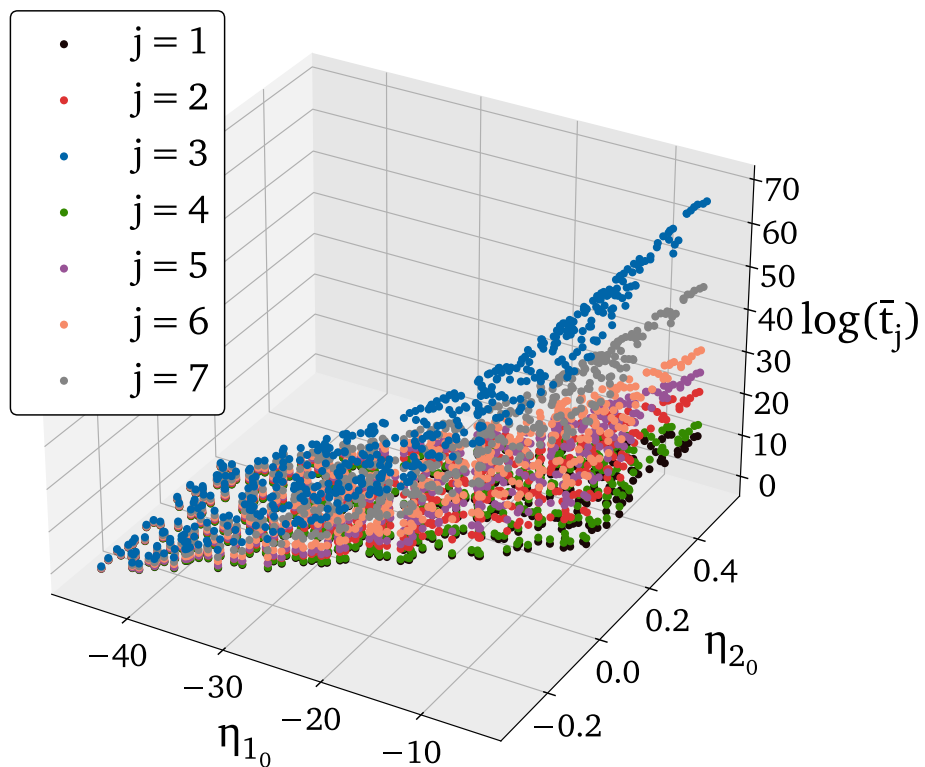
Reconstructed time-dependent temperature and species for test scenarios



A Combustion Chemistry in Isobaric Reactor Test Case



Results from the improved structure





Test Case 10: Data-driven improved deep operator network (DeepONet) for predicting Principal Components

10.1. Copy \$WORKSPACE_PATH/ROMNet/romnet/input/0DReact/DeepONet/TestCase10/ROMNet_Input.py
to \$WORKSPACE_PATH/ROMNet/romnet/input/ROMNet_Input.py

10.2. In \$WORKSPACE_PATH/ROMNet/romnet/input/ROMNet_Input.py, change:
10.2.1. "self.WORKSPACE_PATH = ..."

10.3. Move to \$WORKSPACE_PATH/ROMNet/romnet/app/

10.4. Run: "python3 ROMNet.py ../input/"

10.5. Postprocess results via: \$WORKSPACE_PATH/ROMNet/romnet/scripts/postprocessing/0DReact/DeepONet/Predict_DeepONet.ipynb