



Sandia  
National  
Laboratories

# Test Cases for Combustion Chemistry in a Isobaric Reactor

Simone Venturi, Tiernan Casey

Extreme-Scale Data Science & Analytics (8739)

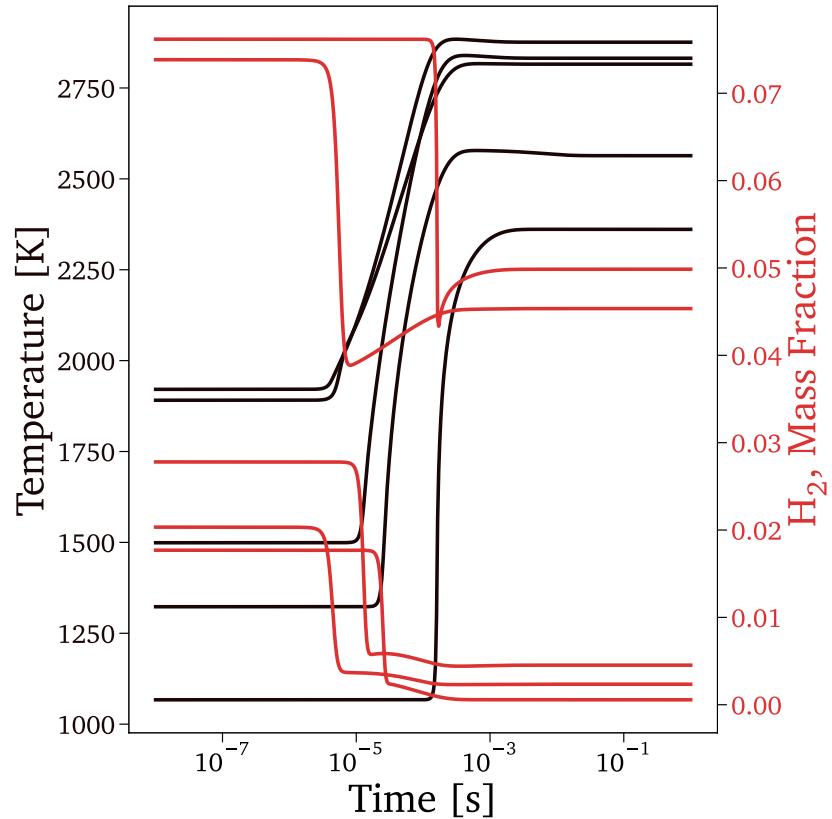


Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

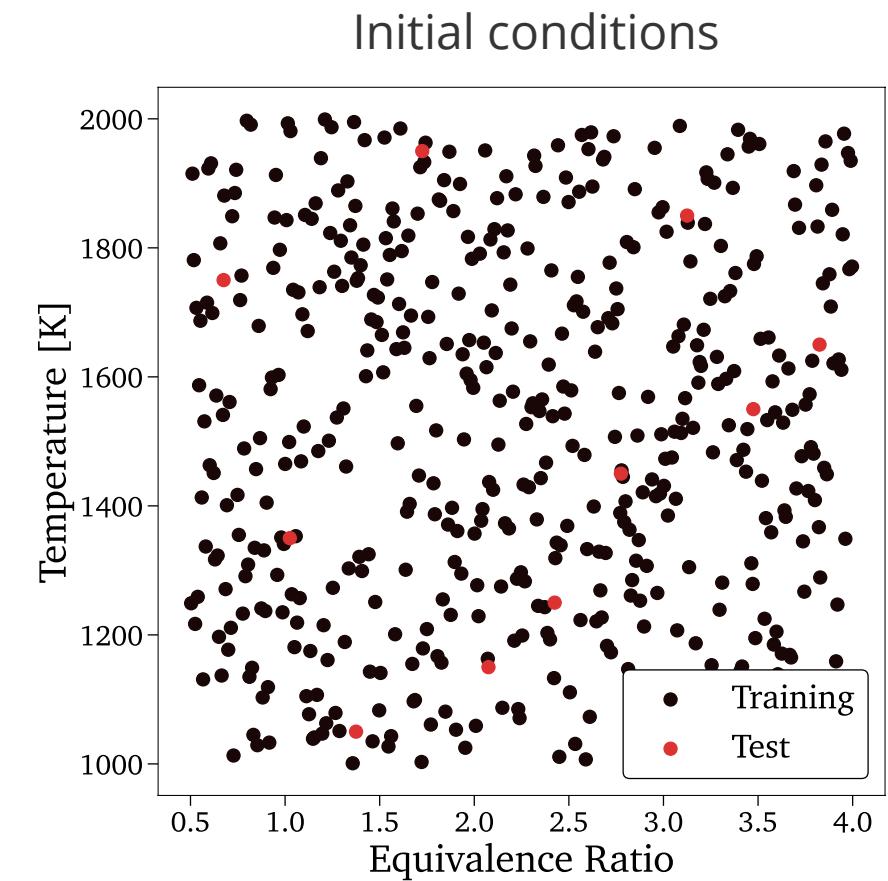
# A Combustion Chemistry in Isobaric Reactor Test Case



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



Some training scenarios

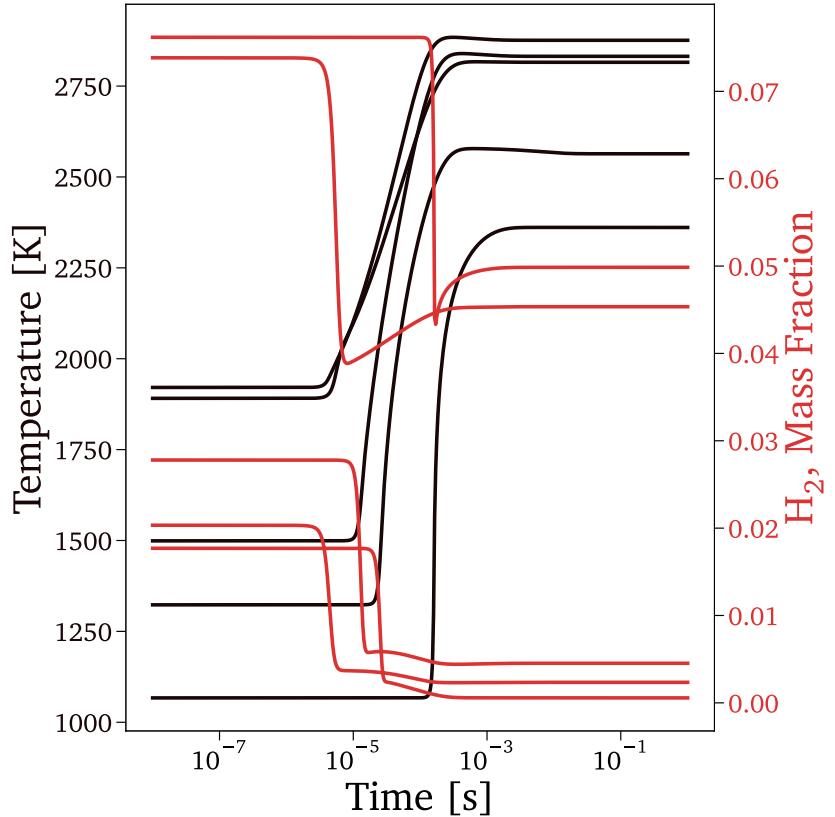


The physical system is implemented in  
`$WORKSPACE_PATH/ROMNet/romnet/romnet/pinn/system/0DReact.py`

# A Combustion Chemistry in Isobaric Reactor Test Case

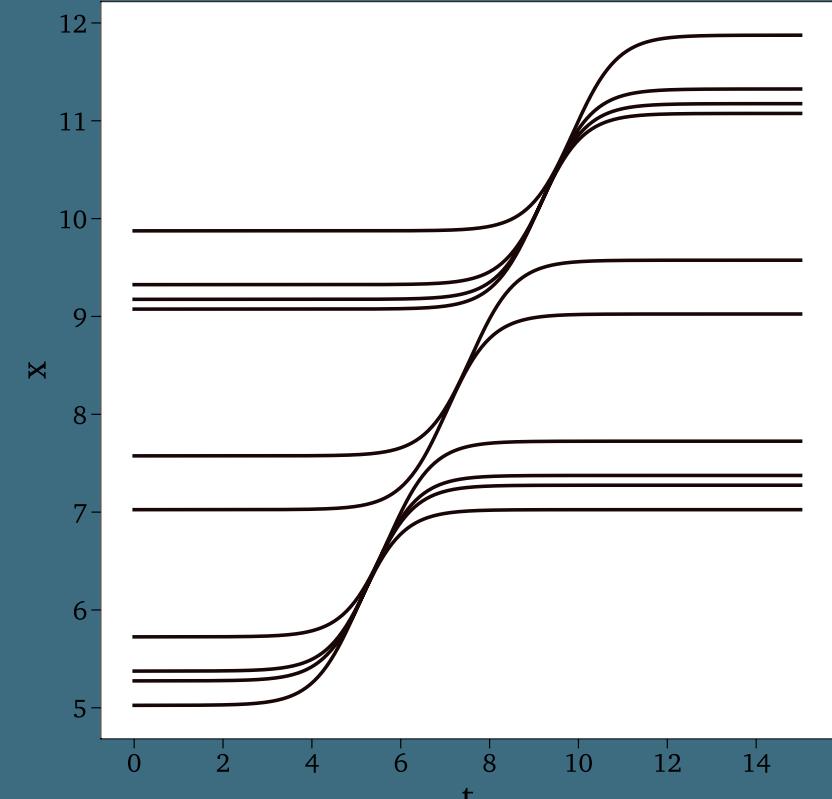


Isobaric 0-D Reactor (Hydrogen-air), 20 state variables  
(i.e., temperature and 19 species mass fractions)



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

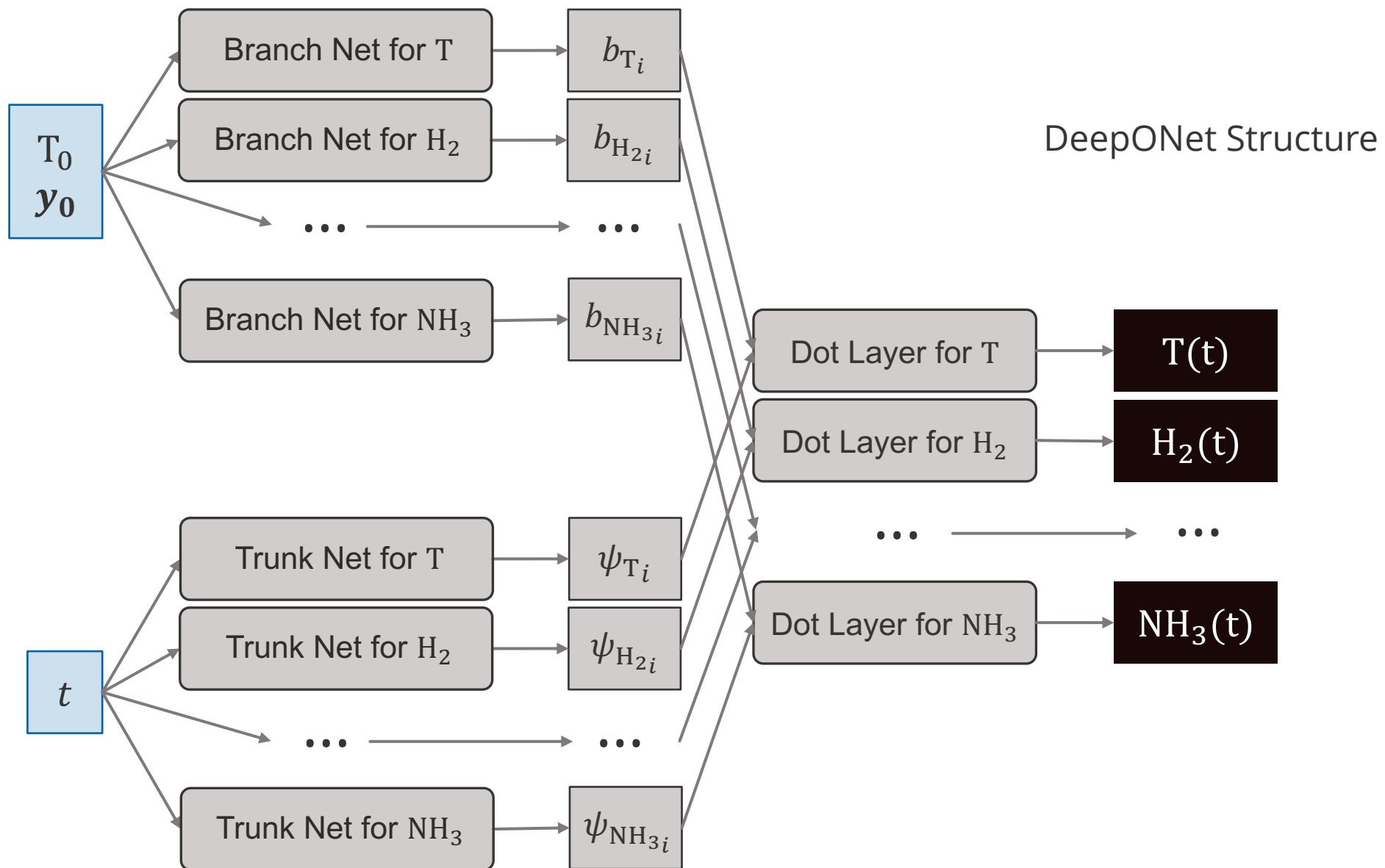
Run Jupyter Notebook:

\$WORKSPACE\_PATH/ROMNet/romnet/scripts/generating\_data/0DReactor/Generate\_Data\_3.ipynb  
for generating training and test data

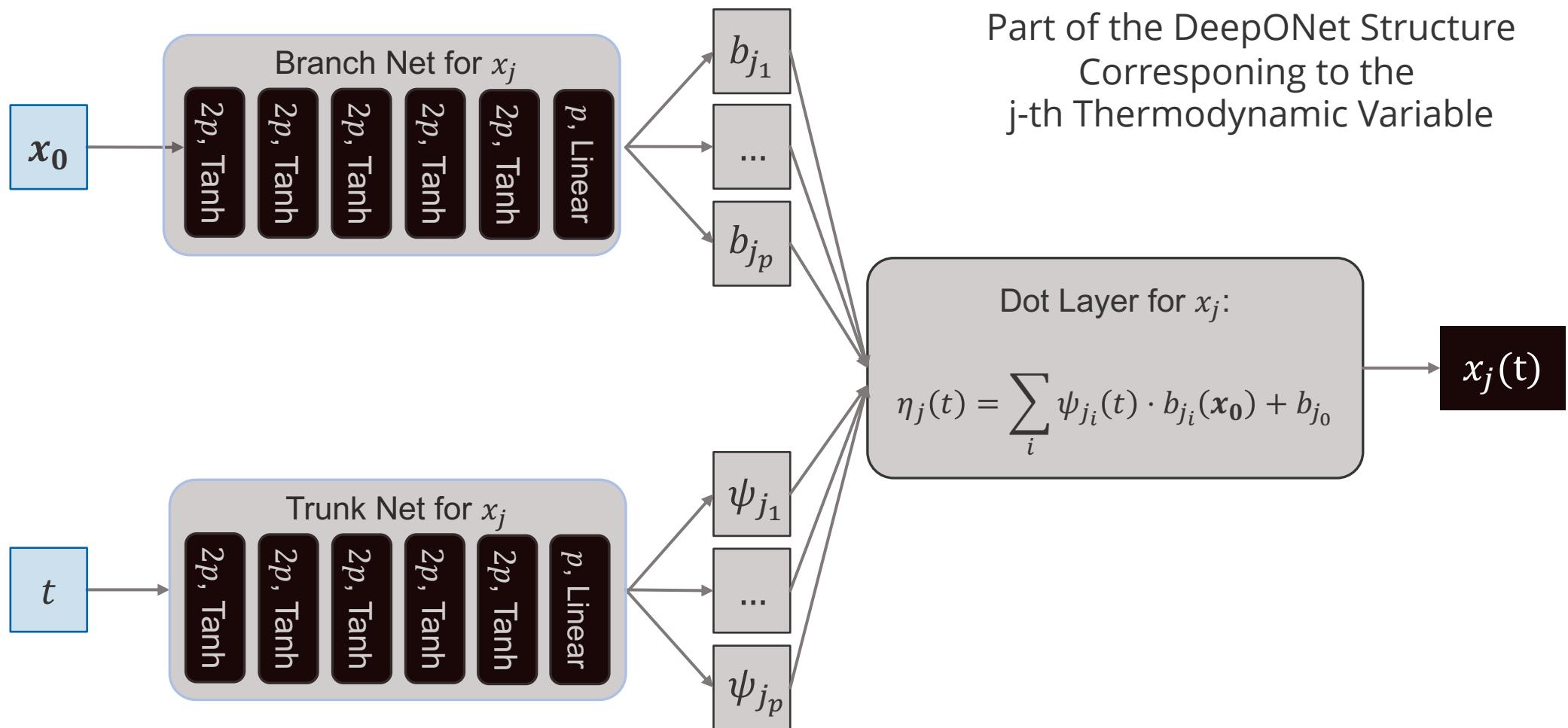


## Test Case 1 Data-Driven Vanilla DeepONet in the Thermodynamic State Space

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

- 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:
  - 3.2.1. "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\_Orig.ipynb

### Relevant Input Variables:

**self.data\_preproc\_type:** if `self.norm_input_flg/self.norm_output_flg == True`, 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.



## Test Case 2 Data-Driven POD-DeepONet in the Thermodynamic State Space

# 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:

$$\mathbf{H}_2 = \begin{bmatrix} | & | & & | & | \\ \mathbf{H}_{2,1} & \mathbf{H}_{2,2} & \dots & \mathbf{H}_{2,499} & \mathbf{H}_{2,500} \\ | & | & & | & | \end{bmatrix}$$

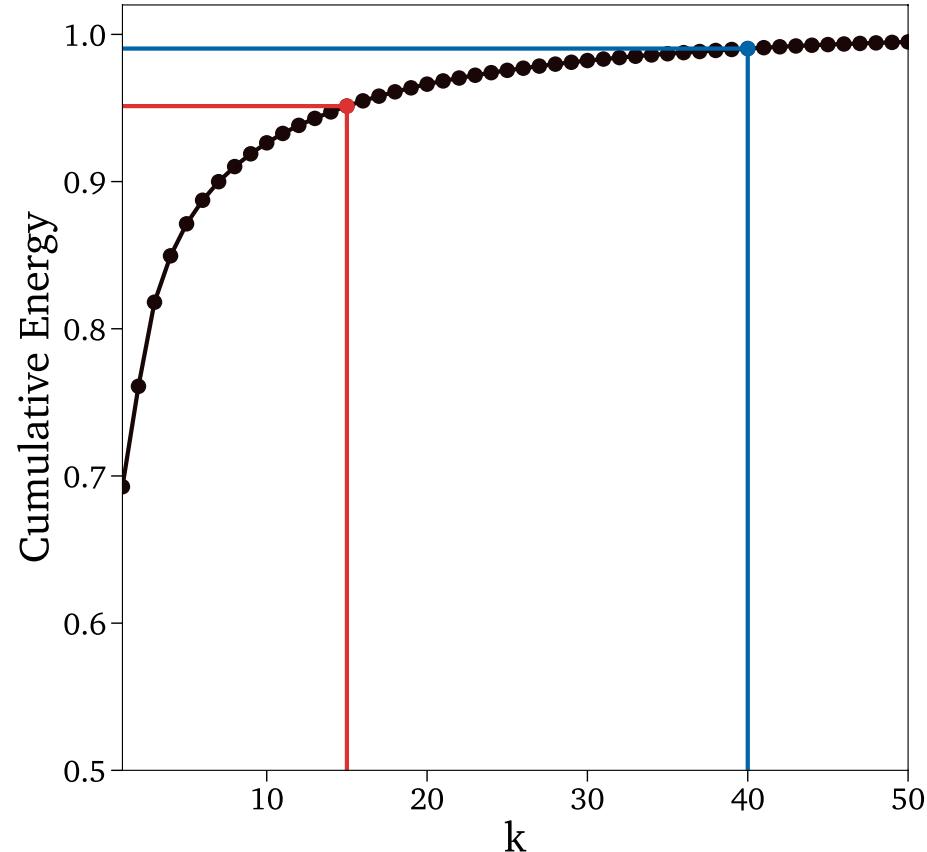
$$\dim(\mathbf{H}_2) = N_t \times N_s$$

No. time  
instants      No. of  
scenarios

# A Combustion Chemistry in Isobaric Reactor Test Case

Eigendecomposition of  $R_{H_2}$ :

$$\Psi_{H_2} = \frac{H_2 - C_{H_2}}{D_{H_2}} A_{H_2}$$

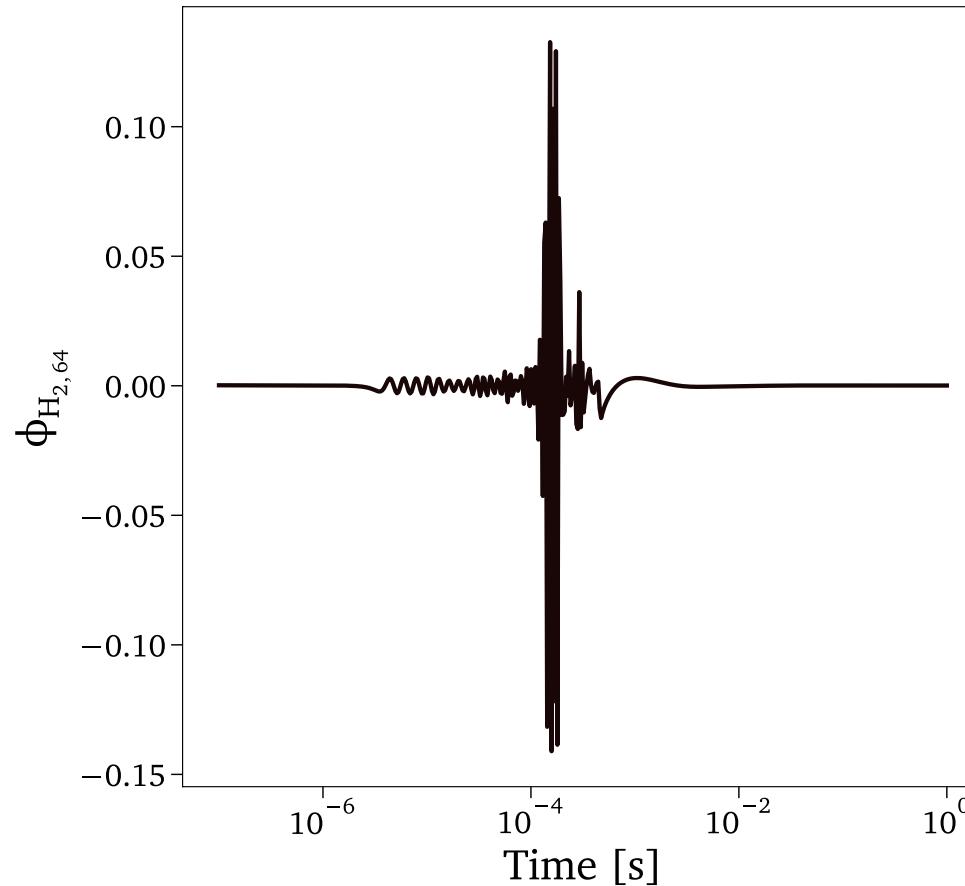
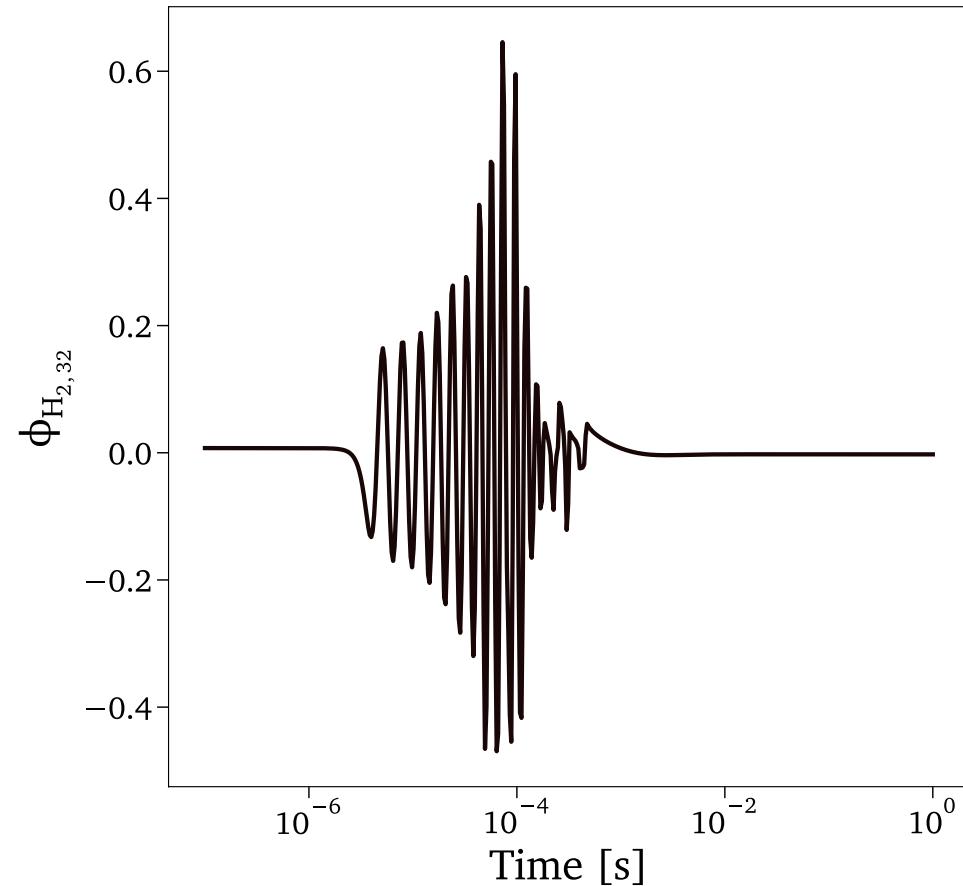


A relatively large number of modes needs to be preserved in order to predict T with good accuracy

# 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/input/ScenarioAggregated_ROMs/0DReactor/FNN/Trunk/Parallelize_ROMNet.py  
for generating Scenario Aggregated SVD data
```

# A Combustion Chemistry in Isobaric Reactor Test Case



## Test Case 2: Data-driven POD-deep operator network (POD-DeepONet) for predicting Thermodynamic Variables

### Train the Trunks:

2.1. Run \$WORKSPACE\_PATH/ROMNet/romnet/input/ScenarioAggregated\_ROMs/0DReactor/FNN/Trunk/Parallelize\_ROMNet.py

### Train the Rest of the DeepONet:

2.2. Copy \$WORKSPACE\_PATH/ROMNet/romnet/input/0DReact/DeepONet/0DReact\_H2\_TestCase2/ROMNet\_Input.py  
to \$WORKSPACE\_PATH/ROMNet/romnet/input/ROMNet\_Input.py

2.3. In \$WORKSPACE\_PATH/ROMNet/romnet/input/ROMNet\_Input.py, change:  
2.3.1. "self.WORKSPACE\_PATH = ..."

2.4. Move to \$WORKSPACE\_PATH/ROMNet/romnet/app/

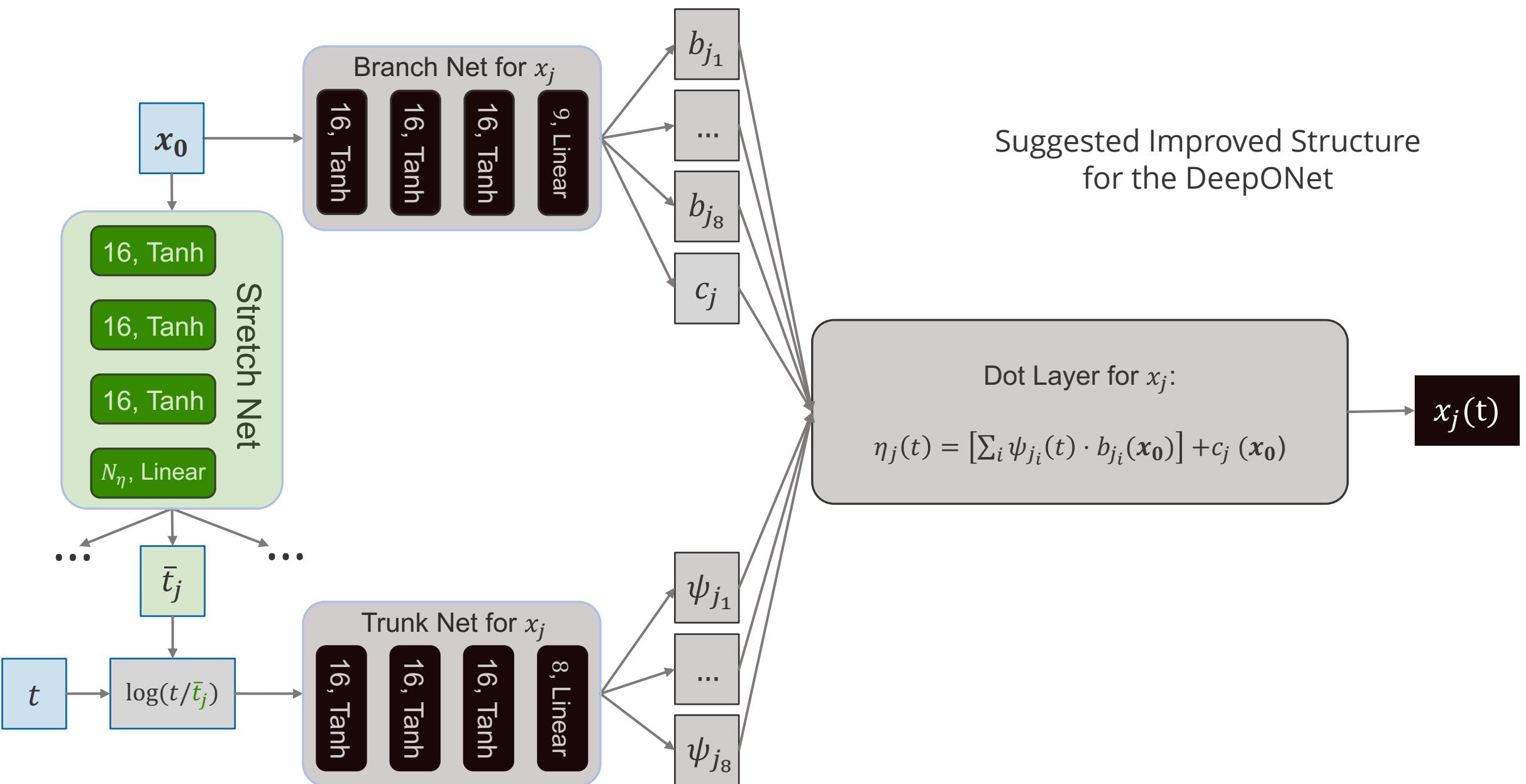
2.5. Run: "python3 ROMNet.py ..input/"

2.6. Postprocess results via: \$WORKSPACE\_PATH/ROMNet/romnet/scripts/postprocessing/0DReact/DeepONet/Predict\_DeepONet\_Orig.ipynb



## Test Case 3 Data-Driven flexDeepONet in the Thermodynamic State Space

# A Combustion Chemistry in Isobaric Reactor Test Case

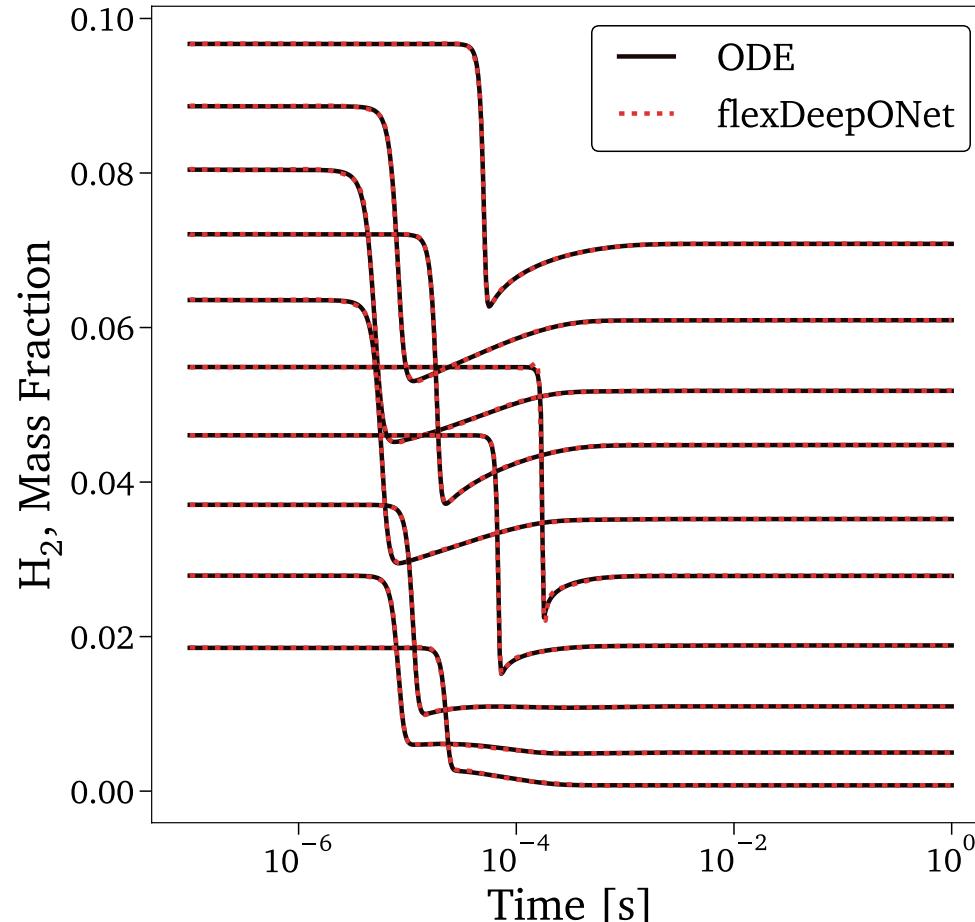
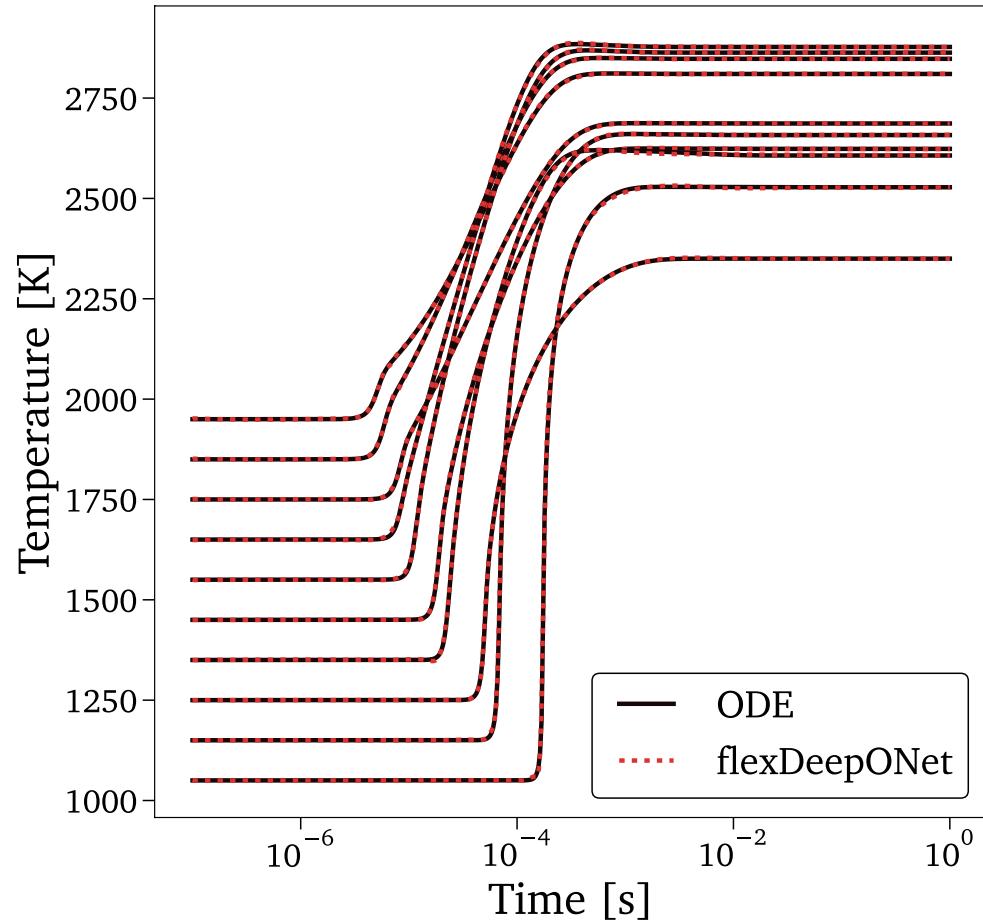


# A Combustion Chemistry in Isobaric Reactor Test Case



Results from the improved structure

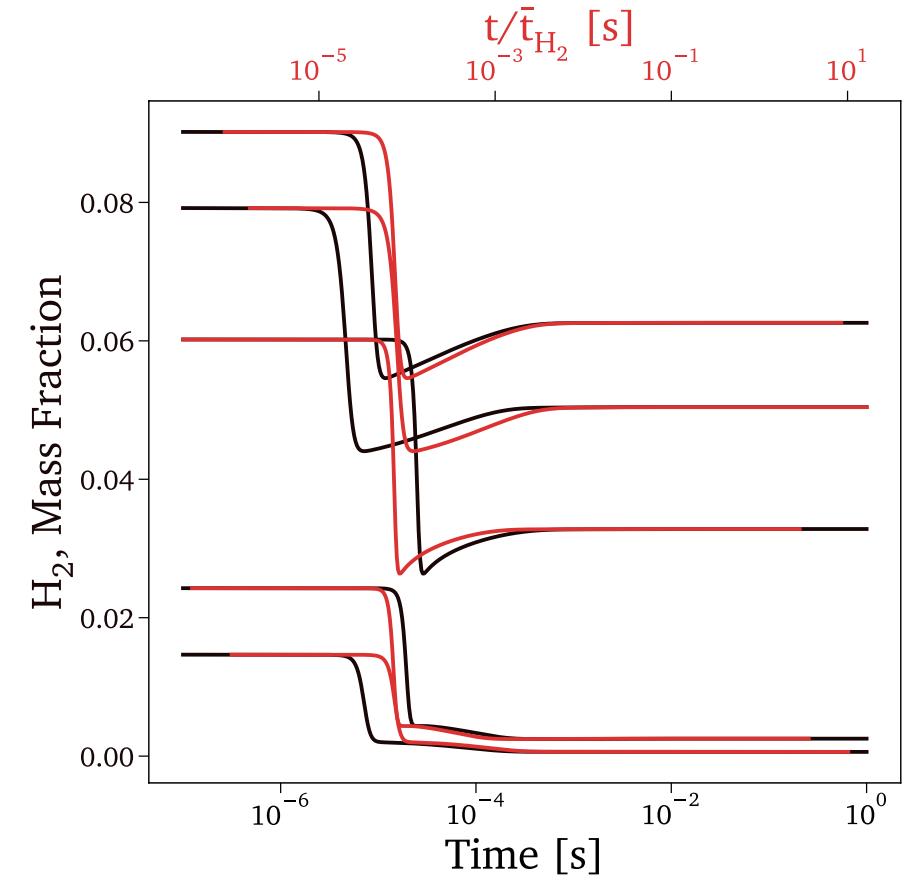
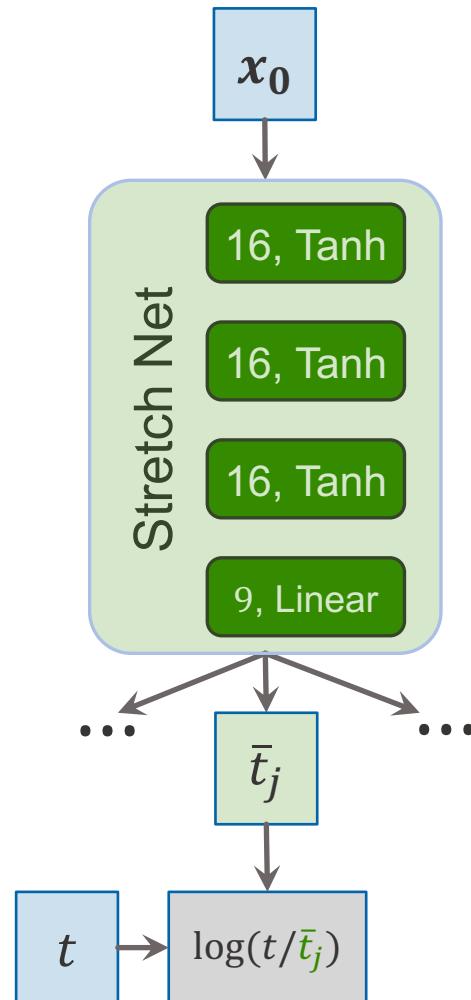
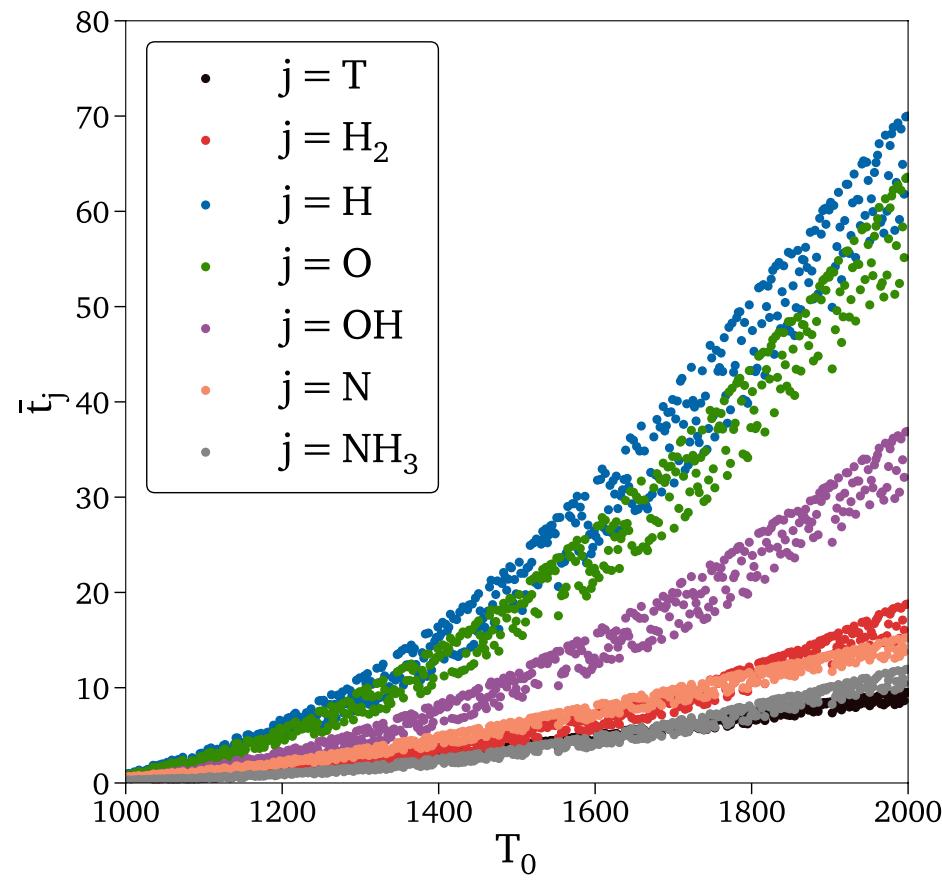
Predicted time-dependent temperature and species for test scenarios



# A Combustion Chemistry in Isobaric Reactor Test Case



Results from the improved structure



# A Combustion Chemistry in Isobaric Reactor Test Case



## **Test Case 3: Data-driven flexible deep operator network (flexDeepONet) 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

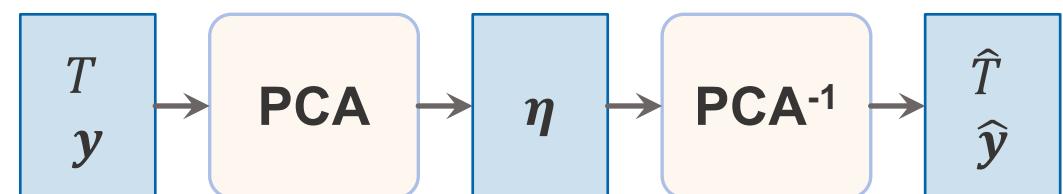
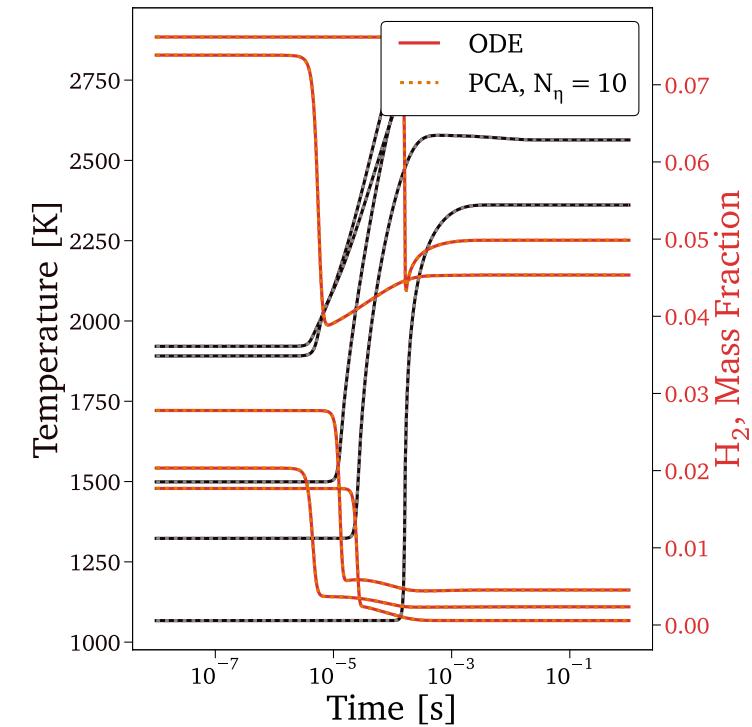
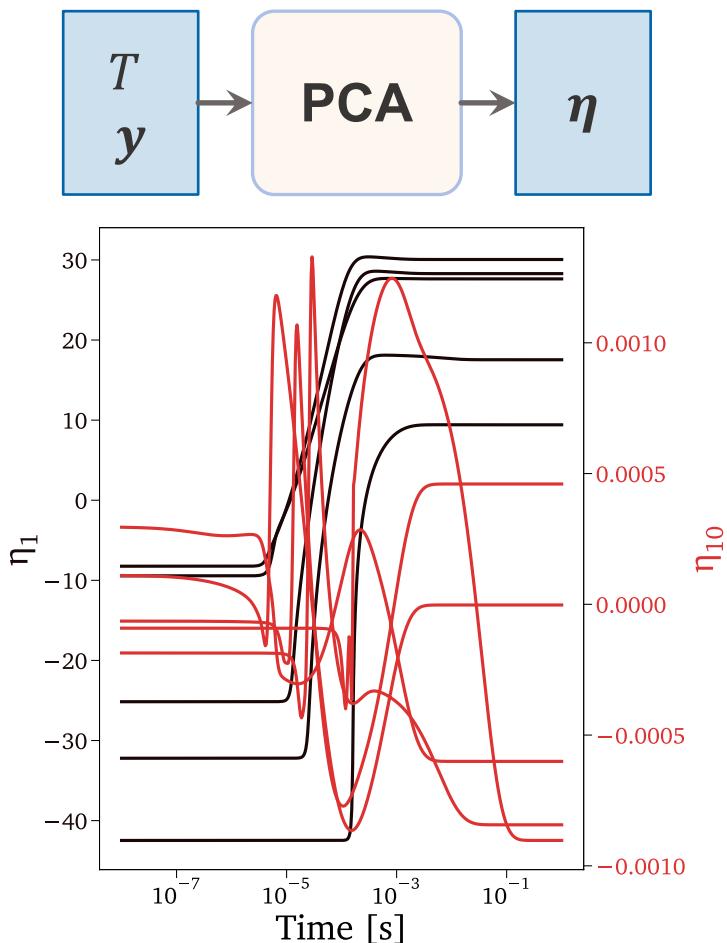


## Test Case 5 Data-Driven Vanilla DeepONet in the Principal Component Space

# A Combustion Chemistry in Isobaric Reactor Test Case



Employed PCA for reducing the dimensionality of the state space



10 principal components ( $\eta$ ) are sufficient for good accuracy

# A Combustion Chemistry in Isobaric Reactor Test Case



Run python scrip:

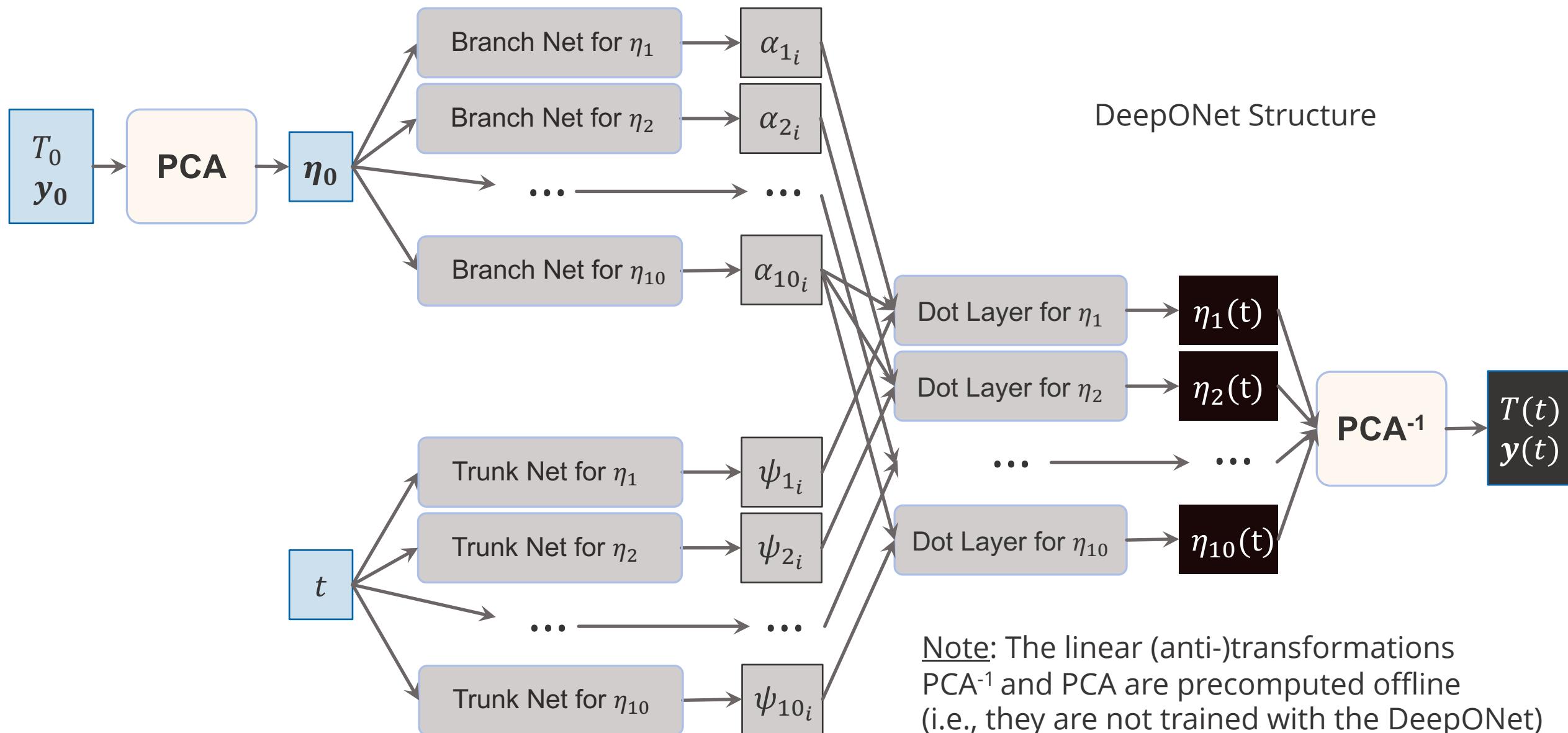
\$WORKSPACE\_PATH/ROMNet/romnet/scripts/generating\_data/0DReactor/Generate\_Data\_2\_Isobaric.py  
for generating PCA simulation data

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

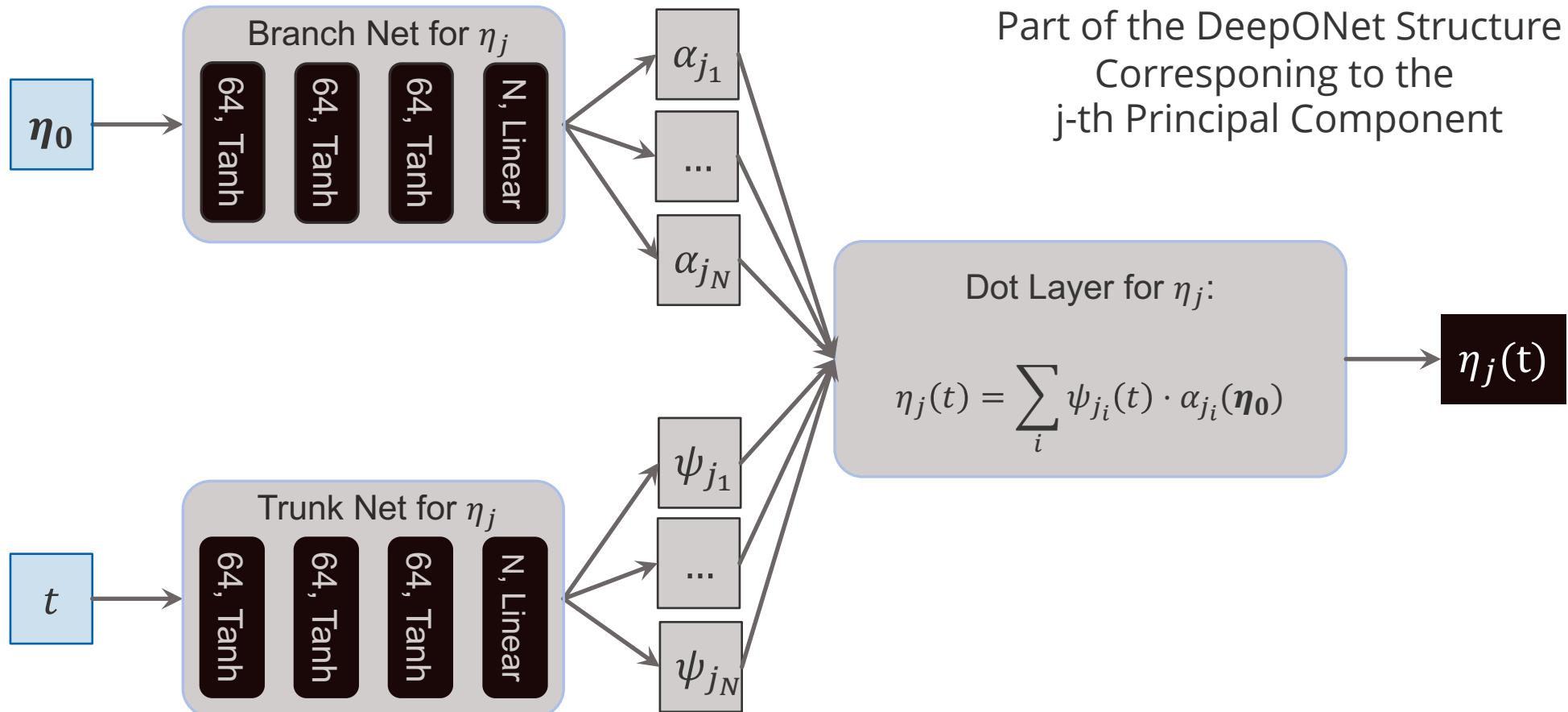
Run Jupyter Notebook:

\$WORKSPACE\_PATH/ROMNet/romnet/scripts/generating\_data/0DReactor/Generate\_Data\_4.ipynb  
for generating PCA training and test data

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

- 5.1. Copy \$WORKSPACE\_PATH/ROMNet/romnet/input/0DReact/DeepONet/0DReact\_H2\_TestCase5/ROMNet\_Input.py to \$WORKSPACE\_PATH/ROMNet/romnet/input/ROMNet\_Input.py
- 5.2. In \$WORKSPACE\_PATH/ROMNet/romnet/input/ROMNet\_Input.py, change:
  - 5.2.1. "self.WORKSPACE\_PATH = ..."
- 5.3. Move to \$WORKSPACE\_PATH/ROMNet/romnet/app/
- 5.4. Run: "python3 ROMNet.py ..//input/"
- 5.5. 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:

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

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

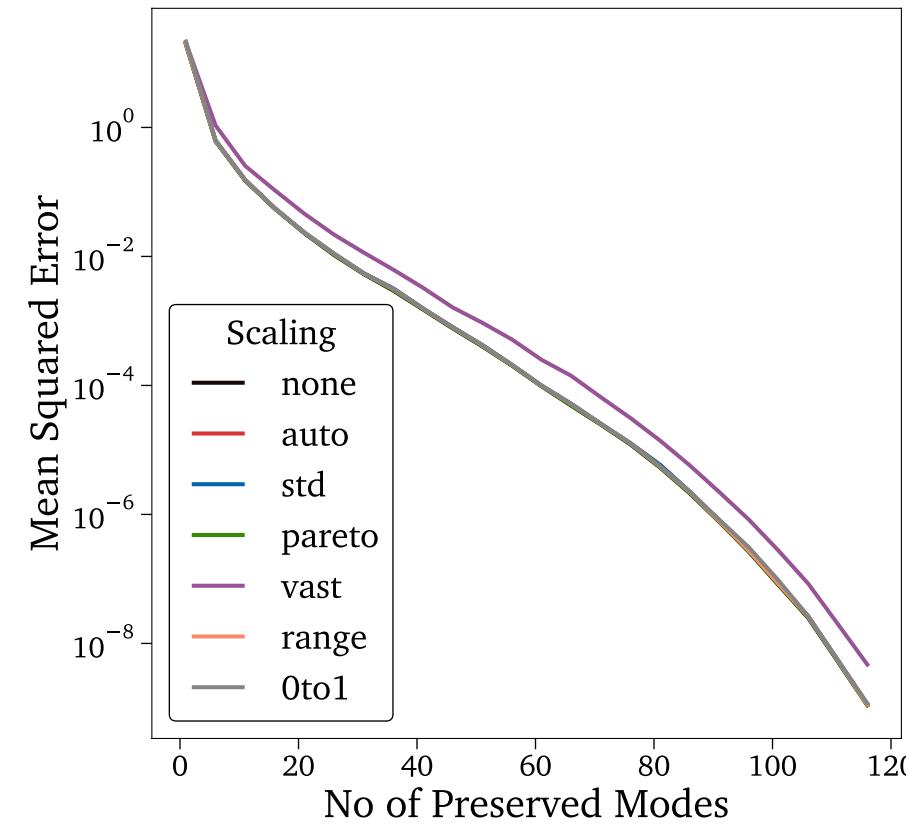
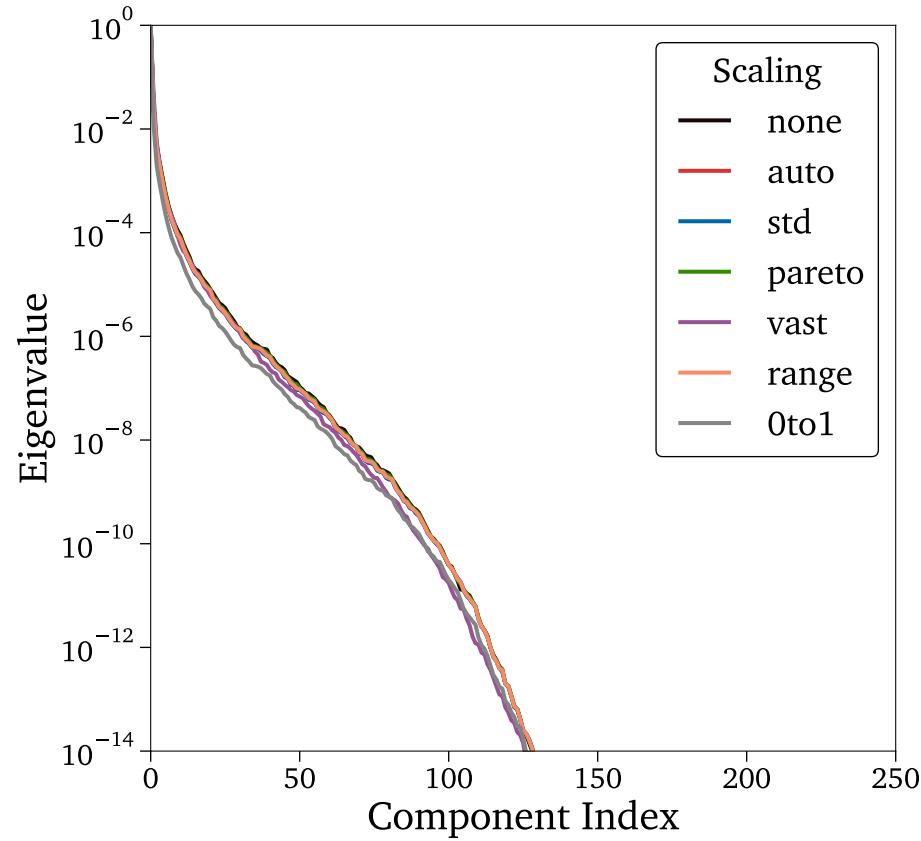
No. time    No. of  
instants    scenarios

# 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  $\eta_1$ ))

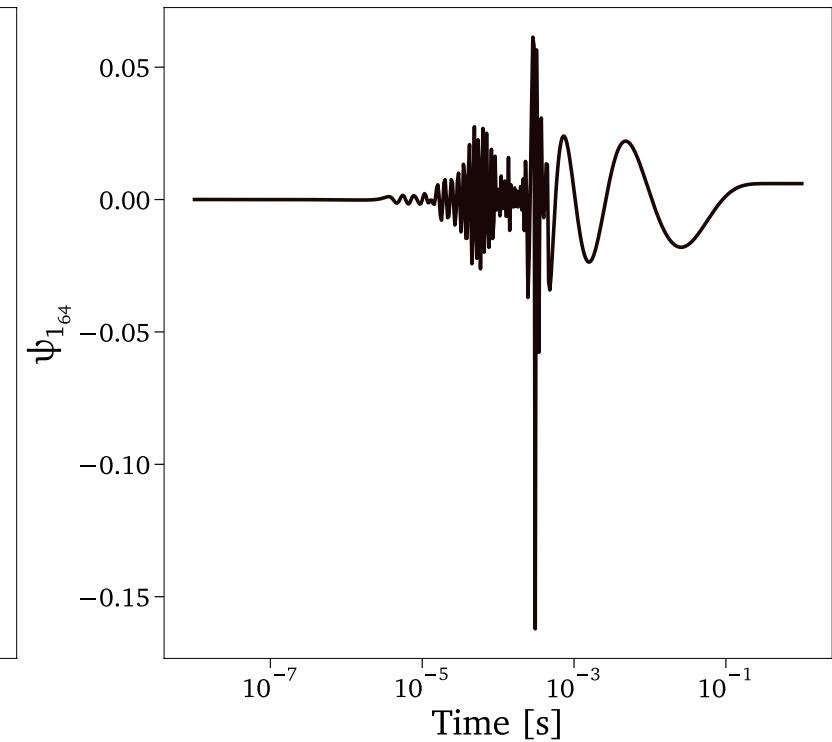
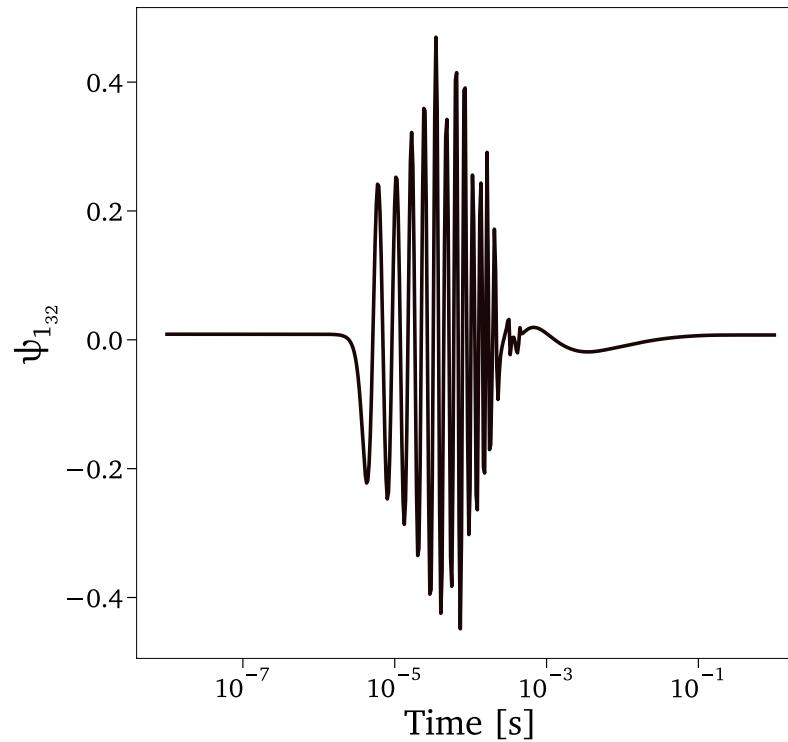
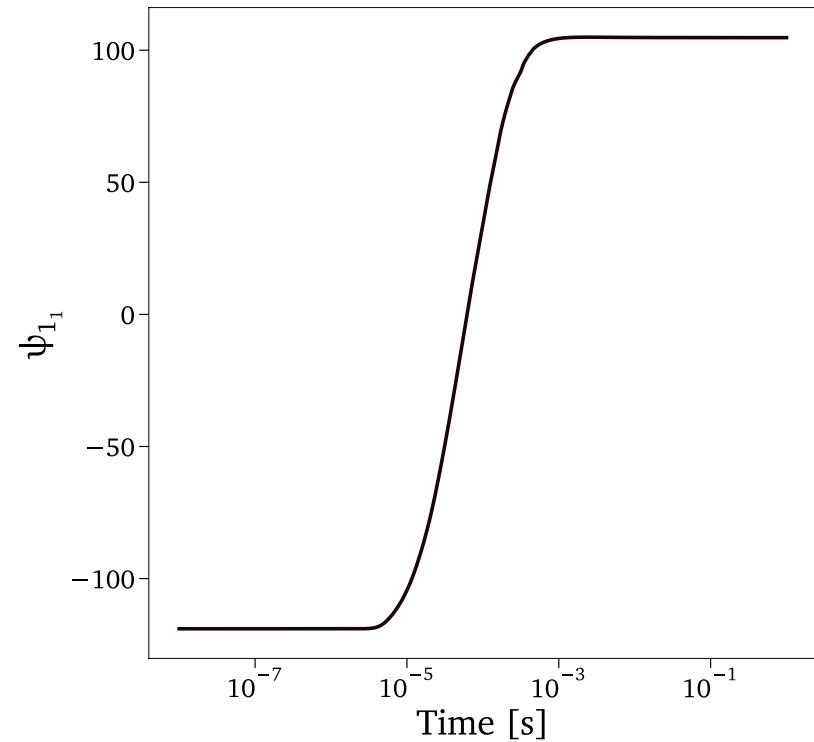


A relatively large number of modes needs to be preserved in order to predict  $\eta_j$  with good accuracy

# A Combustion Chemistry in Isobaric Reactor Test Case



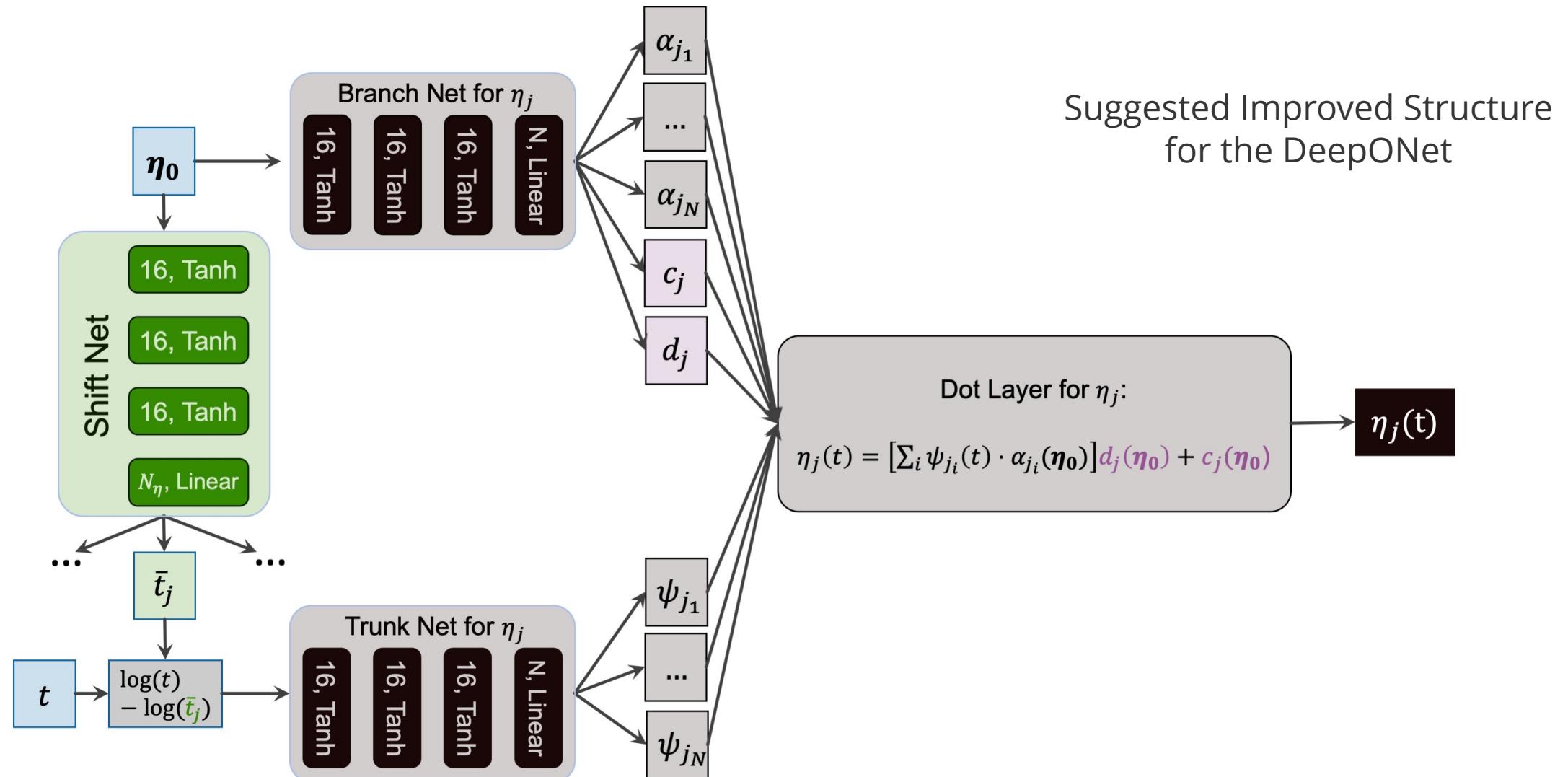
Low energy modes are highly oscillatory and hard to be learnt by the DeepONet's trunk nets





## Test Case 6 Data-Driven flexDeepONet in the Principal Component Space

# A Combustion Chemistry in Isobaric Reactor Test Case

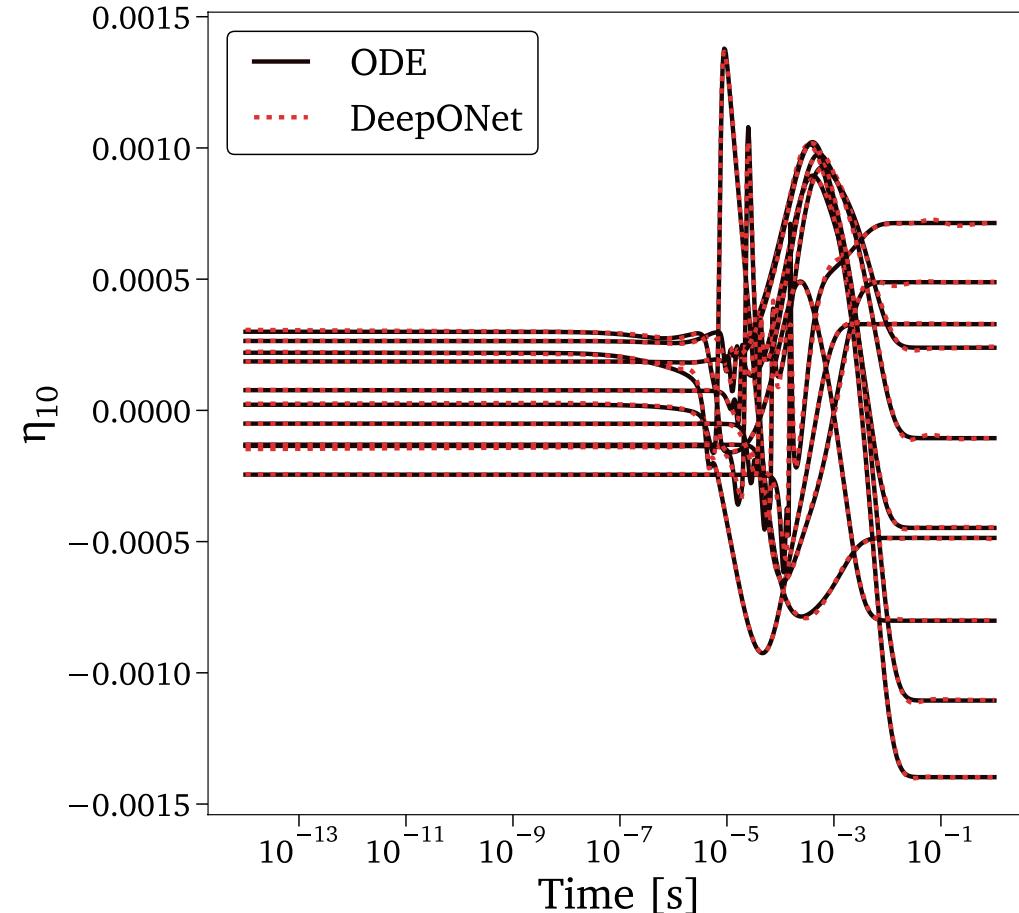
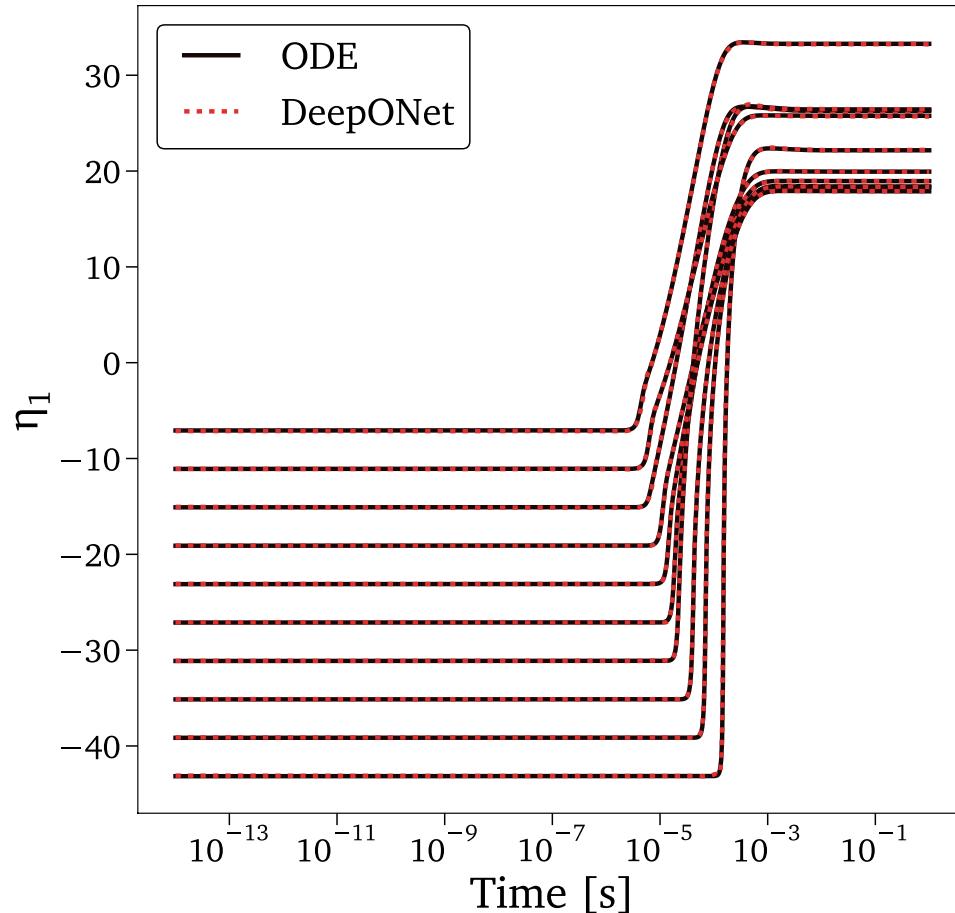


# A Combustion Chemistry in Isobaric Reactor Test Case



Results from the improved structure

Predicted time-dependent principal components for test scenarios

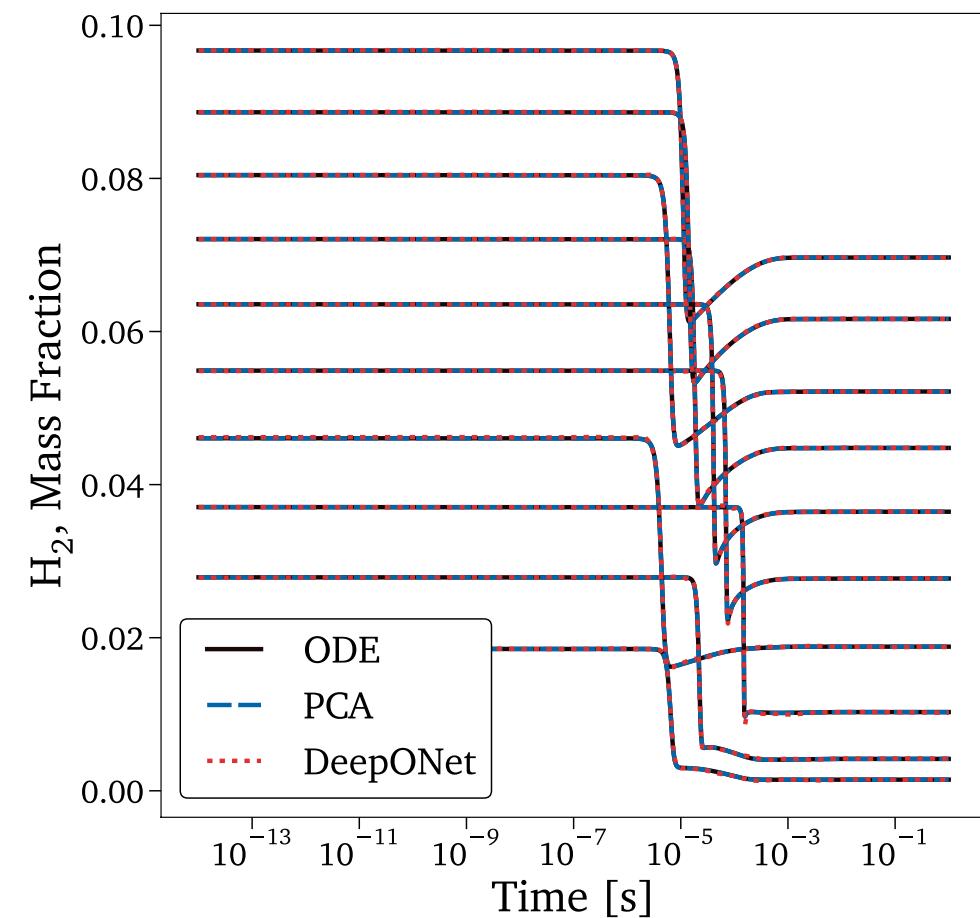
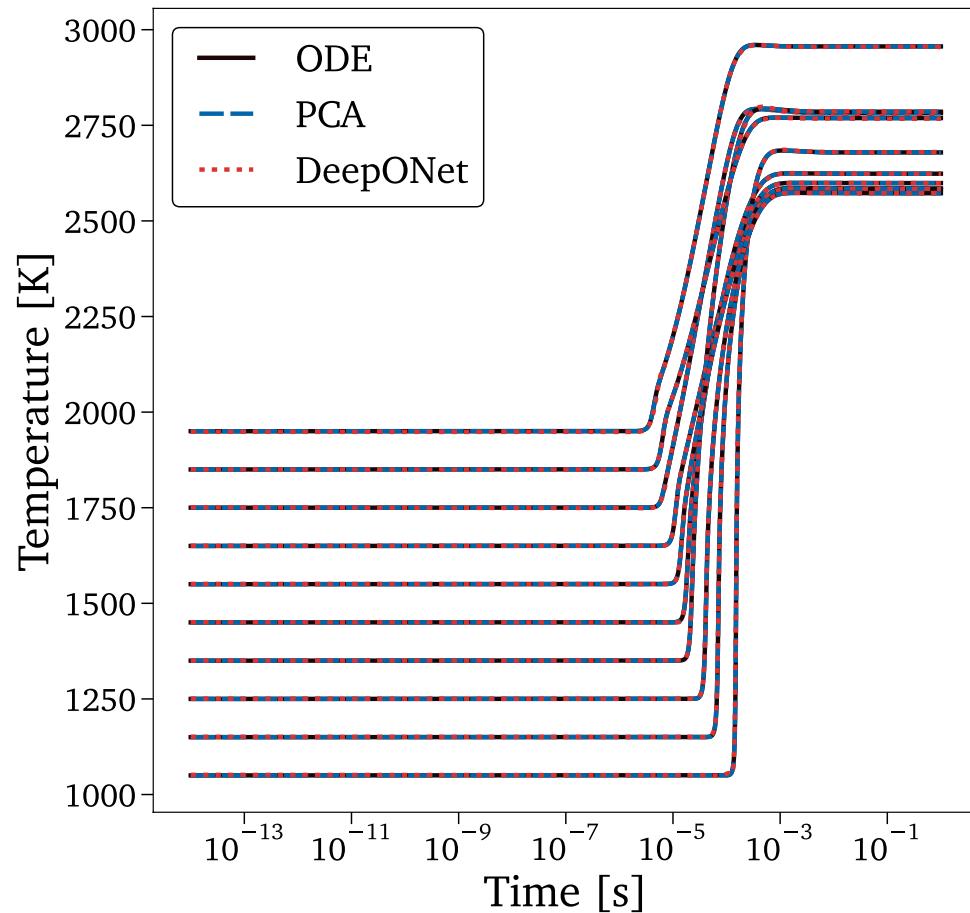


# A Combustion Chemistry in Isobaric Reactor Test Case



Results from the improved structure

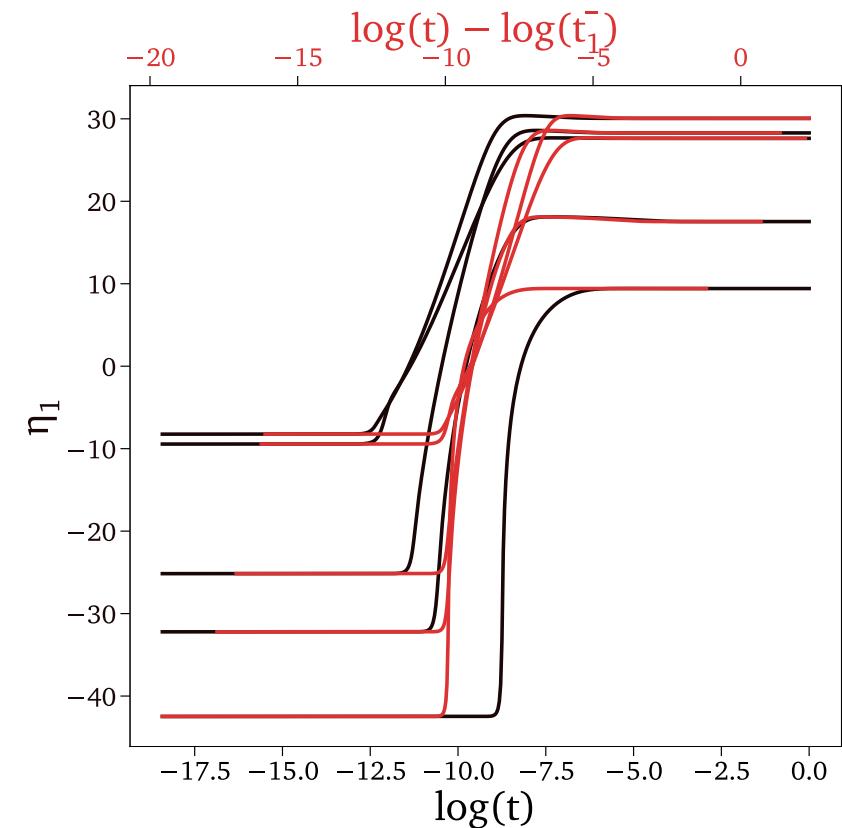
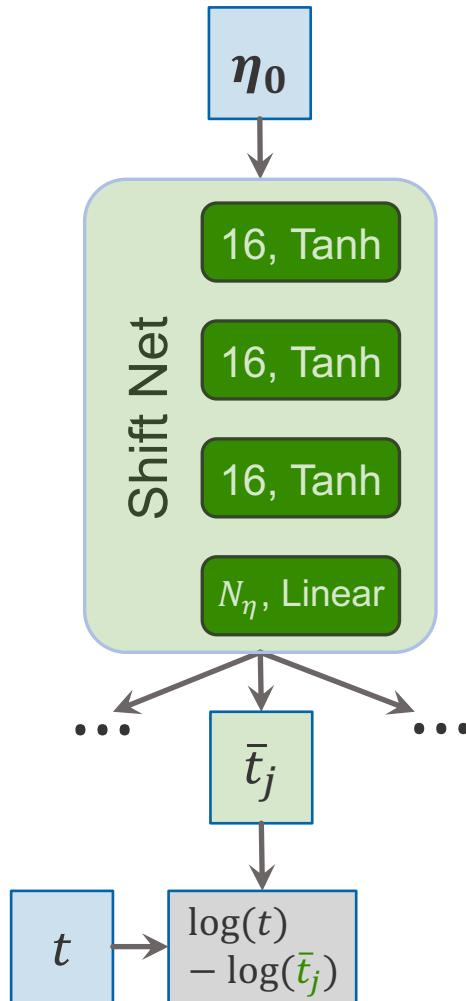
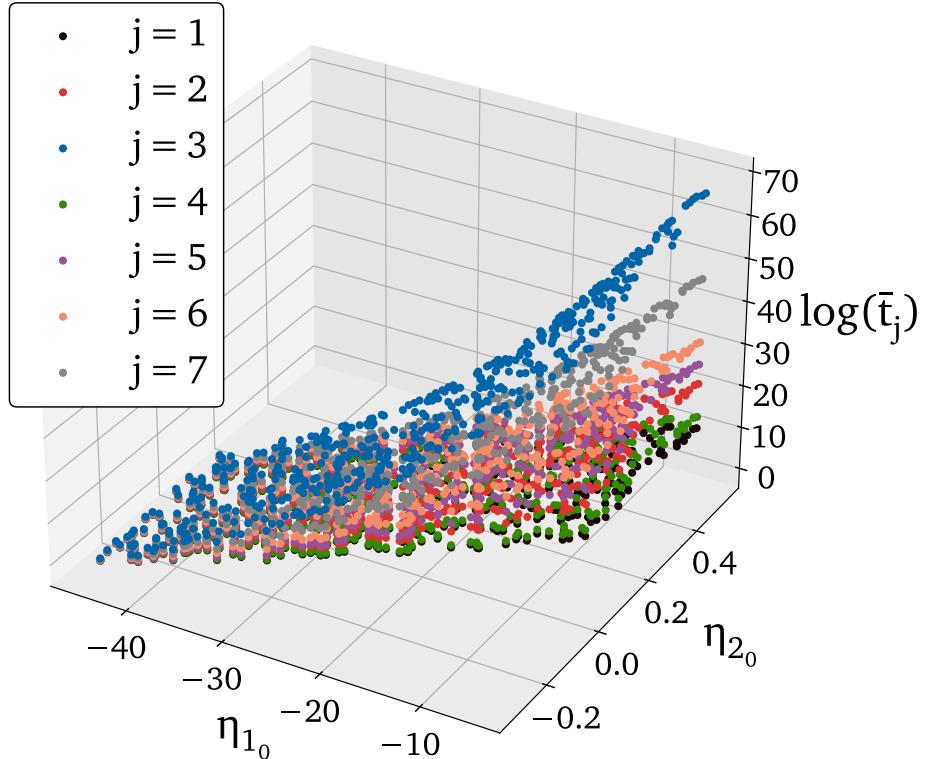
Reconstructed time-dependent temperature and species for test scenarios



# A Combustion Chemistry in Isobaric Reactor Test Case



Results from the improved structure



# A Combustion Chemistry in Isobaric Reactor Test Case



## Test Case 6: Data-driven flexible deep operator network (flexDeepONet) for predicting Principal Components

- 6.1. Copy \$WORKSPACE\_PATH/ROMNet/romnet/input/0DReact/DeepONet/0DReact\_H2\_TestCase6/ROMNet\_Input.py to \$WORKSPACE\_PATH/ROMNet/romnet/input/ROMNet\_Input.py
- 6.2. In \$WORKSPACE\_PATH/ROMNet/romnet/input/ROMNet\_Input.py, change:
  - 6.2.1. "self.WORKSPACE\_PATH = ..."
- 6.3. Move to \$WORKSPACE\_PATH/ROMNet/romnet/app/
- 6.4. Run: "python3 ROMNet.py ..//input/"
- 6.5. Postprocess results via: \$WORKSPACE\_PATH/ROMNet/romnet/scripts/postprocessing/0DReact/DeepONet/Predict\_DeepONet.ipynb



### Relevant Input Variables:

**self.data\_preproc\_type:** if `self.norm_input_flg`/`self.norm_output_flg` == True, 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.



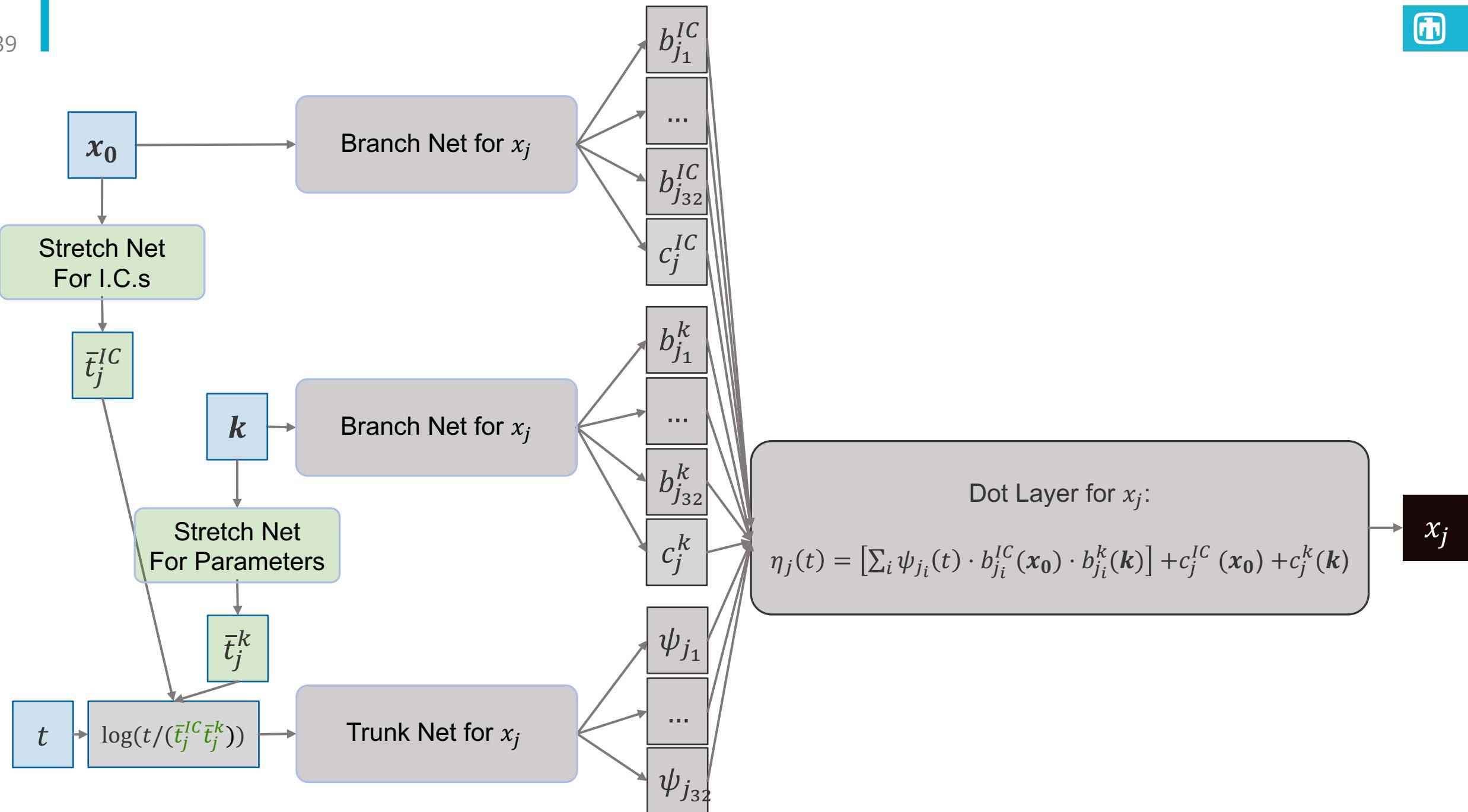
## Test Case 10

# A Combustion Chemistry in Isobaric Reactor Test Case

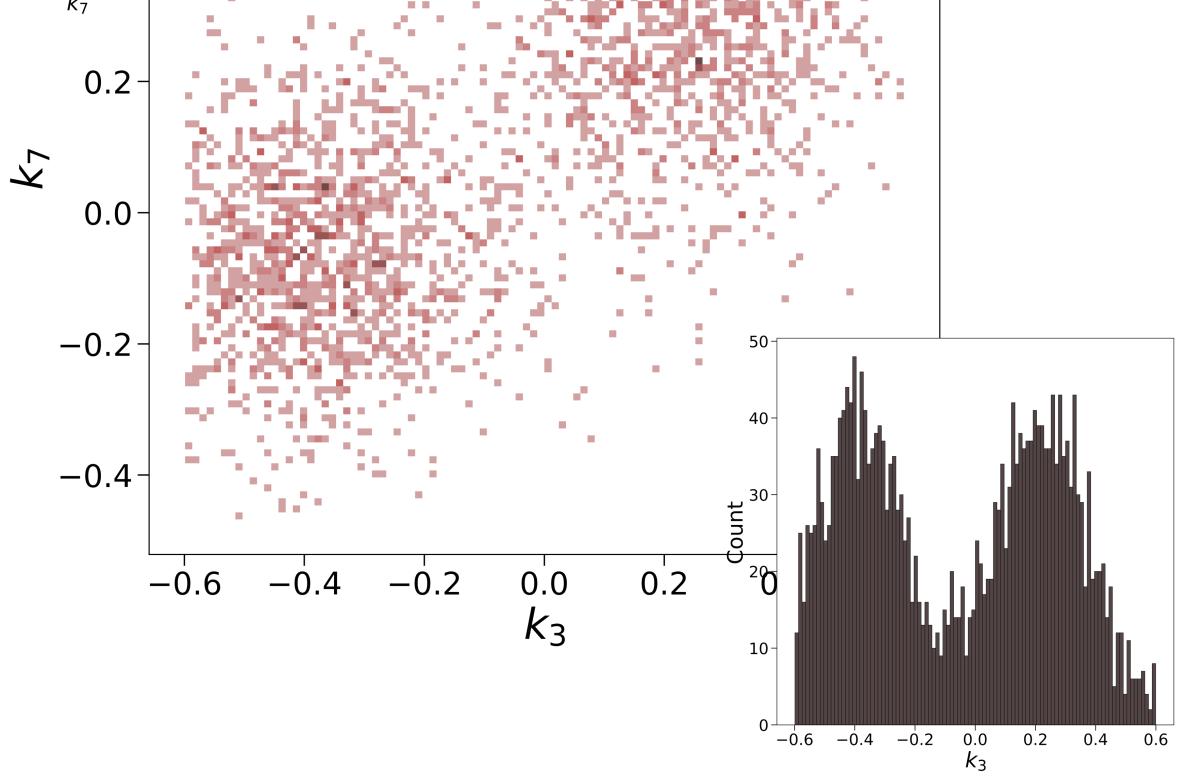
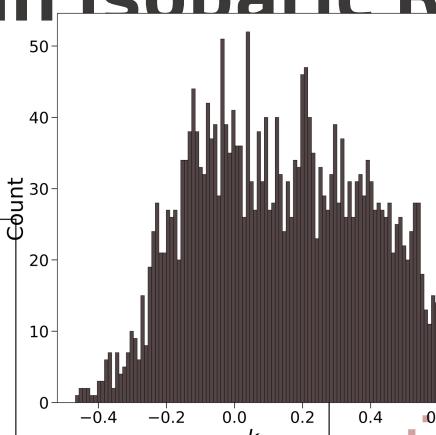
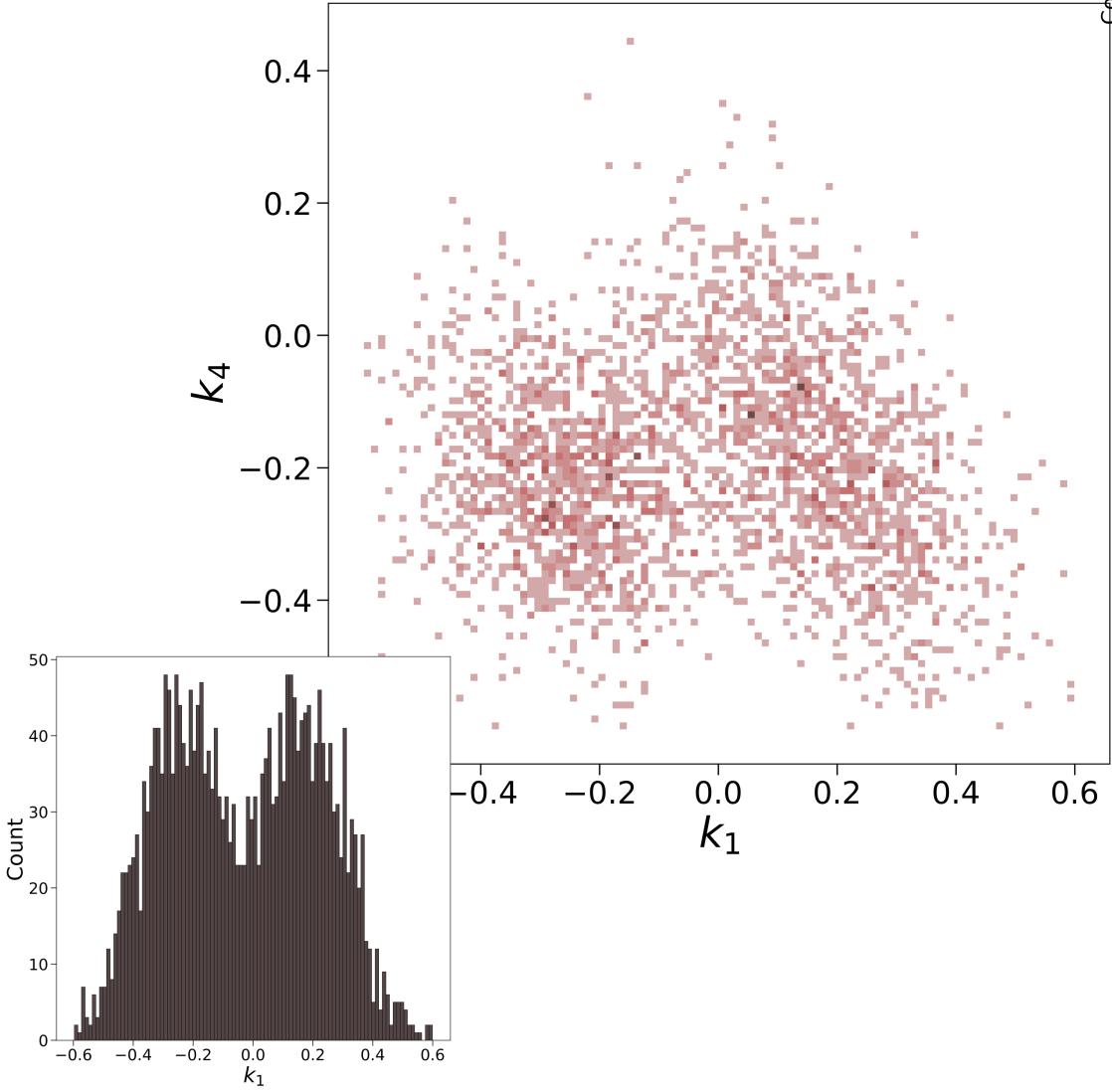


2,000,000 training data points

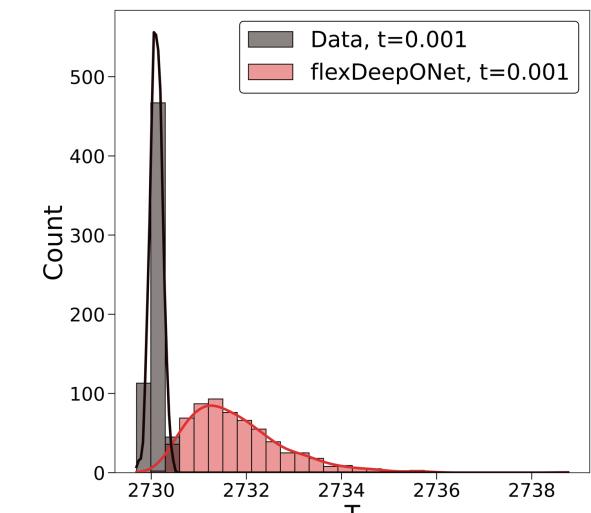
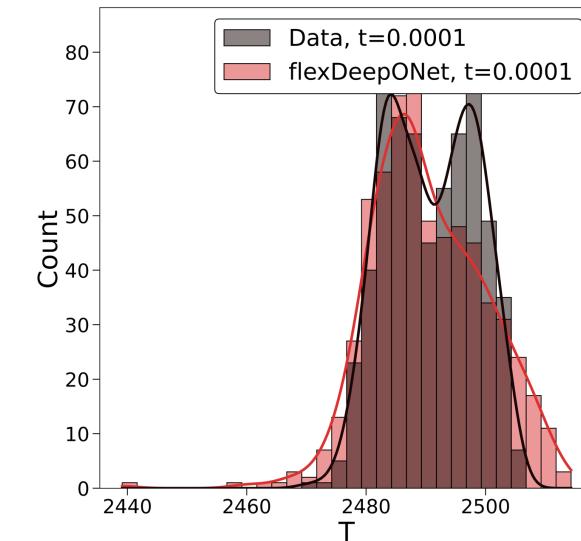
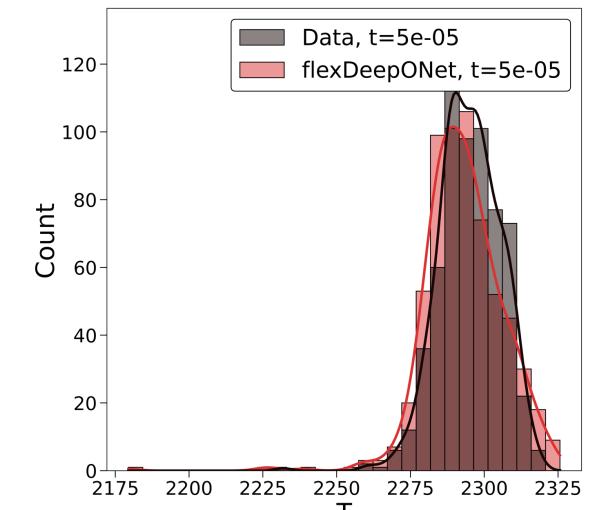
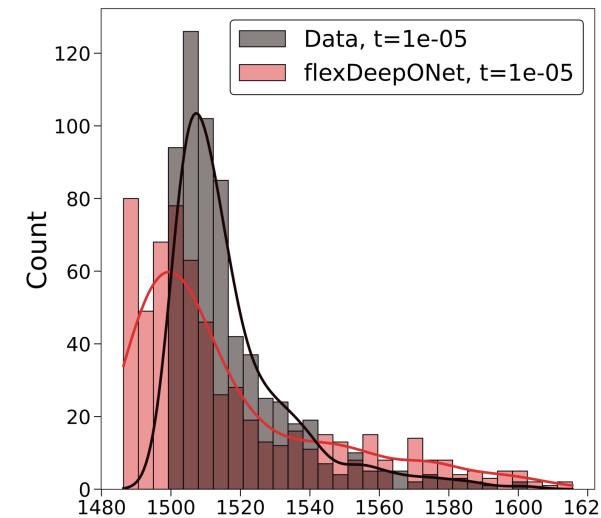
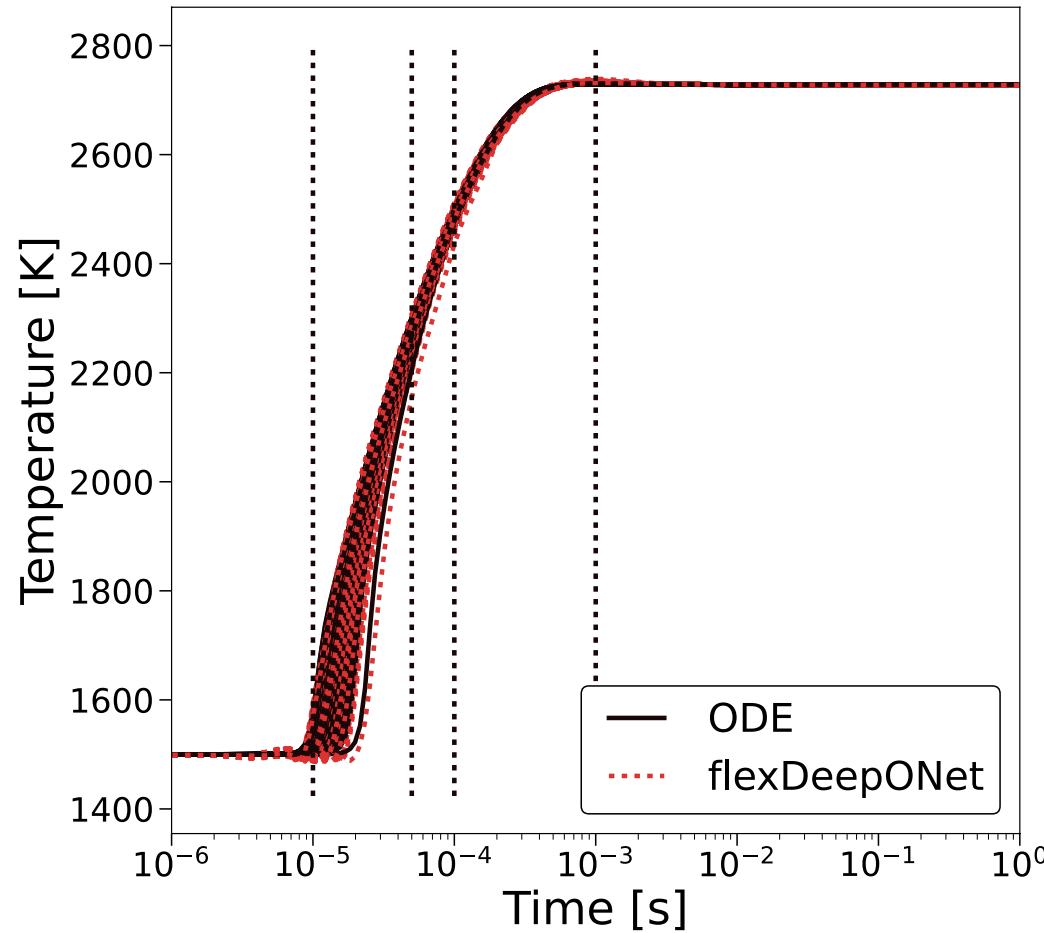
- 10,000 scenarios
  - 100 I.C.s ( $T=1,000\text{-}2,000\text{ K}$  and  $E.F.=0.5\text{-}4$ )
    - x
    - 100 parameter space samples (Perturbed pre-exponential coefficients  $\pm 90\%$ )
      1.  $O + H_2 \rightarrow H + OH$
      2.  $H + O_2 \rightarrow O + OH$
      3.  $2H + M \rightarrow H_2 + M$
      4.  $N + NO \rightarrow N_2 + O$
      5.  $N + O_2 \rightarrow NO + O$
      6.  $N + OH \rightarrow NO + H$
      7.  $N_2O + O \rightarrow N_2 + O_2$
      8.  $NH + N \rightarrow N_2 + H$
    - 200 time instants per scenario



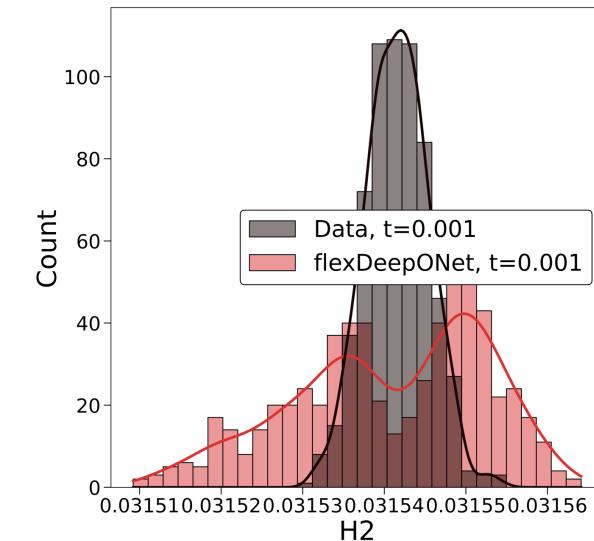
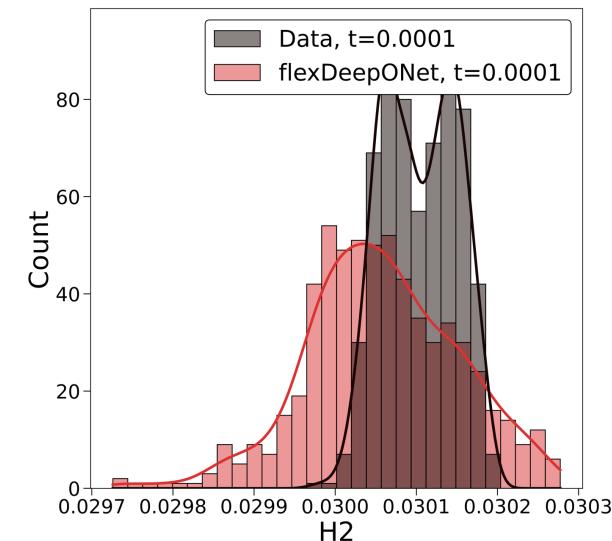
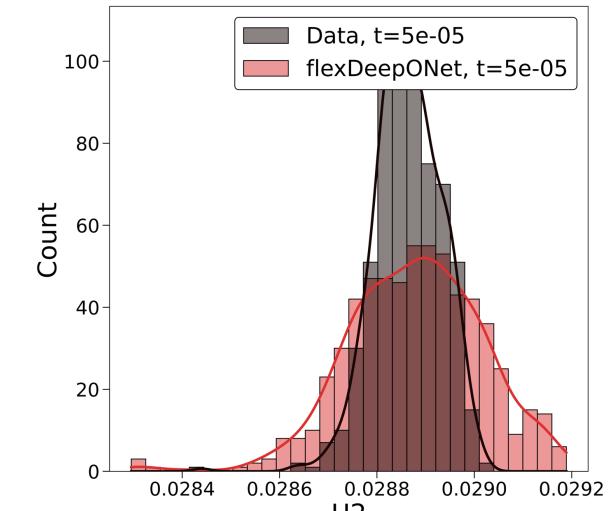
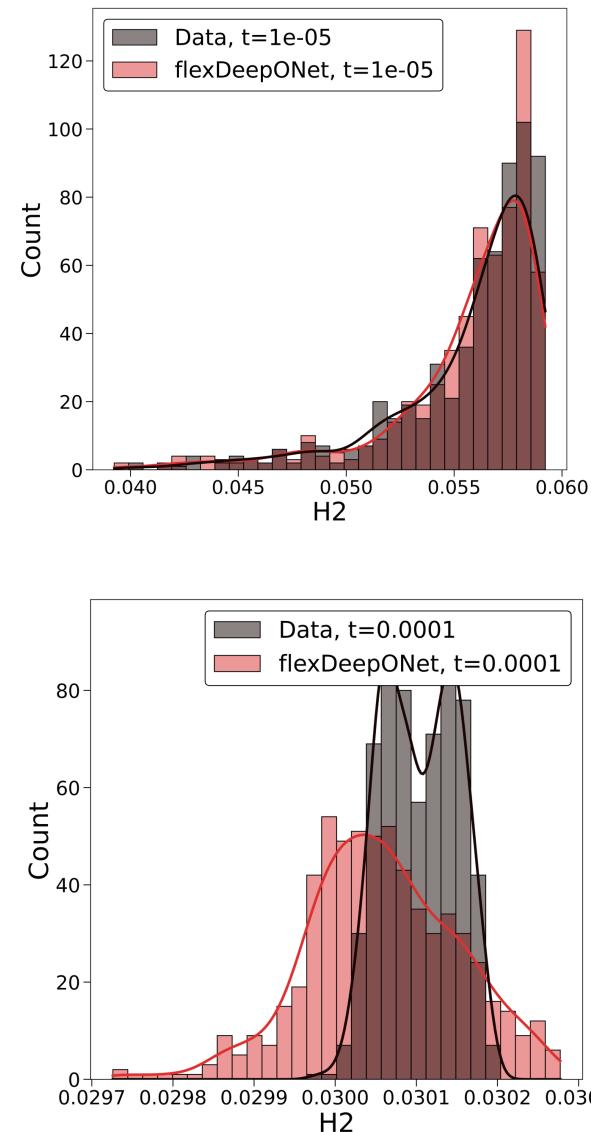
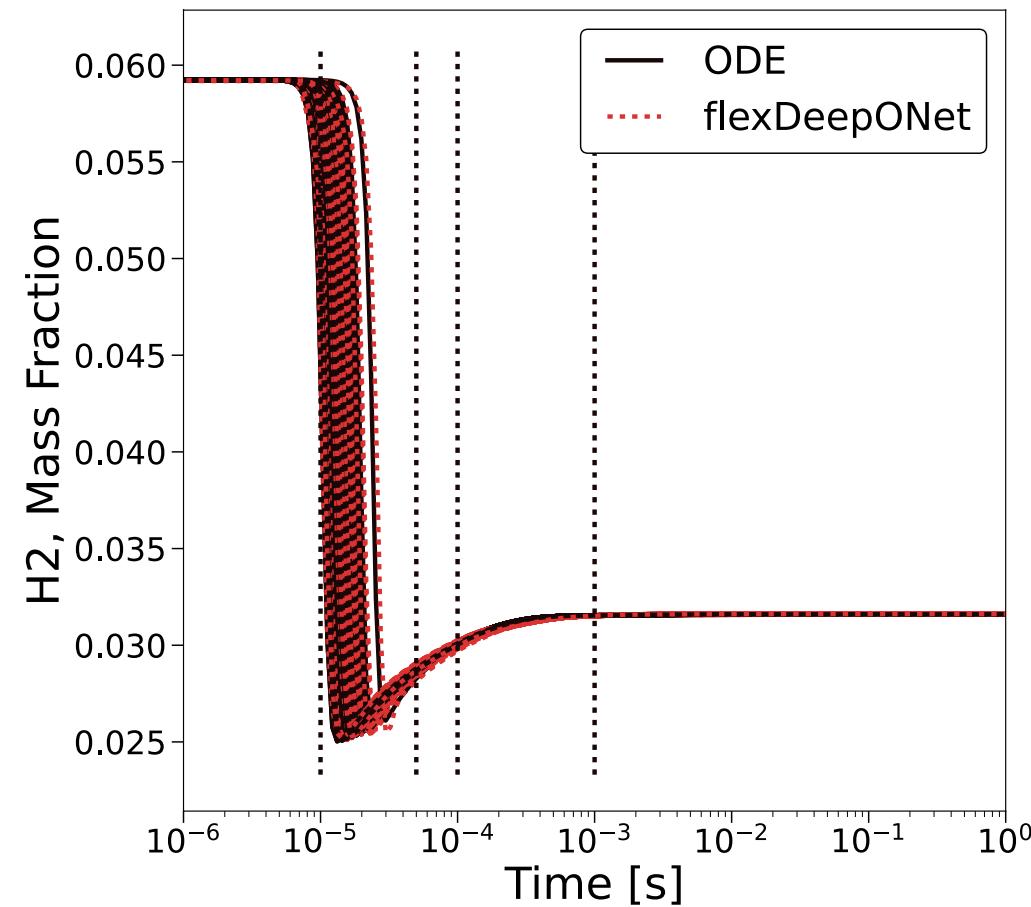
# A Combustion Chemistry in Isobaric Reactor Test Case



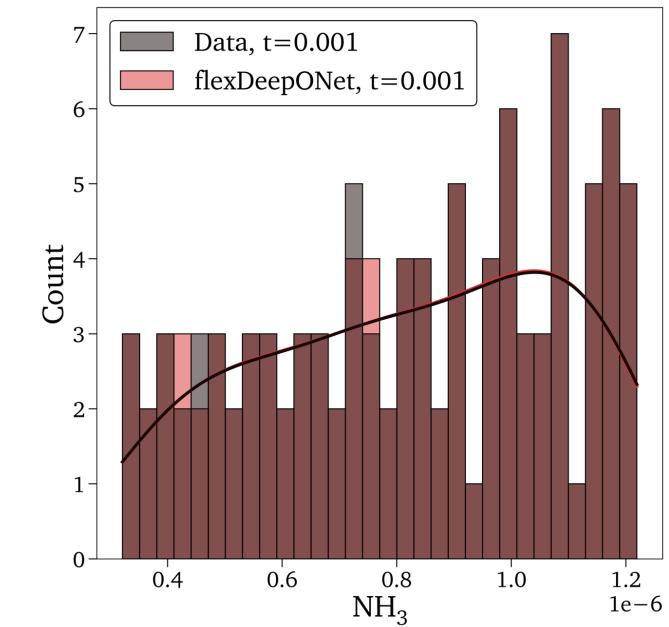
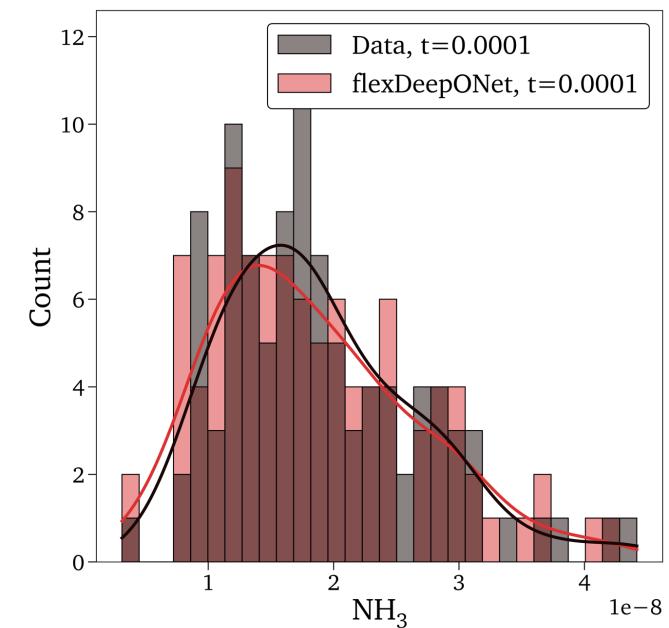
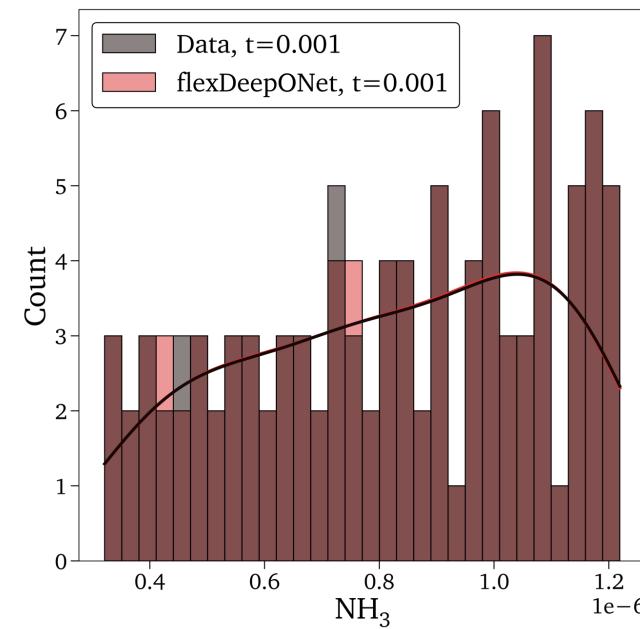
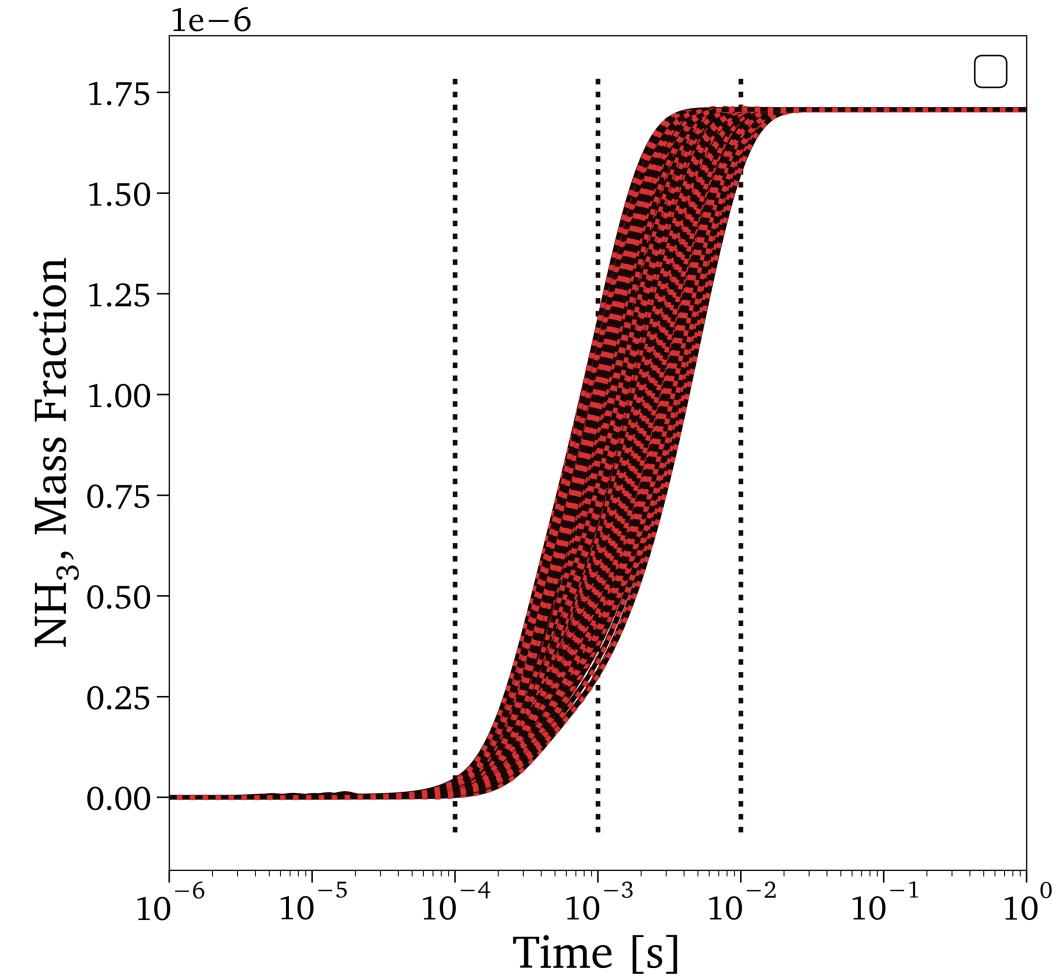
# A Combustion Chemistry in Isobaric Reactor Test Case



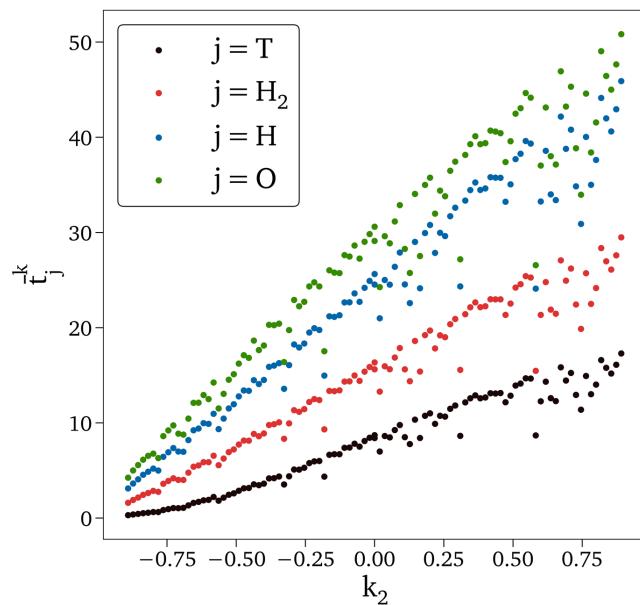
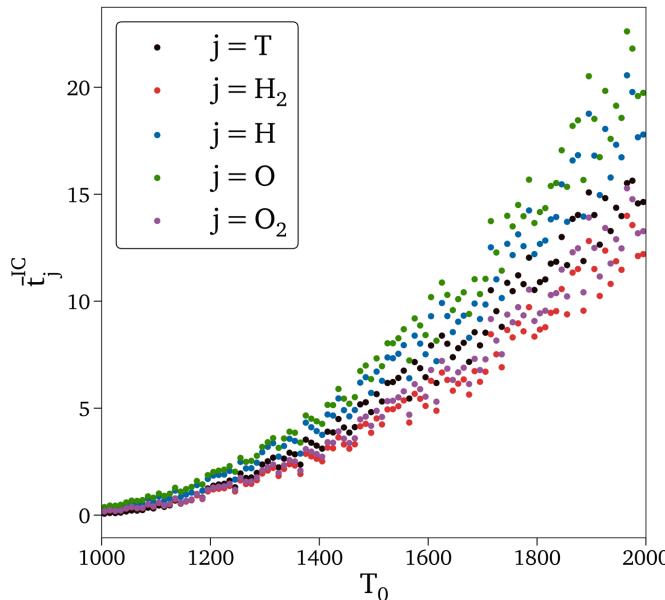
# A Combustion Chemistry in Isobaric Reactor Test Case



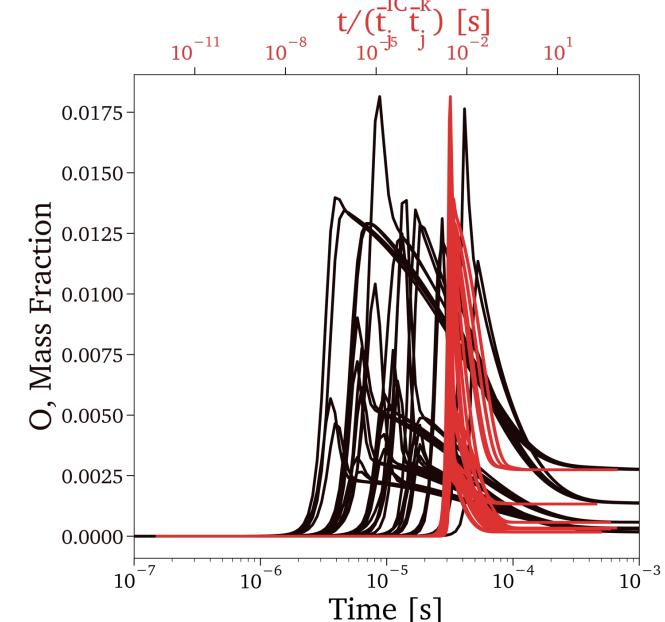
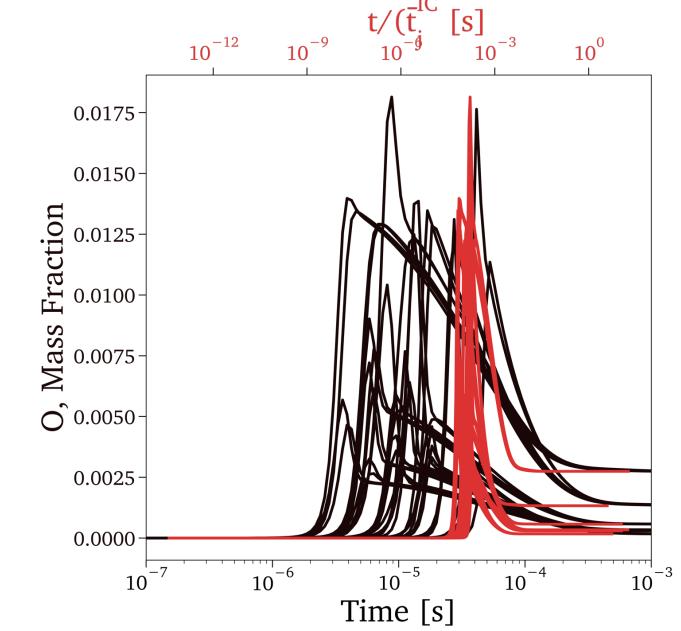
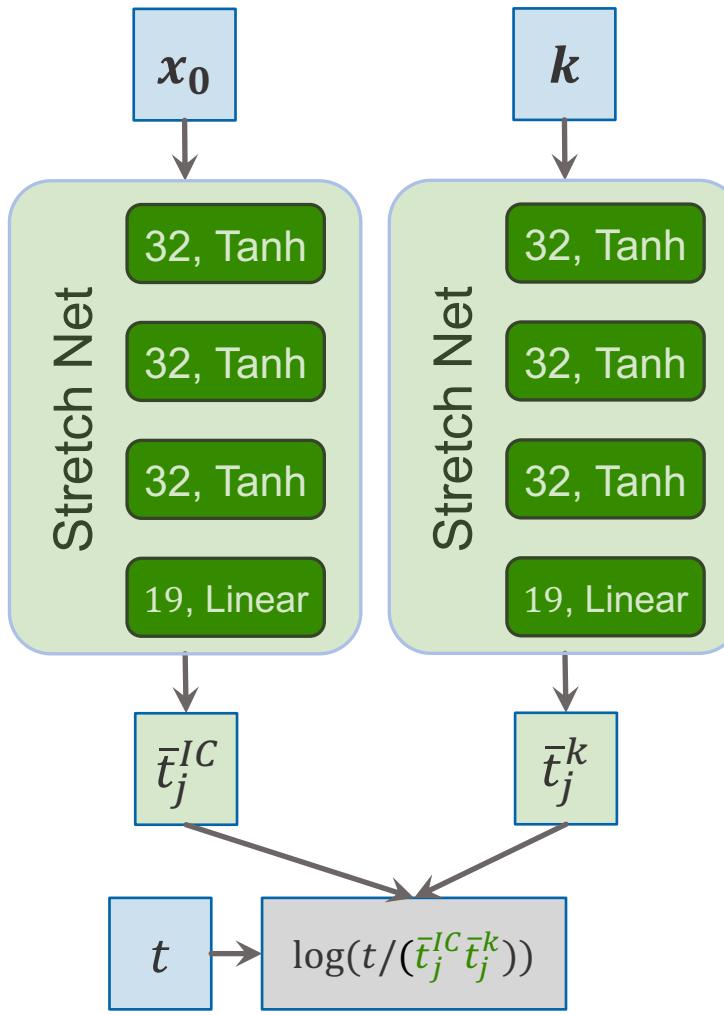
# A Combustion Chemistry in Isobaric Reactor Test Case



# A Combustion Chemistry in Isobaric Reactor Test Case



Results from the improved structure



# A Combustion Chemistry in Isobaric Reactor Test Case



## **Test Case 10: Data-driven MIONet for predicting Thermodynamic Variables**

- 10.1. Copy \$WORKSPACE\_PATH/ROMNet/romnet/input/0DReact/MIONet/0DReact\_H2\_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



## Test Case 11 Data-Driven DeepONet with Frozen Trunk in the Thermodynamic State Space

# A Combustion Chemistry in Isobaric Reactor Test Case



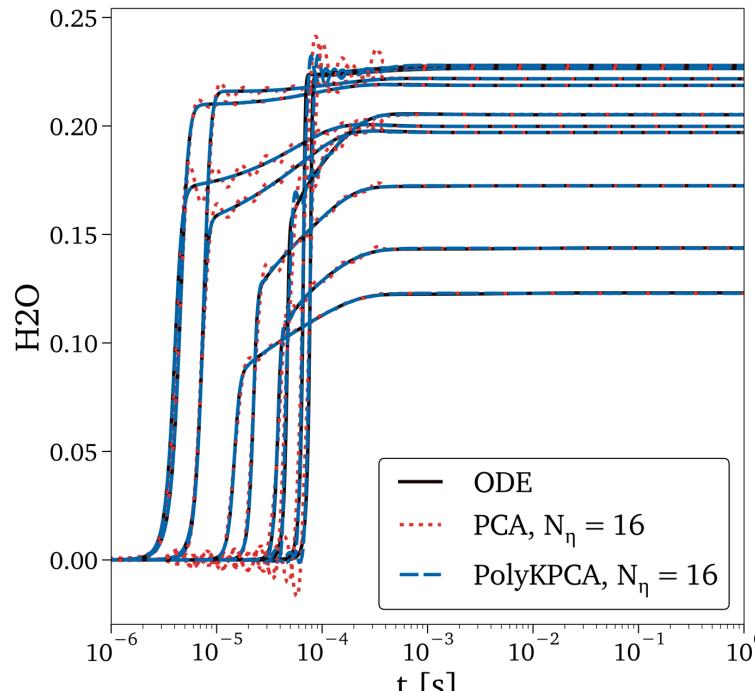
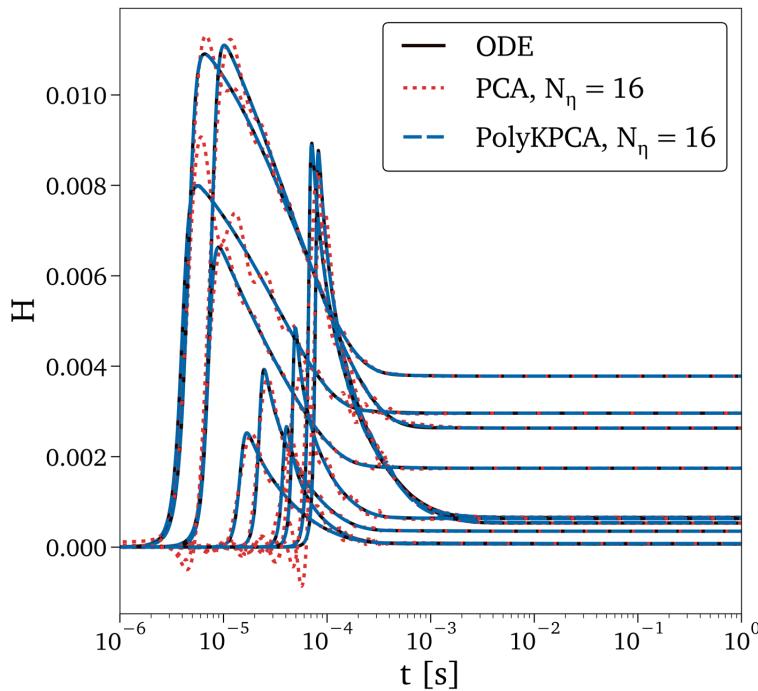
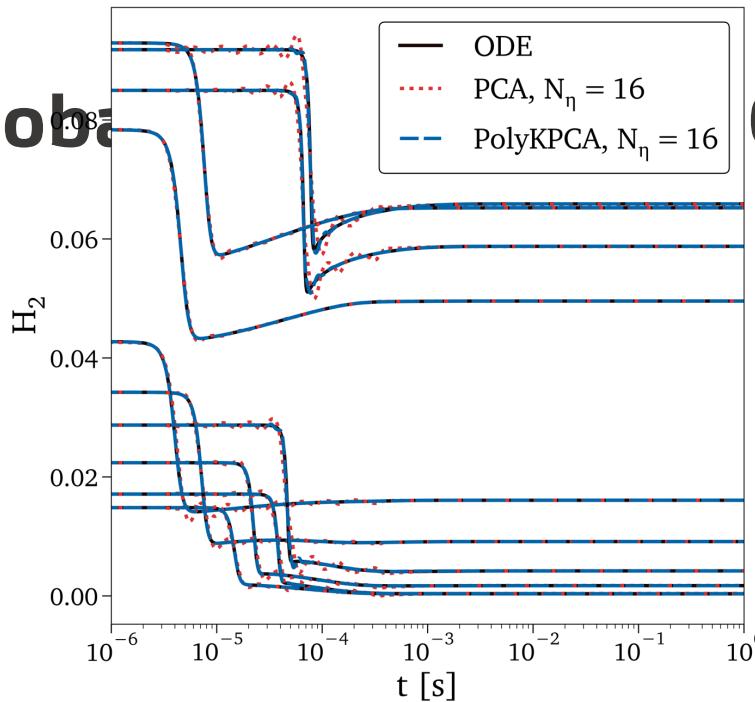
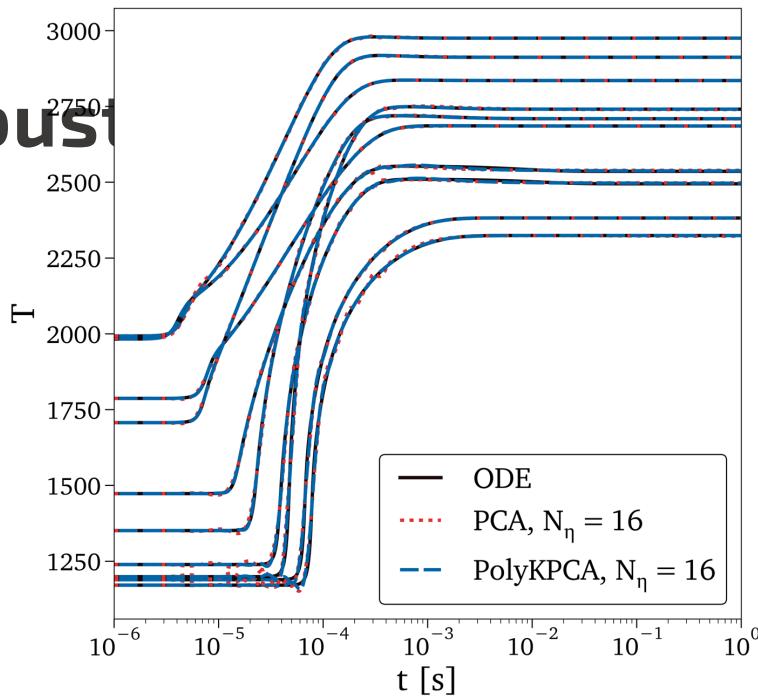
Modify and run Jupyter Notebook:

\$WORKSPACE\_PATH/ROMNet/romnet/input/ScenarioAggregated\_ROMs/0DReactor/FNN/Trunk/Parallelize\_ROMNet.py  
for generating Dimensionality-Reduced training data

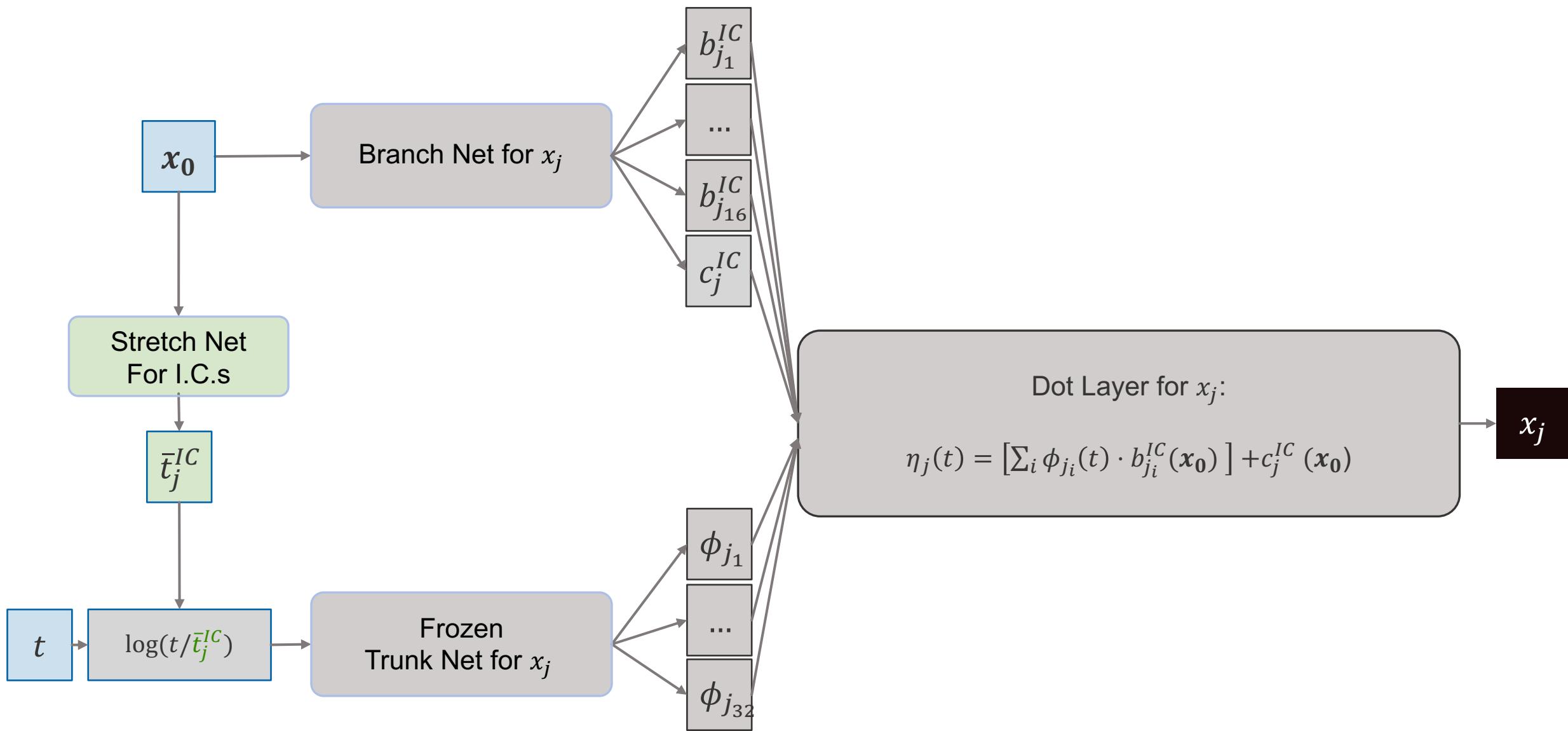
# A Combustion Case



## Isobutane Case



# A Combustion Chemistry in Isobaric Reactor Test Case





# A Combustion Chemistry in Isobaric Reactor Test Case

## Test Case 11: Data-driven DeepONet with frozen trunk for predicting Thermodynamic Variables

### Train the Trunks:

11.1. Run \$WORKSPACE\_PATH/ROMNet/romnet/input/ScenarioAggregated\_ROMs/0DReactor/FNN/Trunk/Parallelize\_ROMNet.py

### Train the Rest of the DeepONet:

11.2. Copy \$WORKSPACE\_PATH/ROMNet/romnet/input/0DReact/DeepONet/0DReact\_H2\_TestCase11/ROMNet\_Input.py  
to \$WORKSPACE\_PATH/ROMNet/romnet/input/ROMNet\_Input.py

11.3. In \$WORKSPACE\_PATH/ROMNet/romnet/input/ROMNet\_Input.py, change:

11.3.1. "self.WORKSPACE\_PATH = ..."

11.4. Move to \$WORKSPACE\_PATH/ROMNet/romnet/app/

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

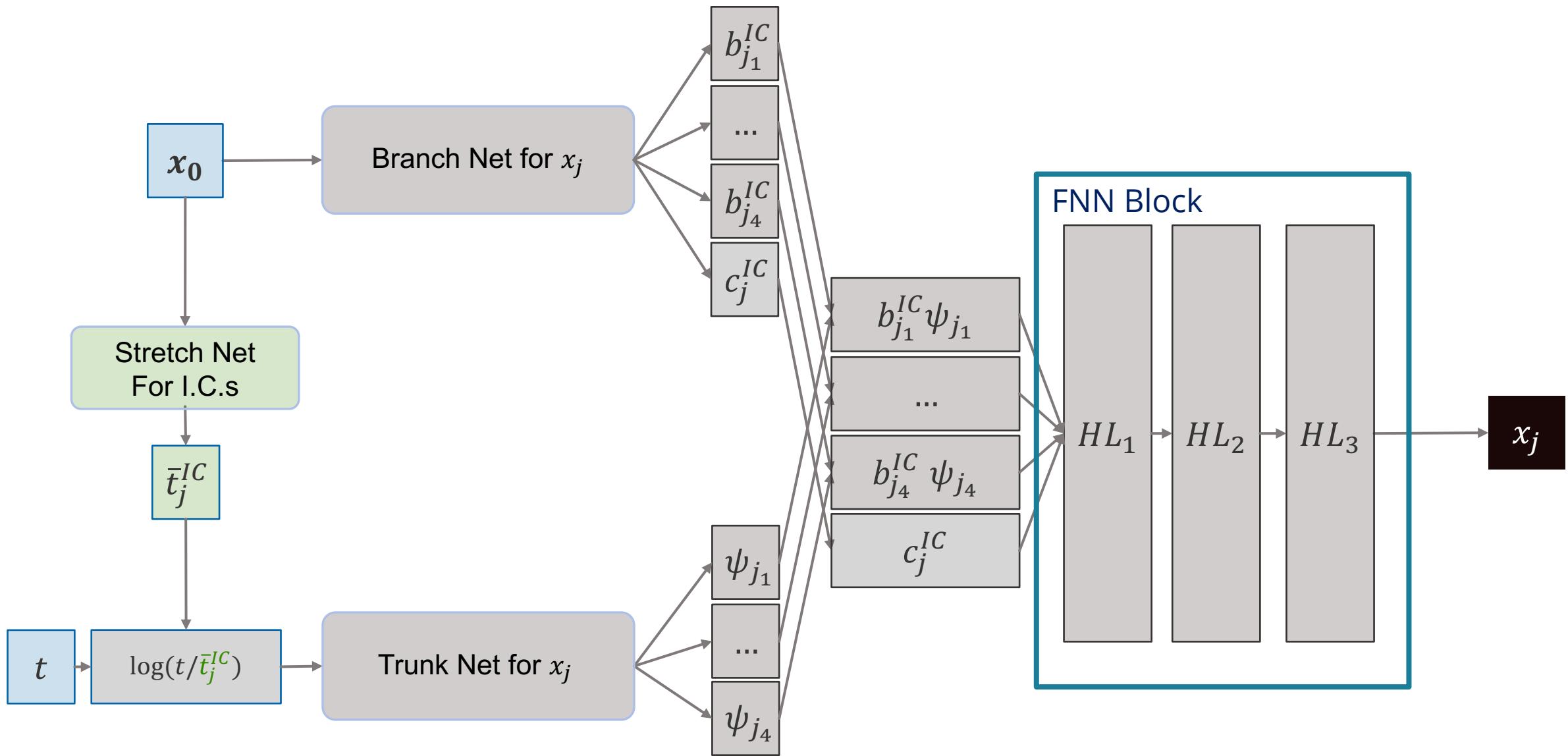
11.6. Postprocess results via: \$WORKSPACE\_PATH/ROMNet/romnet/scripts/postprocessing/0DReact/DeepONet/Predict\_DeepONet\_Orig.ipynb



## Test Case 12

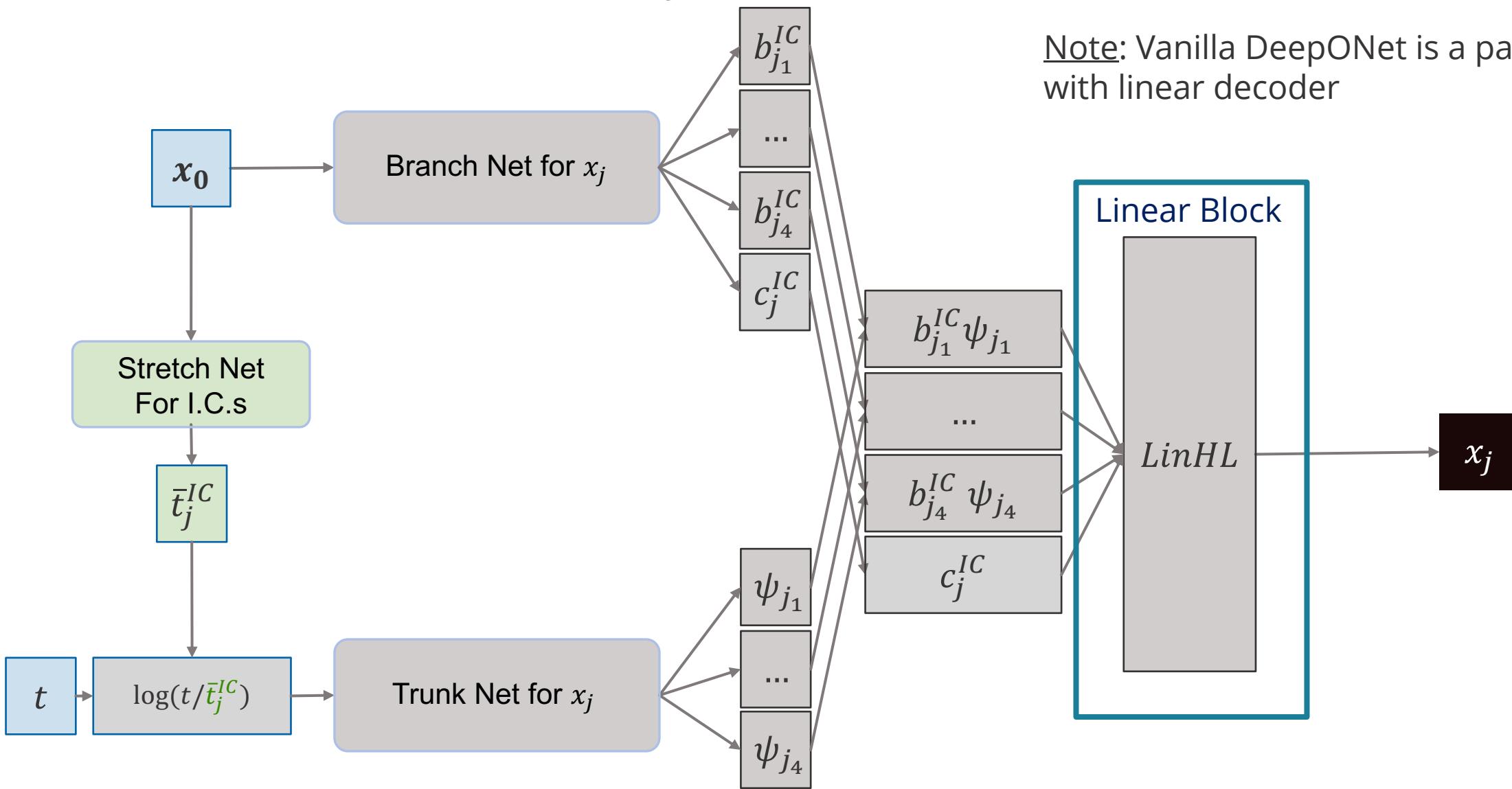
# Data-Driven Modified DeepONet with Non-Linear Projection in the Thermodynamic State Space

# A Combustion Chemistry in Isobaric Reactor Test Case



# A Combustion Chemistry in Isobaric Reactor Test Case

Note: Vanilla DeepONet is a particular case with linear decoder





# A Combustion Chemistry in Isobaric Reactor Test Case

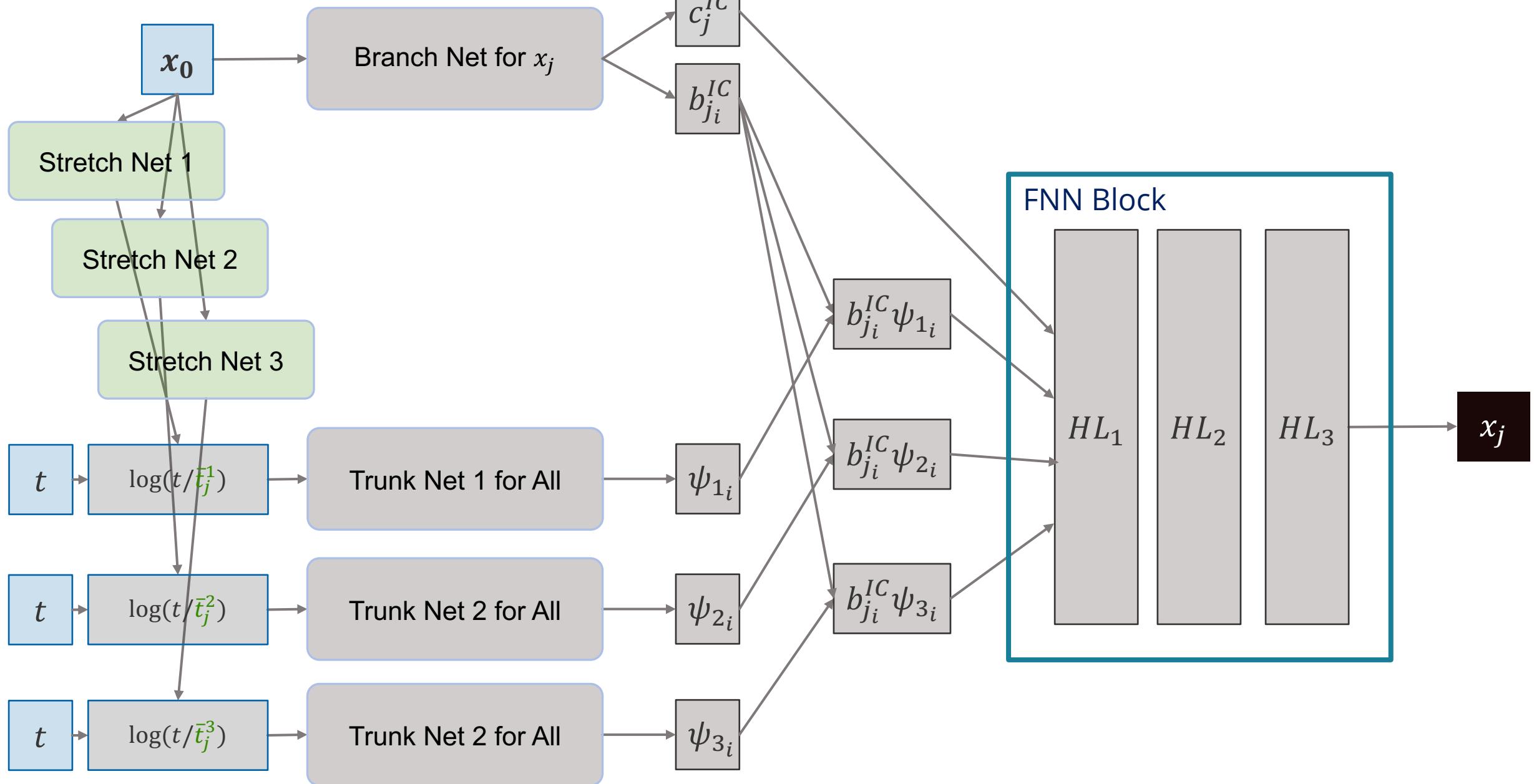
## Test Case 12: Modified DeepONet with Non-Linear Projection for predicting Principal Components

- 12.1. Copy \$WORKSPACE\_PATH/ROMNet/romnet/input/0DReact/DeepONet/0DReact\_H2\_TestCase12/ROMNet\_Input.py to \$WORKSPACE\_PATH/ROMNet/romnet/input/ROMNet\_Input.py
- 12.2. In \$WORKSPACE\_PATH/ROMNet/romnet/input/ROMNet\_Input.py, change:
  - 6.2.1. "self.WORKSPACE\_PATH = ..."
- 12.3. Move to \$WORKSPACE\_PATH/ROMNet/romnet/app/
- 12.4. Run: "python3 ROMNet.py ..//input/"
- 12.5. Postprocess results via: \$WORKSPACE\_PATH/ROMNet/romnet/scripts/postprocessing/0DReact/DeepONet/Predict\_DeepONet.ipynb

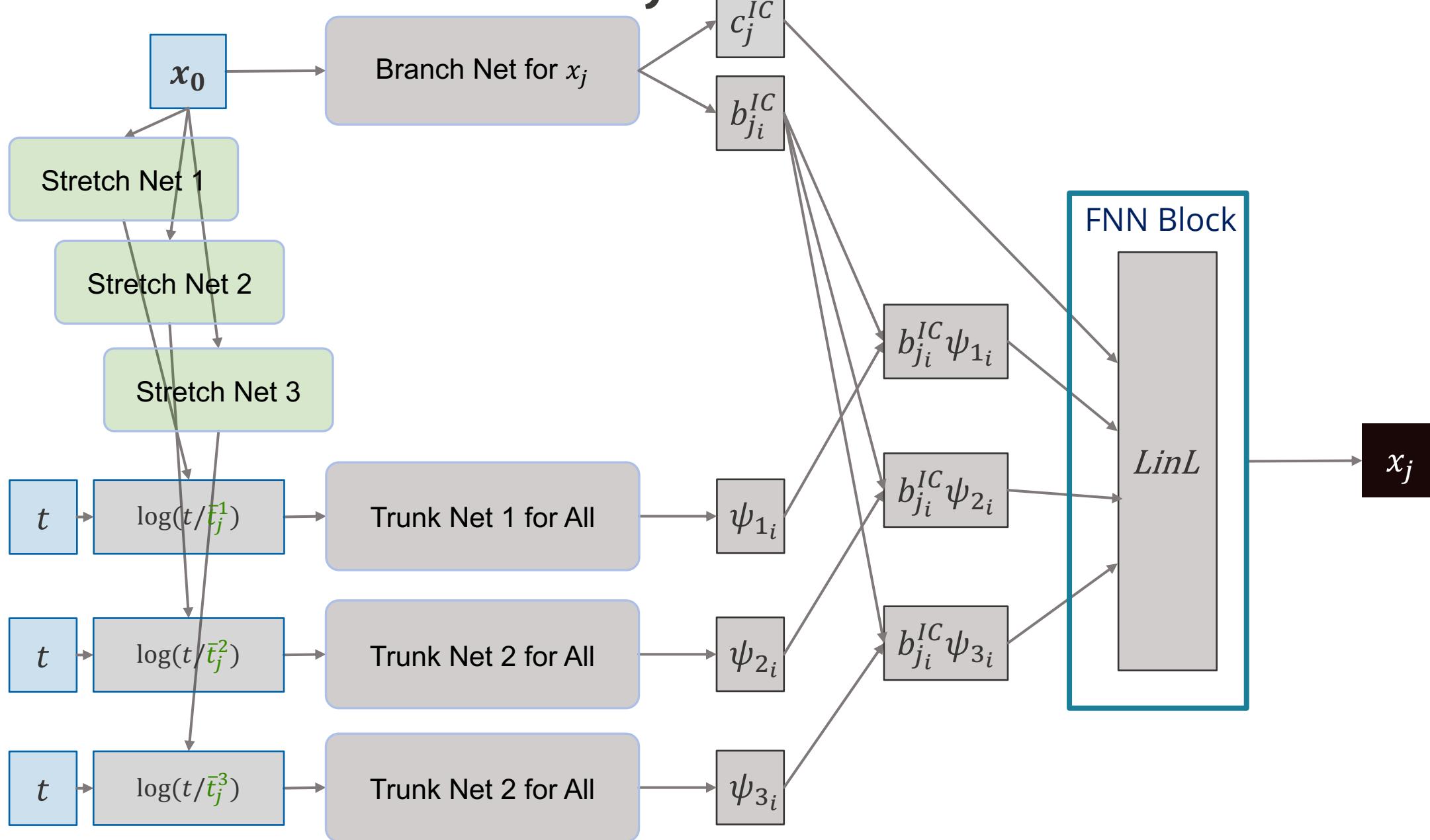


## Test Case 13 Data-Driven Shared-Trunk DeepONet in the Thermodynamic State Space

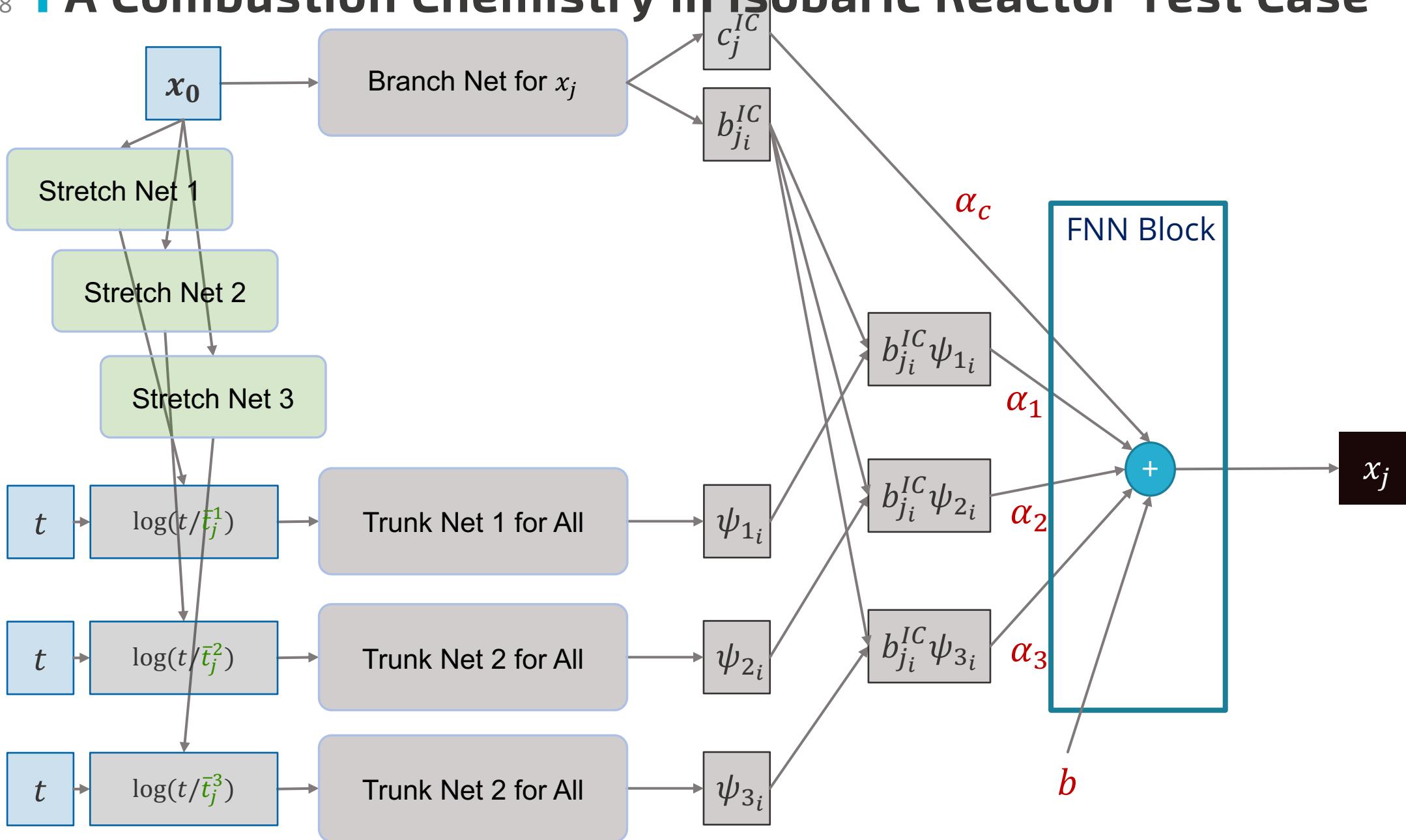
# A Combustion Chemistry in Isobaric Reactor Test Case



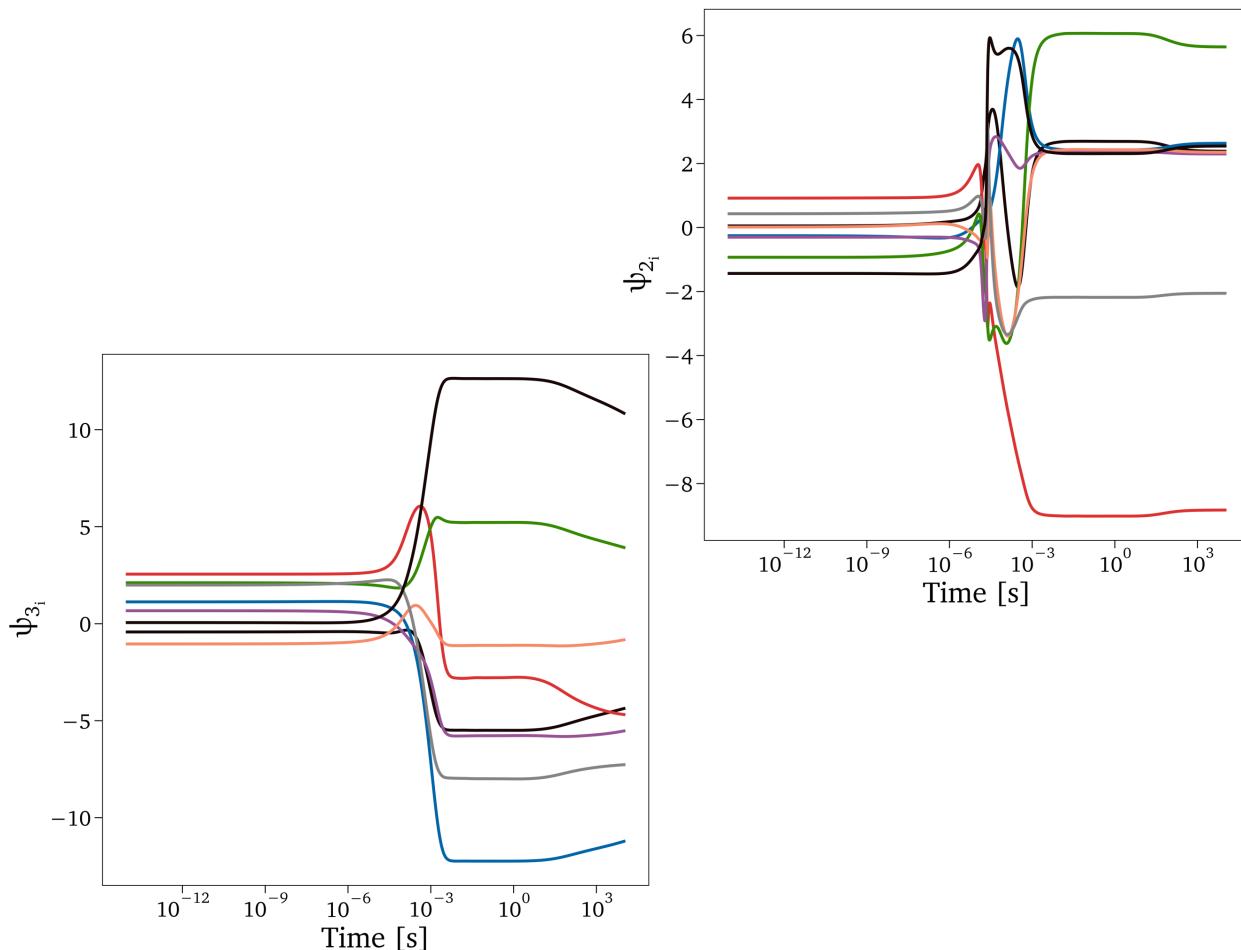
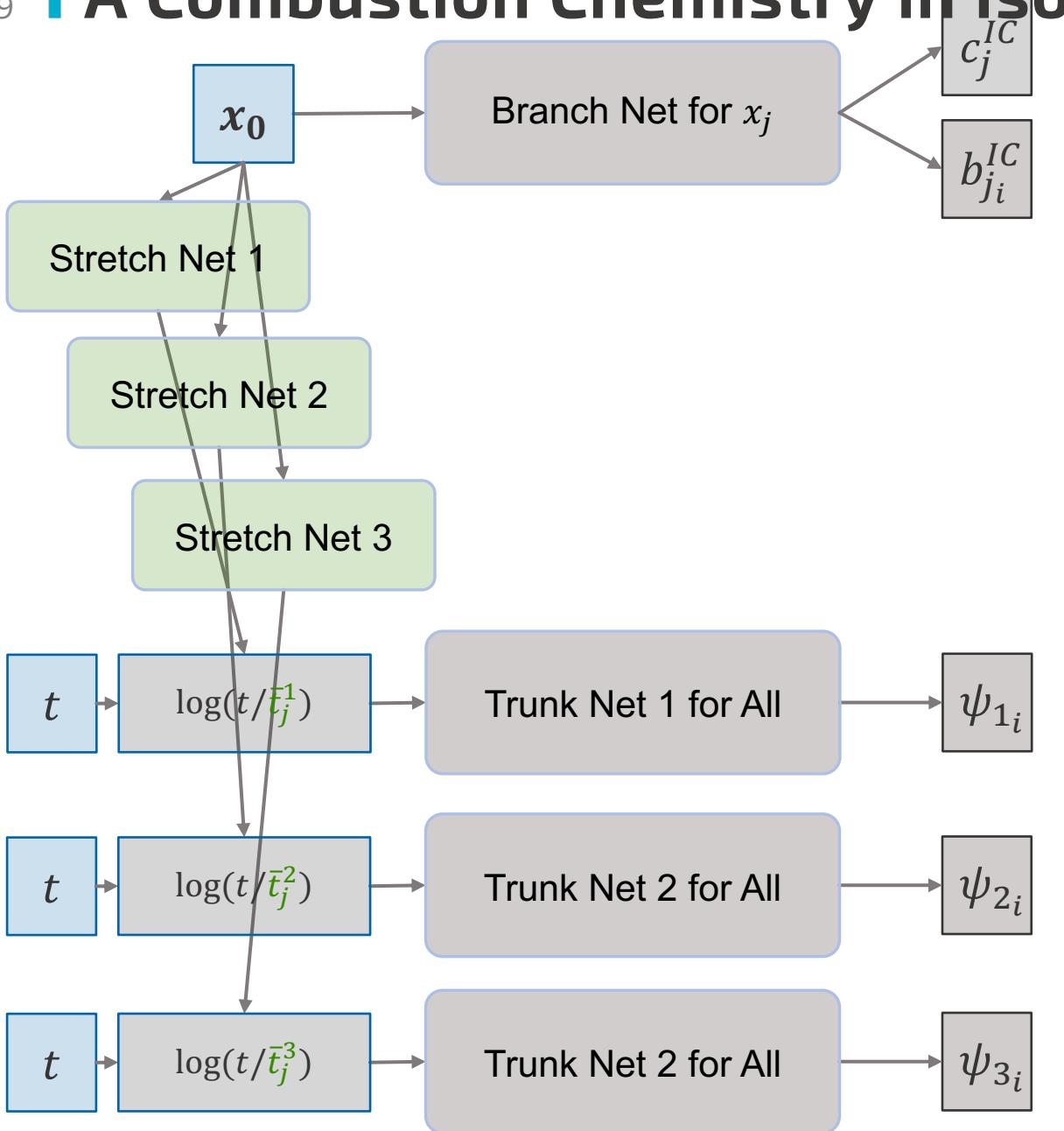
# A Combustion Chemistry in Isobaric Reactor Test Case



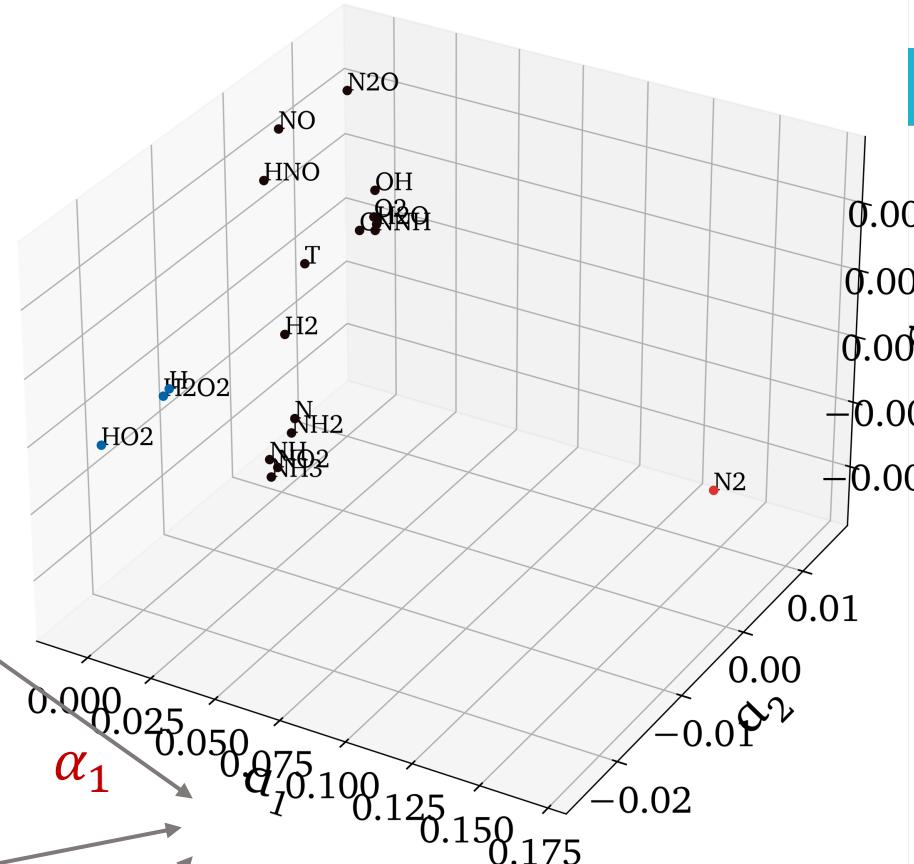
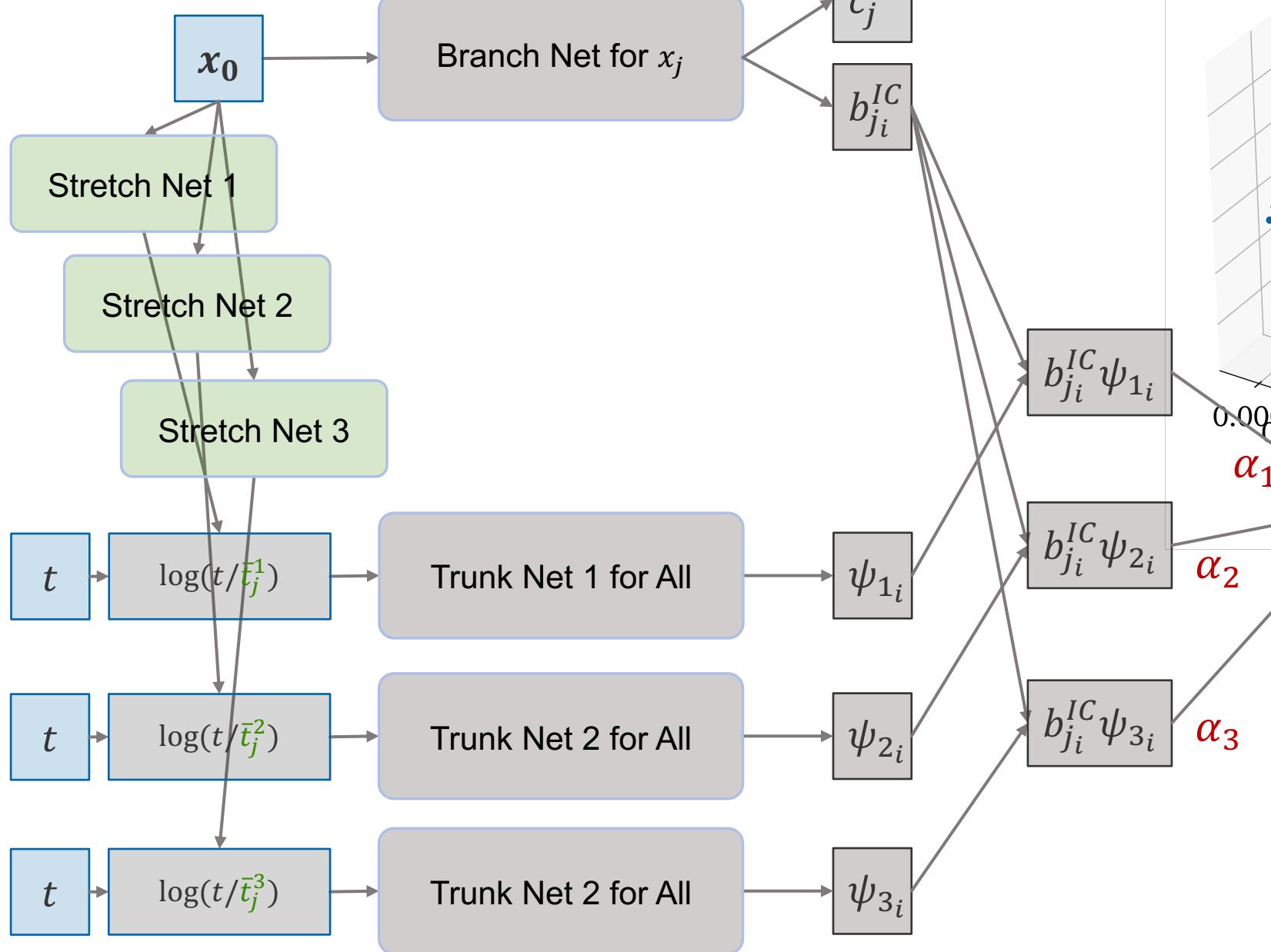
# A Combustion Chemistry in Isobaric Reactor Test Case



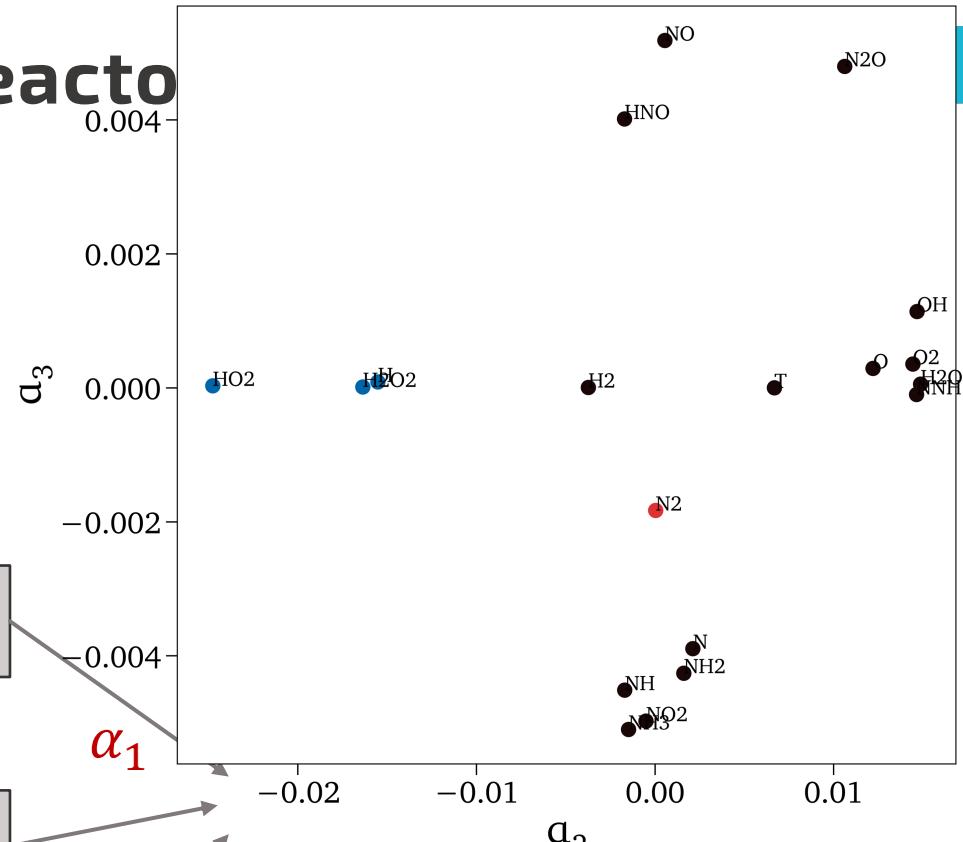
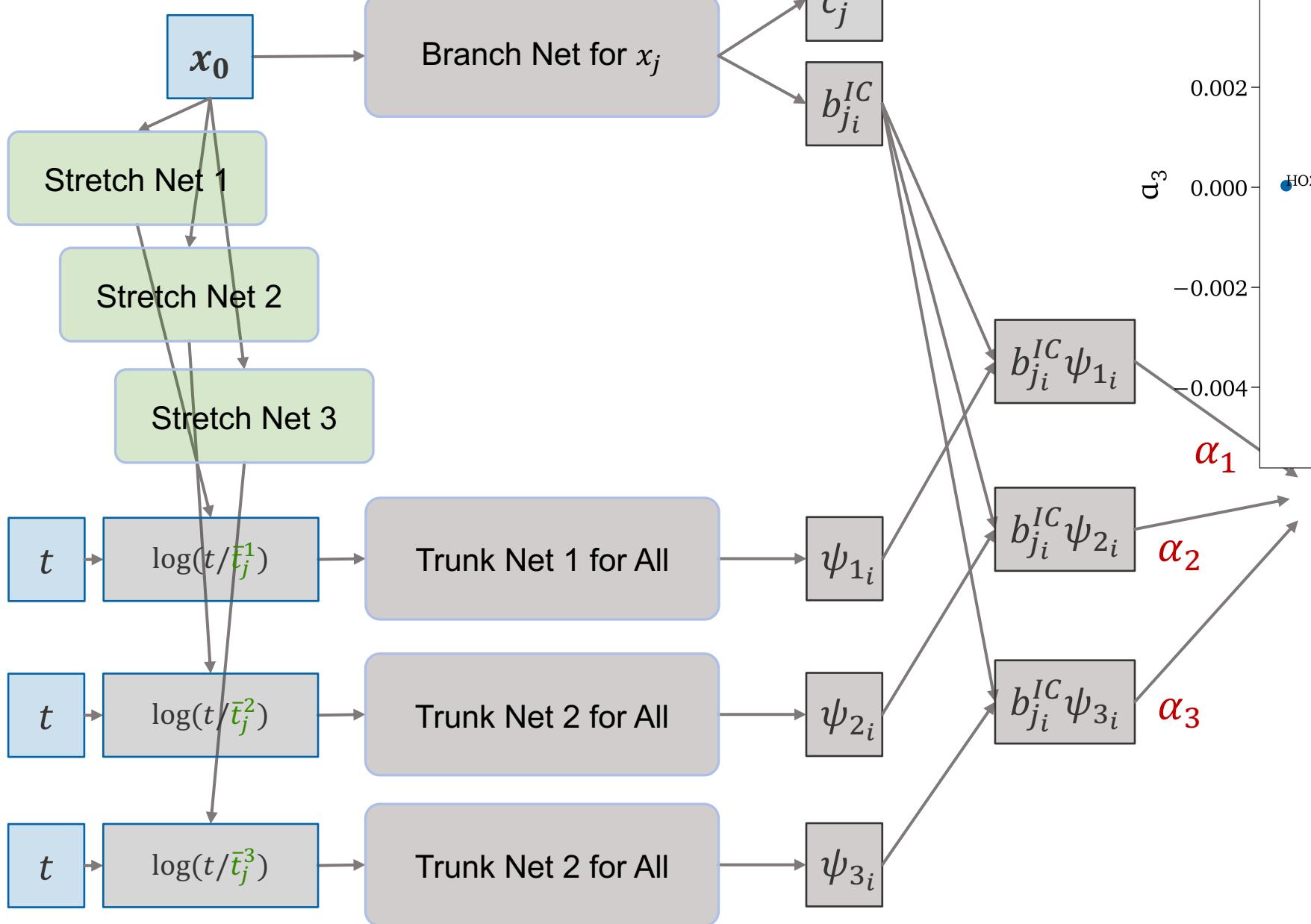
# A Combustion Chemistry in Isobaric Reactor Test Case



# A Combustion Chemistry in Isobaric Re



# A Combustion Chemistry in Isobaric Reactor





# A Combustion Chemistry in Isobaric Reactor Test Case

## **Test Case 13: Shared-Trunk DeepONet for predicting Principal Components**

- 13.1. Copy \$WORKSPACE\_PATH/ROMNet/romnet/input/0DReact/DeepONet/0DReact\_H2\_TestCase13/ROMNet\_Input.py to \$WORKSPACE\_PATH/ROMNet/romnet/input/ROMNet\_Input.py
- 13.2. In \$WORKSPACE\_PATH/ROMNet/romnet/input/ROMNet\_Input.py, change:
  - 6.2.1. "self.WORKSPACE\_PATH = ..."
- 13.3. Move to \$WORKSPACE\_PATH/ROMNet/romnet/app/
- 13.4. Run: "python3 ROMNet.py ..input/"
- 13.5. Postprocess results via: \$WORKSPACE\_PATH/ROMNet/romnet/scripts/postprocessing/0DReact/DeepONet/Predict\_DeepONet.ipynb