Test Cases for Combustion Chemistry in a Isobaric Reactor

Simone Venturi, Tiernan Casey

Extreme-Scale Data Science & Analytics (8739)

Sandia National Laboratories

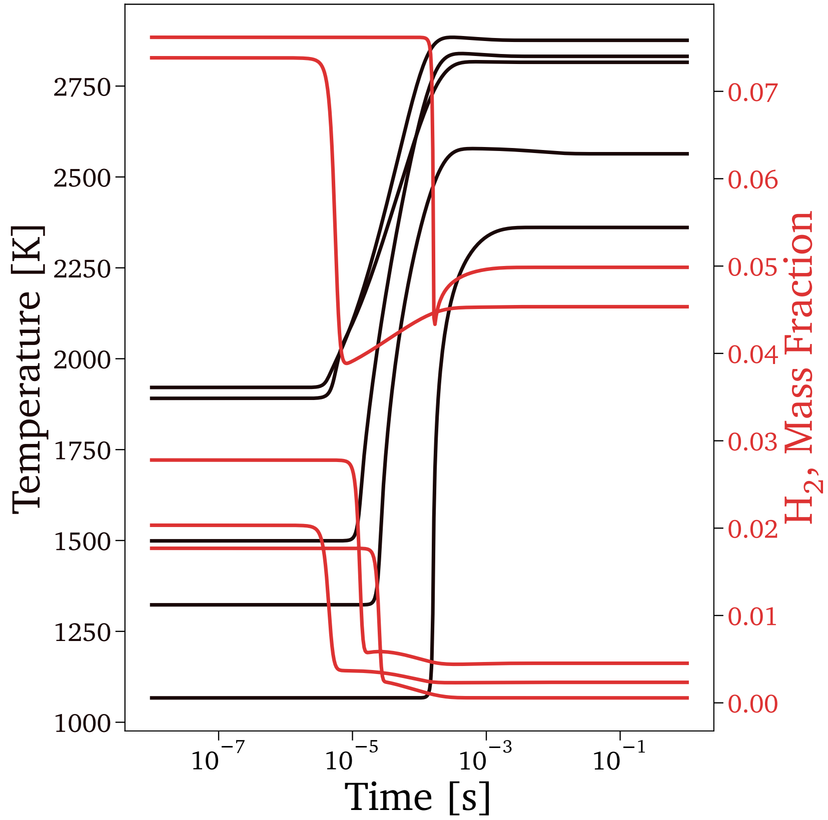
U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

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.

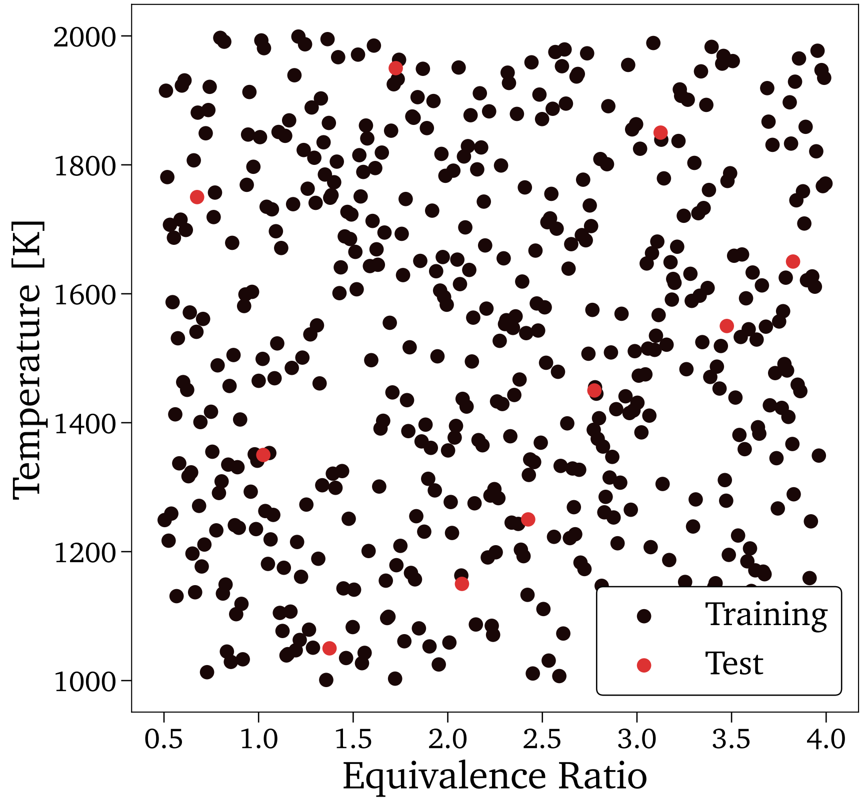
A Combustion Chemistry in Isobaric Reactor Test Case

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



Some training scenarios

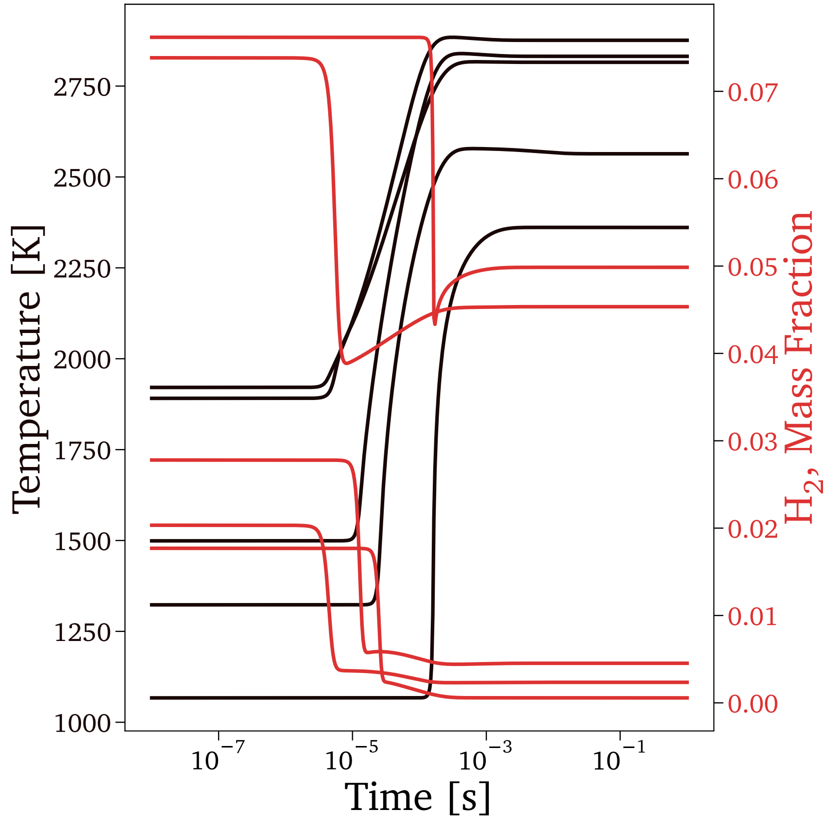
Initial conditions



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

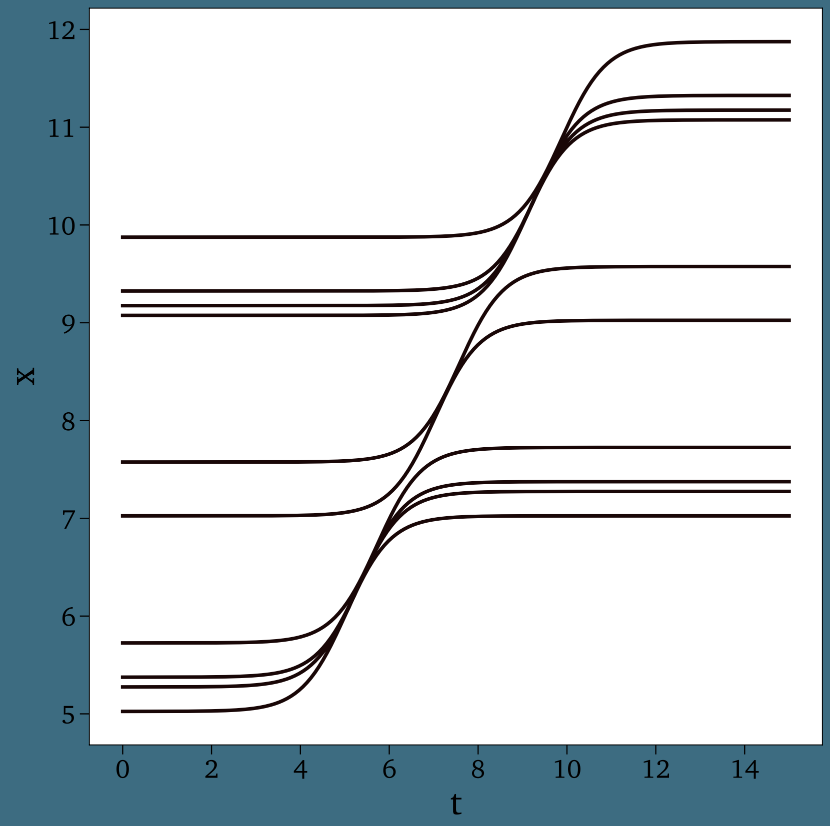
A Combustion Chemistry in Isobaric Reactor Test Case

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 implem
$WORKSPACE_PATH/ROMNet/romnet/romnet/pinn/system/0DReact.py

A Combustion Chemistry in Isobaric Reactor Test Case

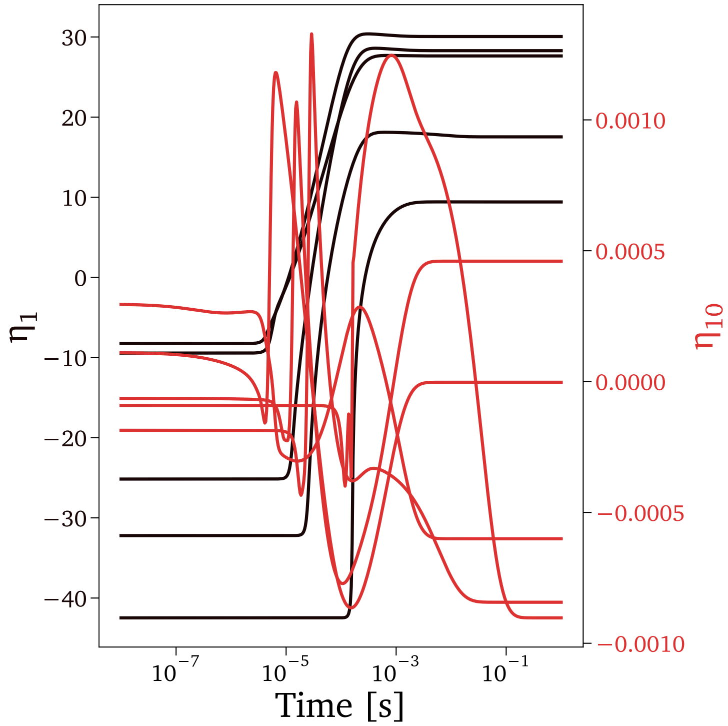
Run python scrip:
$WORKSPACE_PATH/ROMNet/romnet/scripts/generating_data/0DReactor/Generate_Data_1_Isobaric.py
for generating simulation data

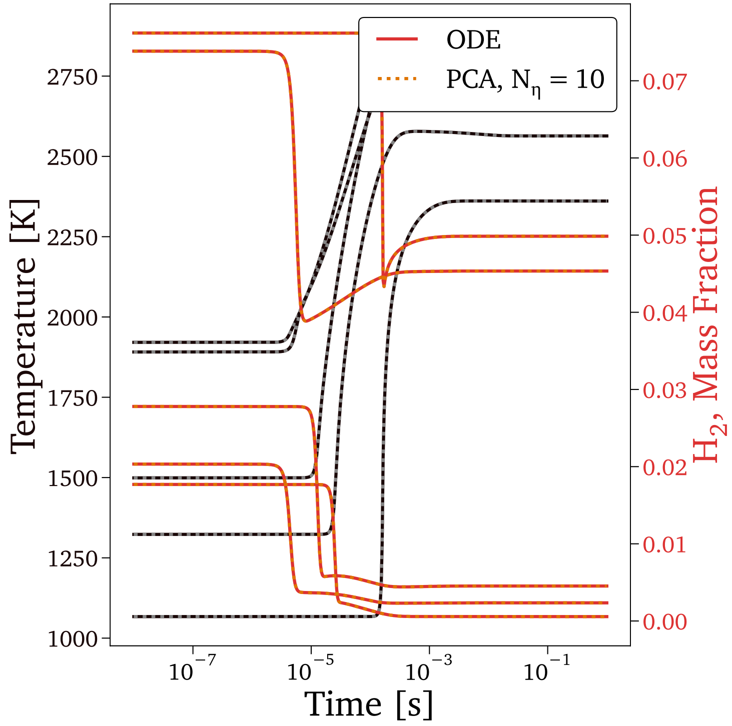
Note: The script needs to be run twice, the second time for generating test data

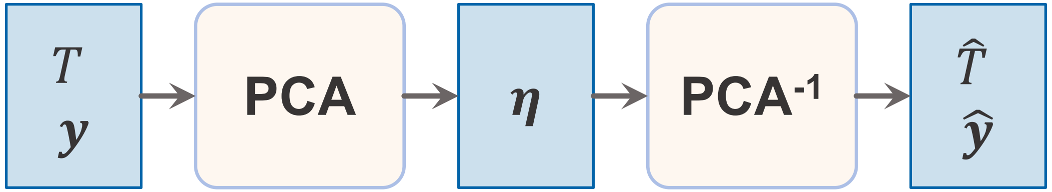
Run Jupyter Notebook:
$WORKSPACE_PATH/ROMNet/romnet/scripts/generating_data/0DReactor/Generate_Data_3_Isobaric.ipynb
for generating training and test data

A Combustion Chemistry in Isobaric Reactor Test Case

Employed PCA for reducing the dimensionality of the state space







10 principal components ($\boldsymbol{\eta}$) 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

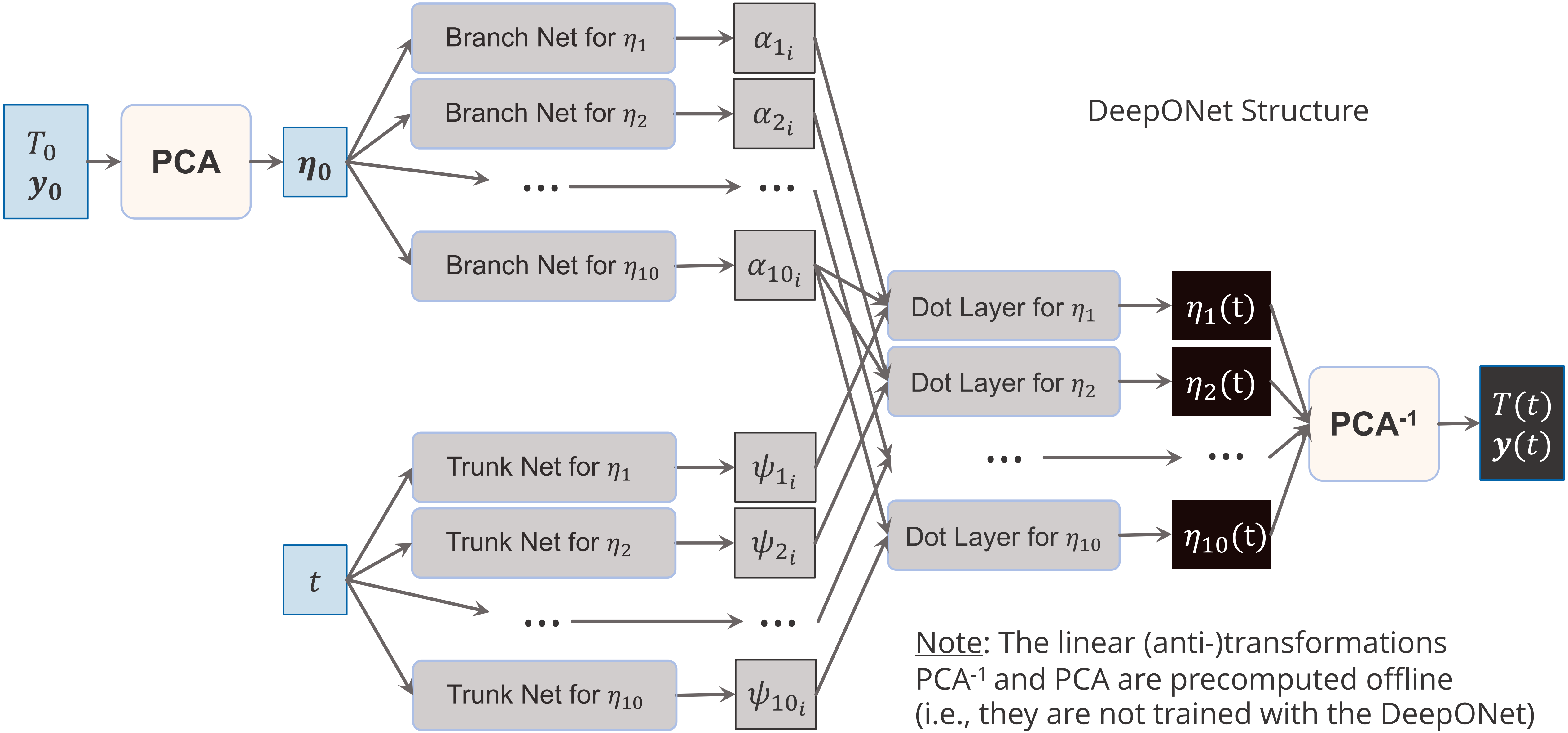
Note: The script needs to be run twice, the second time for generating test data

Run Jupyter Notebook:
$WORKSPACE_PATH/ROMNet/romnet/scripts/generating_data/0DReactor/Generate_Data_4_Isobaric.ipynb
for generating PCA training and test data

A Combustion Chemistry in Isobaric Reactor Test Case

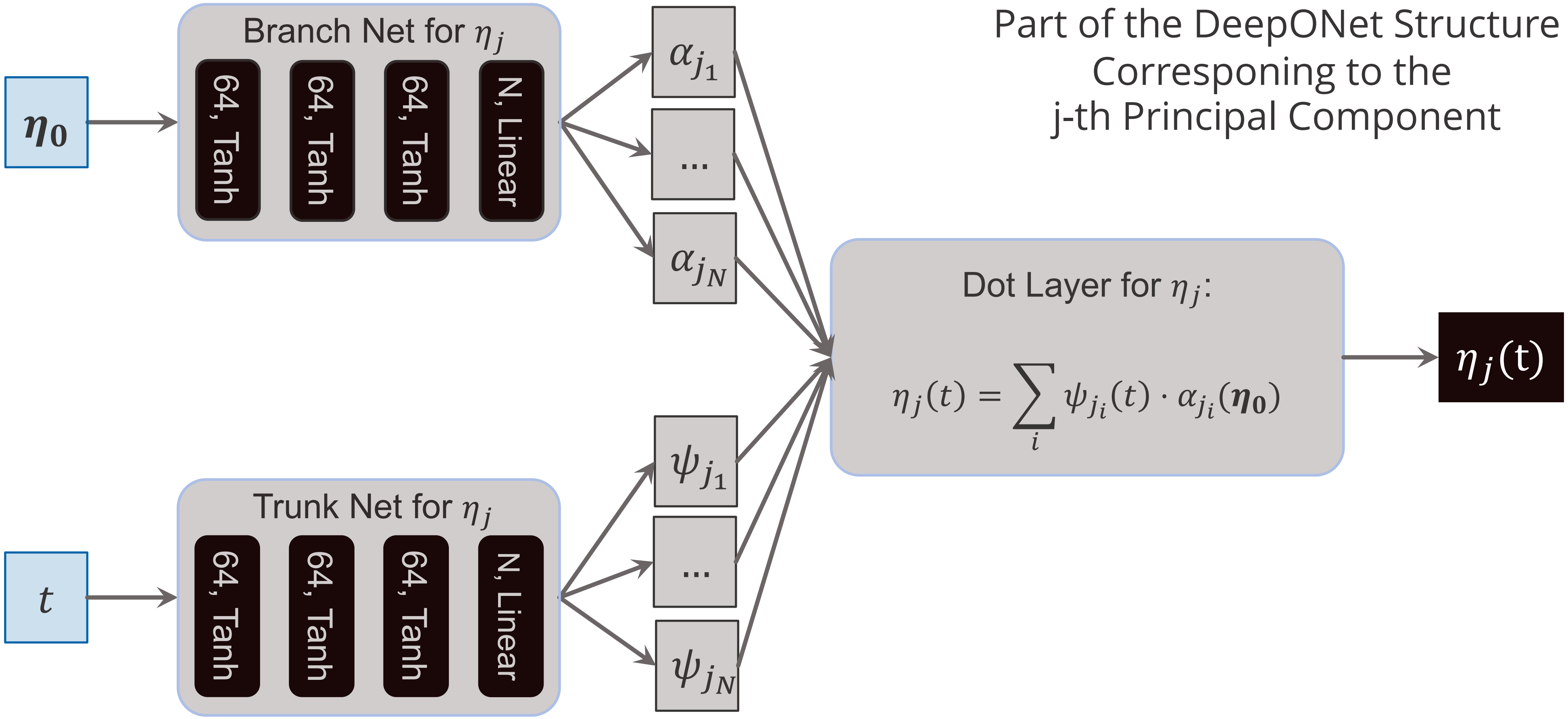
Test Case 1

A Combustion Chemistry in Isobaric Reactor Test Case



Note: The linear (anti-)transformations PCA^{-1} and PCA are precomputed offline (i.e., they are not trained with the DeepONet)

A Combustion Chemistry in Isobaric Reactor Test Case



Part of the DeepONet Structure
Corresponing to the
j-th Principal Component

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 1: Data-driven deep operator network (DeepONet) for predicting Principal Components

1.1. Copy $WORKSPACE_PATH/ROMNet/romnet/input/0DReact/DeepONet/0DReact_H2_TestCase1/ROMNet_Input.py
to $WORKSPACE_PATH/ROMNet/romnet/input/ROMNet_Input.py

1.2. In $WORKSPACE_PATH/ROMNet/romnet/input/ROMNet_Input.py, change:
1.2.1. "self.WORKSPACE_PATH = …"

1.3. Move to $WORKSPACE_PATH/ROMNet/romnet/app/

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

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

A Combustion Chemistry in Isobaric Reactor Test Case

Investigating the issue: a principal component analysis

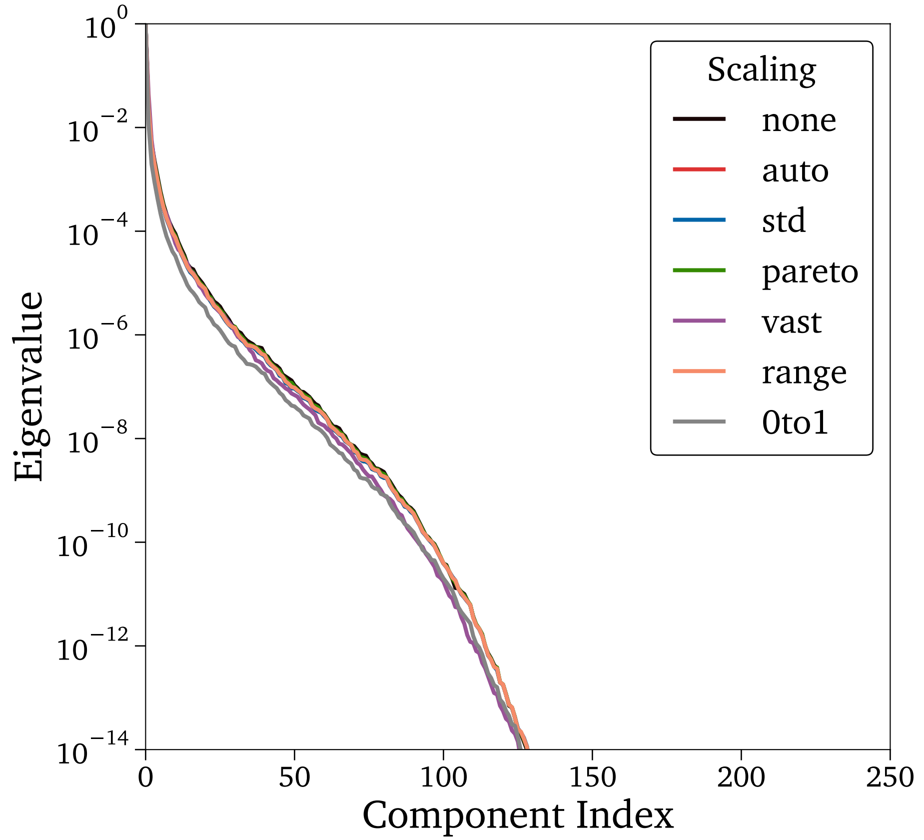
Aggregation of training scenarios for $\eta_j(t)$, where i represents the scenario index:

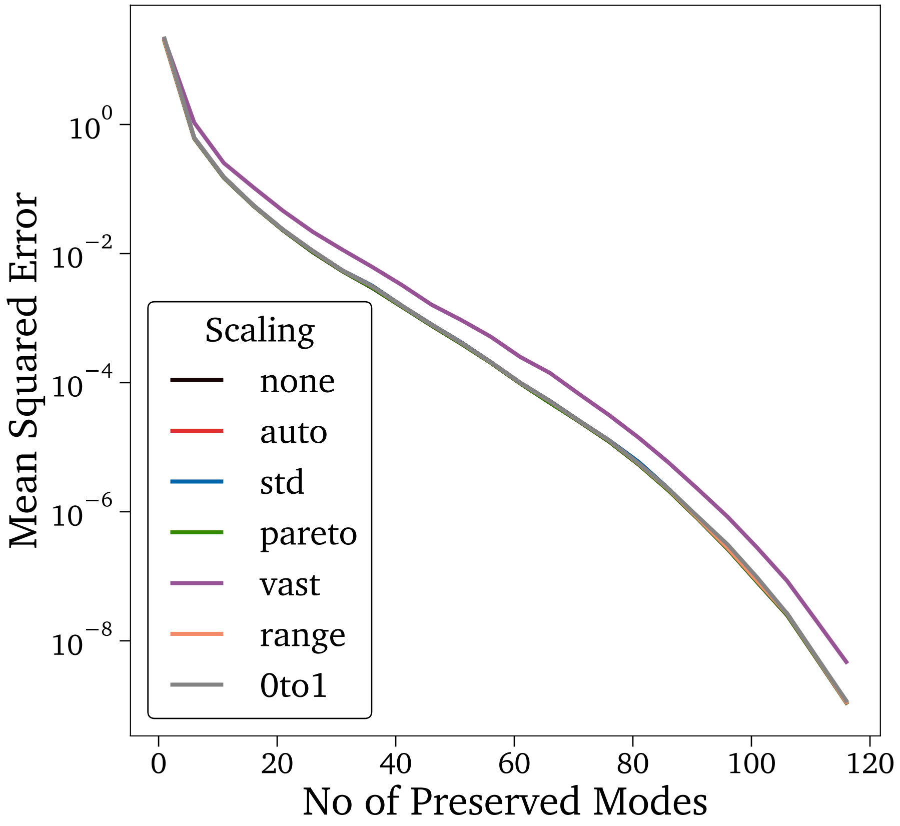
$$\mathbf{H_j} = \begin{bmatrix} | & | & & | & | \\ \boldsymbol{\eta_{j_1}} & \boldsymbol{\eta_{j_2}} & \cdots & \boldsymbol{\eta_{j_{499}}} & \boldsymbol{\eta_{j_{500}}} \\ | & | & & | & | \end{bmatrix}$$

$$\mathbf{dim}(\mathbf{H_j}) = \underbrace{\boldsymbol{N_t}}_{\text{No. time instants}} \times \underbrace{\boldsymbol{N_s}}_{\text{No. of scenarios}}$$

A Combustion Chemistry in Isobaric Reactor Test Case

Eigendecomposition of $\boldsymbol{R}_{\mathbf{H}_j}$: $$\boldsymbol{\Psi}_j = \frac{\mathbf{H_j} - \boldsymbol{C}_j}{\boldsymbol{D}_j}\boldsymbol{A}_j$$ (Note: results are shown for $j = 1$ (i.e., for η_1))

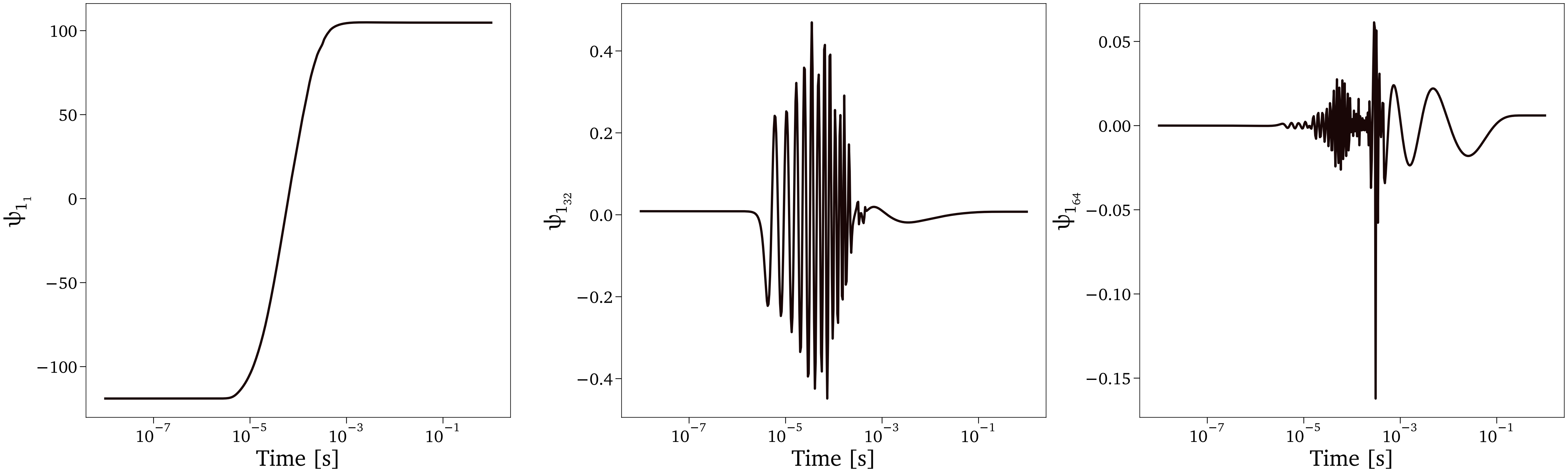




A relatively large number of modes needs to be preserved in order to predict η_j with good accuracy

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



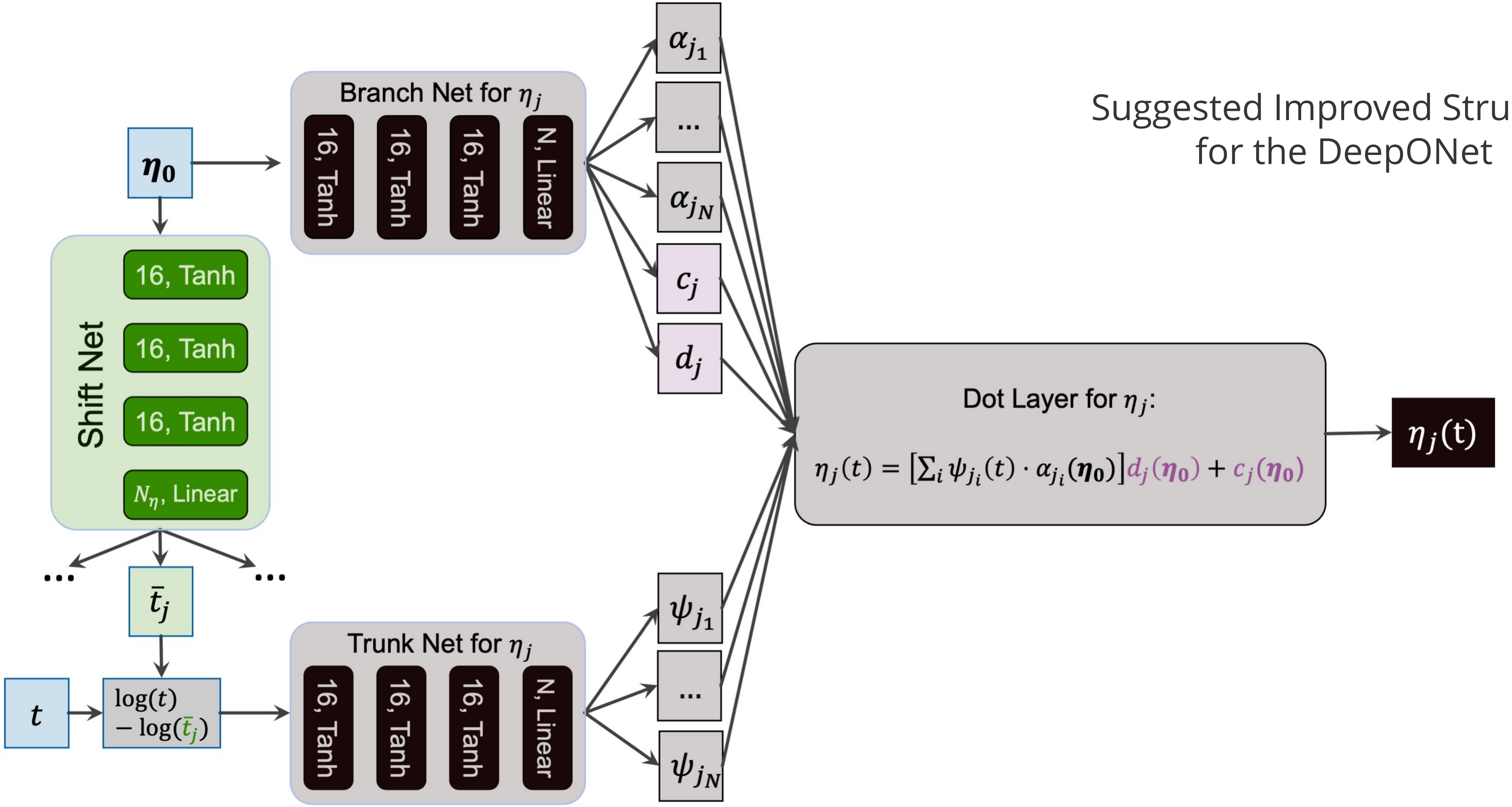
A Combustion Chemistry in Isobaric Reactor Test Case

Run Jupyter Notebook:
$WORKSPACE_PATH/ROMNet/romnet/scripts/generating_data/0DReactor/Generate_Data_6.ipynb
for generating scenario-aggregated SVD for PCA data

A Combustion Chemistry in Isobaric Reactor Test Case

Test Case 2

A Combustion Chemistry in Isobaric Reactor Test Case

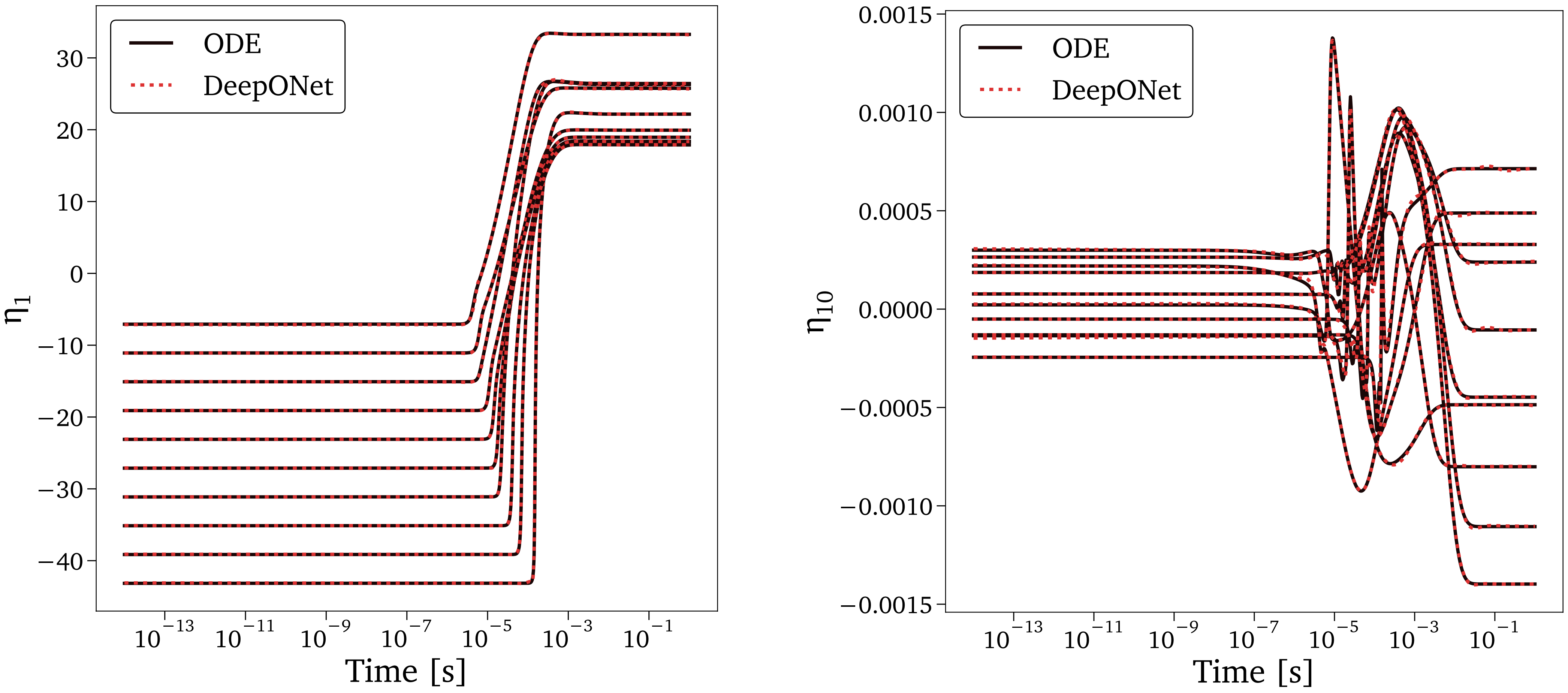


Suggested Improved Structure for the DeepONet

A Combustion Chemistry in Isobaric Reactor Test Case

Results from the improved structure

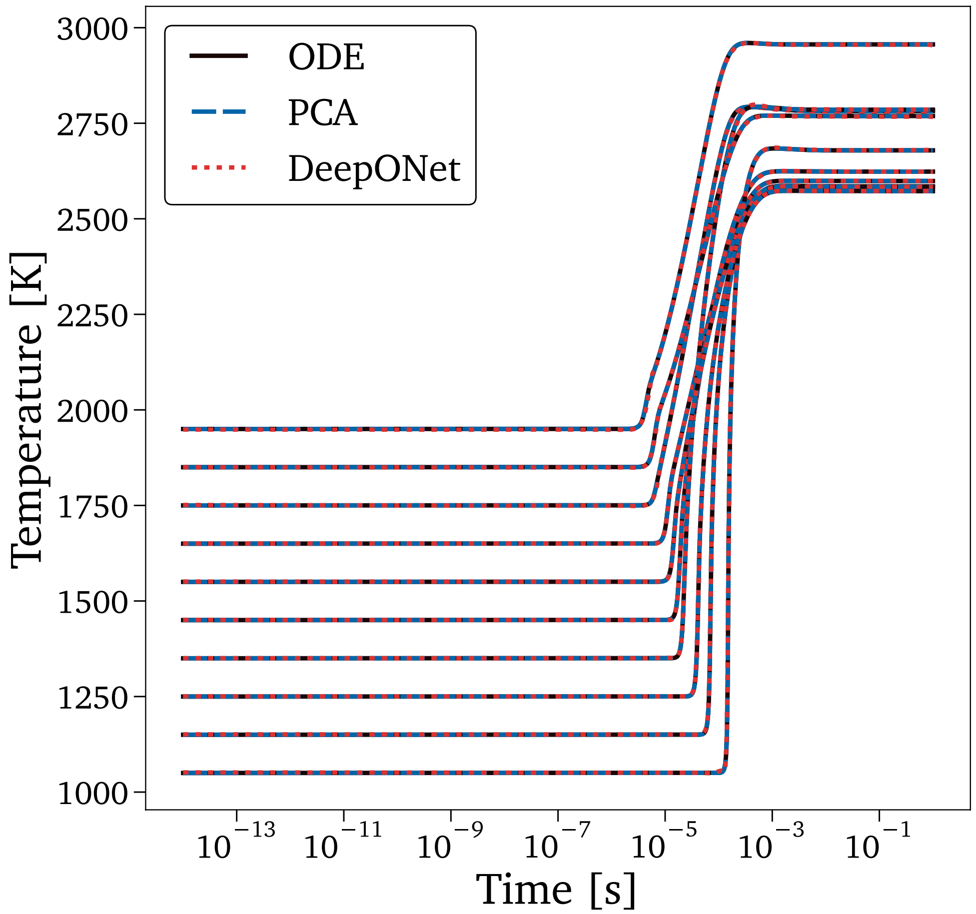
Predicted time-dependent principal components for test scenarios

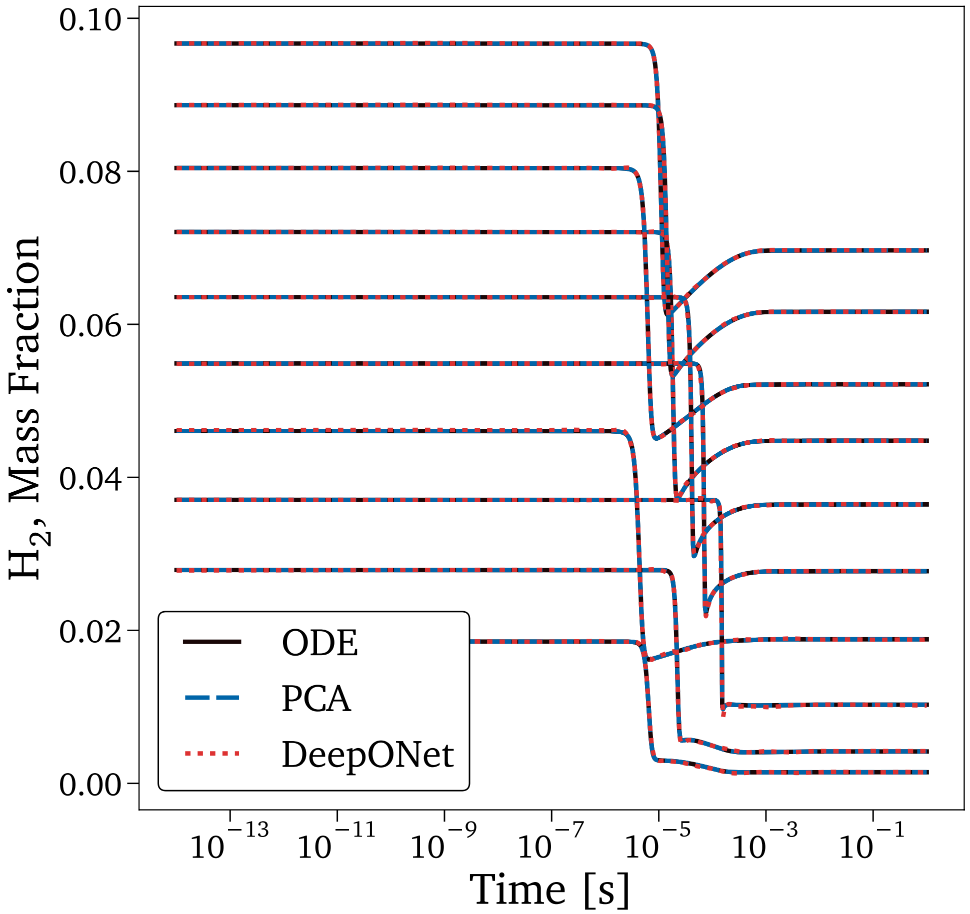


A Combustion Chemistry in Isobaric Reactor Test Case

Results from the improved structure

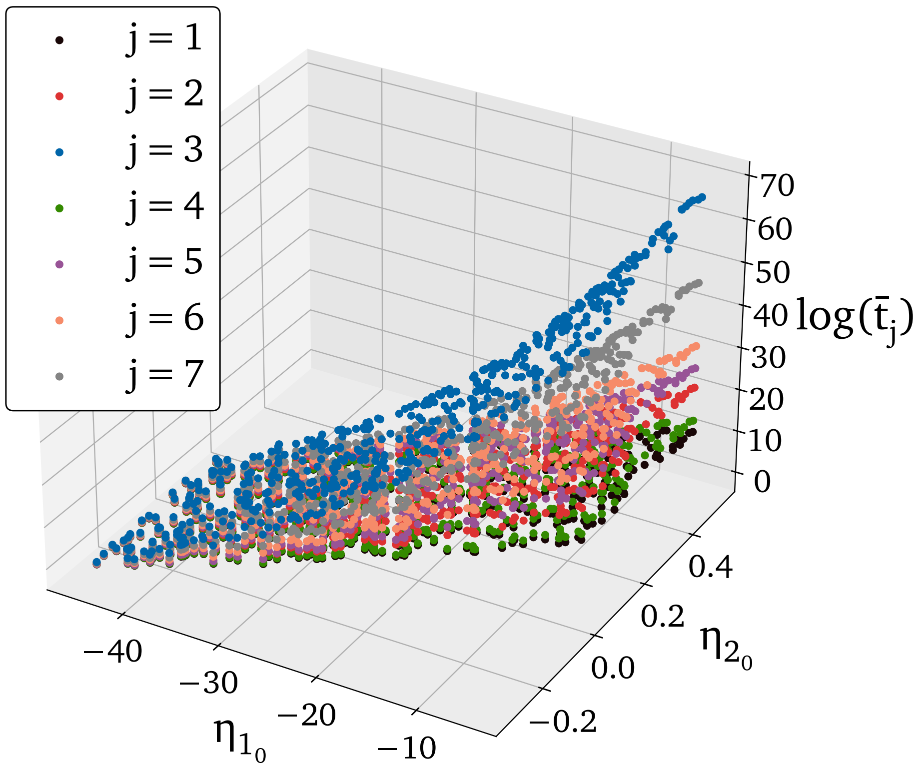
Reconstructed time-dependent temperature and species for test scenarios

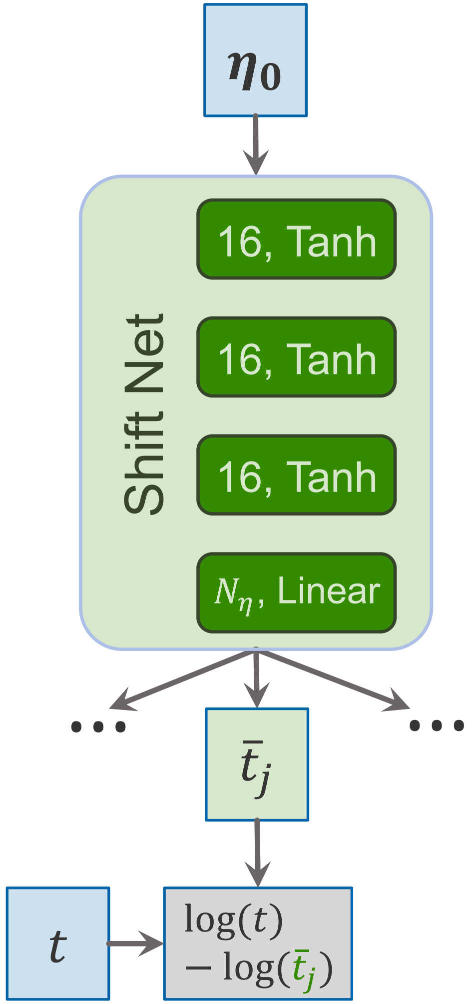


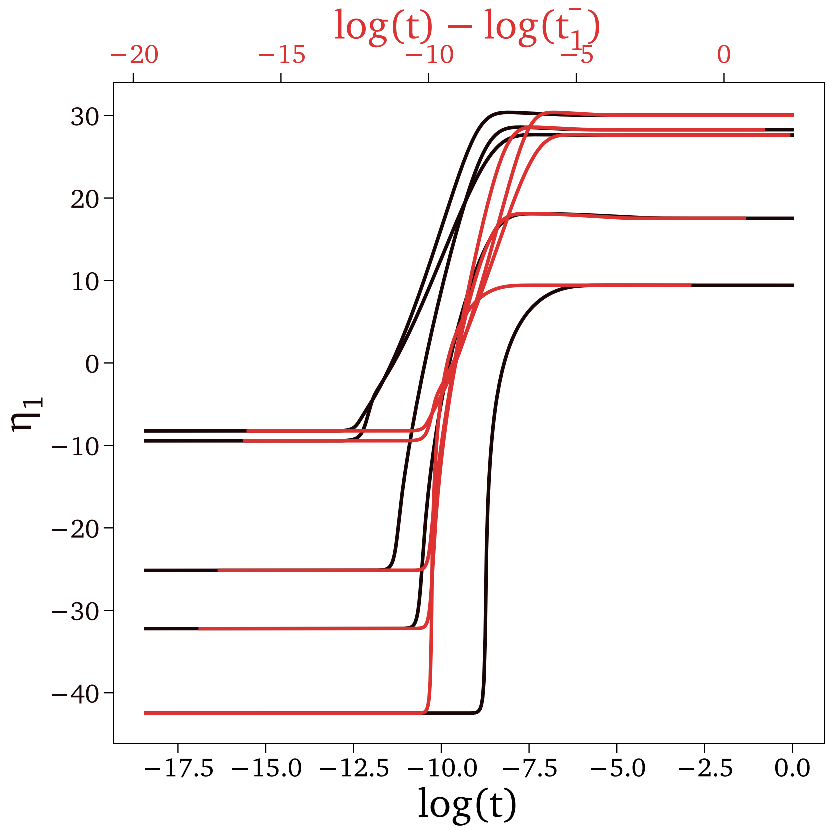


A Combustion Chemistry in Isobaric Reactor Test Case

Results from the improved structure







A Combustion Chemistry in Isobaric Reactor Test Case

Test Case 2: Data-driven improved deep operator network (DeepONet) for predicting Principal Components

2.1. Copy $WORKSPACE_PATH/ROMNet/romnet/input/0DReact/DeepONet/0DReact_H2_TestCase2/ROMNet_Input.py
to $WORKSPACE_PATH/ROMNet/romnet/input/ROMNet_Input.py

2.2. In $WORKSPACE_PATH/ROMNet/romnet/input/ROMNet_Input.py, change:
2.2.1. "self.WORKSPACE_PATH = …"

2.3. Move to $WORKSPACE_PATH/ROMNet/romnet/app/

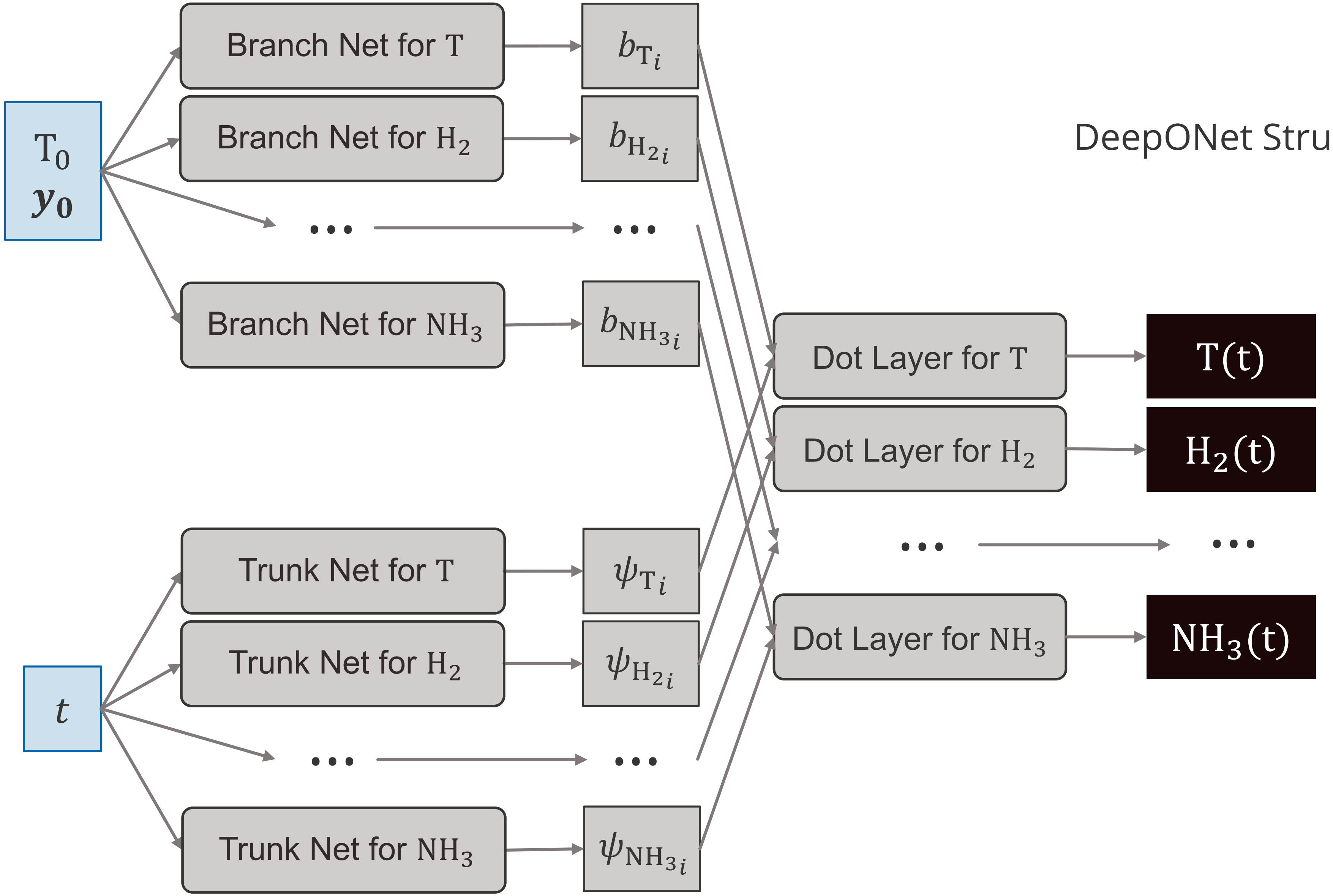
2.4. Run: "python3 ROMNet.py ../input/"

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

A Combustion Chemistry in Isobaric Reactor Test Case

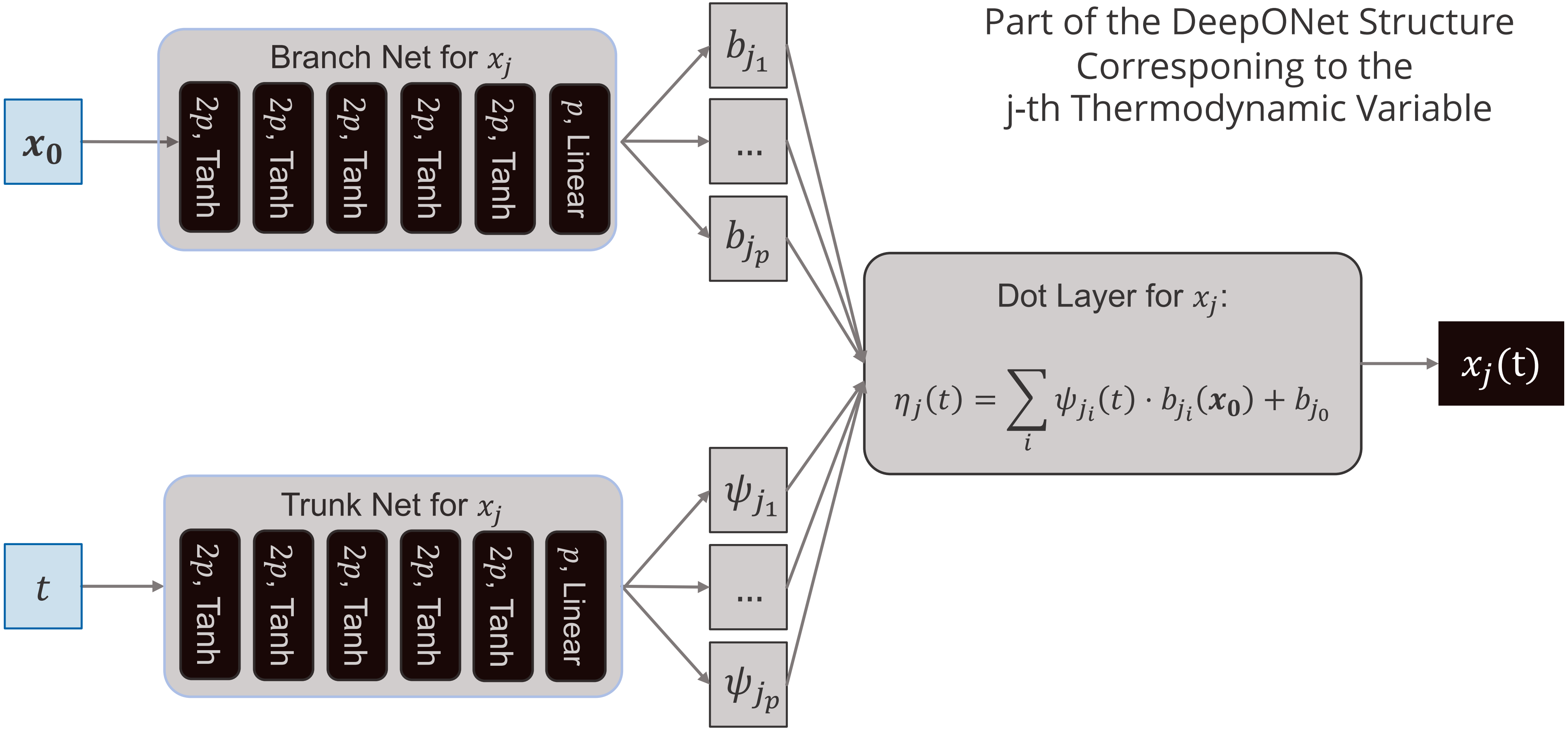
Test Case 3

A Combustion Chemistry in Isobaric Reactor Test Case



DeepONet Structure

A Combustion Chemistry in Isobaric Reactor Test Case



Part of the DeepONet Structure
Corresponing to the
j-th Thermodynamic Variable

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 3: Data-driven deep operator network (DeepONet) for predicting Thermodynamic Variables

3.1. Copy $WORKSPACE_PATH/ROMNet/romnet/input/0DReact/DeepONet/0DReact_H2_TestCase3/ROMNet_Input.py
  to $WORKSPACE_PATH/ROMNet/romnet/input/ROMNet_Input.py

3.2. In $WORKSPACE_PATH/ROMNet/romnet/input/ROMNet_Input.py, change:
  3.2.1. "self.WORKSPACE_PATH = …"

3.3. Move to $WORKSPACE_PATH/ROMNet/romnet/app/

3.4. Run: "python3 ROMNet.py ../input/”

3.5. Postprocess results via: $WORKSPACE_PATH/ROMNet/romnet/scripts/postprocessing/0DReact/DeepONet/Predict_DeepONet_Orig.ipynb

Relevant Input Variables:

self.**data_preproc_type**: if self.norm_input_flg/self.norm_output_flg == True, then then input/data is center and/or scaled based on the technique specified by **data_preproc_type**. (Note: auto-scaling is the preset centering and scaling)

self.**rectify_flg**: If set to True in order to guarantee the positivity of the outputs, a ReLu postprocessing layer is applied at the end of DeepONet.

A Combustion Chemistry in Isobaric Reactor Test Case

Investigating the issue: a principal component analysis

Aggregation of training scenarios for $\mathbf{H_2}(t)$, where i represents the scenario index:

$$\mathbf{H_2} = \begin{bmatrix} | & | & & | & | \\ \mathbf{H_2}_1 & \mathbf{H_2}_2 & \cdots & \mathbf{H_2}_{499} & \mathbf{H_2}_{500} \\ | & | & & | & | \end{bmatrix}$$

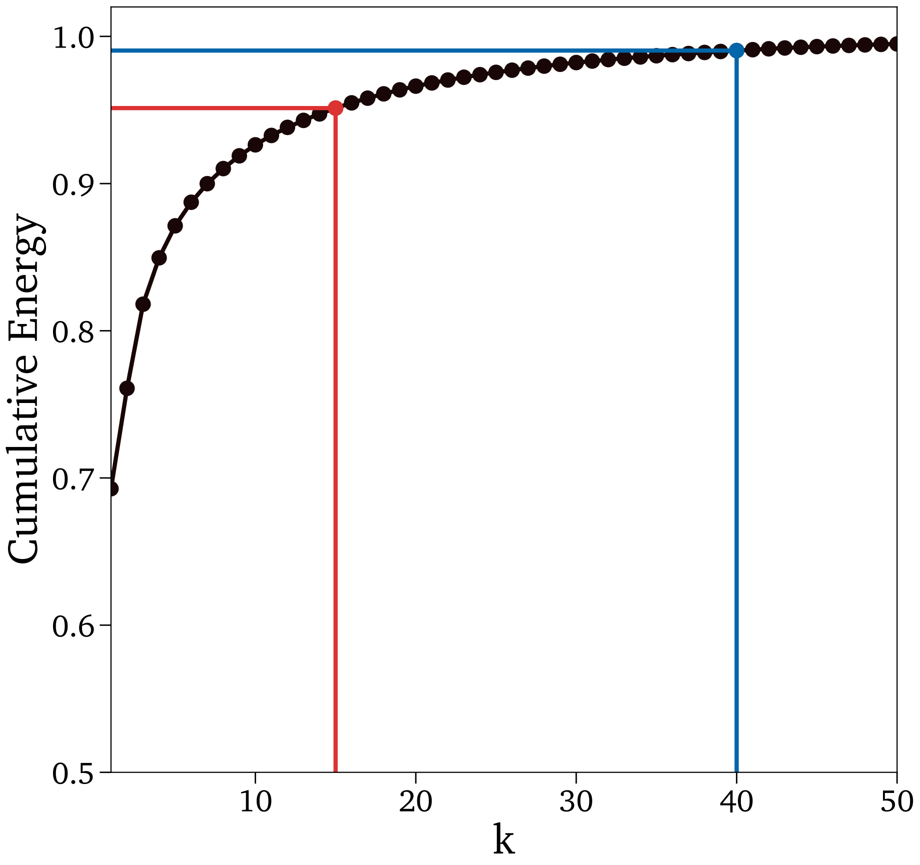
$$\mathbf{dim}(\mathbf{H_2}) = \boldsymbol{N_t} \times \boldsymbol{N_s}$$

No. time instants

No. of scenarios

A Combustion Chemistry in Isobaric Reactor Test Case

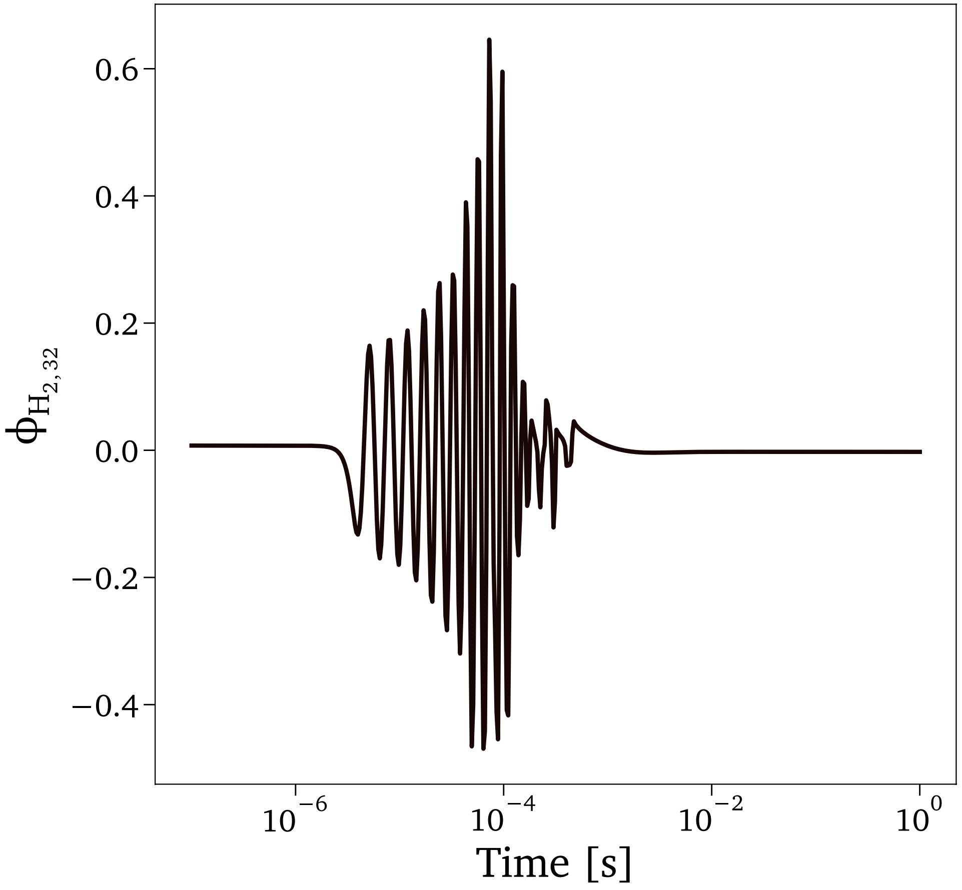
Eigendecomposition of $\boldsymbol{R}_{H_2}$: $$\boldsymbol{\Psi}_{H_2} = \frac{\mathbf{H}_2 - \boldsymbol{C}_{H_2}}{\boldsymbol{D}_{H_2}} \boldsymbol{A}_{H_2}$$

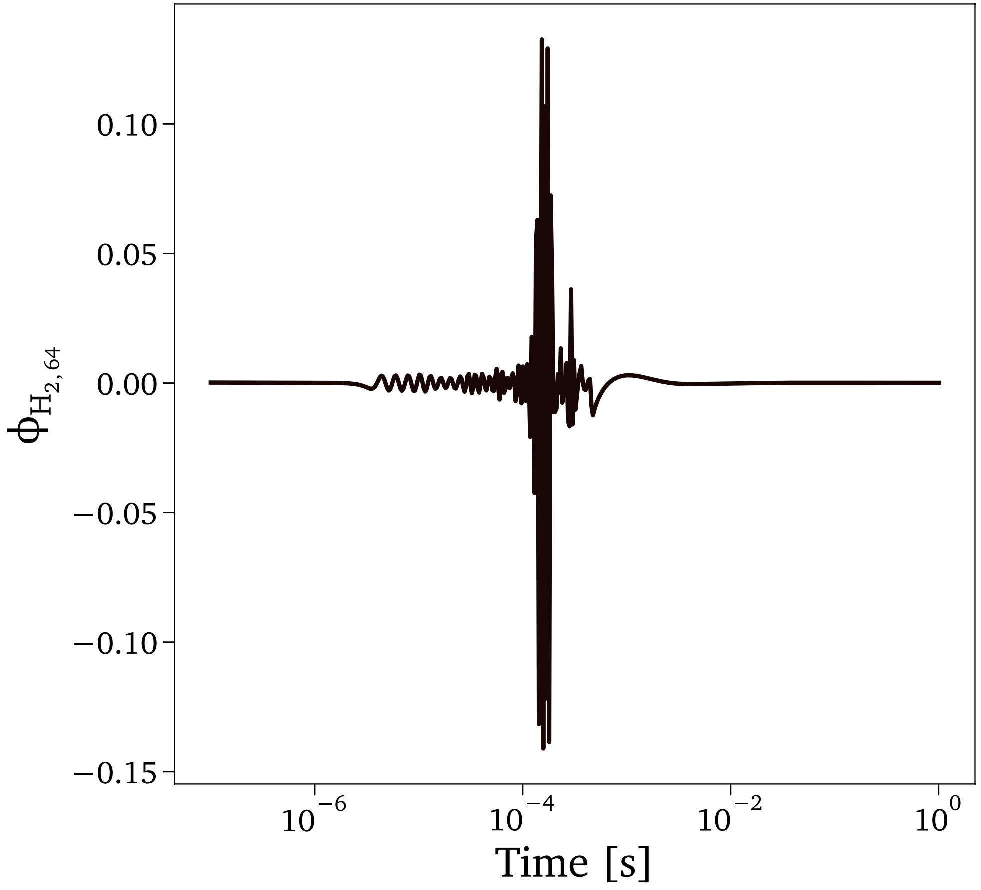


A relatively large number of modes needs to be preserved in order to predict T with good accuracy

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





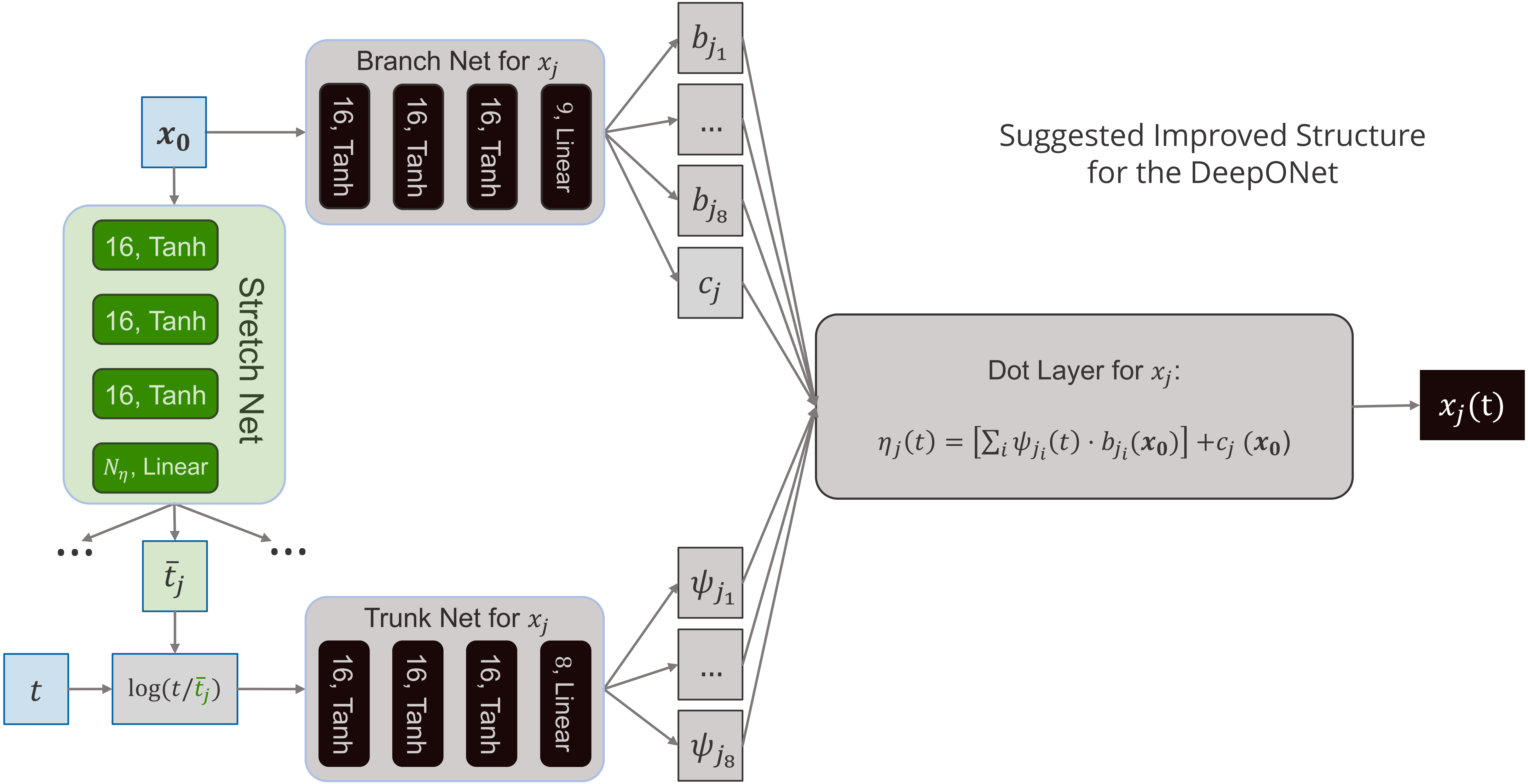
A Combustion Chemistry in Isobaric Reactor Test Case

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

A Combustion Chemistry in Isobaric Reactor Test Case

Test Case 4

A Combustion Chemistry in Isobaric Reactor Test Case

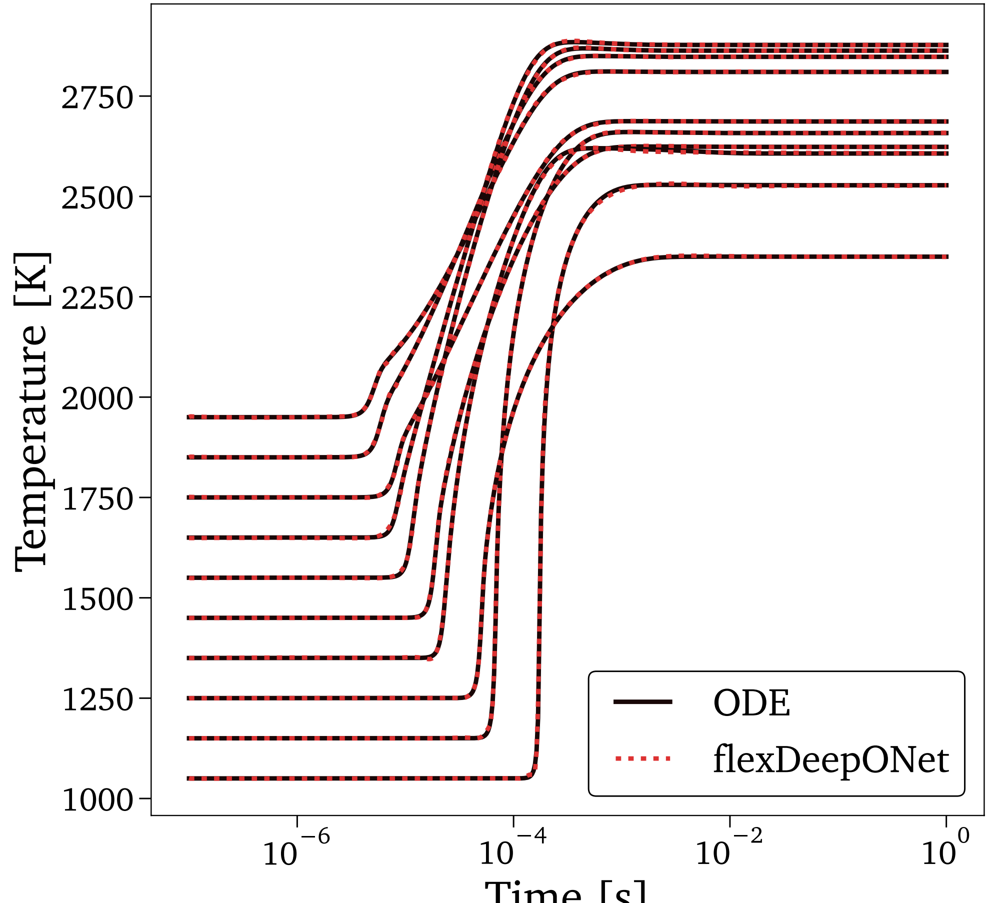


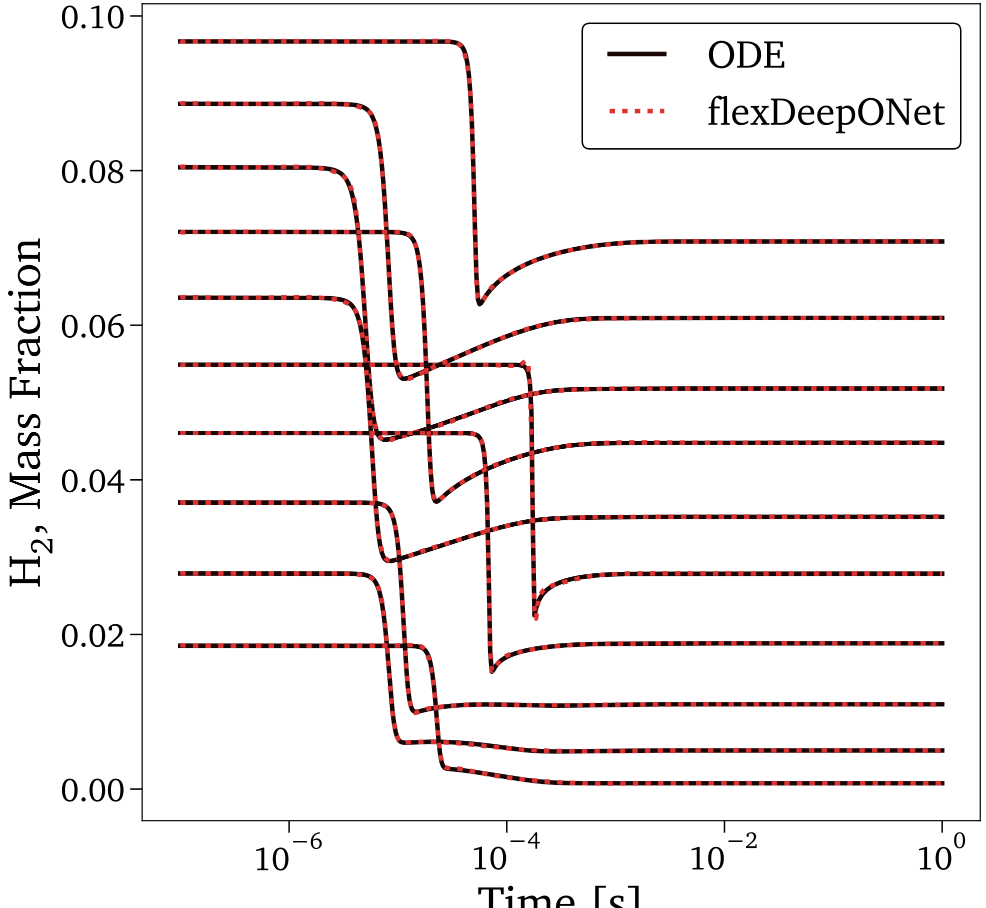
Suggested Improved Structure for the DeepONet

A Combustion Chemistry in Isobaric Reactor Test Case

Results from the improved structure

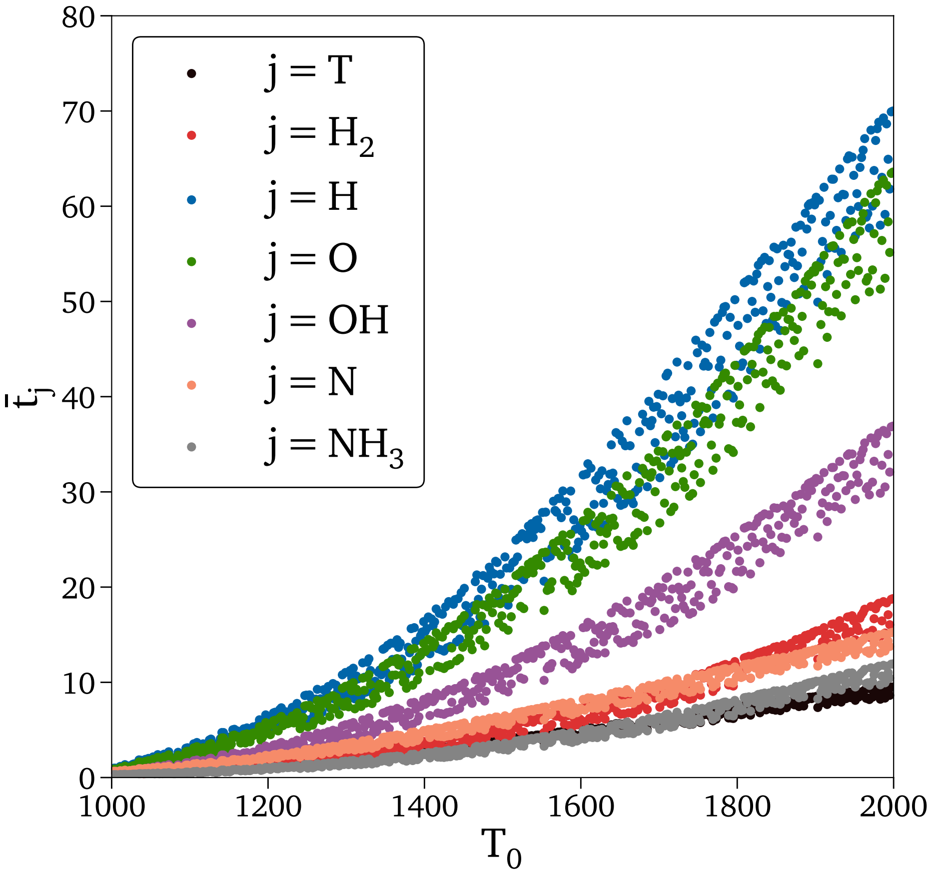
Predicted time-dependent temperature and species for test scenarios

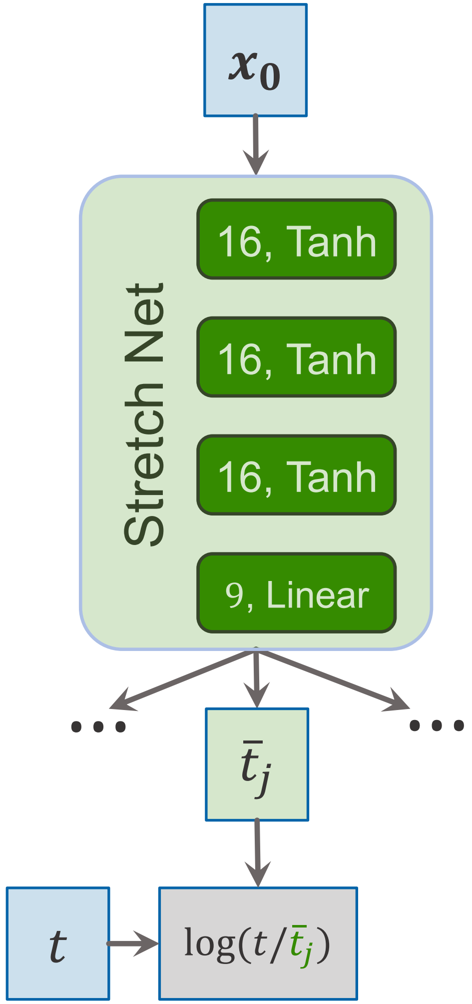


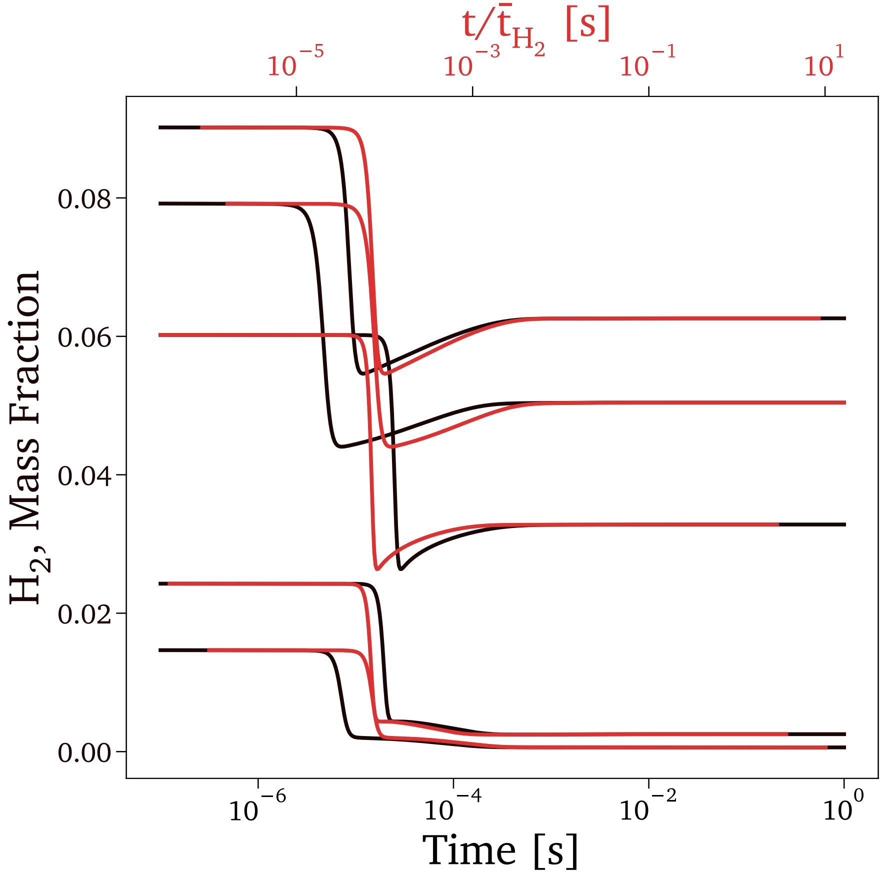


A Combustion Chemistry in Isobaric Reactor Test Case

Results from the improved structure







A Combustion Chemistry in Isobaric Reactor Test Case

Test Case 4: Data-driven improved deep operator network (DeepONet) for predicting Thermodynamic Variables

4.1. Copy $WORKSPACE_PATH/ROMNet/romnet/input/0DReact/DeepONet/0DReact_H2_TestCase4/ROMNet_Input.py
to $WORKSPACE_PATH/ROMNet/romnet/input/ROMNet_Input.py

4.2. In $WORKSPACE_PATH/ROMNet/romnet/input/ROMNet_Input.py, change:
4.2.1. "self.WORKSPACE_PATH = …"

4.3. Move to $WORKSPACE_PATH/ROMNet/romnet/app/

4.4. Run: "python3 ROMNet.py ../input/”

4.5. Postprocess results via: $WORKSPACE_PATH/ROMNet/romnet/scripts/postprocessing/0DReact/DeepONet/Predict_DeepONet_Orig.ipynb

Relevant Input Variables:

self.**data_preproc_type**: if self.norm_input_flg/self.norm_output_flg == True, then then input/data is center and/or scaled based on the technique specified by **data_preproc_type**. (Note: auto-scaling is the preset centering and scaling)

self.**rectify_flg**: If set to True in order to guarantee the positivity of the outputs, a ReLu postprocessing layer is applied at the end of DeepONet.