Sandia National Laboratories

Test Cases for the Mass-Spring-Damper System

Simone Venturi, Tiernan Casey

Extreme-Scale Data Science & Analytics (8739)

Part of the Code Documentation for Neural Networks for Reduced Order Modeling (ROMNet)

U.S. DEPARTMENT OF ENERGY | 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 Mass-Spring-Damper Test Case

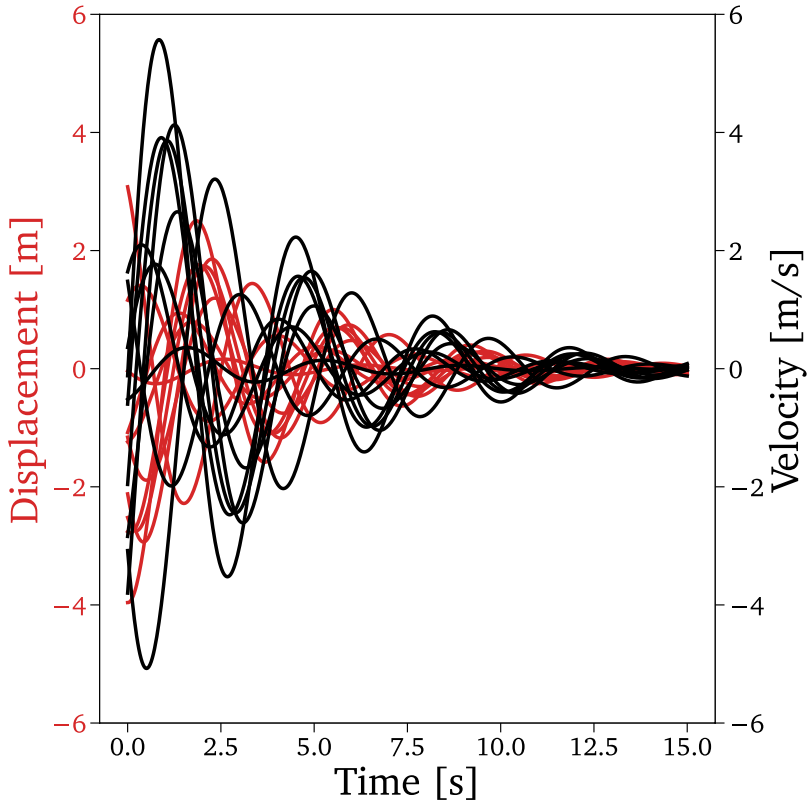
Equations of motion:

$$\begin{cases} m\ddot{x} + c\dot{x} + kx = 0, \\ x(t=0) = x_0, \\ \dot{x}(t=0) = v_0, \end{cases}$$

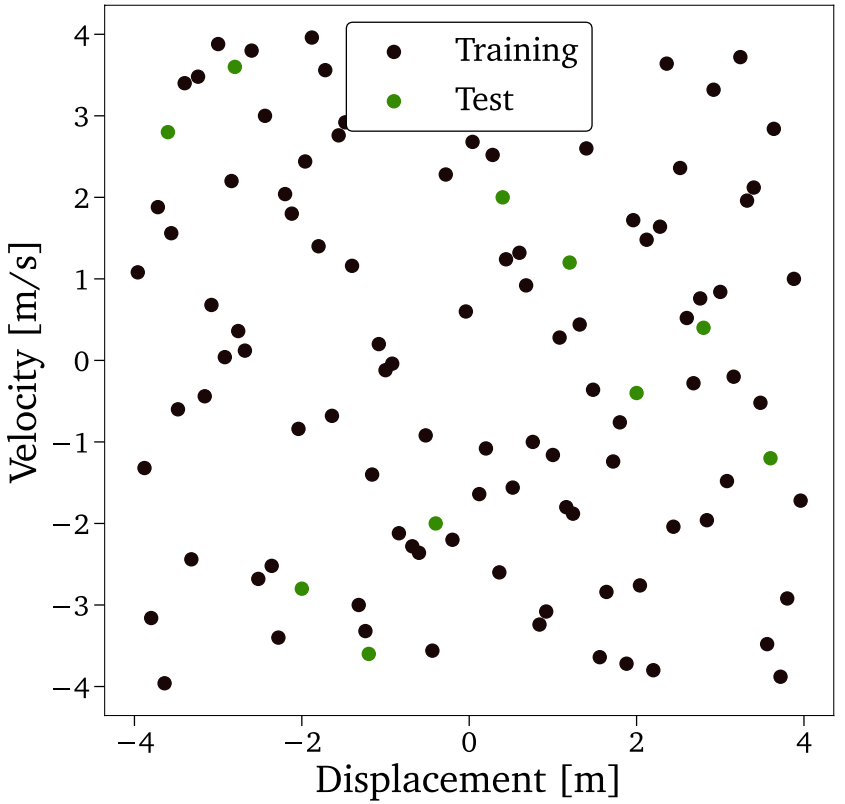
which can be rewritten as:

$$\begin{cases} \begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \\ x(t=0) = x_0, \\ \dot{x}(t=0) = v_0. \end{cases}$$

Some training scenarios



Initial conditions



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

The m, c, and k parameters can be found in
$WORKSPACE_PATH//ROMNet/romnet/database/MassSpringDamper/Params/

A Mass-Spring-Damper Test Case

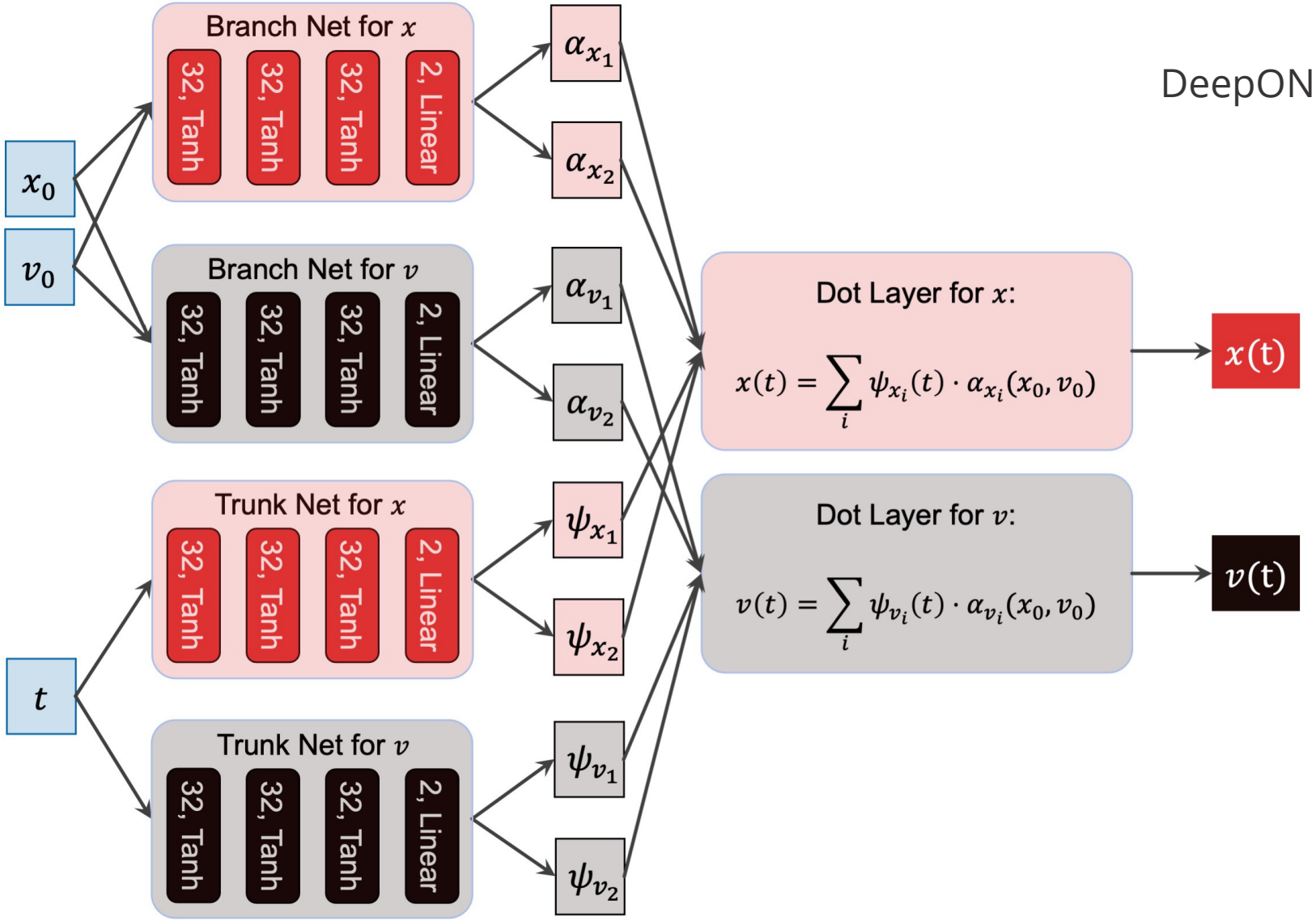
Run Jupyter Notebook
$WORKSPACE_PATH/ROMNet/romnet/scripts/generating_data/MassSpringDamper/Generate_Data_1.ipynb
for generating training and test data

A Mass-Spring-Damper Test Case

Test Cases 1 & 4
Data-Driven and Physics-Informed Vanilla DeepONets

A Mass-Spring-Damper Test Case

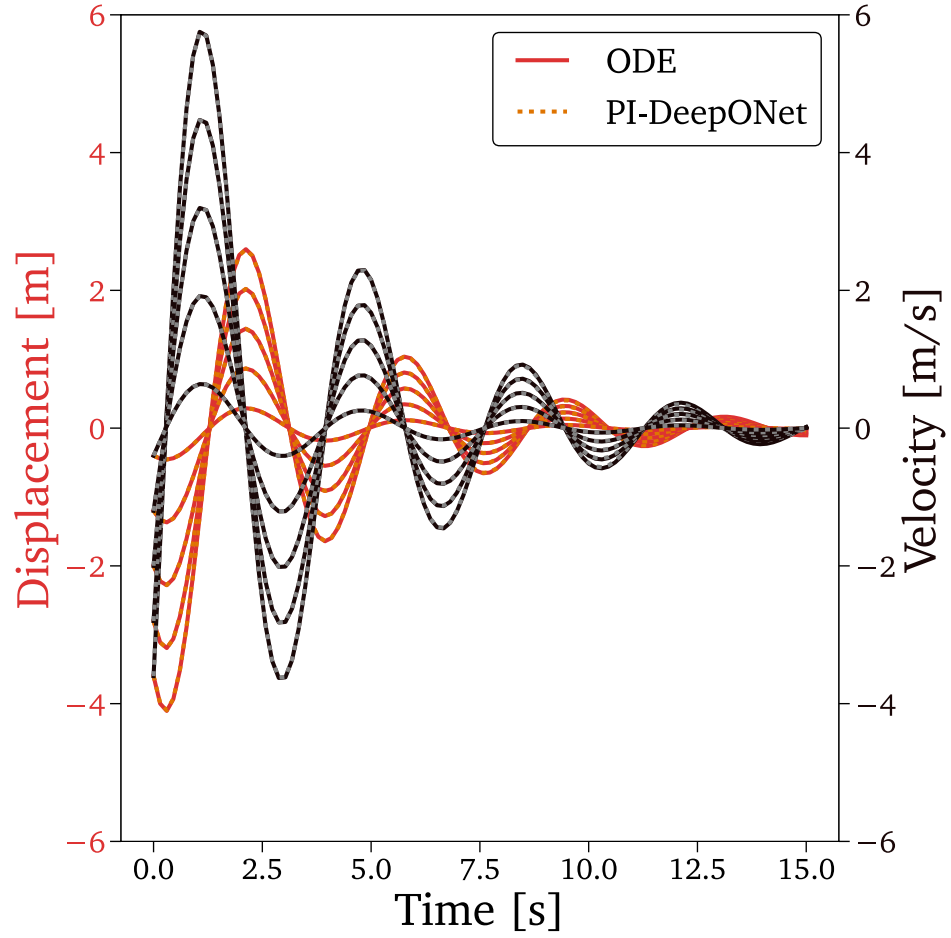
DeepONet Structure



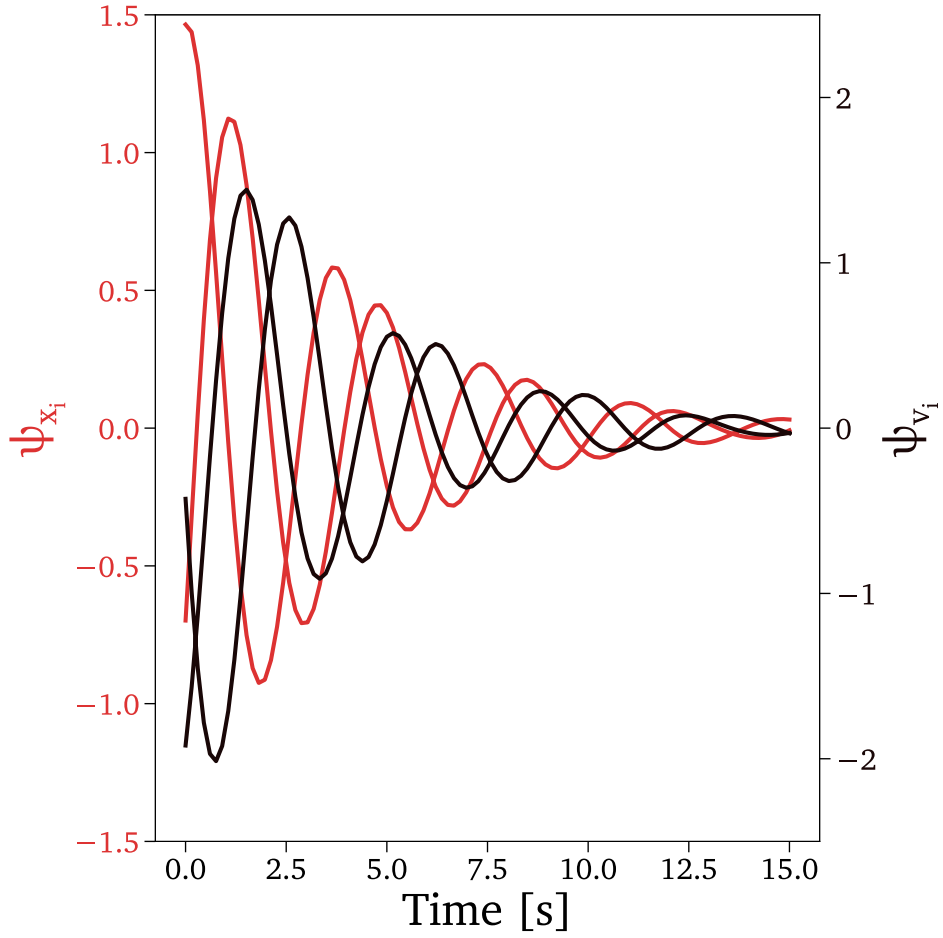
A Mass-Spring-Damper Test Case

Test Case 1: Fully data-driven training
Test Case 2: Trained with a physics-informed loss

Test cases



Trunk nets' outputs (~dynamical modes)



A Mass-Spring-Damper Test Case

Test Case 1: Data-driven deep operator network (DeepONet) for predicting position and velocity

1.1. Copy $WORKSPACE_PATH/ROMNet/romnet/input/MassSpringDamper/DeepONet/MSD_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/MassSpringDamper/DeepONet/Predict_DeepONet.ipynb

A Mass-Spring-Damper Test Case

Test Case 4: Physics Informed deep operator network (DeepONet) for predicting position and velocity

4.1. Copy $WORKSPACE_PATH/ROMNet/romnet/input/MassSpringDamper/DeepONet/MSD_TestCase2/ROMNet_Input.py
to $WORKSPACE_PATH/ROMNet/romnet/input/ROMNet_Input.py

4.2. In $WORKSPACE_PATH/ROMNet/romnet/input/ROMNet_Input.py, change:
2.2.1. "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/MassSpringDamper/DeepONet/Predict_DeepONet.ipynb

In the input file:

self. **surrogate_type** controls the type of surrogate

self.**structure** is a dictionary that controls the structure of the surrogate

Surrogates:

- FNN: Feed-Forward Neural Network
- DeepONet: Deep Operator Network
- Double_DeepONet: Two DeepONets in Series

System of Components:

- FNN
- DeepONet

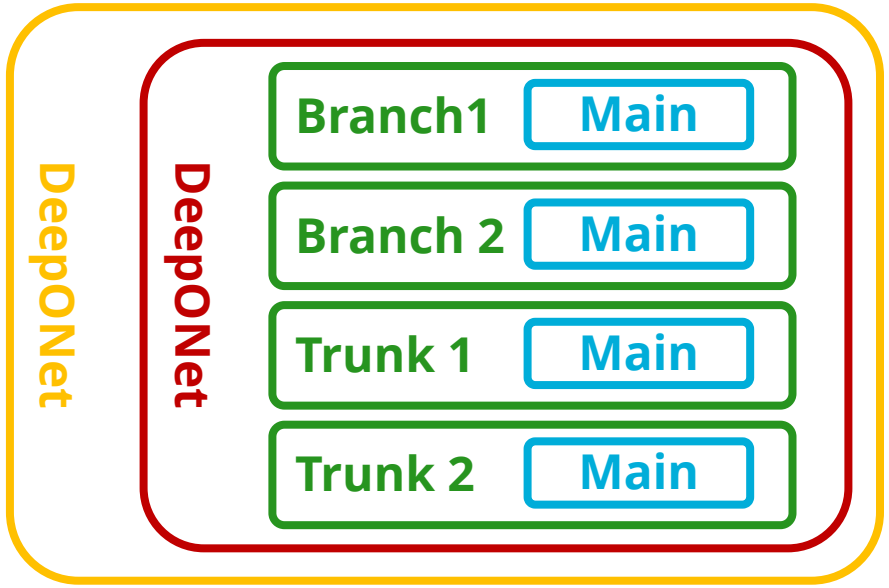
Components:

- FNN
- Branch
- Branch_i
- Trunk
- Trunk_i

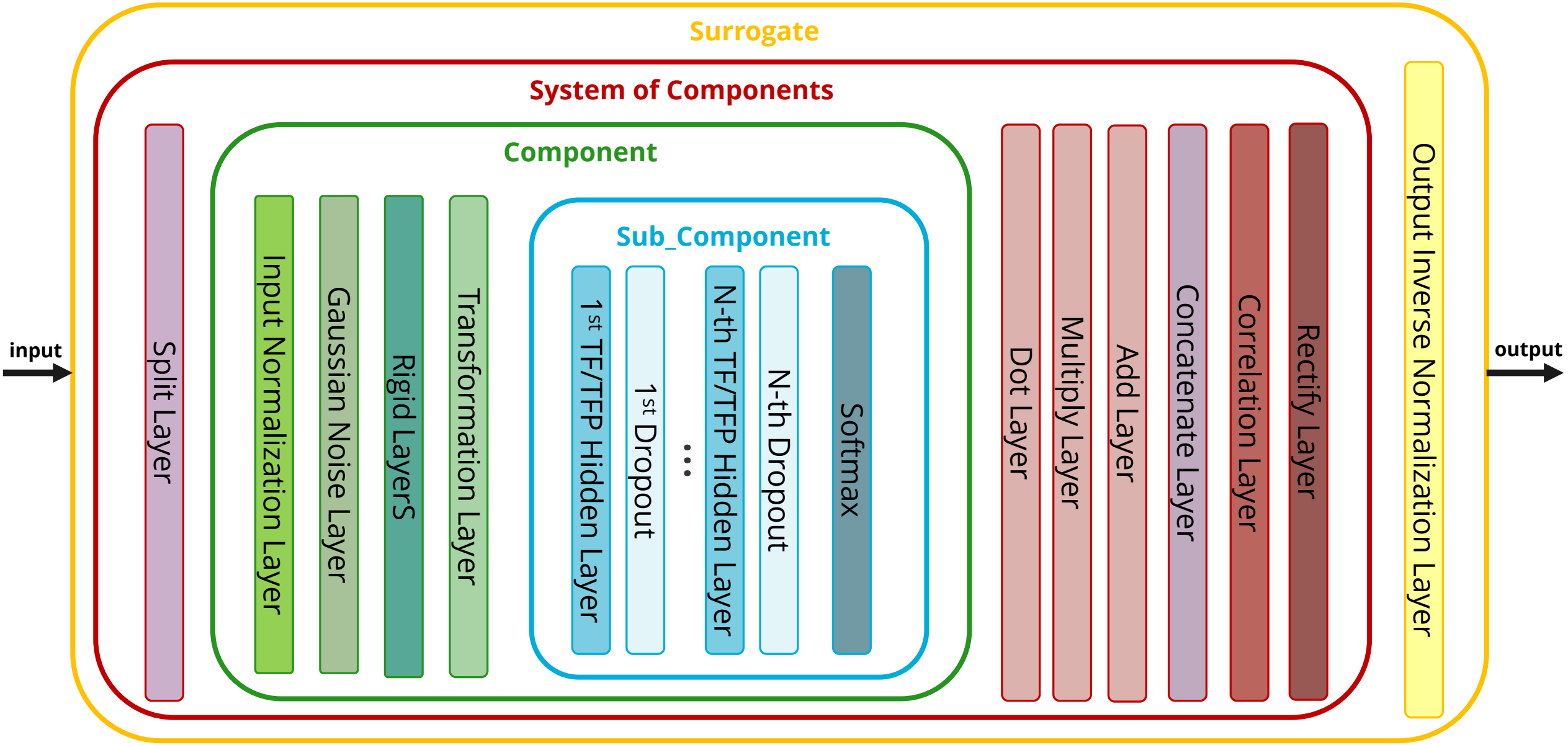
Sub-Components:

- Main
- U
- V

In these test cases:



The related classes can be found in
$WORKSPACE_PATH/ROMNet/romnet/romnet/nn/



The input files for Test Case 1 and 4 differ only for:

self.**n_train (**i.e. Type/No of Data Points)

- 'pts': data point
- 'ics': ODE's initial conditions
- 'res': ODE residual

self. **losses** (i.e., Dictionary Containing Loss Functions for Each Data Type)

self.**loss_weights** (i.e., Dictionary Containing Weights for Each Data Type)

A Mass-Spring-Damper Test Case

Test Case 2
Data-Driven POD-DeepONet

A Mass-Spring-Damper Test Case

A scenario-aggregated Singular Value Decomposition (SVD) analogy

By aggregating the training scenarios for $x_i(t)$ and $v_i(t)$, where i represents the scenario index:

$$\boldsymbol{X} = \begin{bmatrix} | & | & & | & | \\ \boldsymbol{x_1} & \boldsymbol{x_2} & \cdots & \boldsymbol{x_{99}} & \boldsymbol{x_{100}} \\ | & | & & | & | \end{bmatrix}$$

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

No of time instants

No of scenarios

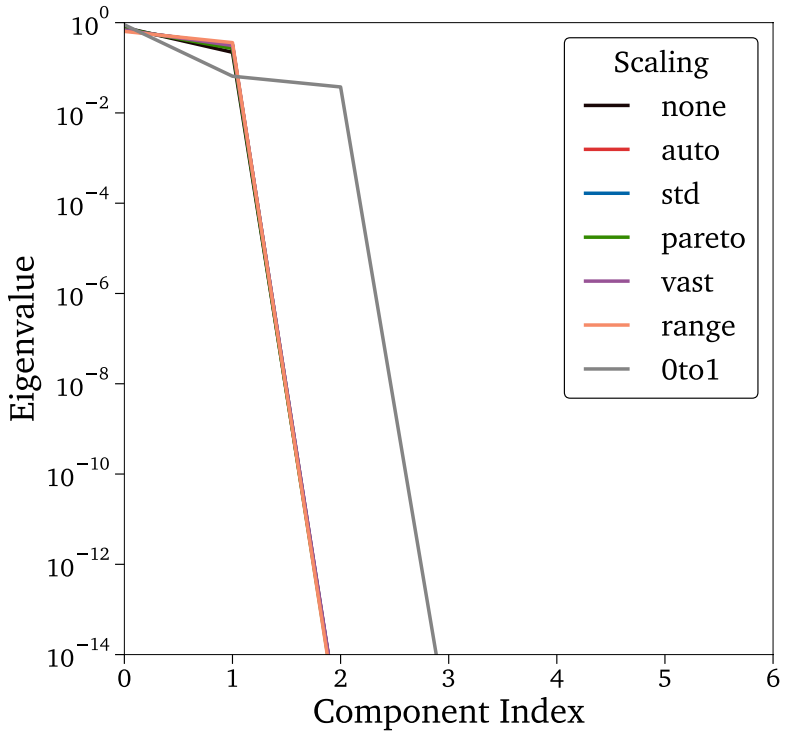
$$\boldsymbol{V} = \begin{bmatrix} | & | & & | & | \\ \boldsymbol{v_1} & \boldsymbol{v_2} & \cdots & \boldsymbol{v_{99}} & \boldsymbol{v_{100}} \\ | & | & & | & | \end{bmatrix}$$

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

A Mass-Spring-Damper Test Case

Eigenvector Decomposition of the Covariance Matrix of $\mathbf{X}$ ($\mathbf{R_X} = \frac{\mathbf{X}\mathbf{X}^\mathbf{T}}{N_t - 1}$):

Analyzing the eigenvalues:



Eigenvalues of $\mathbf{R_x}$

$$\mathbf{\Lambda_x} = \mathbf{\Psi_x^{-1} R_x \Psi_x}$$

Orthonormal eigenvectors of $\mathbf{R_x}$

Two principal components ($N_\psi = 2$) are sufficient for fully characterizing all the 100 scenarios

Note: Equivalent results obtained for the SVD of $\mathbf{V}$

A Mass-Spring-Damper Test Case

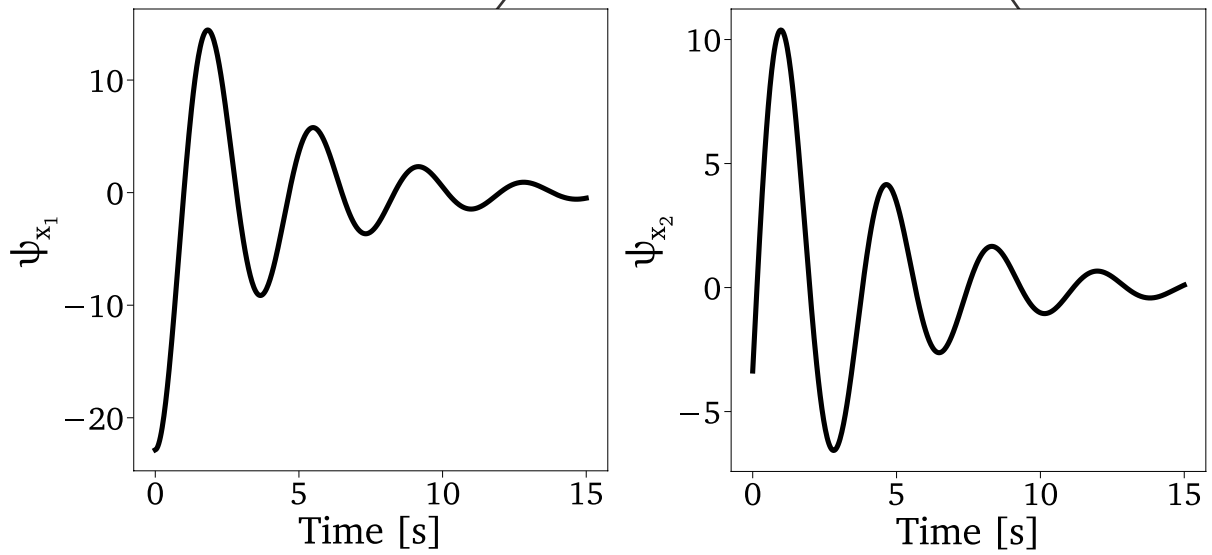
Eigenvector Decomposition of the Covariance Matrix of X ($R_x = \frac{XX^T}{N_S - 1}$):

Analyzing the modes and the projection matrix

$$\Psi_x = (X - C_x) \cdot D_x^{-1} A_x = \begin{bmatrix} \psi_{x_1} & \psi_{x_2} \end{bmatrix}$$

Where:

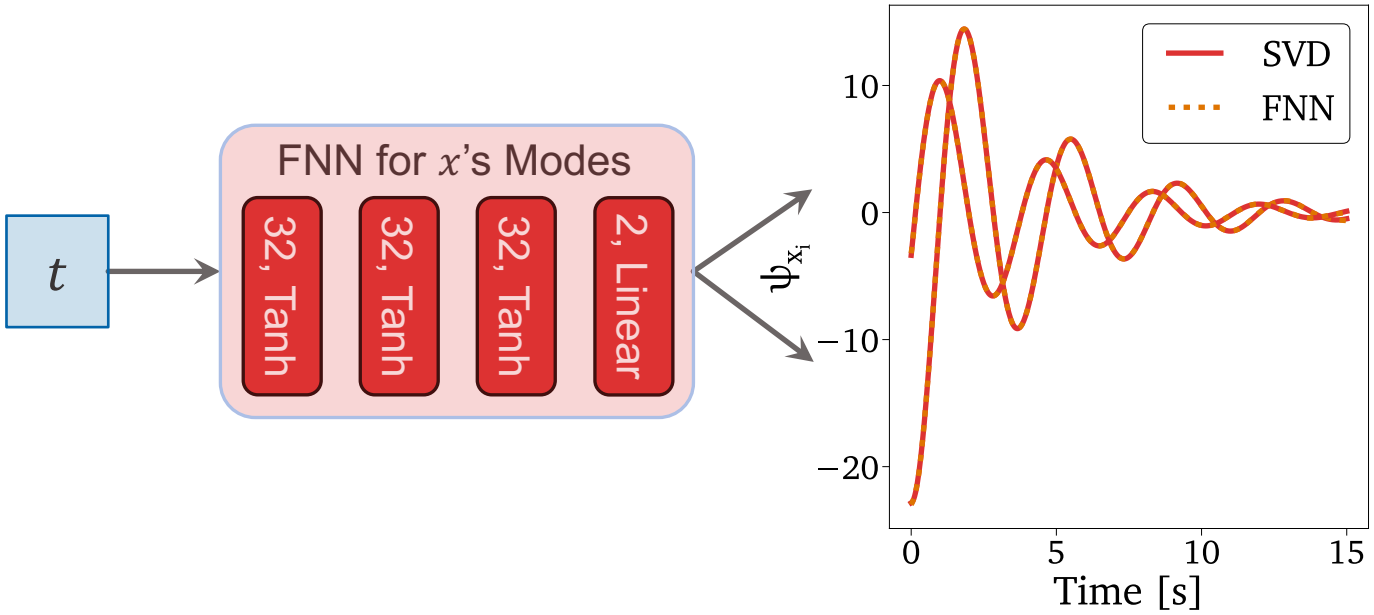
$$\dim(\Psi_x) = N_t \times N_\psi = 500 \times 2$$

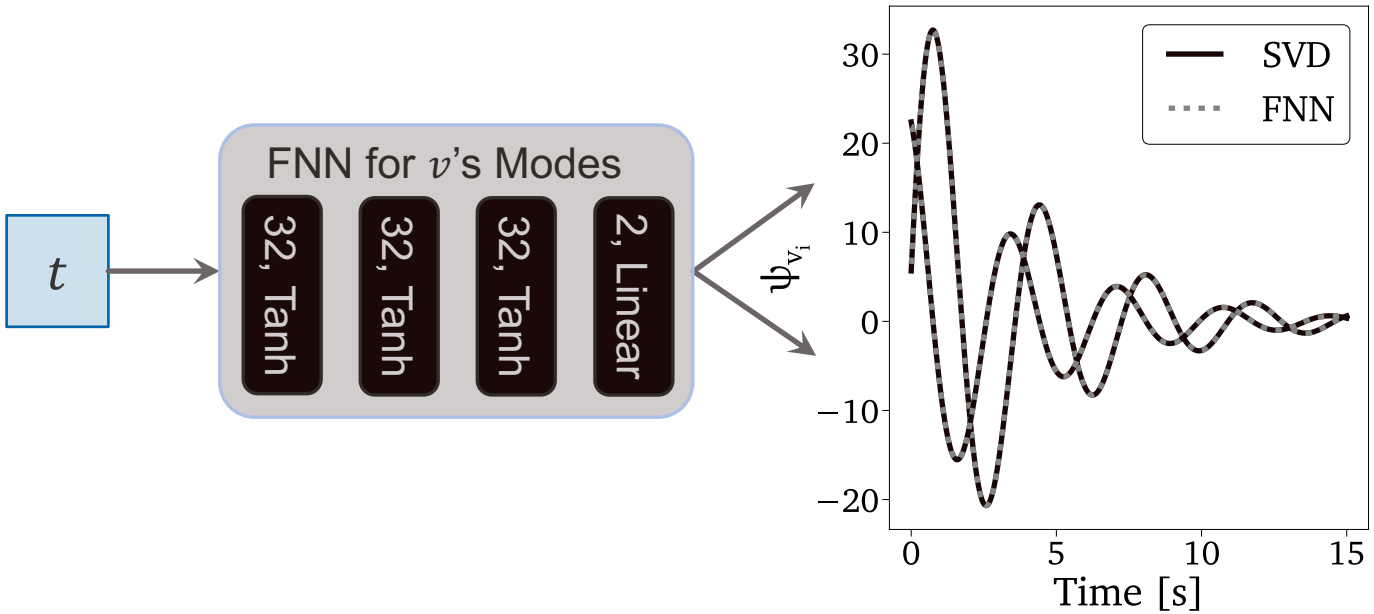


A Mass-Spring-Damper Test Case

Run Python Script
$WORKSPACE_PATH/ROMNet/romnet/scripts/generating_data/ScenarioAggregated_ROMS/MassSpringDamper/Generate_Data.py
for generating SVD training and test data

A Mass-Spring-Damper Test Case





Fitted the modes of x and v with two independent feed-forward neural networks

A Mass-Spring-Damper Test Case

Test Case 2: POD-DeepONet:

Train Trunks and POD-DeepONet

2.1. Run $WORKSPACE_PATH/ROMNet/romnet/input/ScenarioAggregated_ROMs/MassSpringDamper/FNN/Trunk/Parallelize_ROMNet.py

2.2. Copy $WORKSPACE_PATH/ROMNet/romnet/input/MassSpringDamper/DeepONet/MSD_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/SVD/MassSpringDamper/FNN/Predict_FNN_Trunk.ipynb

The input files for Test Case 1 differs from the one of Test Case 2:

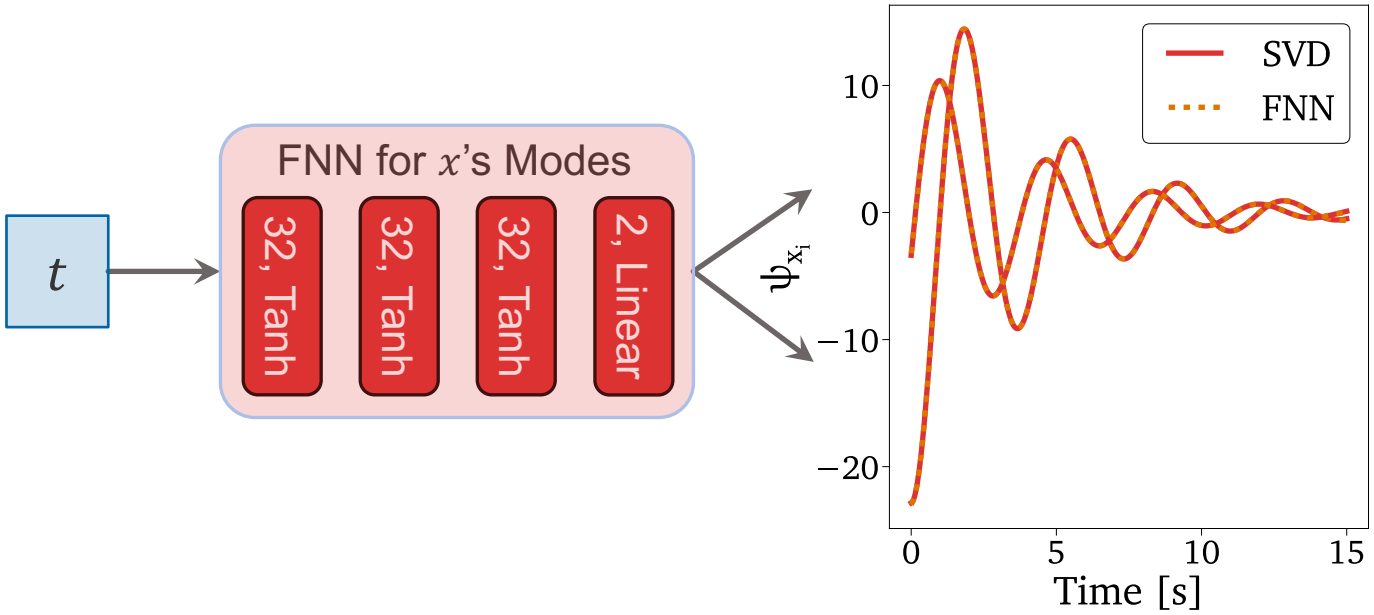
self.**path_to_load_fld**: we are now including a path for uploading pre-trained weights

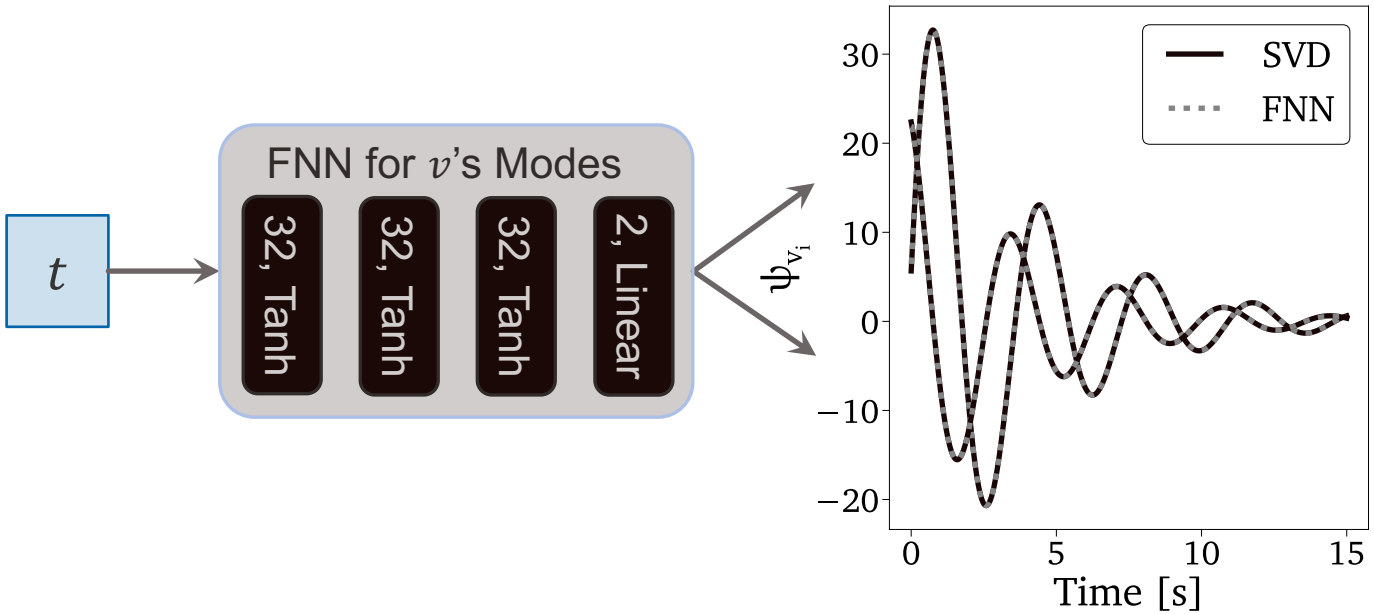
self.**trainable_flg**: we are not training the parameters

A Mass-Spring-Damper Test Case

Test Case 3
Data-Driven SVD-DeepONet

A Mass-Spring-Damper Test Case





Fitted the modes of x and v with two independent feed-forward neural networks

A Mass-Spring-Damper Test Case

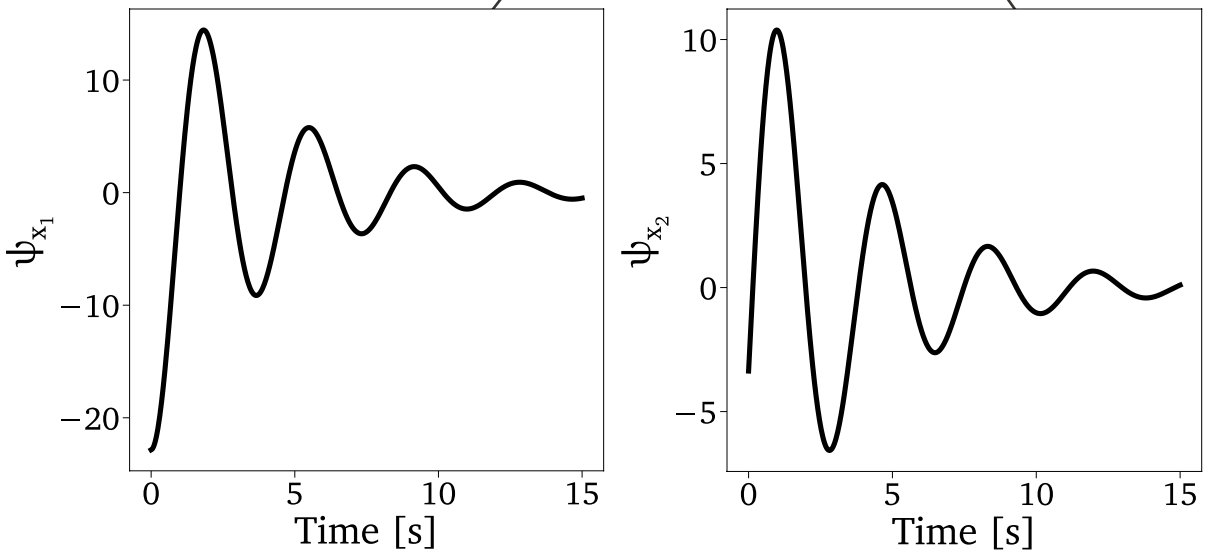
Eigenvector Decomposition of the Covariance Matrix of X ($R_x = \frac{XX^T}{N_S - 1}$):

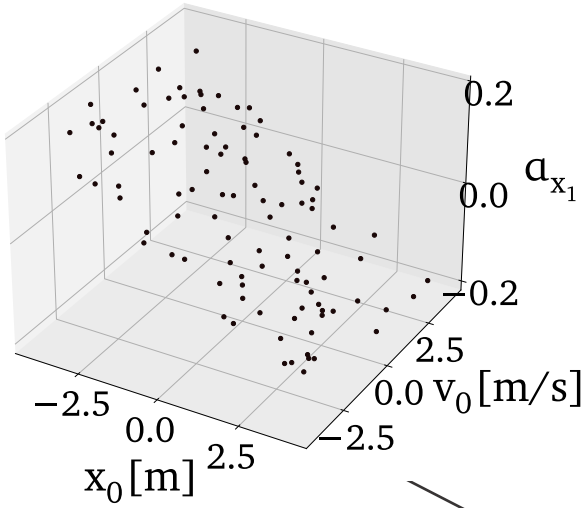
Analyzing the modes and the projection matrix

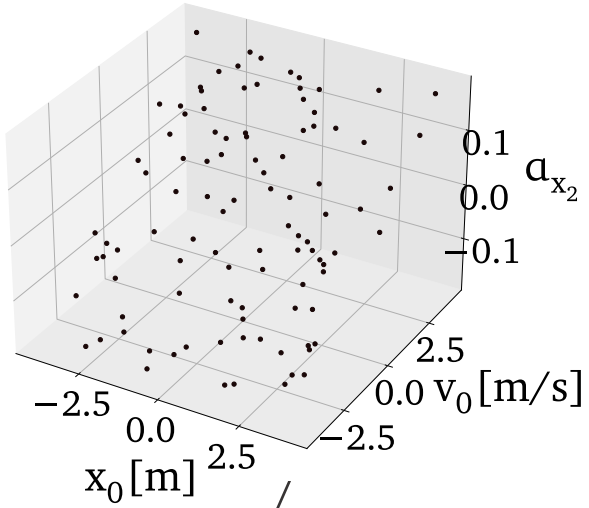
$$\Psi_x = (X - C_x) \cdot D_x^{-1} A_x = \begin{bmatrix} \psi_{x_1} & \psi_{x_2} \end{bmatrix}$$

Where:

$$\dim(\Psi_x) = N_t \times N_\psi = 500 \times 2$$





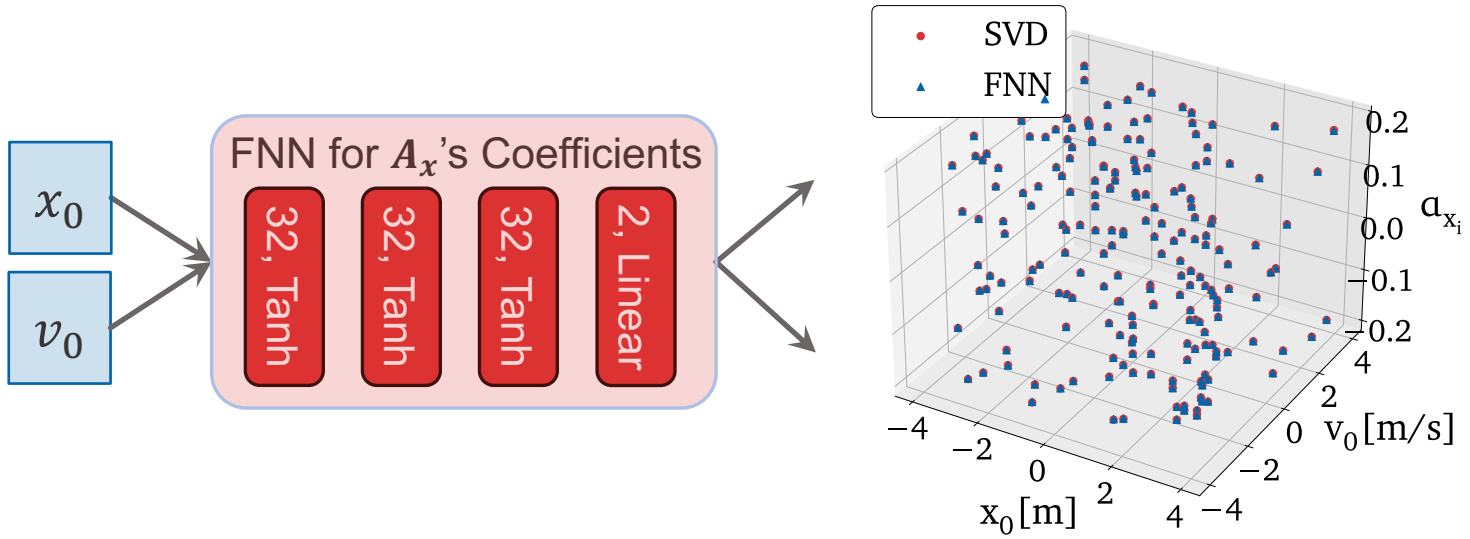


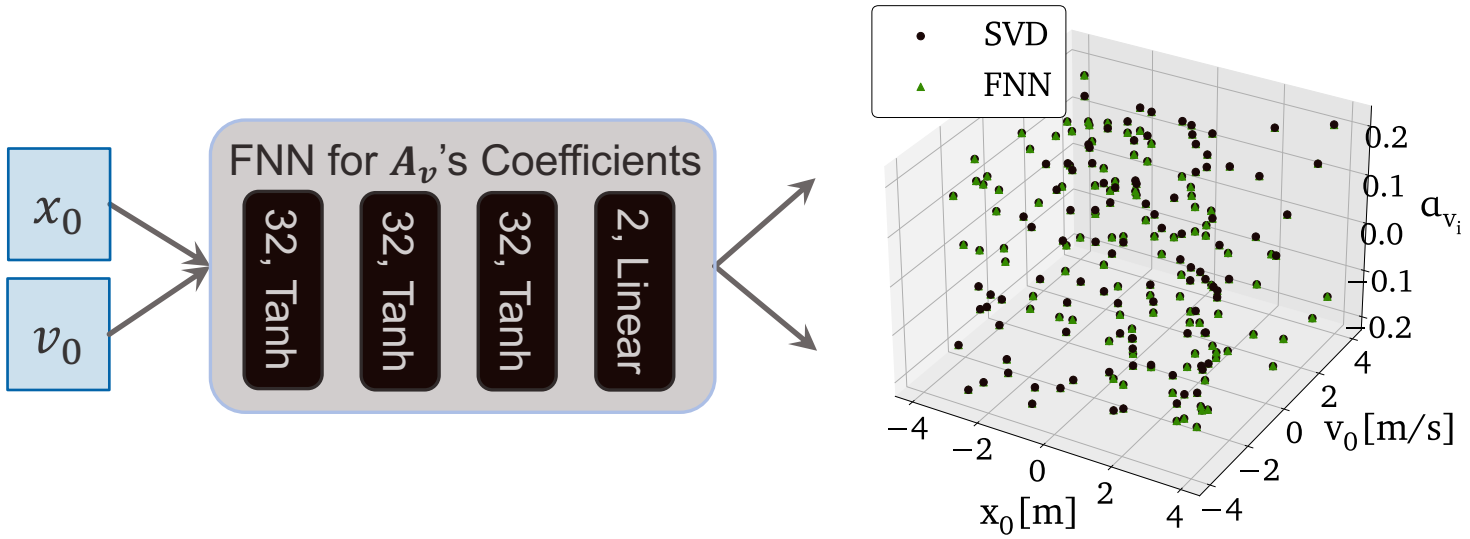
$$A_x = \begin{bmatrix} \alpha_{x_1} & \alpha_{x_2} \end{bmatrix}$$

Where:

$$\dim(A_x) = N_s \times N_\psi = 100 \times 2$$

A Mass-Spring-Damper Test Case





Fitted the $\mathbf{A}_x$ and $\mathbf{A}_v$ components with two independent feed-forward neural networks

A Mass-Spring-Damper Test Case

Test Case 3: SVD-DeepONet:

Train Trunks, Branches and Produce SVD-DeepONet

3.1. Run $WORKSPACE_PATH/ROMNet/romnet/input/ScenarioAggregated_ROMs/MassSpringDamper/FNN/Trunk/Parallelize_ROMNet.py

3.2. Run $WORKSPACE_PATH/ROMNet/romnet/input/ScenarioAggregated_ROMs/MassSpringDamper/FNN/Branch/Parallelize_ROMNet.py

3.3. Copy $WORKSPACE_PATH/ROMNet/romnet/input/MassSpringDamper/DeepONet/MSD_TestCase3/ROMNet_Input.py
to $WORKSPACE_PATH/ROMNet/romnet/input/ROMNet_Input.py

3.4. In $WORKSPACE_PATH/ROMNet/romnet/input/ROMNet_Input.py, change:
3.4.1. "self.WORKSPACE_PATH = ..."

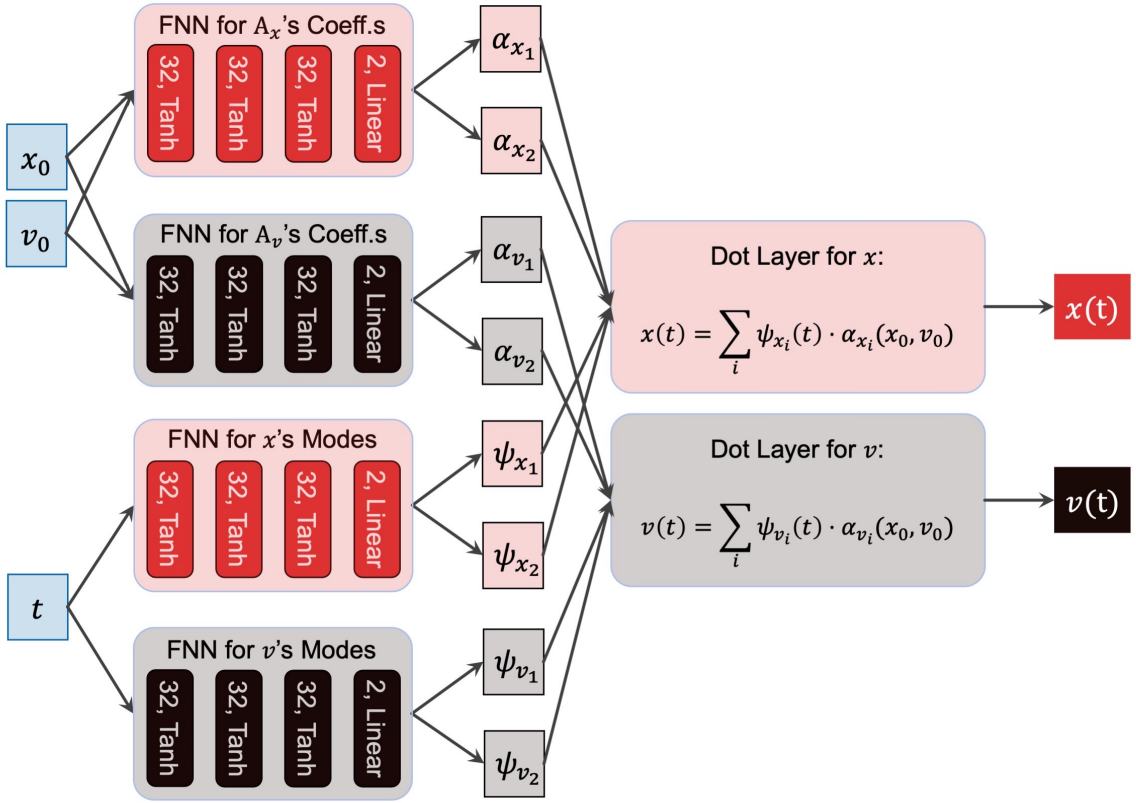
3.5. Move to $WORKSPACE_PATH/ROMNet/romnet/app/

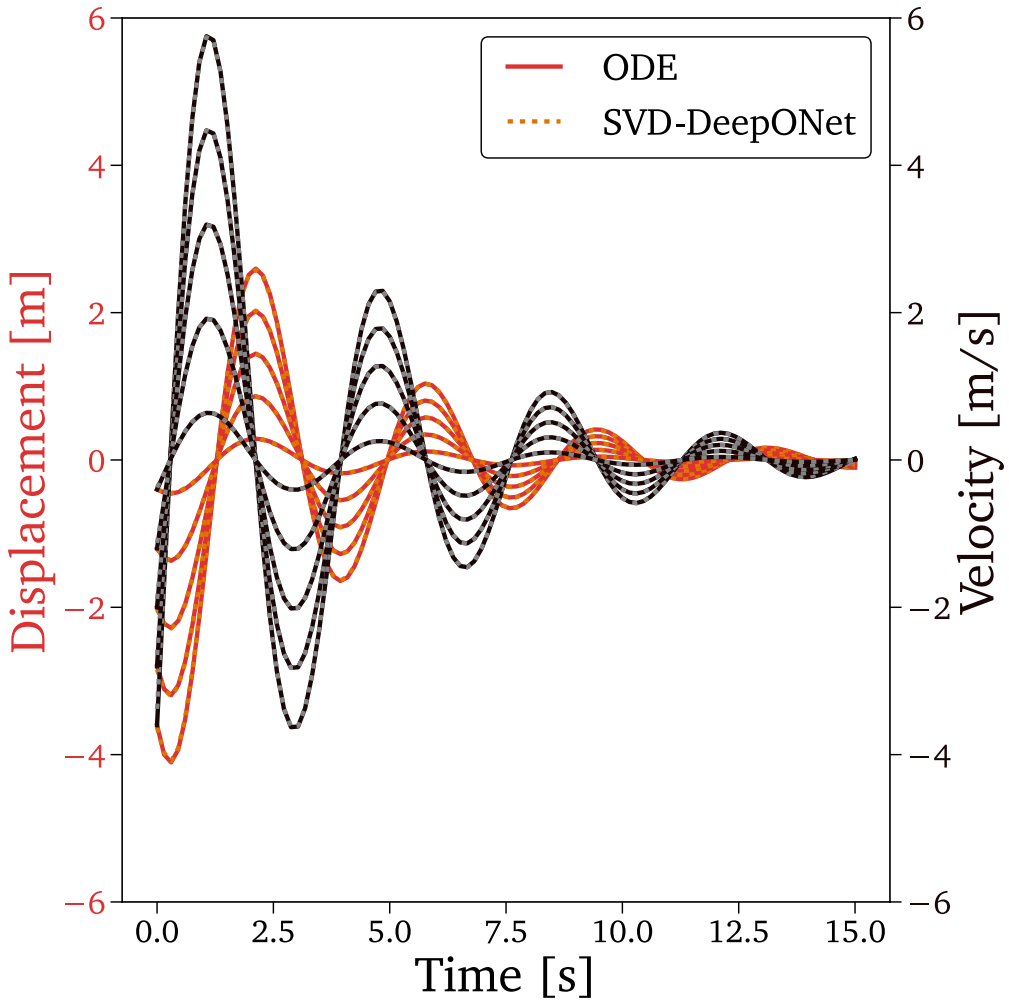
3.6. Run: "python3 ROMNet.py ../input/"

3.7. Postprocess results via: $WORKSPACE_PATH/ROMNet/romnet/scripts/postprocessing/SVD/MassSpringDamper/FNN/Predict_FNN_Trunk.ipynb

A Mass-Spring-Damper Test Case

Generative DeepONet:
uploaded the parameters of the four FNN's
as weights and biases of
the corresponding trunk and branch nets
and predicted test scenarios (w/o any additional training)





PROBABILISTIC EXTENSIONS

A Mass-Spring-Damper Test Case

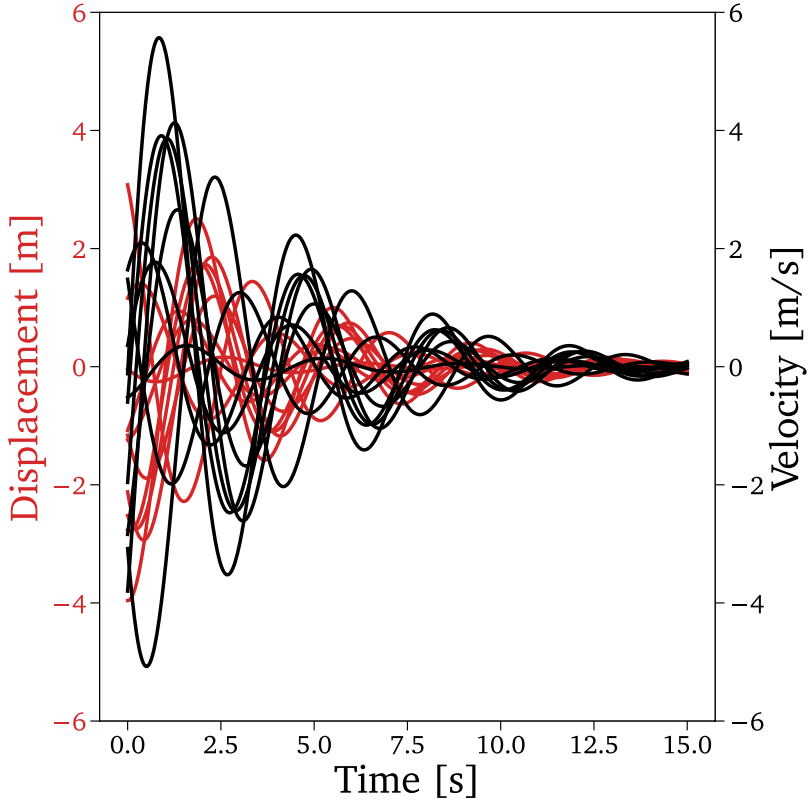
Equations of motion:

$$\begin{cases} m\ddot{x} + c\dot{x} + kx = 0, \\ x(t=0) = x_0, \\ \dot{x}(t=0) = v_0, \end{cases}$$

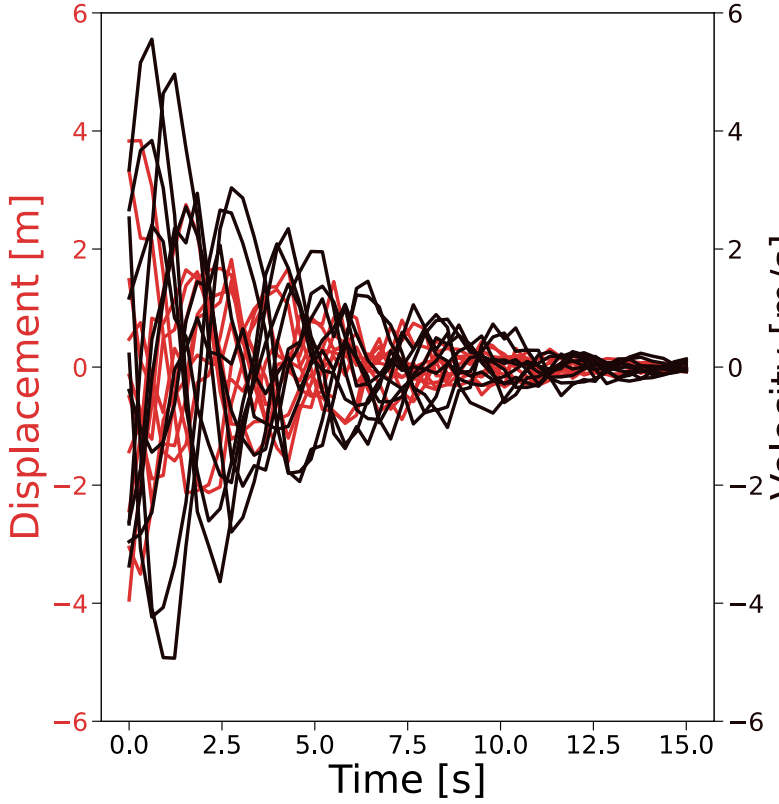
which can be rewritten as:

$$\begin{cases} \begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \\ x(t=0) = x_0, \\ \dot{x}(t=0) = v_0. \end{cases}$$

Some training scenarios
No noise, 500 scenarios, 500 time steps



Some training scenarios
With Noise, 50 scenarios, 50 time steps



$$y_{Data}(t) = y_{ODE}(t) + \mathcal{N}(0,0.05^2)\frac{15-t}{2}$$

A Mass-Spring-Damper Test Case

Run Jupyter Notebook
$WORKSPACE_PATH/ROMNet/romnet/scripts/generating_data/MassSpringDamper/Generate_Data_1_wNoise.ipynb
for generating training and test data

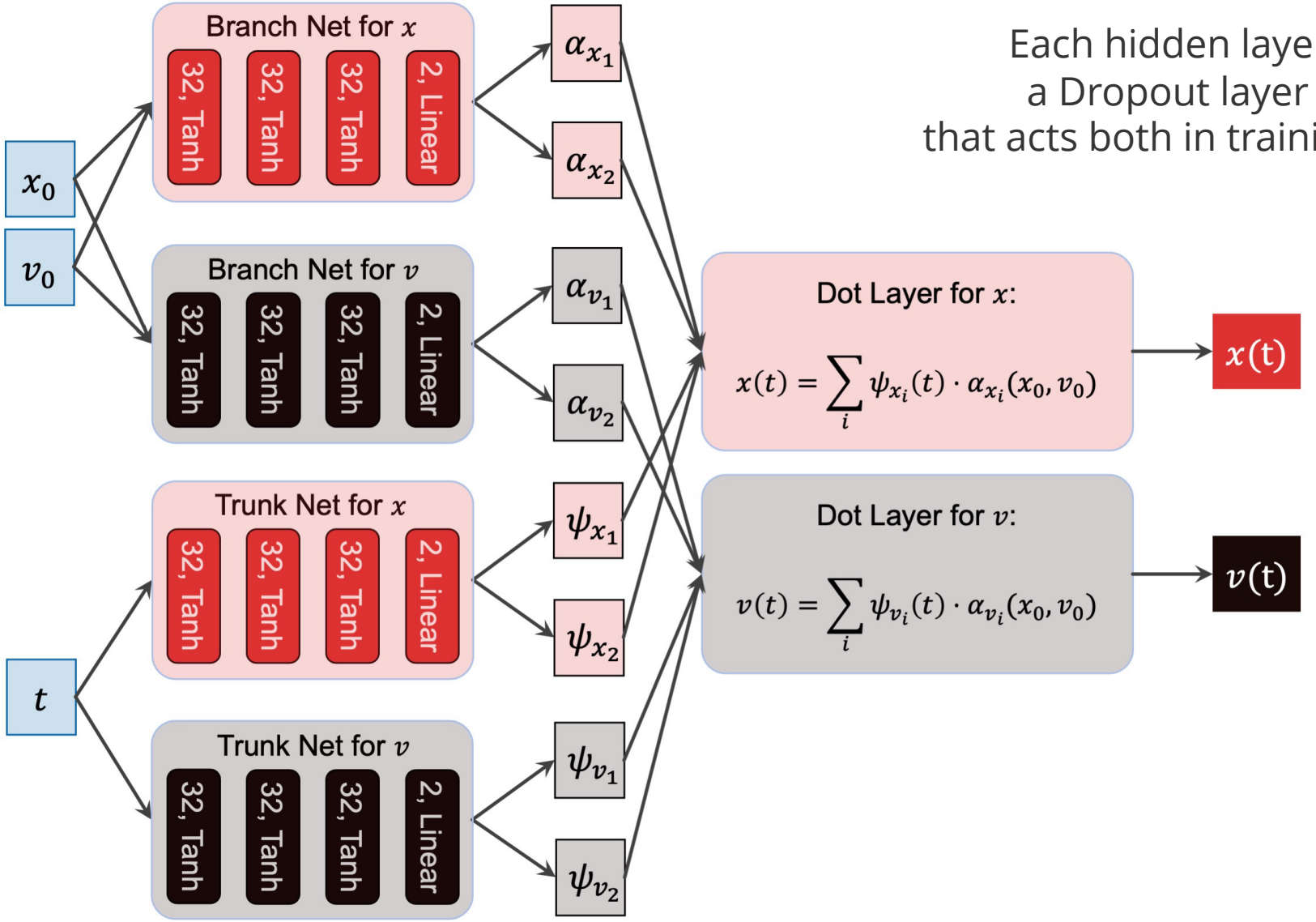
A Mass-Spring-Damper Test Case

Test Cases 5
Data-Driven DeepONet with MC Dropout

A Mass-Spring-Damper Test Case

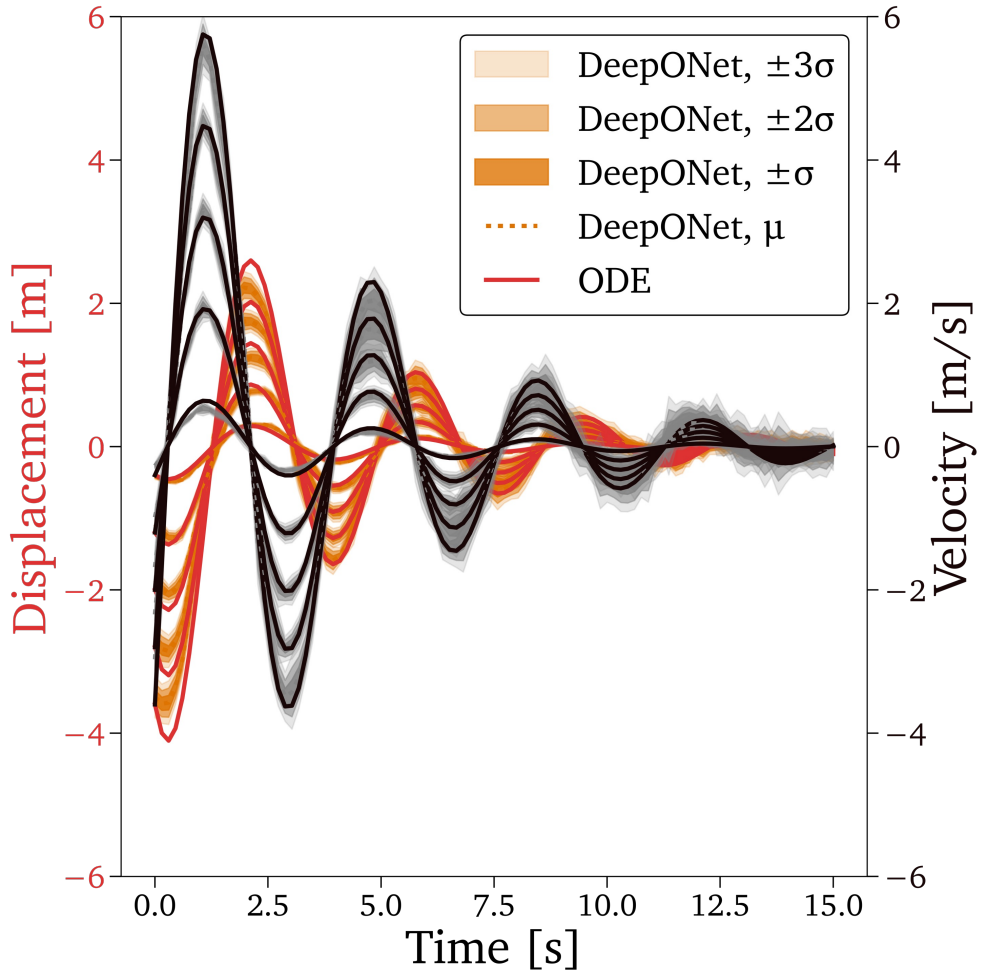
DeepONet Structure

Each hidden layer is followed by
a Dropout layer with rate 1.e-3
that acts both in training and in prediction

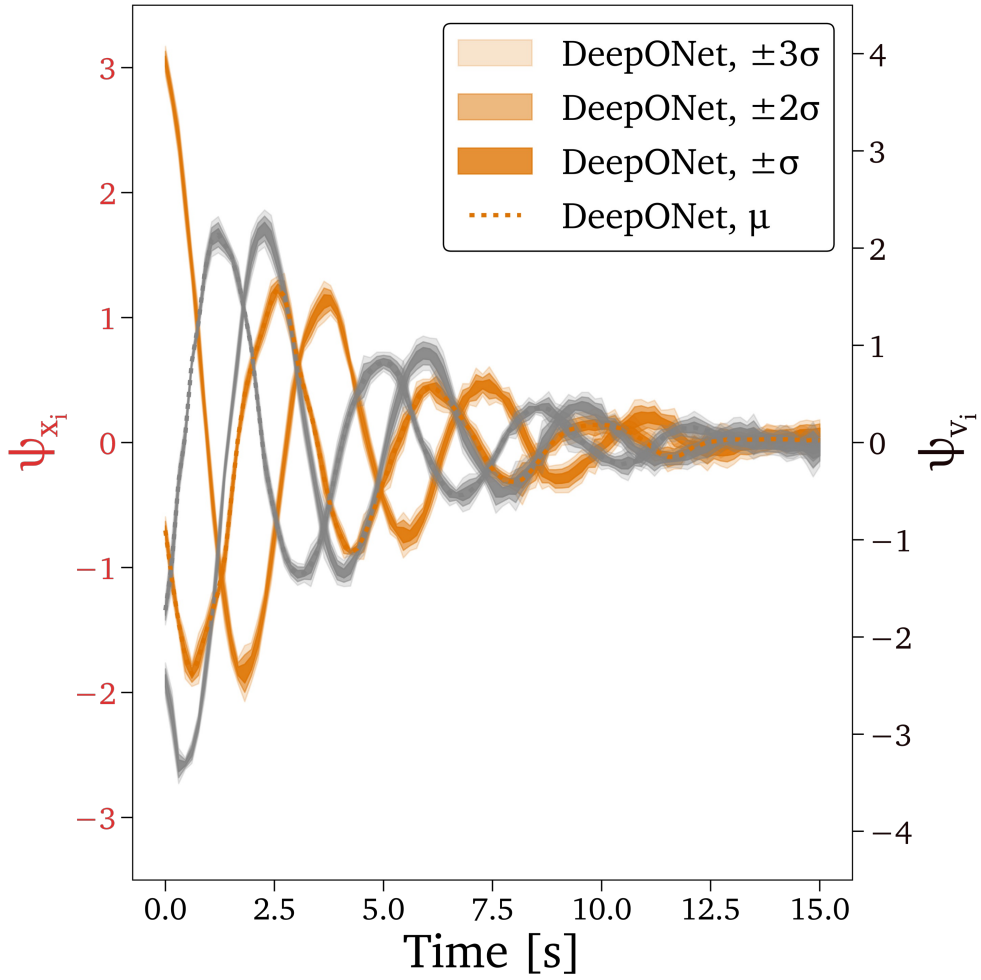


A Mass-Spring-Damper Test Case

Test cases



Modes



A Mass-Spring-Damper Test Case

Test Case 5: Data-driven deep operator network (DeepONet) for predicting position and velocity from corrupted data based on MC Dropout

5.1. Copy $WORKSPACE_PATH/ROMNet/romnet/input/MassSpringDamper/DeepONet/MSD_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/MassSpringDamper/DeepONet/Predict_ProbDeepONet.ipynb

In the input file:

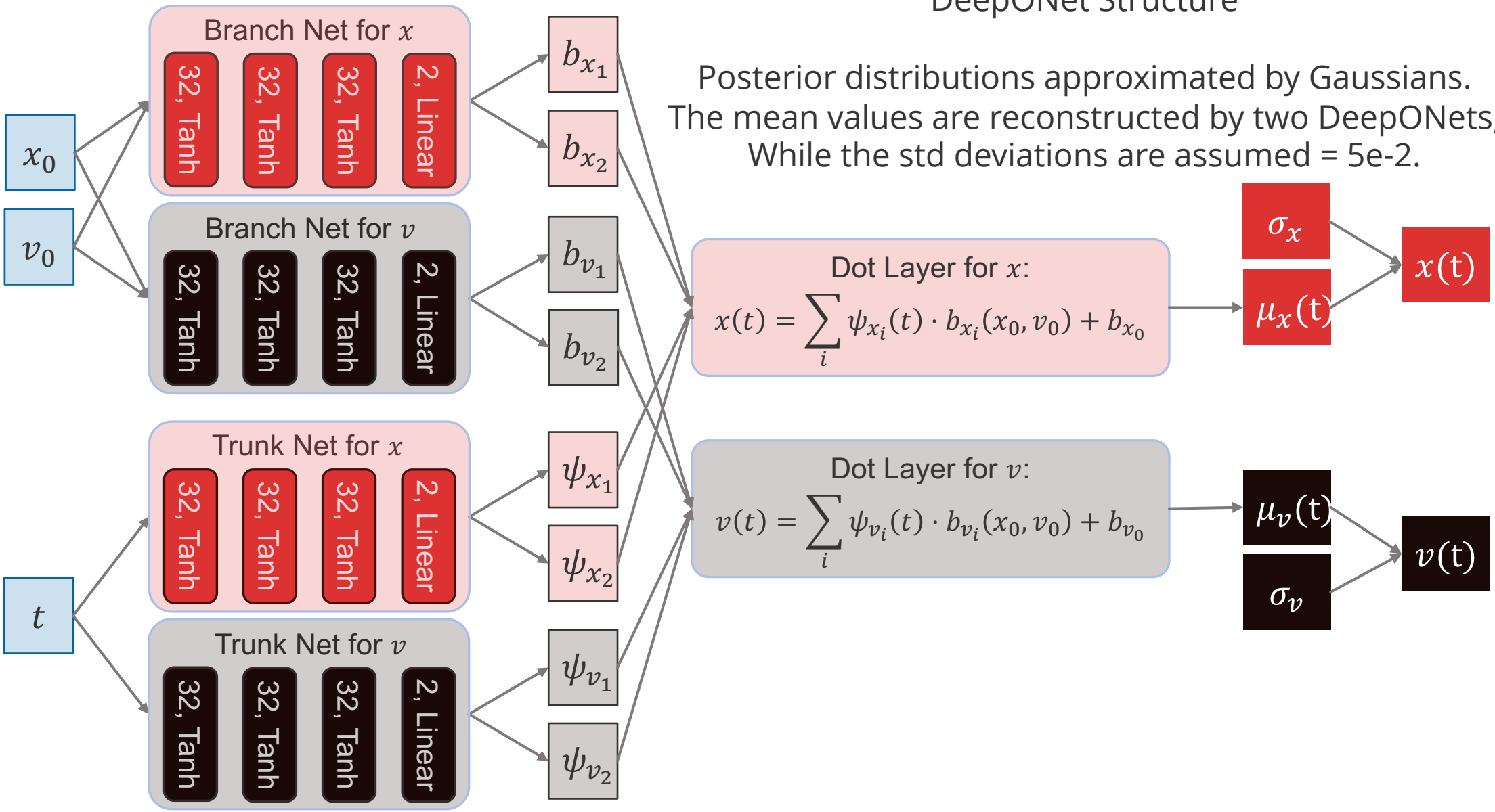
self.**dropout_rate** is a dictionary that controls the dropout rate

self.**dropout_pred_flg** is a dictionary that controls wether to use dropout also in the prediction phase

A Mass-Spring-Damper Test Case

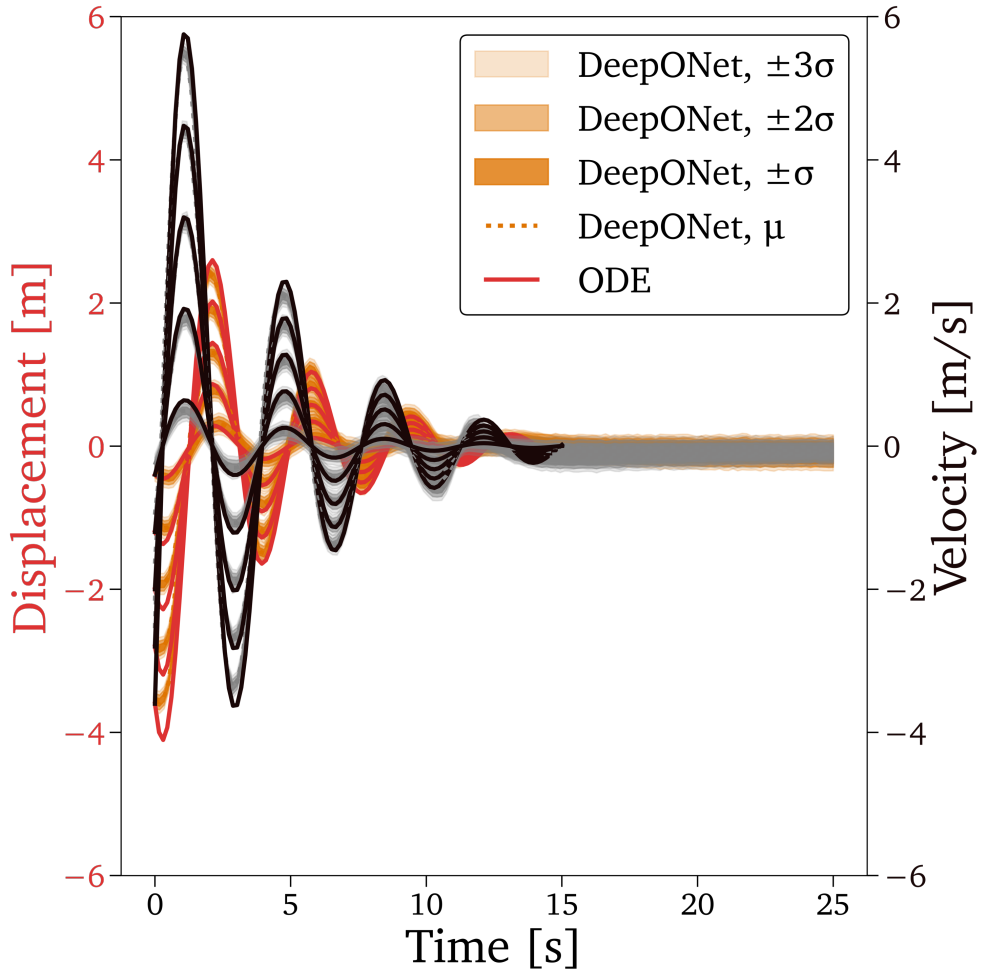
Test Cases 6
Data-Driven Variational-Inference Deterministic DeepONets with Fixed Standard Deviations

A Mass-Spring-Damper Test Case

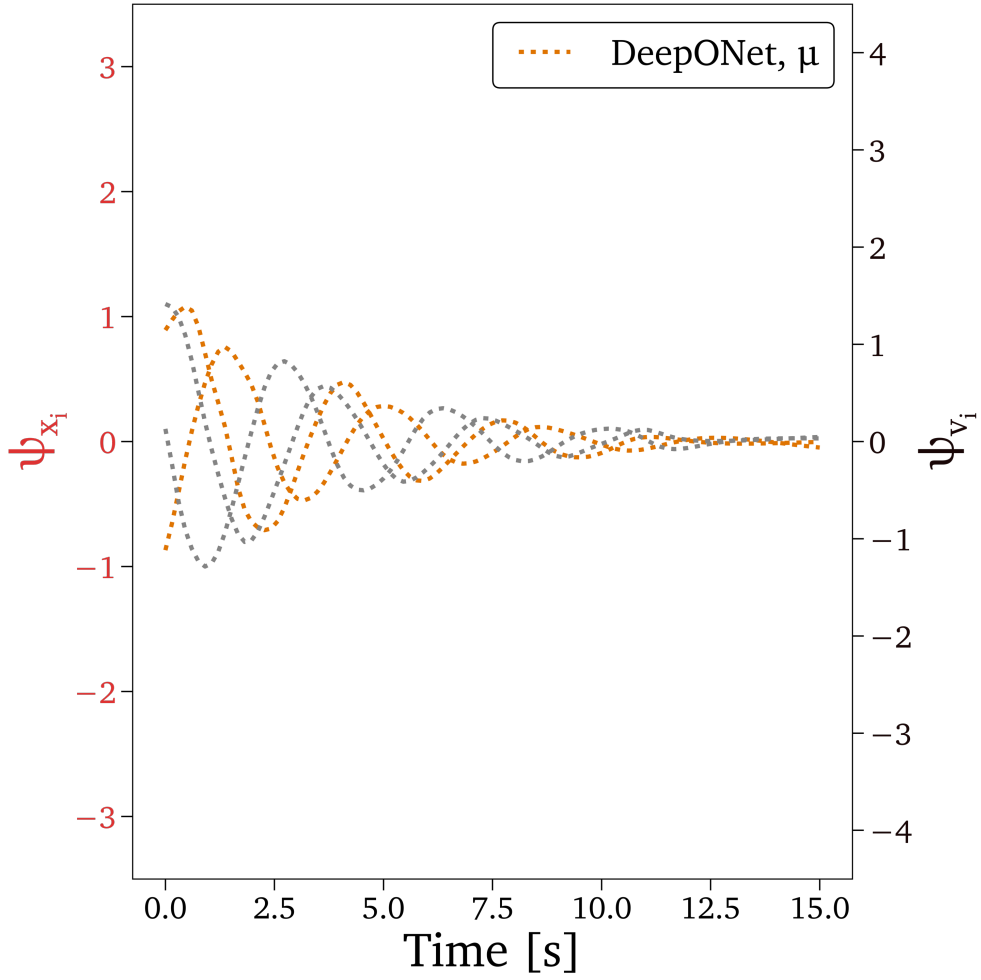


A Mass-Spring-Damper Test Case

Test cases



Modes



A Mass-Spring-Damper Test Case

Test Case 6: Data-driven deep operator network (DeepONet) for predicting position and velocity from corrupted data based on Variational Inference with fixed Std. Deviation

6.1. Copy $WORKSPACE_PATH/ROMNet/romnet/input/MassSpringDamper/DeepONet/MSD_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/MassSpringDamper/DeepONet/Predict_ProbDeepONet.ipynb

In the input file:

self.**surrogate_type** is set to 'VI_DeepONet'

self.**sigma_like** is a vector containing the values of the likelihood standard deviations

self.**losses** is set to negative log-likelihood ('NLL')

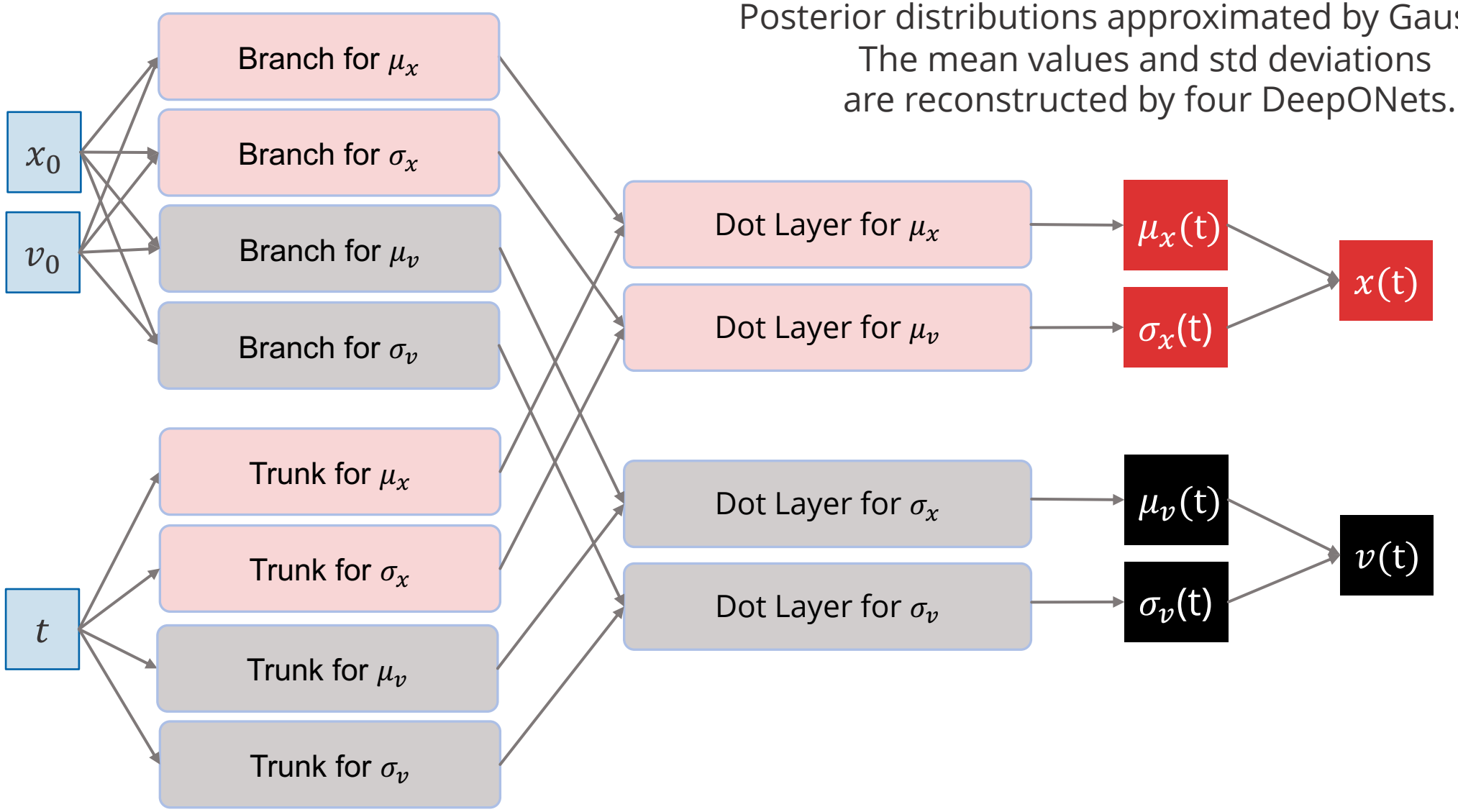
A Mass-Spring-Damper Test Case

Test Cases 7
Data-Driven Variational-Inference Deterministic DeepONets with Calibrated Standard Deviations

A Mass-Spring-Damper Test Case

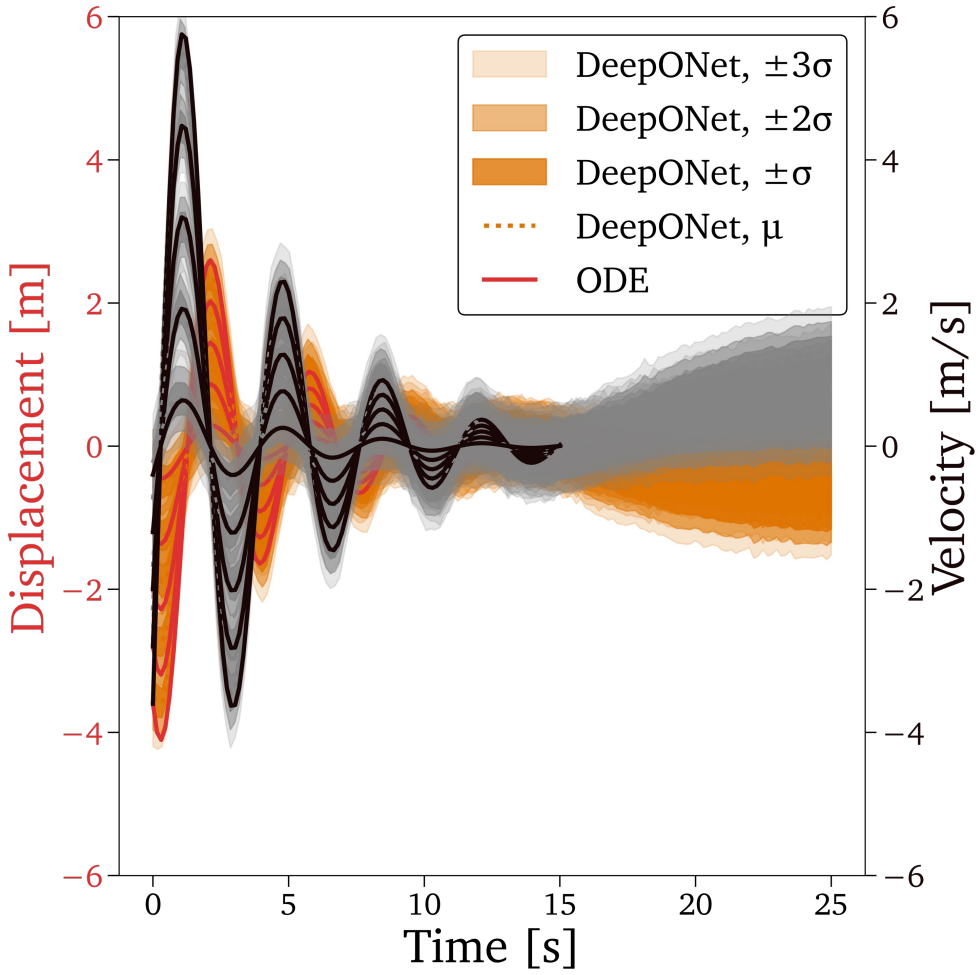
DeepONet Structure

Posterior distributions approximated by Gaussians.
The mean values and std deviations
are reconstructed by four DeepONets.

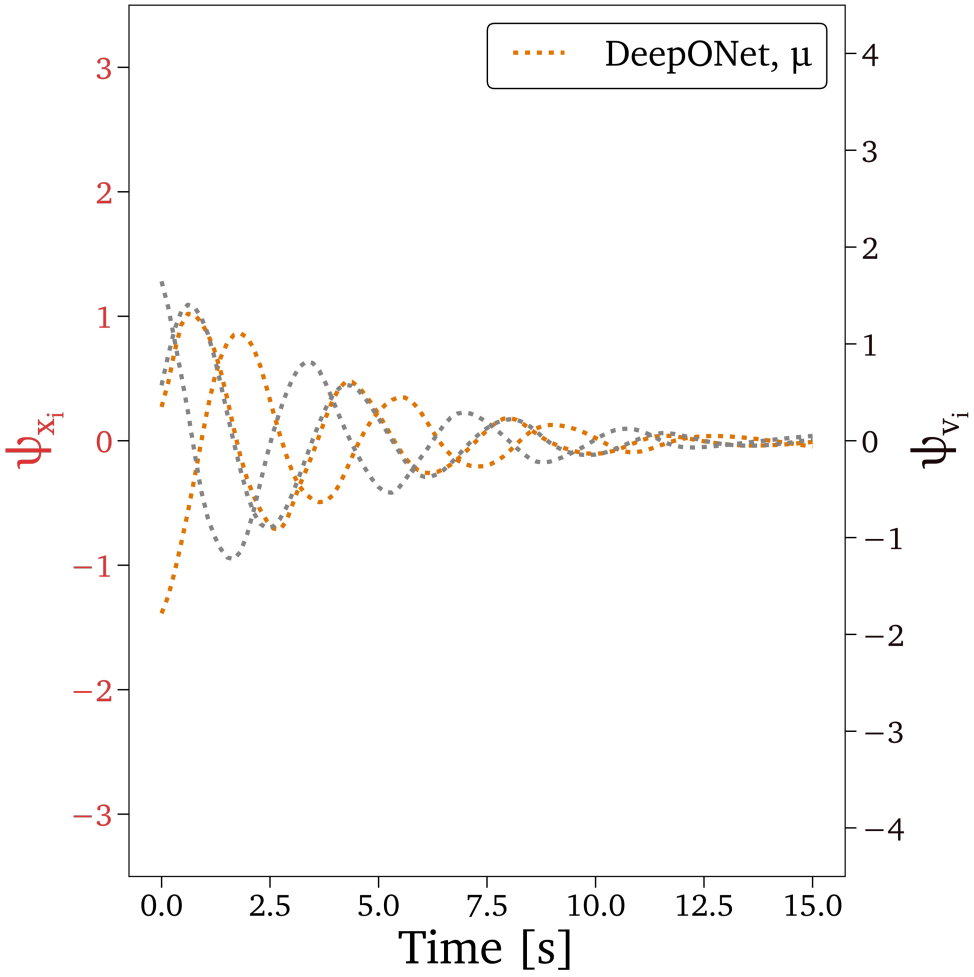


A Mass-Spring-Damper Test Case

Test cases



Modes



A Mass-Spring-Damper Test Case

Test Case 7: Data-driven deep operator network (DeepONet) for predicting position and velocity from corrupted data based on Variational Inference

7.1. Copy $WORKSPACE_PATH/ROMNet/romnet/input/MassSpringDamper/DeepONet/MSD_TestCase7/ROMNet_Input.py
to $WORKSPACE_PATH/ROMNet/romnet/input/ROMNet_Input.py

7.2. In $WORKSPACE_PATH/ROMNet/romnet/input/ROMNet_Input.py, change:
7.2.1. "self.WORKSPACE_PATH = ..."

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

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

7.5. Postprocess results via: $WORKSPACE_PATH/ROMNet/romnet/scripts/postprocessing/MassSpringDamper/DeepONet/Predict_ProbDeepONet.ipynb

In the input file:

self.**surrogate_type** is set to 'VI_DeepONet'

self.**structure** contains two system_of_components: one for the means, the other for the stds

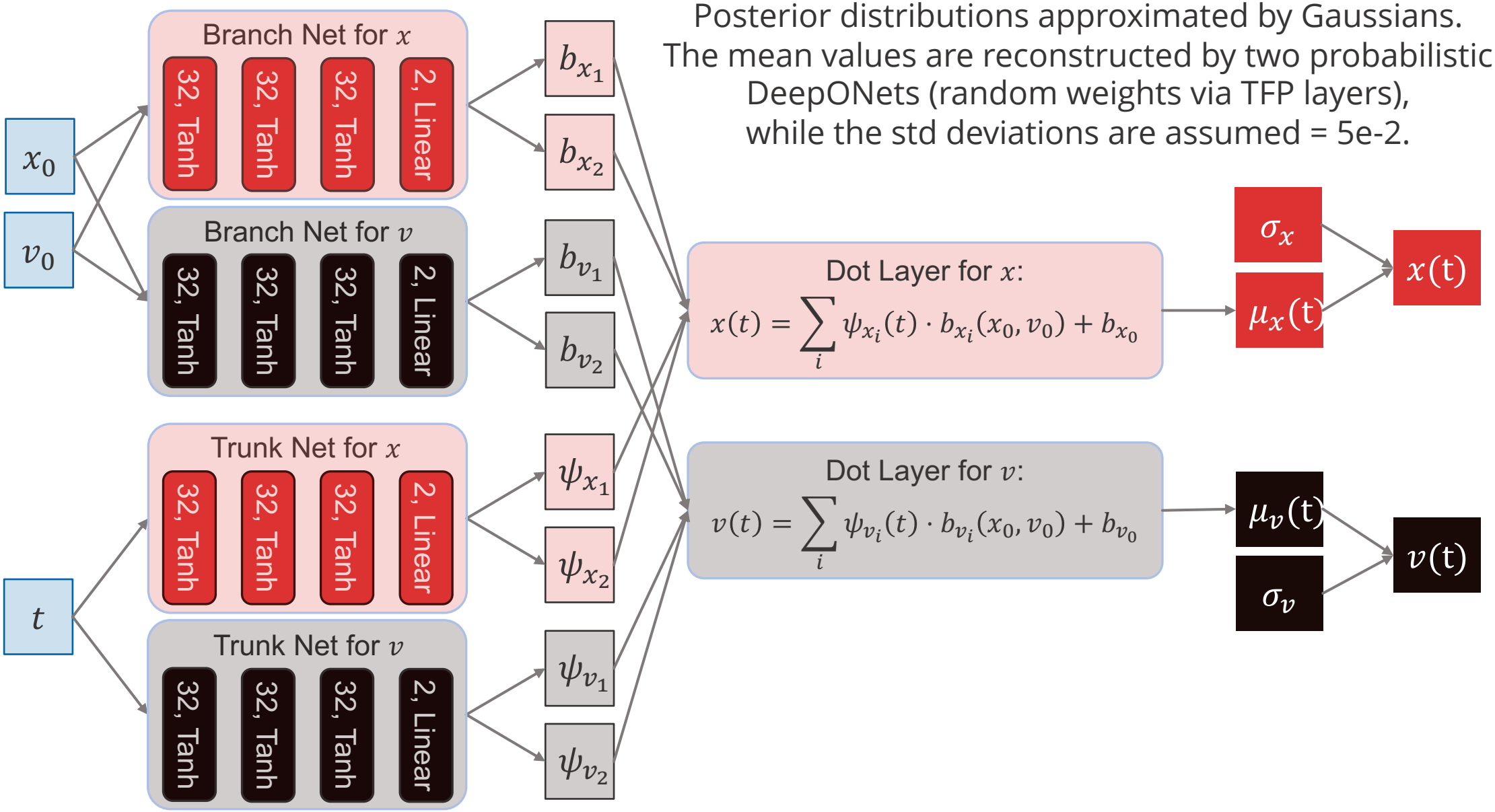
self.**sigma_like** is None

self.**losses** is set to negative log-likelihood ('NLL')

A Mass-Spring-Damper Test Case

Test Cases 8
Data-Driven Variational-Inference Probabilistic DeepONet with Fixed Standard Deviations

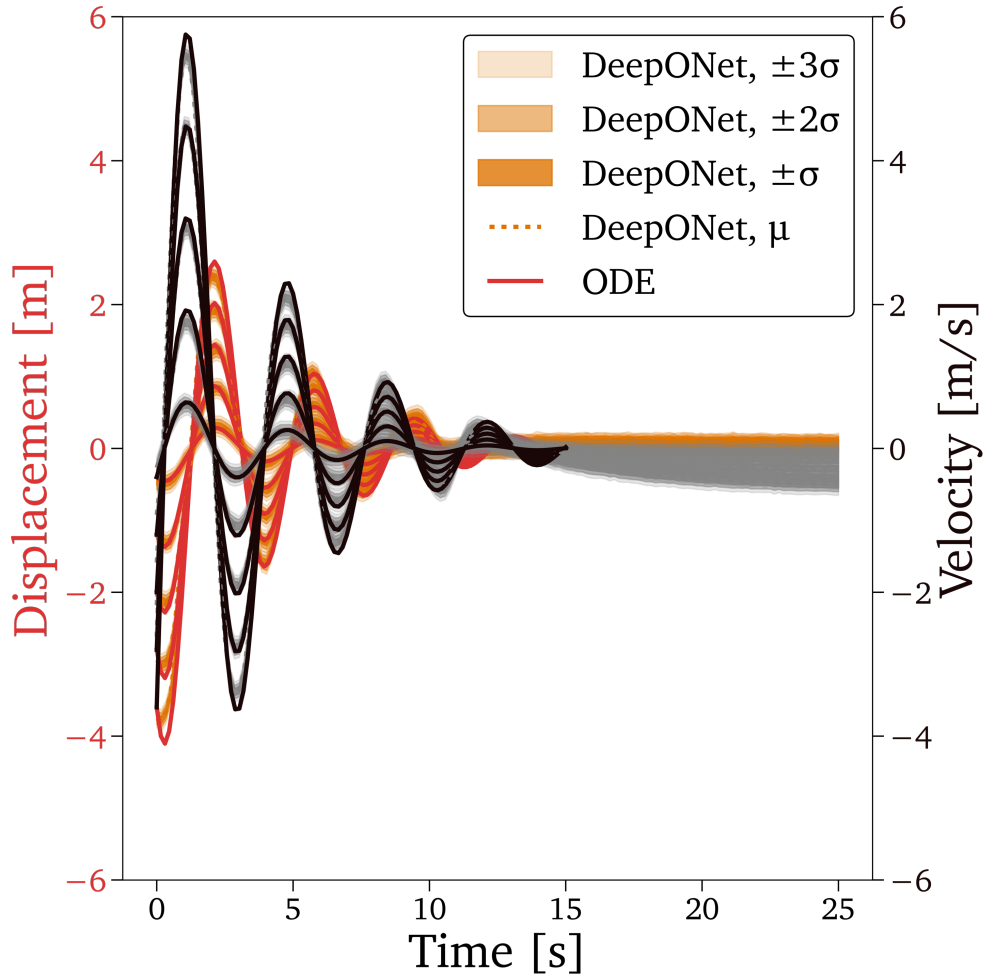
A Mass-Spring-Damper Test Case



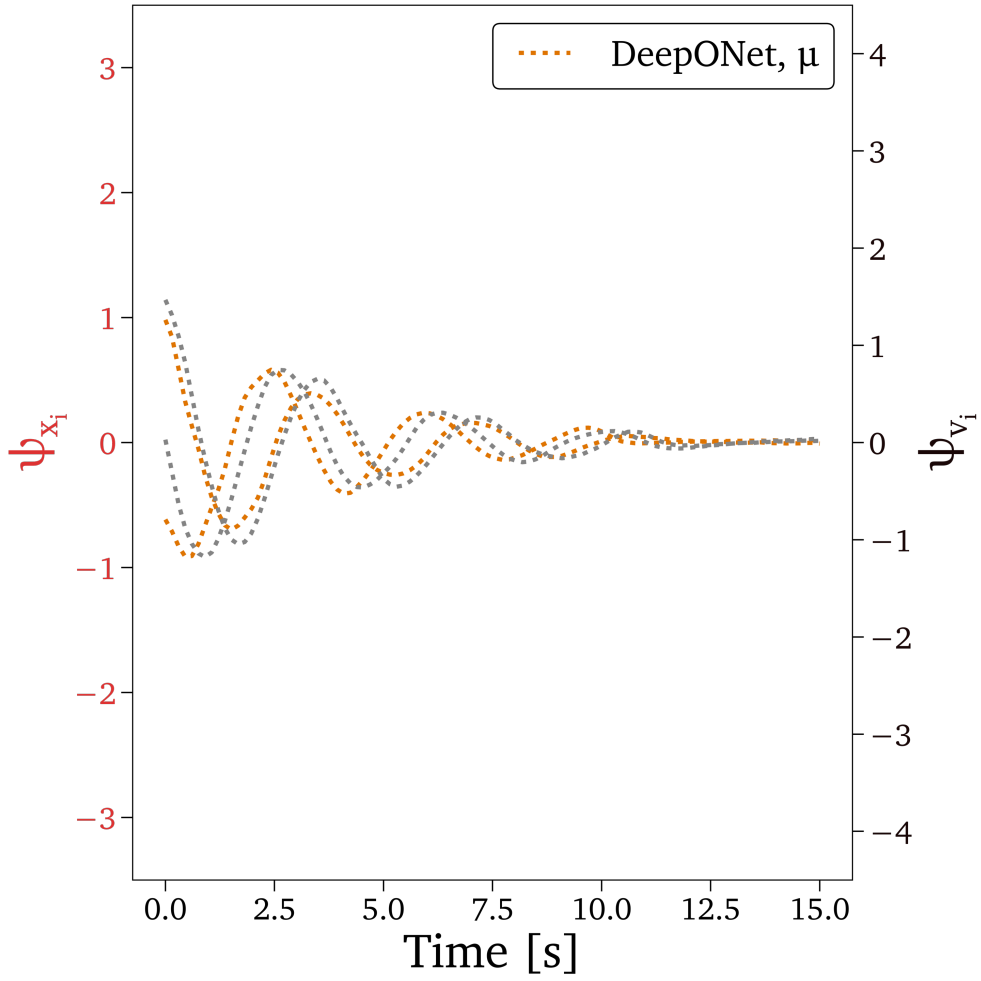
Posterior distributions approximated by Gaussians.
The mean values are reconstructed by two probabilistic DeepONets (random weights via TFP layers), while the std deviations are assumed = 5e-2.

A Mass-Spring-Damper Test Case

Test cases



Modes



A Mass-Spring-Damper Test Case

Test Case 8: Data-driven deep operator network (DeepONet) for predicting position and velocity from corrupted data based on Variational Inference with fixed Std. Deviation

8.1. Copy $WORKSPACE_PATH/ROMNet/romnet/input/MassSpringDamper/DeepONet/MSD_TestCase6/ROMNet_Input.py
to $WORKSPACE_PATH/ROMNet/romnet/input/ROMNet_Input.py

8.2. In $WORKSPACE_PATH/ROMNet/romnet/input/ROMNet_Input.py, change:
8.2.1. "self.WORKSPACE_PATH = ..."

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

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

8.5. Postprocess results via: $WORKSPACE_PATH/ROMNet/romnet/scripts/postprocessing/MassSpringDamper/DeepONet/Predict_ProbDeepONet.ipynb

In the input file:

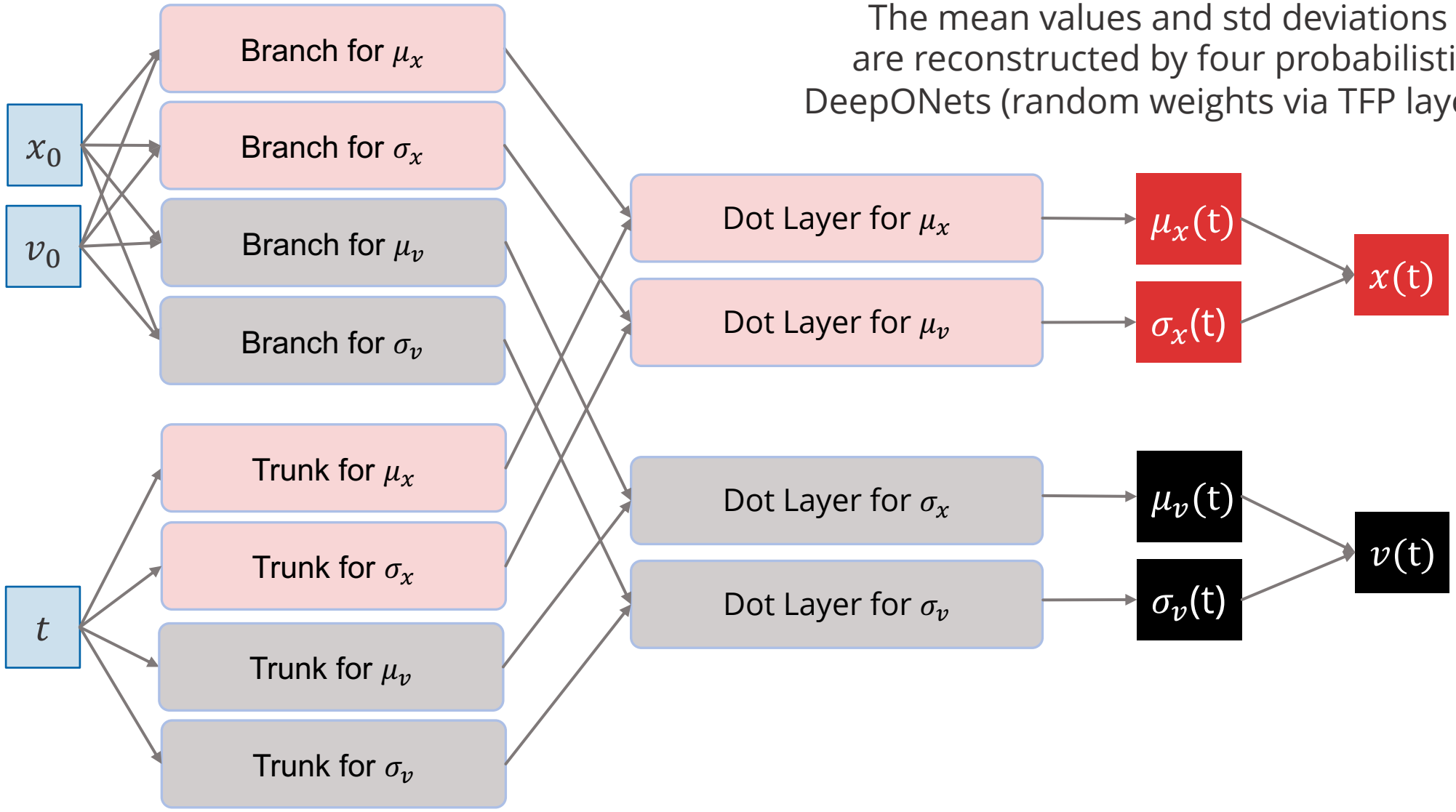
self.**surrogate_type** is set to 'VI_DeepONet'
self.**sigma_like** is a vector containing the values of the likelihood standard deviations
self.**layer_type** is a dictionary containing the layers types (TFP -> tensorflow_probability)
self.**losses** is set to negative log-likelihood ('NLL')

A Mass-Spring-Damper Test Case

Test Cases 9
Data-Driven Variational-Inference Probabilistic DeepONet with Calibrated Standard Deviations

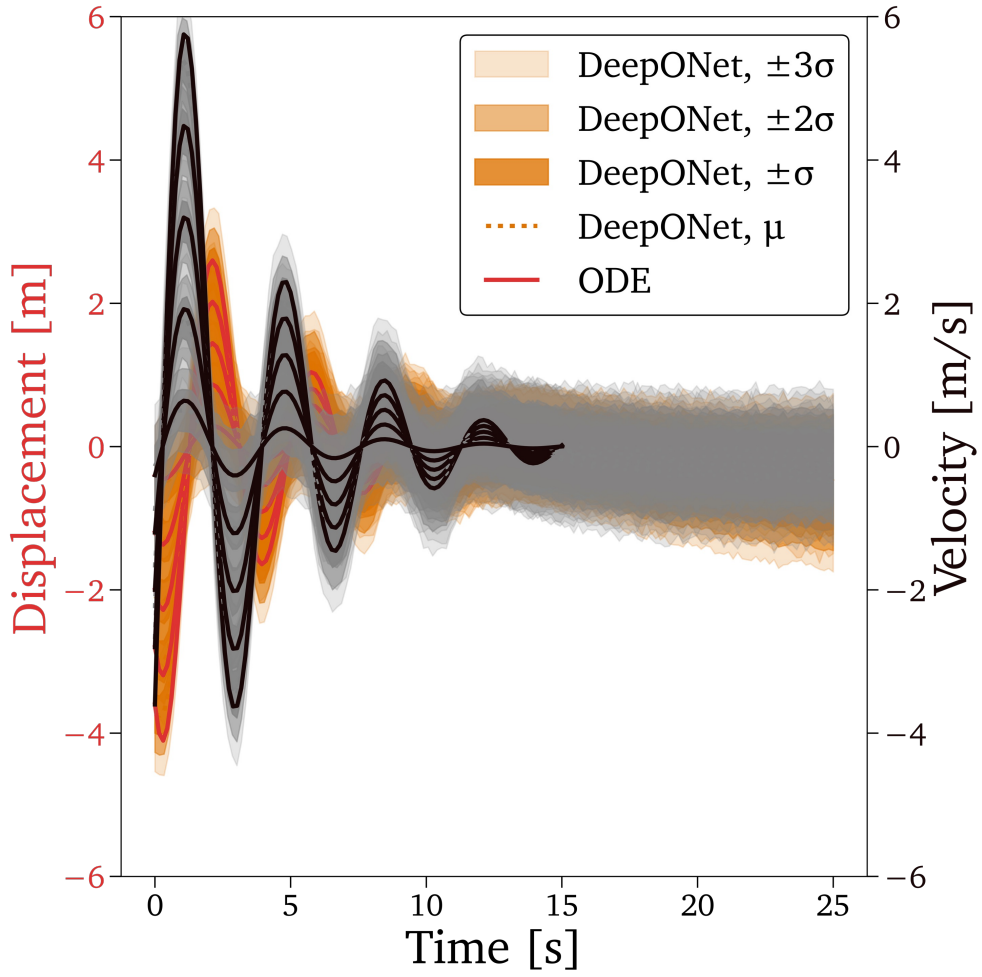
A Mass-Spring-Damper Test Case

Posterior distributions approximated by Gaussians.
The mean values and std deviations
are reconstructed by four probabilistic
DeepONets (random weights via TFP layers).

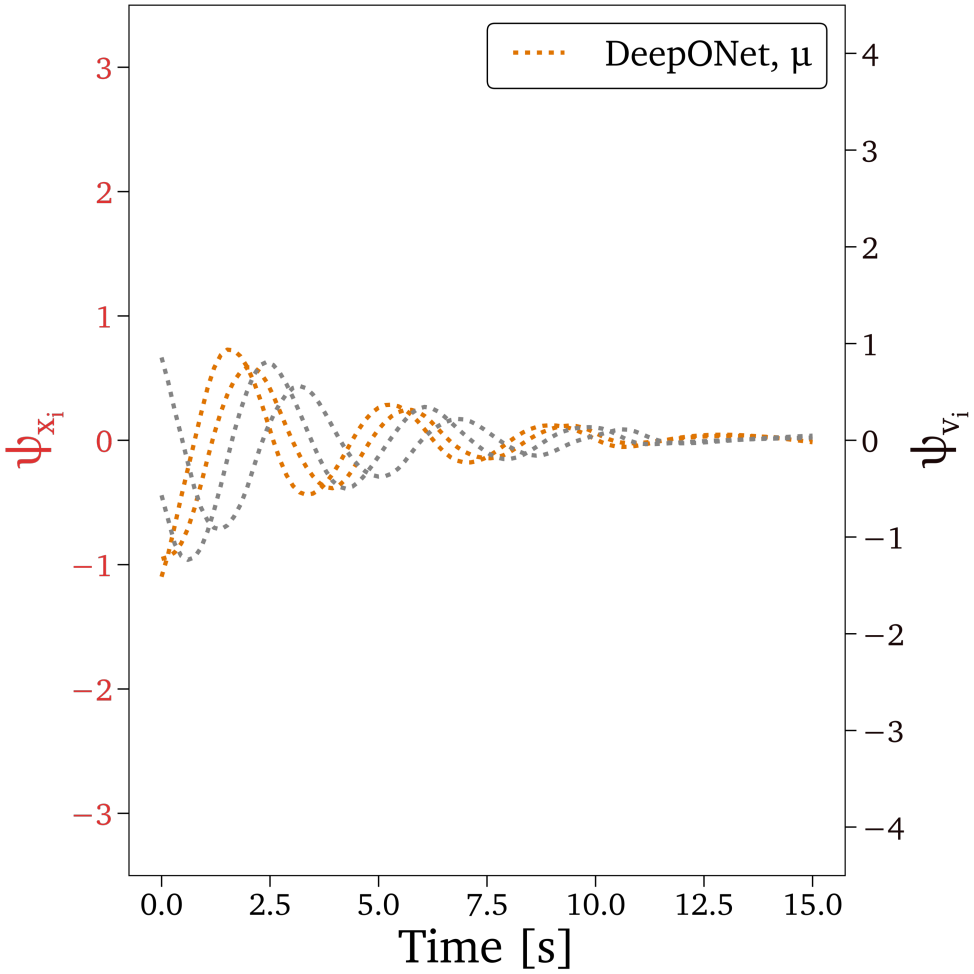


A Mass-Spring-Damper Test Case

Test cases



Modes



A Mass-Spring-Damper Test Case

Test Case 9: Data-driven deep operator network (DeepONet) for predicting position and velocity from corrupted data based on Variational Inference

9.1. Copy $WORKSPACE_PATH/ROMNet/romnet/input/MassSpringDamper/DeepONet/MSD_TestCase7/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/MassSpringDamper/DeepONet/Predict_ProbDeepONet.ipynb

In the input file:

self.**surrogate_type** is set to 'VI_DeepONet'
self.**structure** contains two system_of_components: one for the means, the other for the stds
self.**layer_type** is a dictionary containing the layers types (TFP -> tensorflow_probability)
self.**sigma_like** is None
self.**losses** is set to negative log-likelihood ('NLL')

A Mass-Spring-Damper Test Case

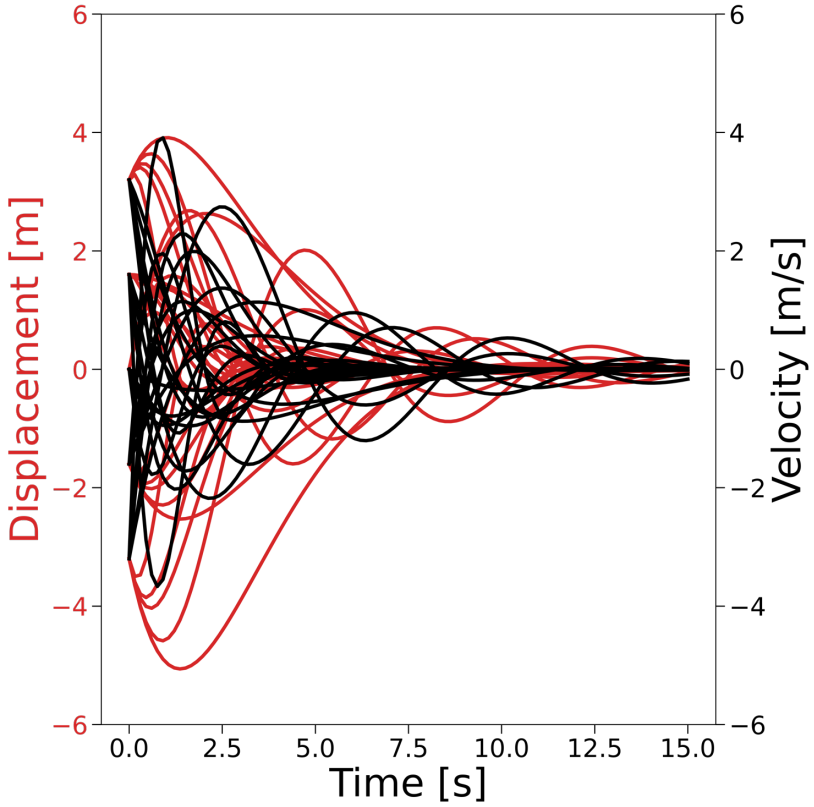
Test Cases 10
Data-Driven MIONet

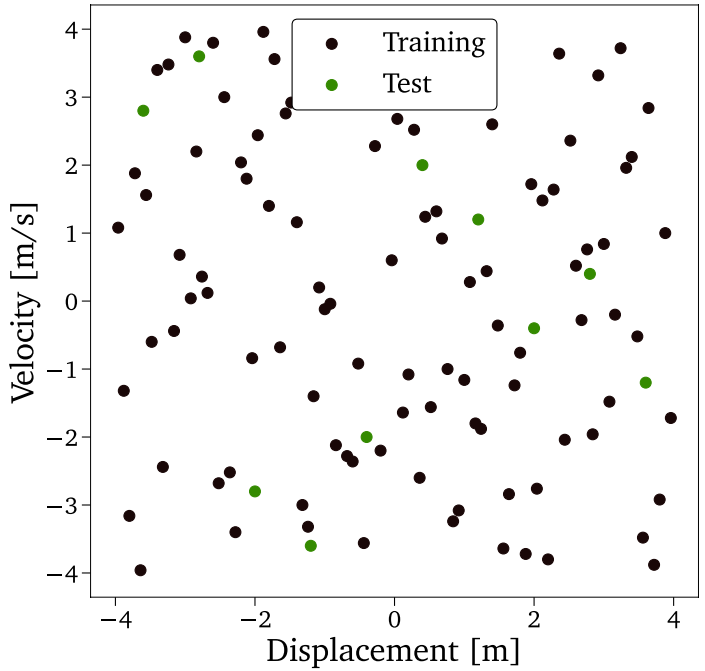
A Mass-Spring-Damper Test Case

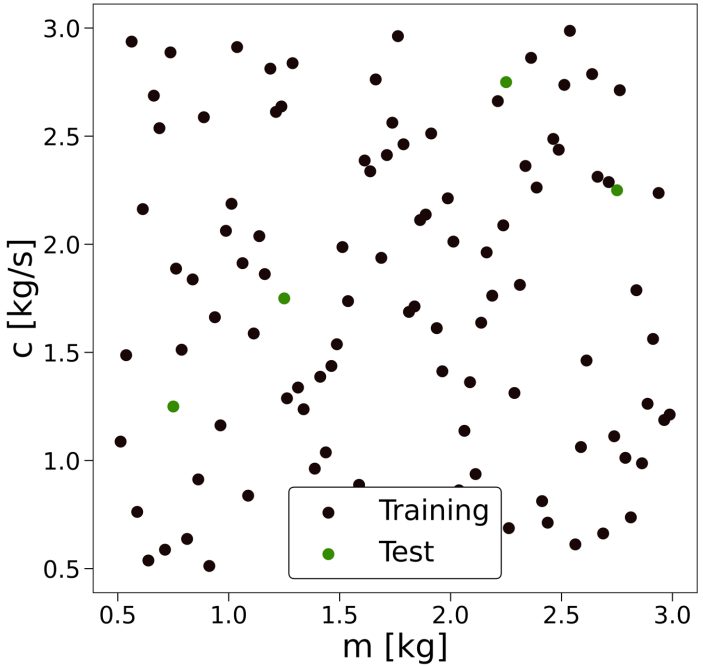
Training – 1,000,000 Data Points:

- 100 different I.C.s
  (LHS in $[-4.0 - 4.0]^2$)
  - **For each I.C., 100 different sets of parameters (LHS in $[0.5 - 3.0]^3$)**
    - For each I.C. and parameter set, 100 time instants

Some training scenarios



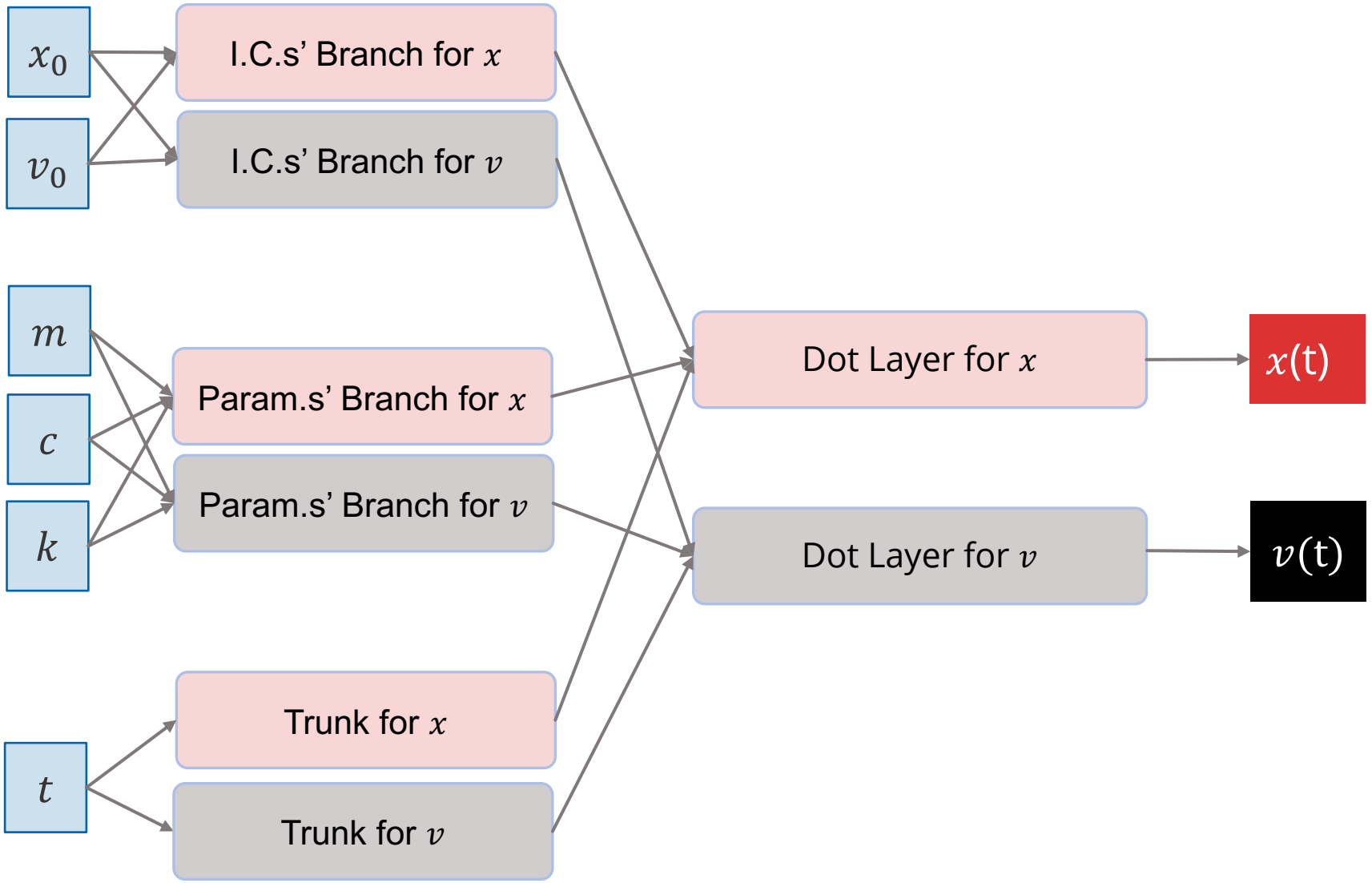




A Mass-Spring-Damper Test Case

Run Jupyter Notebook
$WORKSPACE_PATH/ROMNet/.../generating_data/MassSpringDamper/Generate_Data_1_UncertainParams.ipynb
for generating training and test data

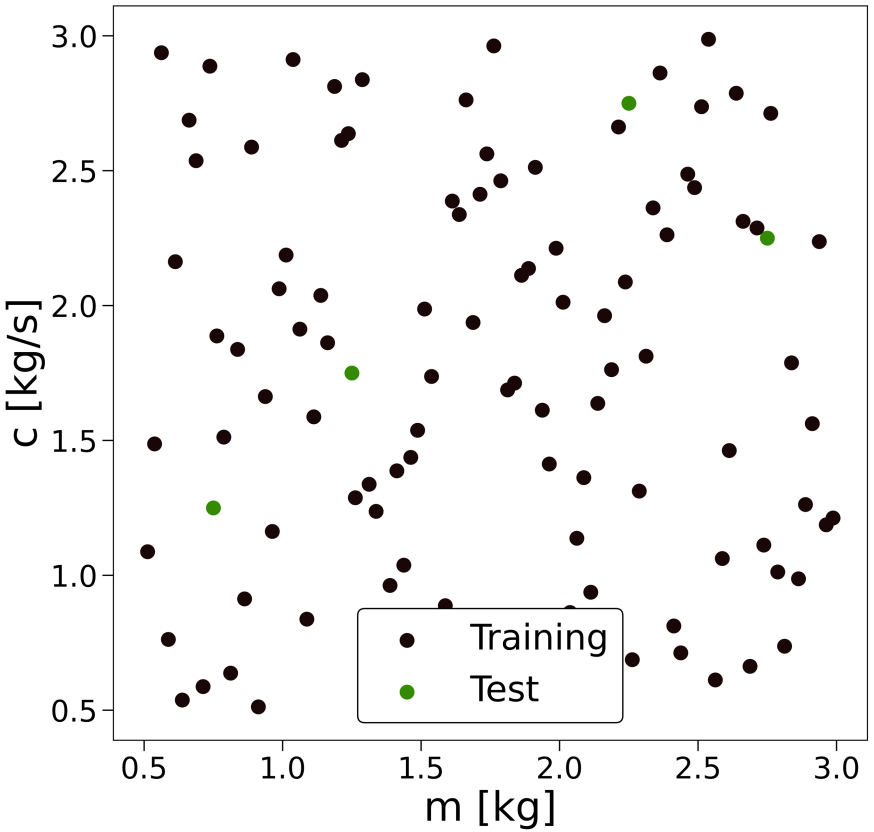
A Mass-Spring-Damper Test Case



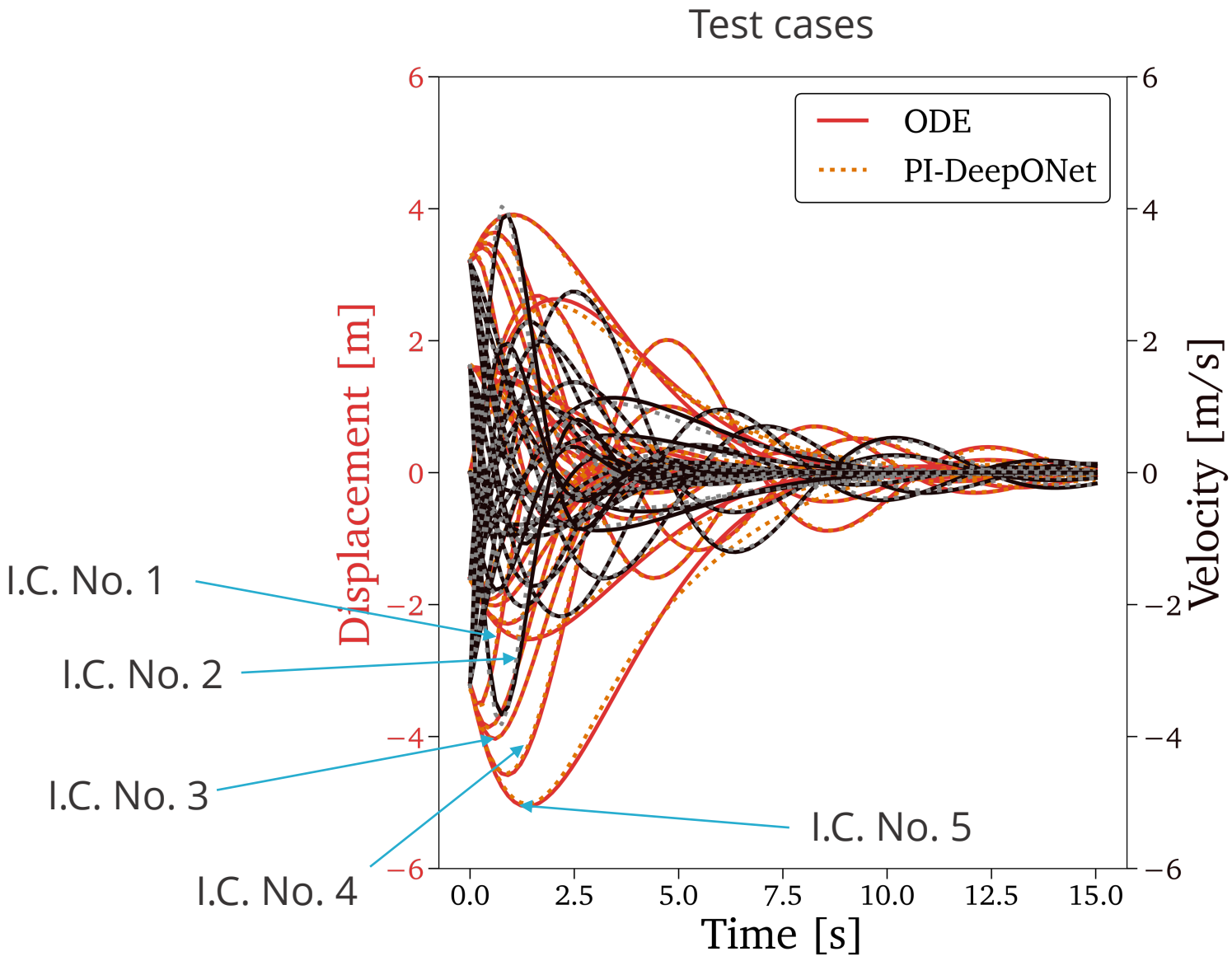
A Mass-Spring-Damper Test Case

Test:

- 5 different I.C.s
  - For each I.C., 5 different sets of parameters
    - For each I.C. and parameter set, 100 time instants



A Mass-Spring-Damper Test Case



A Mass-Spring-Damper Test Case

Test Case 10: Data-driven MIONet for predicting position and velocity for varying operator parameters

9.1. Copy $WORKSPACE_PATH/ROMNet/romnet/input/MassSpringDamper/MIONet/MSD_TestCase10/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/MassSpringDamper/MIONet/Predict_MIONet.ipynb

In the input file:

self.**surrogate_type** is set to ‘MIONet’

self.**structure** contains two branches: one for the I.C.s, the other for the parameters