Applying Time Shift Symmetries in Normalizing Flows for Likelihood Free Inference

Maanas Goel
Advisors: Dr. Deep Chatterjee, Dr. Erik Katsavounidis

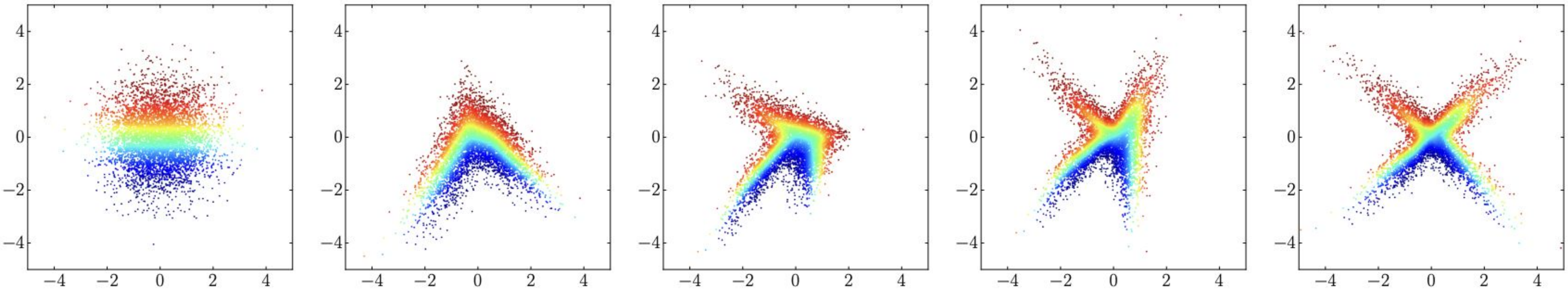
Research Internship
(June 2023 - Present)

Motivation

- ❖ Parameter estimation takes time
- ❖ Likelihood free inference using normalizing flows can accelerate this process, but
  - ➢ Learning symmetries like time translation is hard for naive neural networks
  - ➢ Computationally more time and more parameters
- ❖ Build a more informed normalizing flow by embedding time translation symmetry
- ❖ Consider sine gaussian parameter estimation
  - ➢ Frequency, duration, skylocation are important parameters
  - ➢ Central time of arrival not so much
- ❖ Make a neural network be invariant to time translations
  - ➢ Infer parameters such as frequency, duration, skylocation with a smaller model

Normalizing Flow

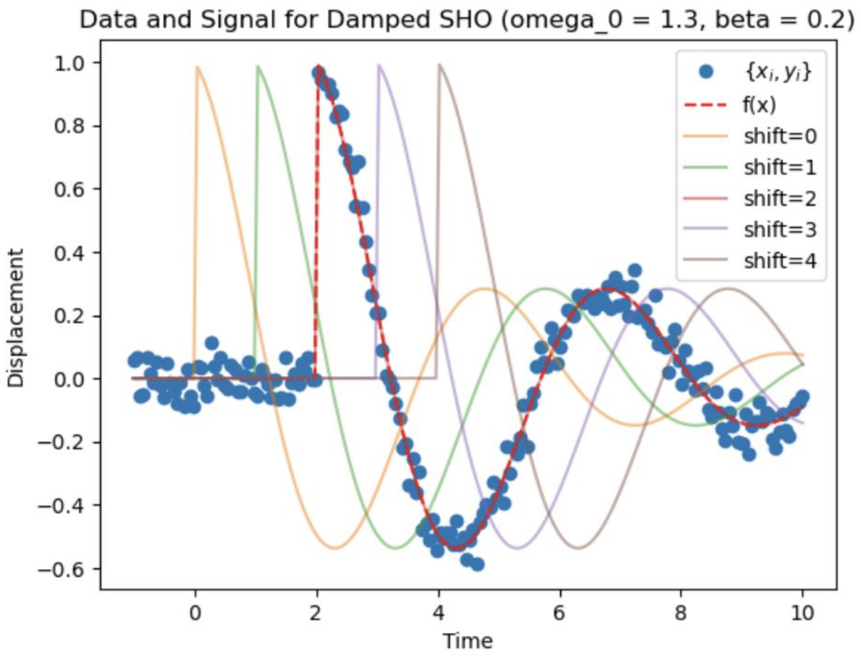
- ❖ Apply a transformation T to some vector u, sampled from a base distribution $p_u(u)$ to represent the target distribution x
- ❖ Flow: the invertible transformations can be composed with other transformations to create multidimensional complex transformations
- ❖ Normalizing: change of variables gives a normalized probability density after applying an invertible transformation [2]

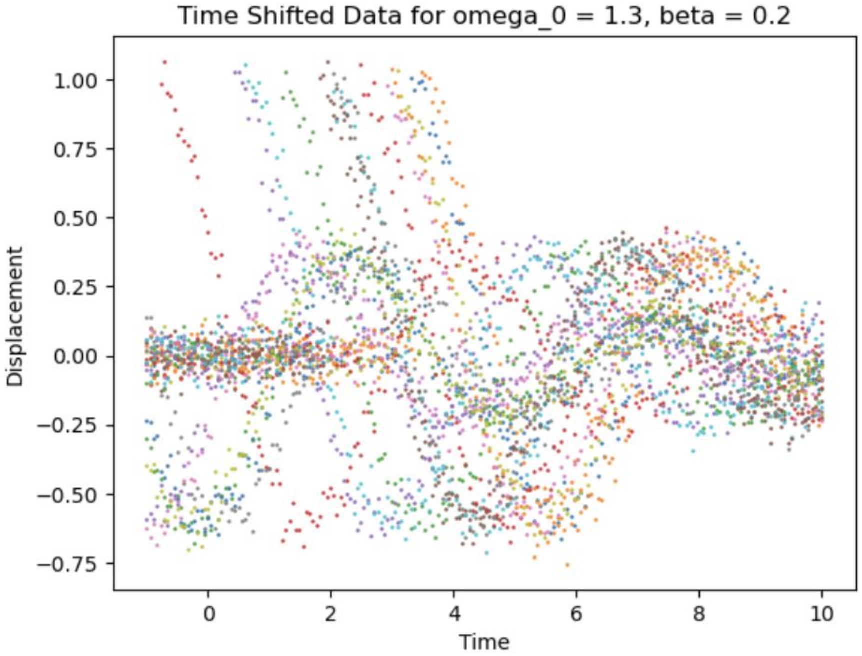


Damped Harmonic Oscillator

$$\ddot{x}(t) + 2\beta\omega_0\dot{x}(t) + \omega_0^2 x(t) = 0$$

$$, x = \exp(\gamma t); \gamma = -\omega_0 \left[\beta \pm i\sqrt{1-\beta^2}\right]$$





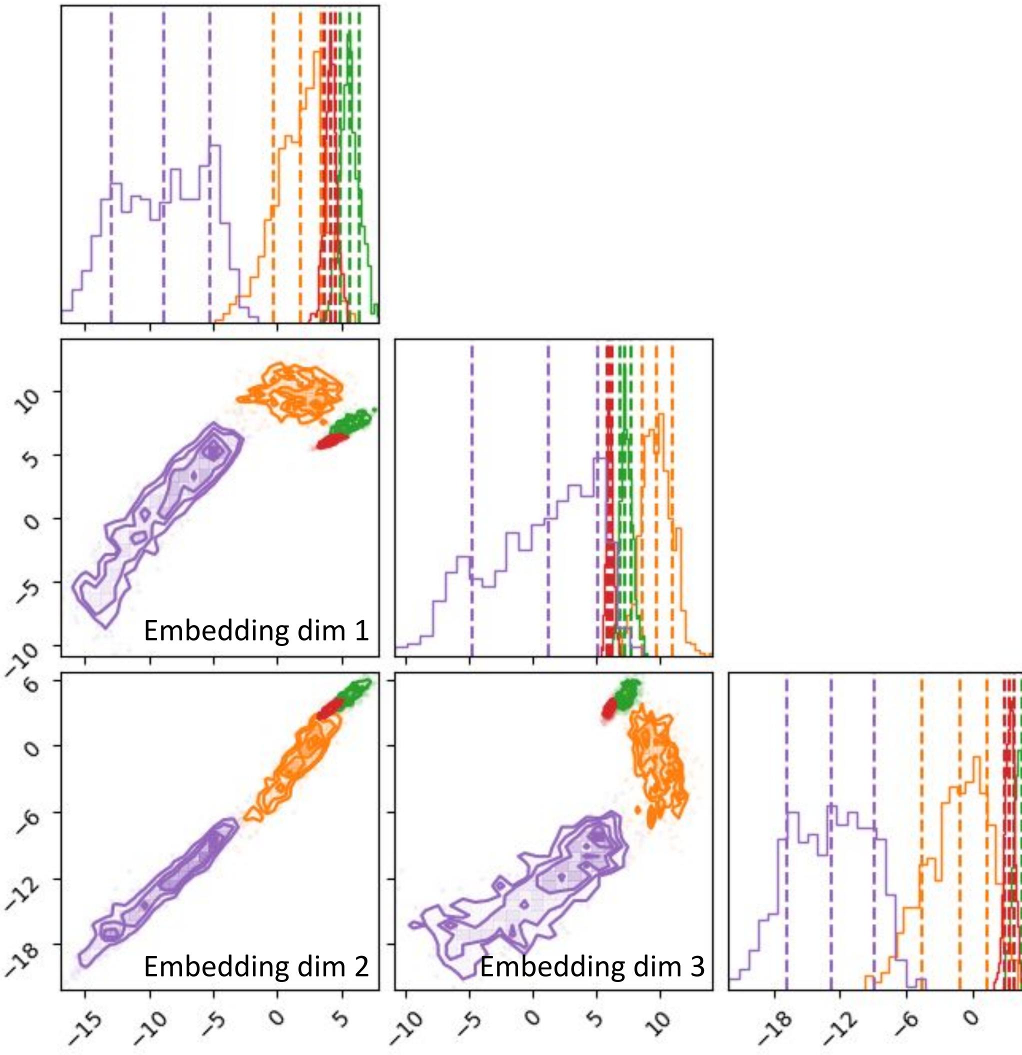
Inferring on Omega (natural frequency), Beta (damping coefficient)

Method to implement Similarity Embedding

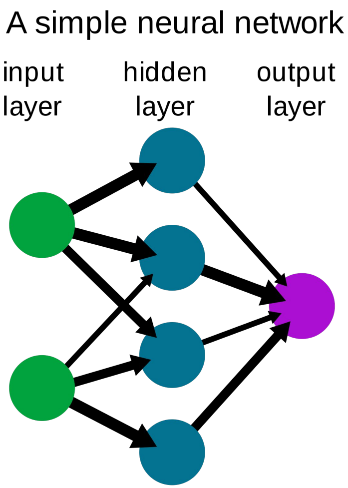
- VICReg Loss (Variance-Invariance-Covariance Regularization)

  https://arxiv.org/pdf/2105.04906.pdf

- Weighted sum of 3 loss components
  - Variance: makes dense
  - Invariance: makes same data closer
  - Covariance: makes circular [1]



Why implement a Neural Network



- ❖ Simply using a normalizing flow will learn all the time shifted data as unique data

- ❖ We're more interested in determining the signal's frequency, duration, skylocation
  - ➢ We want to be agnostic to time shifts

- ❖ If we simply train a normalizing flow, we'd need a <u>bigger model</u> and <u>more data</u> to train
  - ➢ More computational cost

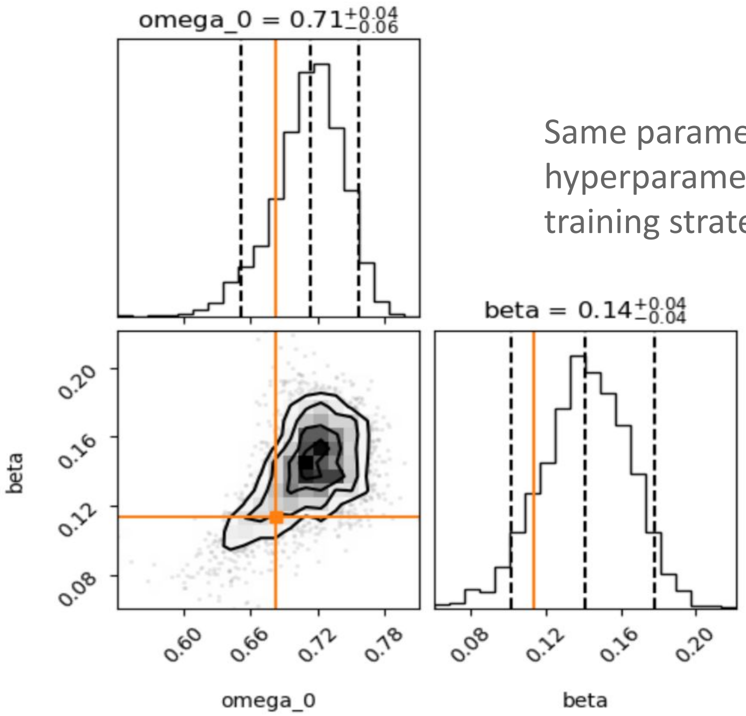
- ❖ Therefore, we used a symmetry informed normalizing flow, which requires a <u>smaller model</u> and <u>less data</u>

Embedding Net in the Flow

- ❖ With the similarity embedding
  - ➢ Freeze the weights from the similarity embedding training
  - ➢ Pass the weights through the Masked Affine Autoregressive Transform
- ❖ Without the similarity embedding
  - ➢ Do not train similarity embedding
  - ➢ Do not freeze the parameters
  - ➢ Let the training of the normalizing flow change the weights of the similarity embedding

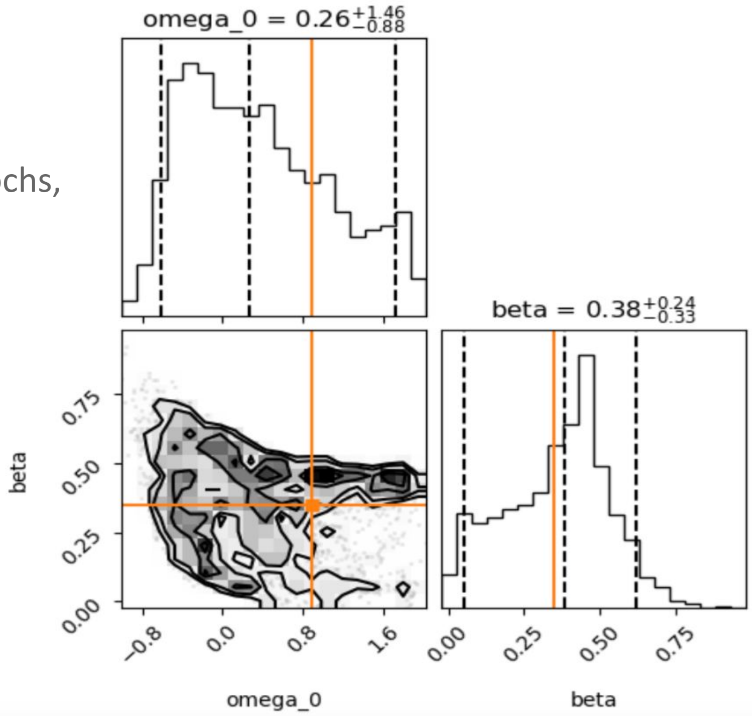
```
layers_f.initial_layer.weight
2500
layers_f.initial_layer.bias
10
layers_f.blocks.0.batch_norm_layers.0.weight
10
layers_f.blocks.0.batch_norm_layers.0.bias
10
layers_f.blocks.0.batch_norm_layers.1.weight
10
layers_f.blocks.0.batch_norm_layers.1.bias
10
layers_f.blocks.0.conv_layers.0.weight
500
layers_f.blocks.0.conv_layers.0.bias
10
layers_f.blocks.0.conv_layers.1.weight
500
layers_f.blocks.0.conv_layers.1.bias
10
layers_f.final_layer.weight
10
layers_f.final_layer.bias
1
contraction_layer.weight
600
contraction_layer.bias
3
expander_layer.weight
60
expander_layer.bias
20
layers_h.0.weight
400
layers_h.0.bias
20
final_layer.weight
240
final_layer.bias
12
4936
```

Flow with Similarity Embedding



Same parameters, hyperparameters, epochs, training strategies

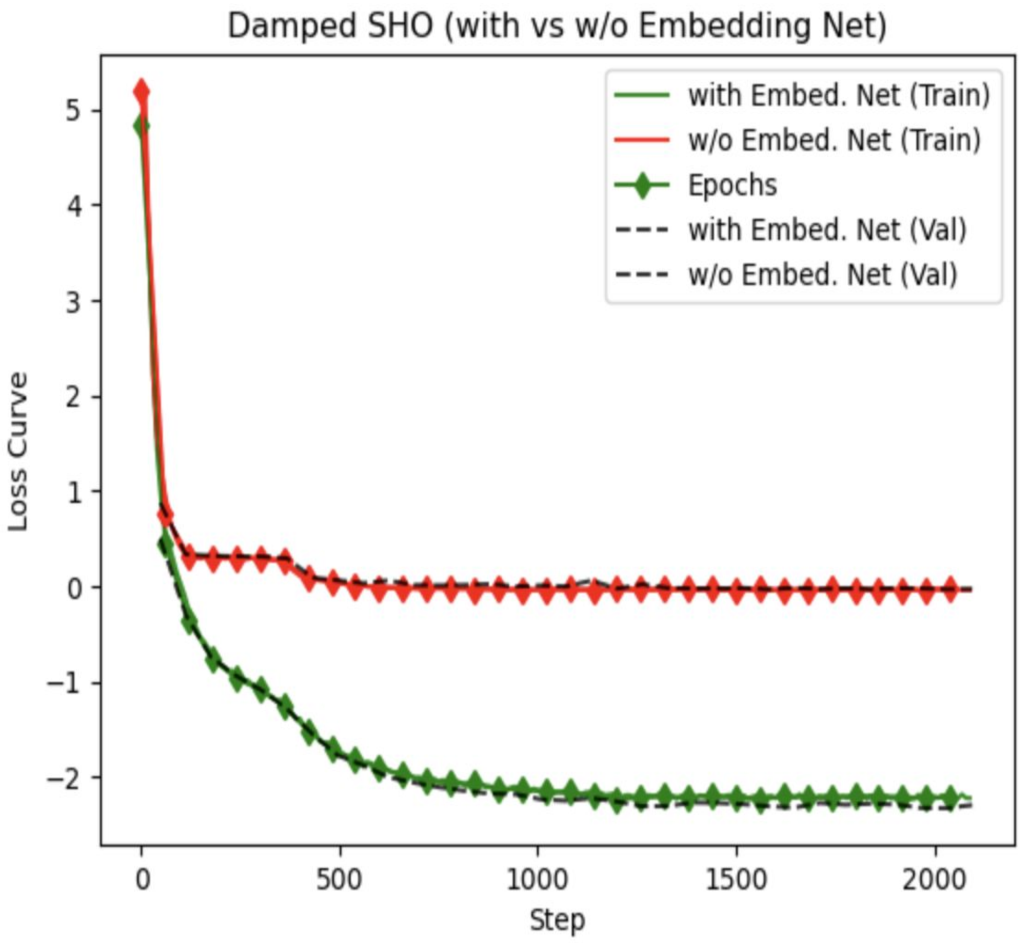
Flow without Similarity Embedding



Efficiency with Similarity Embedding

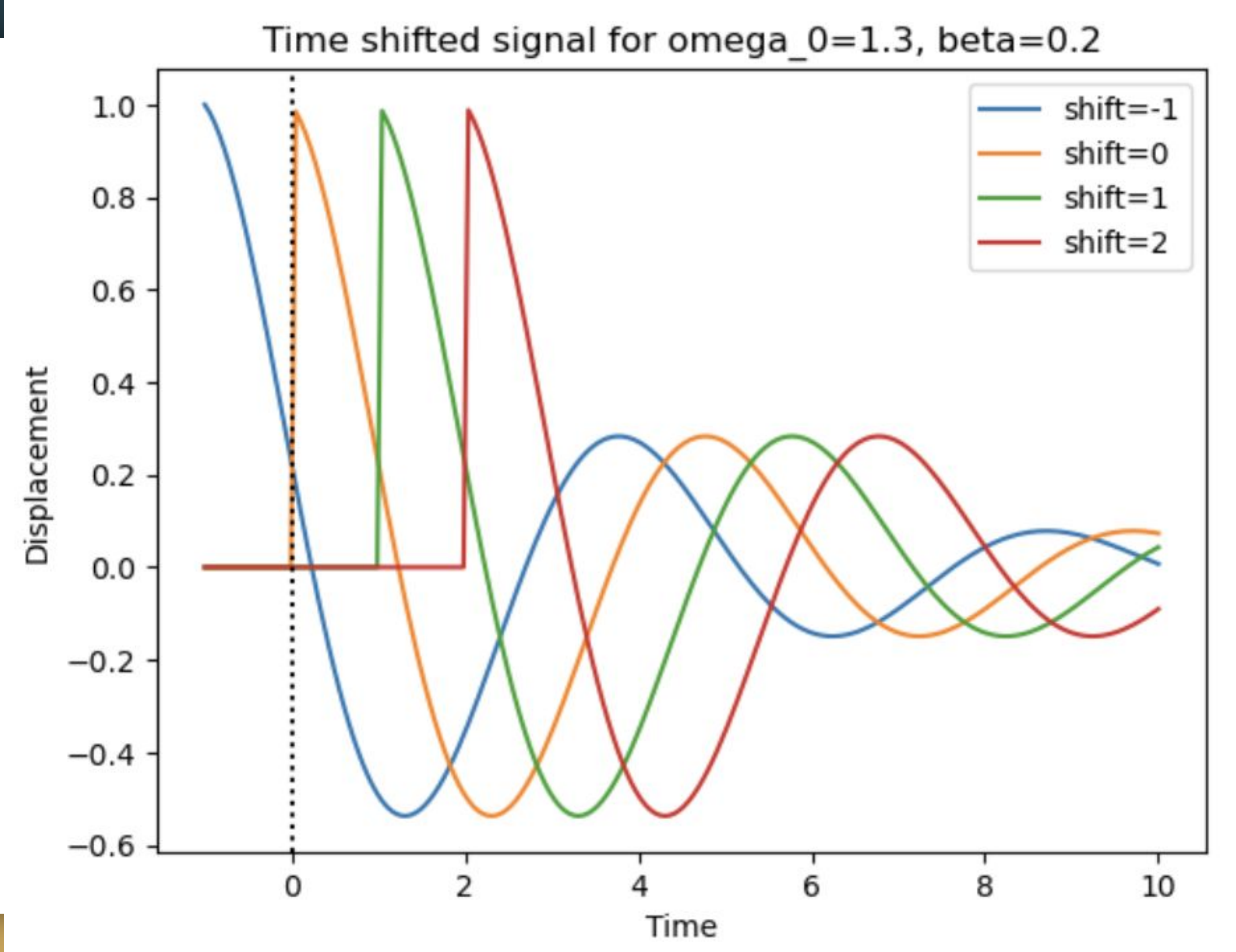
- Losses dive deeper
- More accurate and efficient results that are computationally less demanding
- Get the same results with a smaller number of parameters compared to results without using the similarity embedding

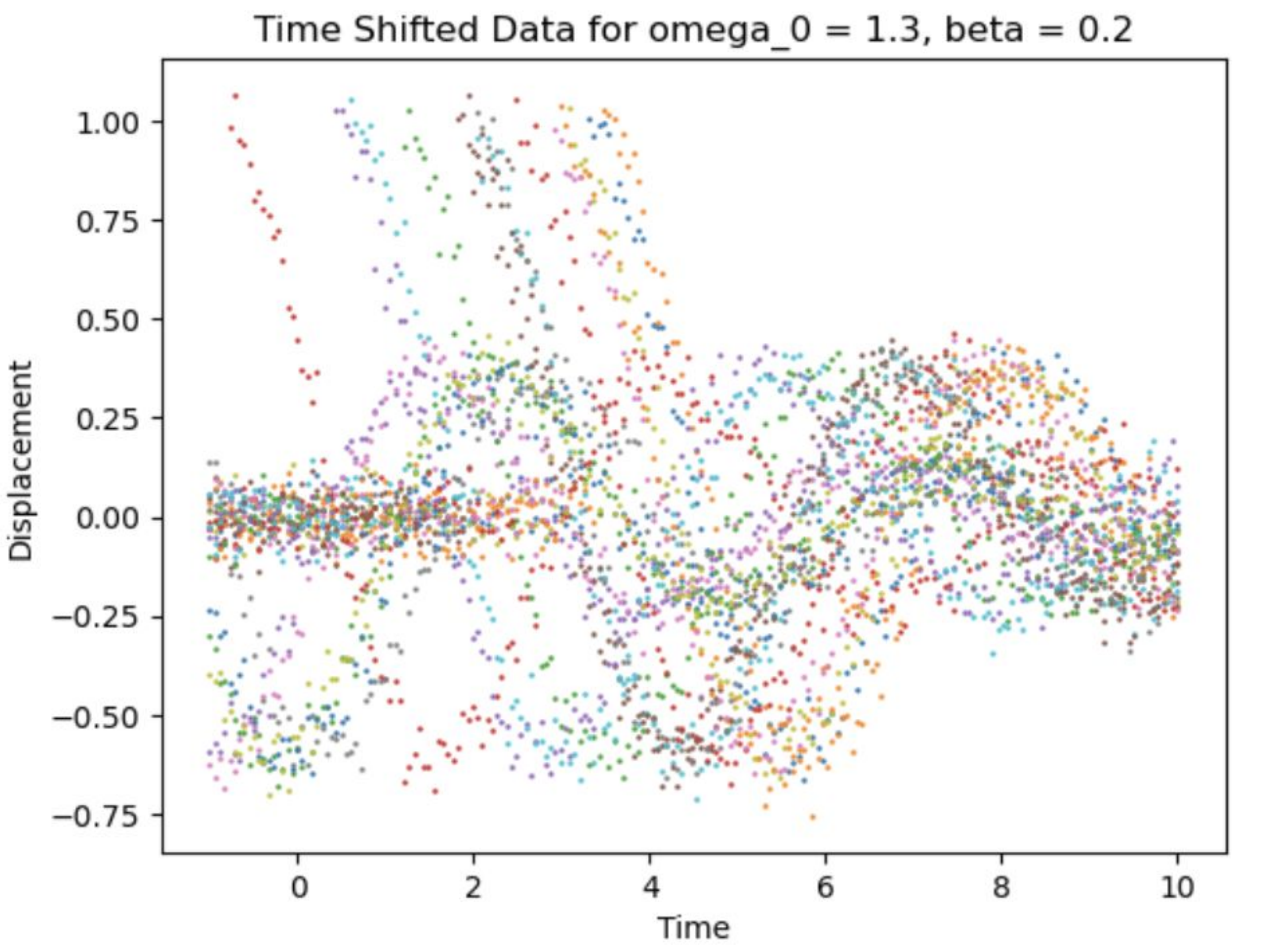
Same number of parameters with / without Embedding Net



Bibliography

- [1] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning, 2022.
- [2] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference, 2021.
- [3] https://en.wikipedia.org/wiki/Neural_network#/media/File:Neural_network_example.svg

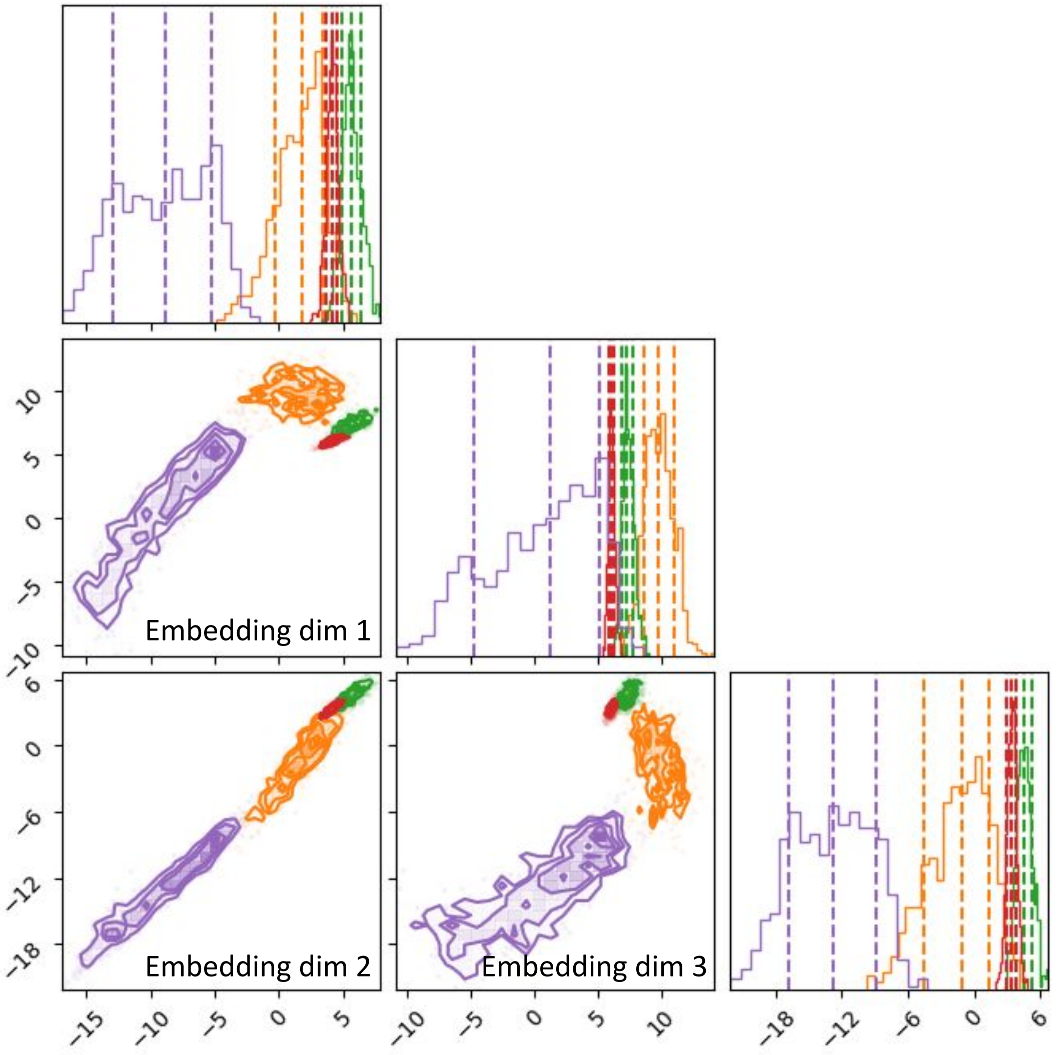

Time shifted signal for omega_0=1.3, beta=0.2
shift=-1
shift=0
shift=1
shift=2
Displacement
Time
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
0
2
4
6
8
10


Time Shifted Data for omega_0 = 1.3, beta = 0.2
Displacement
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
0
2
4
6
8
10
Time

Neural Network: Similarity Embedding

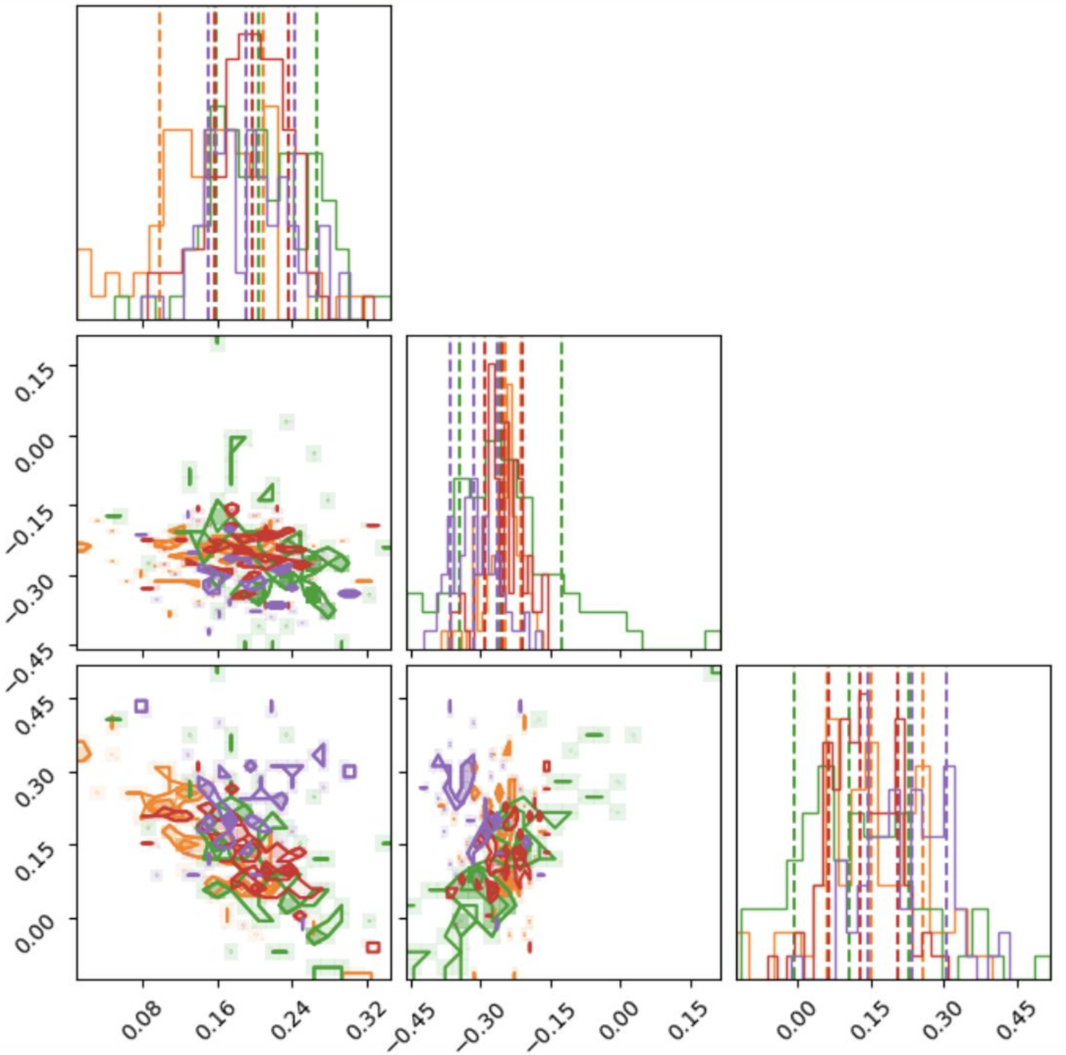
- ❖ Maps time shifted data down to cluster of points in 3D space
    - ➢ Each cluster contains points with the same Damped SHO (same omega and beta value) with the different time shifts
    - ➢ Useful to eliminate time variation]

X, Y: sets of 2D projections

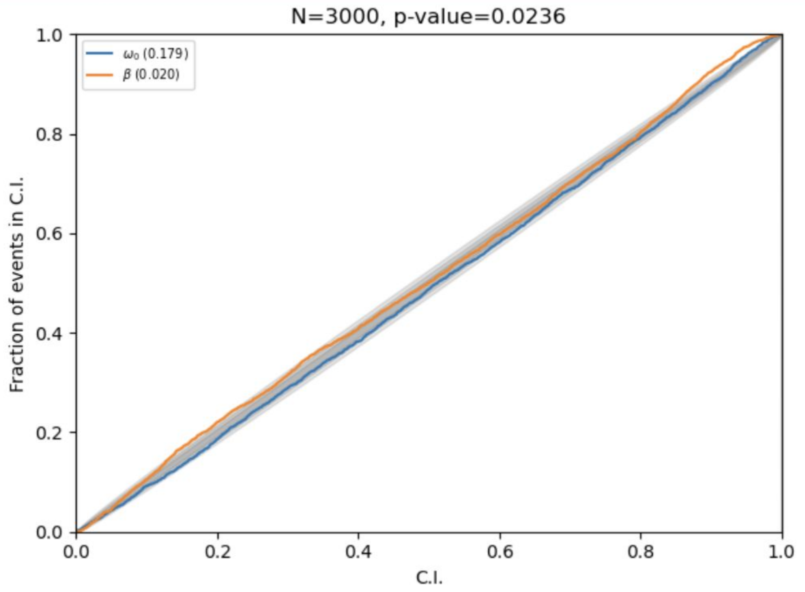


Without Neural Network: Similarity Embedding

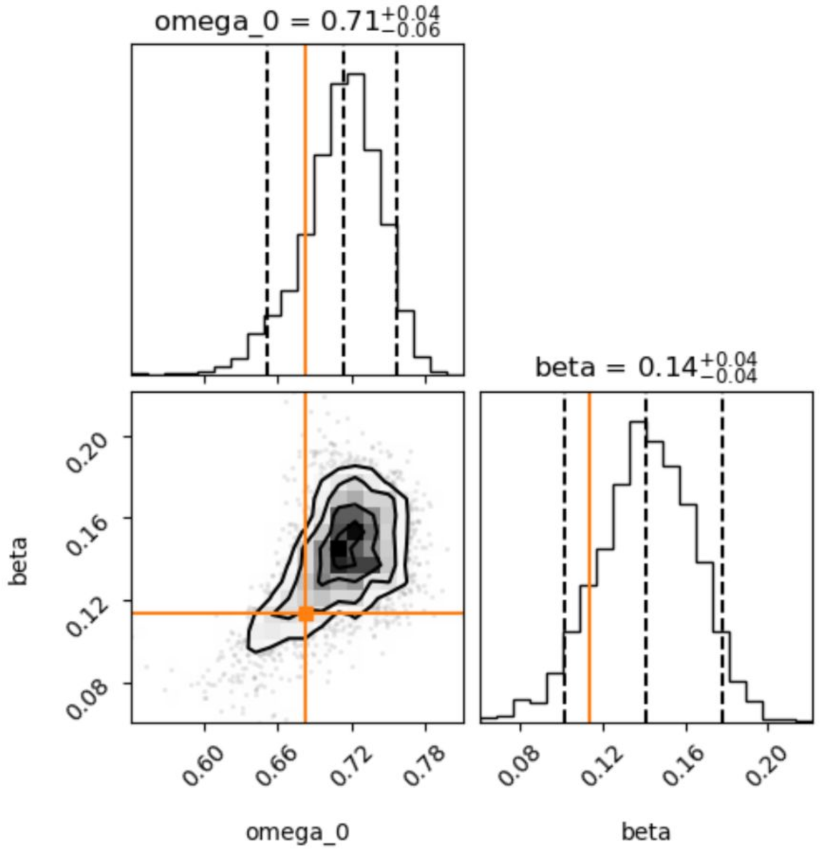
- ❖ No clear separation of different data points
- ❖ Time shifts is still an extra parameter present



Flow with Similarity Embedding



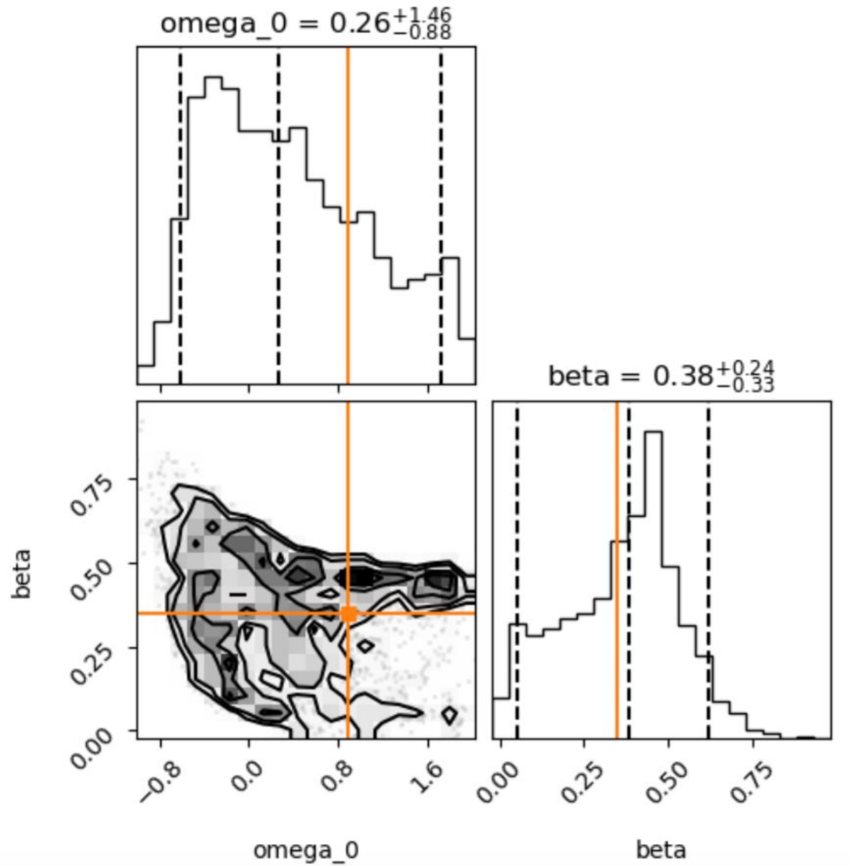
PP Plot



Posterior Widths

Flow without Similarity Embedding

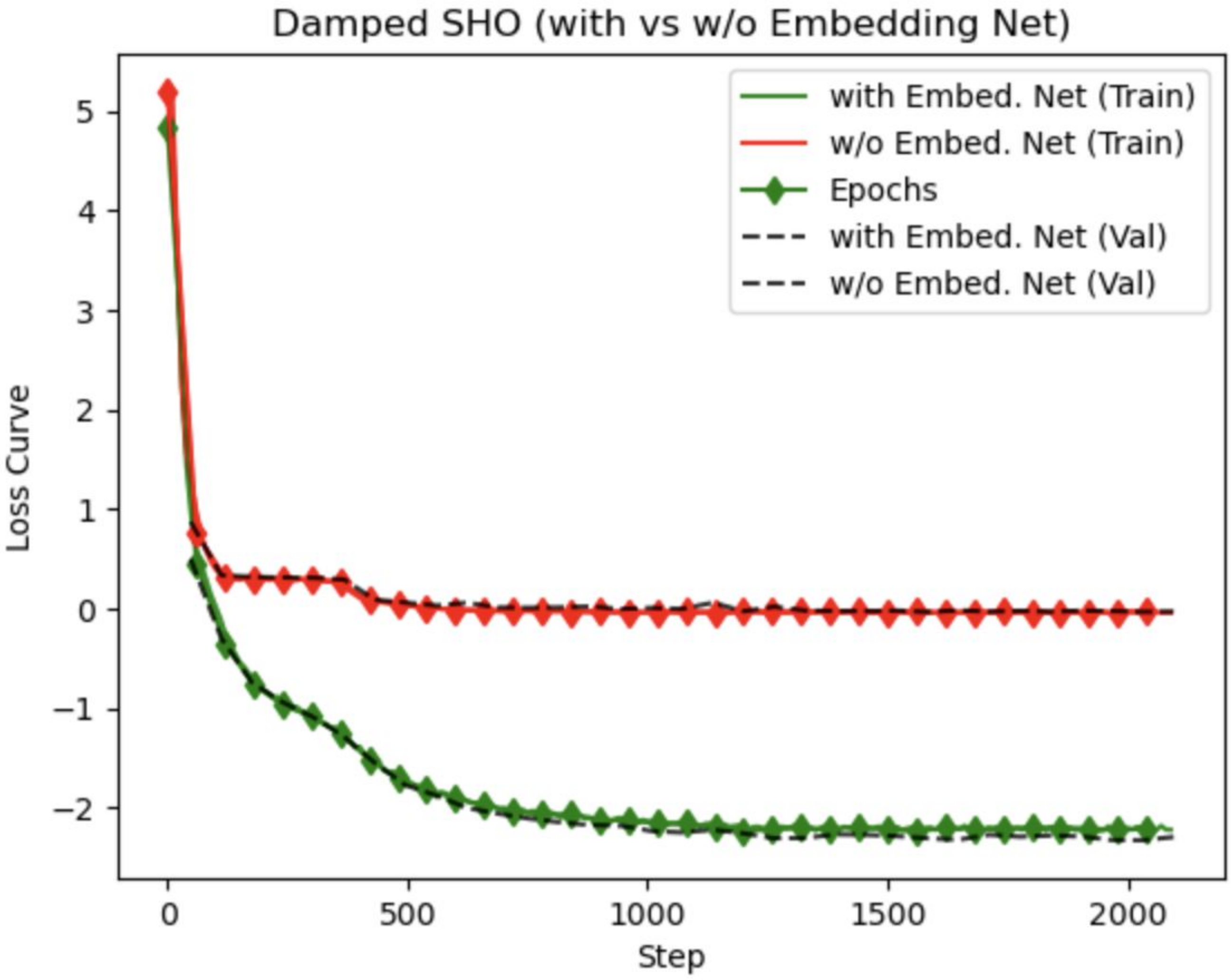
- ❖ For the same
  - ➢ number of parameters, hyperparameters, epochs, training strategies



Posterior Widths

Efficiency with Similarity Embedding

❖ Same number of parameters with / without Embedding Net



Wide posterior widths of Flow without Similarity Embedding