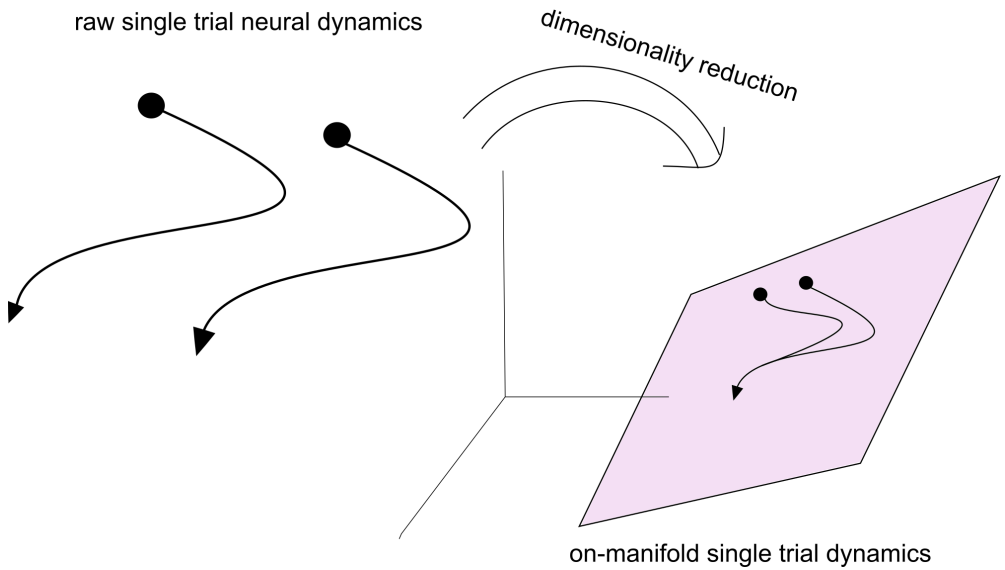
Deep Learning of Contraction Metrics from Neural Data

Leo Kozachkov

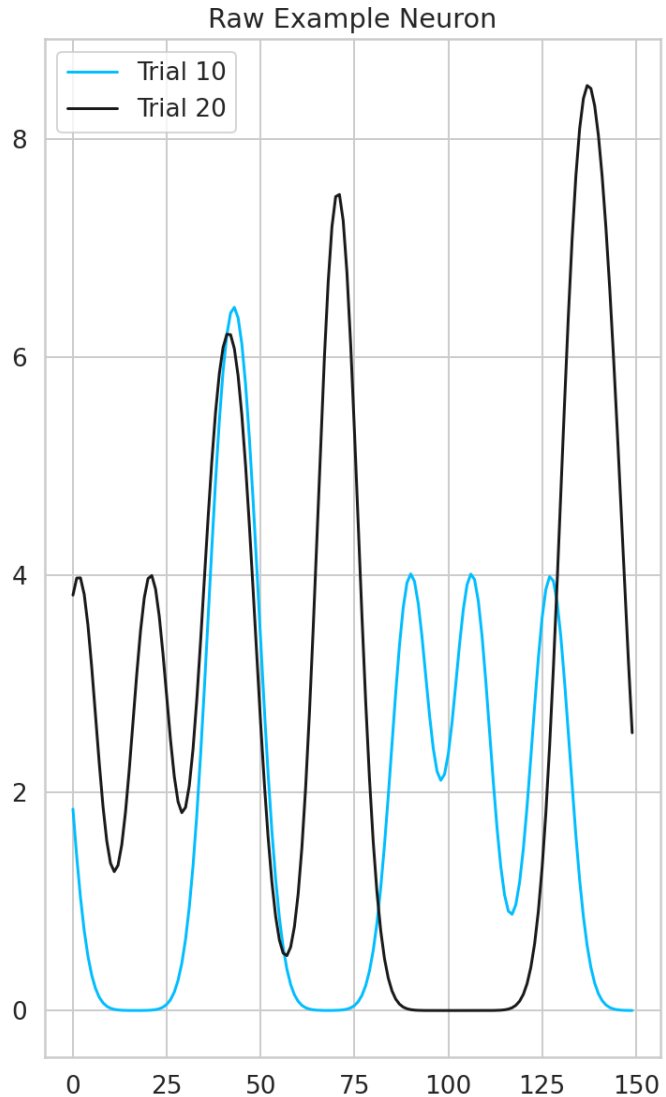
MIT

2020

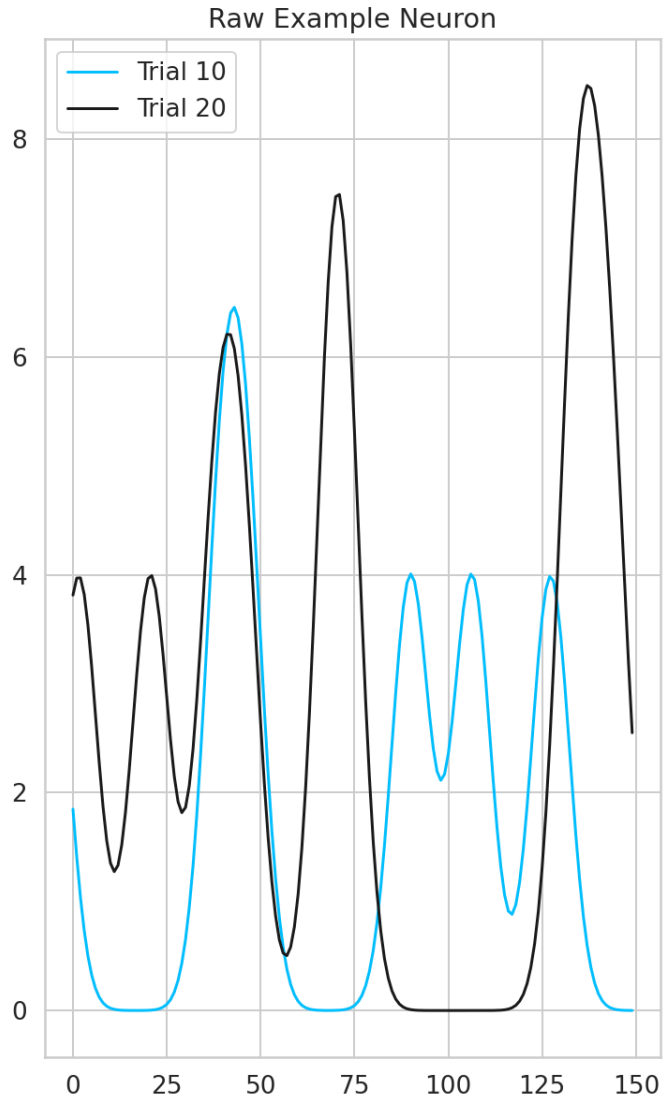
Interestingly, if we project the neural data into a lower-dimensional space (given by PCA of trial-average), we can make the situation better.

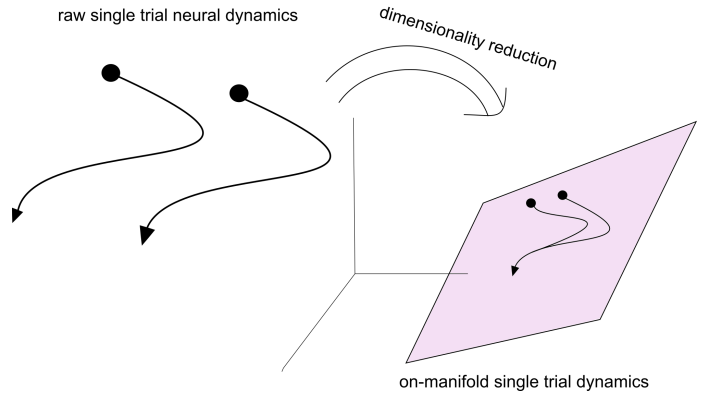


Raw neural data is not obviously stable (leftmost panel). But it is also not obviously *unstable*.



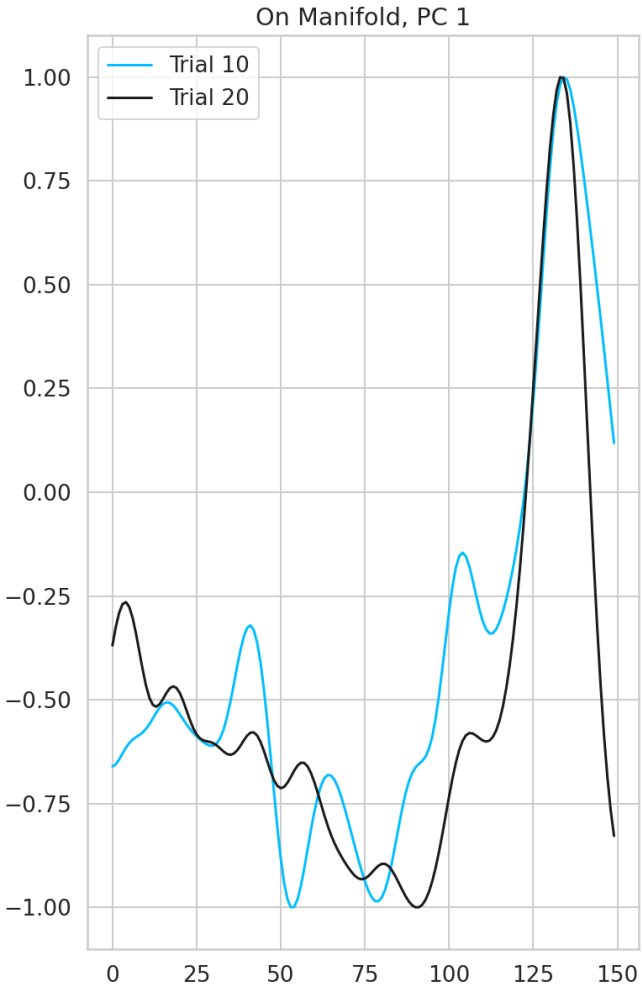
Raw neural data is not obviously stable (leftmost panel). But it is also not obviously *unstable*.



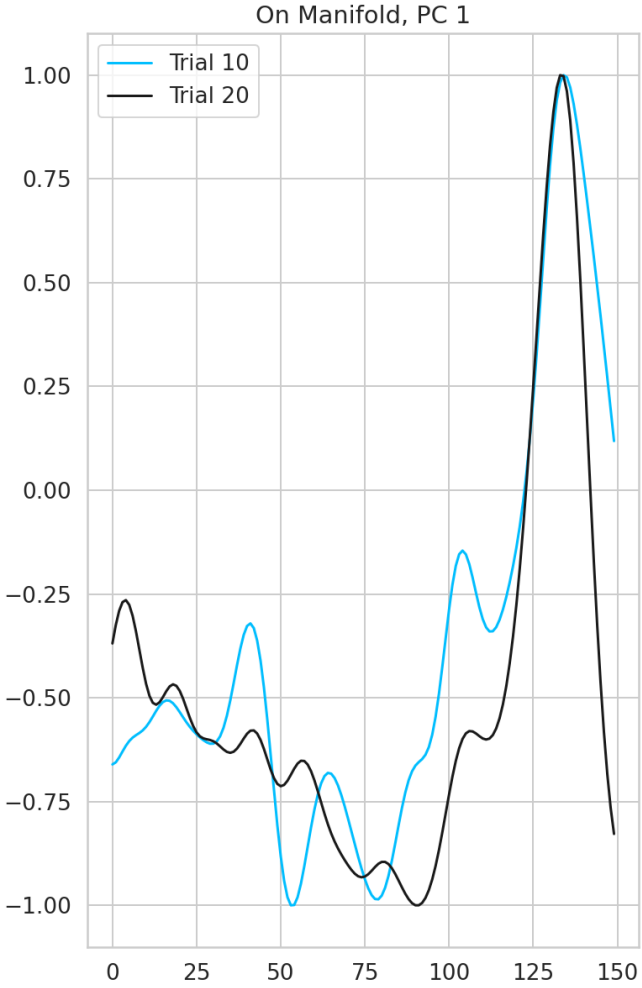


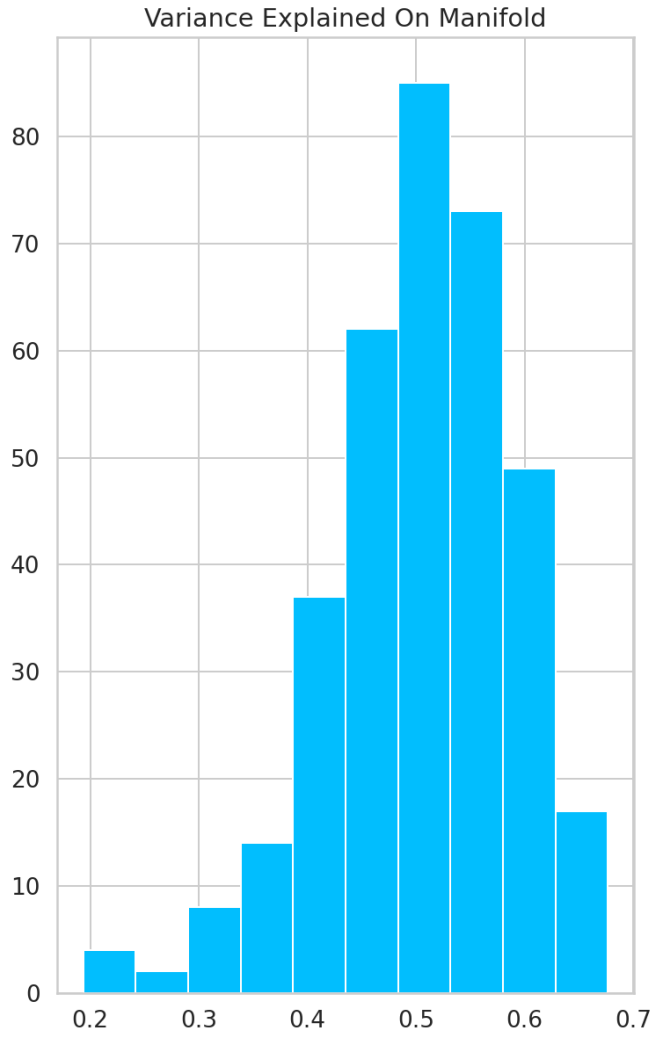
Are there stable manifolds inside the raw neural state space?

Projecting single trials onto manifold spanned by the PCA axes helps. We find these axis based on the average neural response.

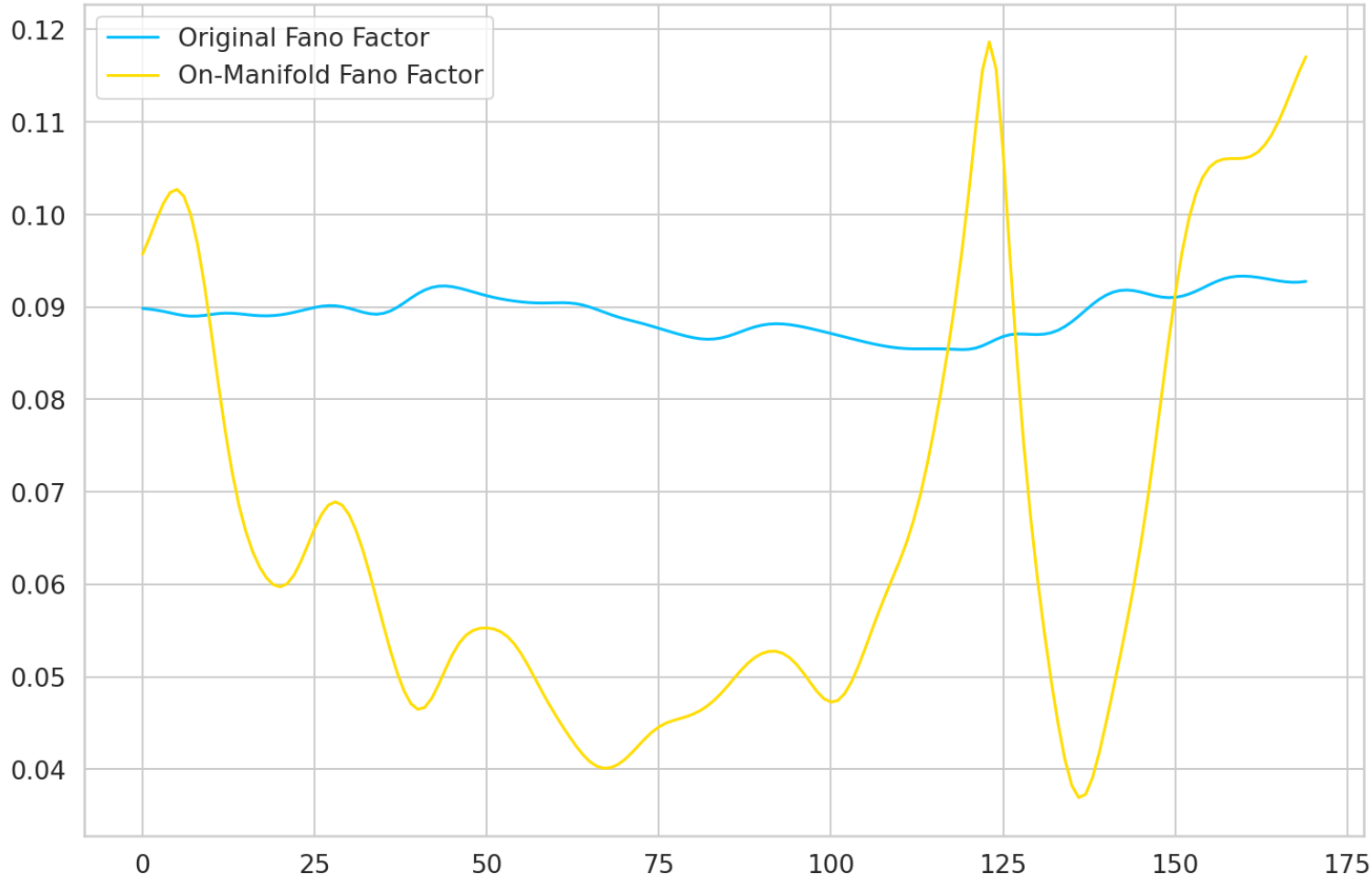


Projecting single trials onto manifold spanned by the PCA axes helps. We find these axis based on the average neural response.





Lower fano factor on manifold.



Contraction Analysis

Look at 'on manifold' contraction. Differential dynamics.

$$\delta\dot{\mathbf{x}} = \mathbf{J}\delta\mathbf{x} \tag{1}$$

Contraction Analysis

Look at 'on manifold' contraction. Differential dynamics.

$$\delta\dot{\mathbf{x}} = \mathbf{J}\delta\mathbf{x} \tag{1}$$

We don't really have access to $\mathbf{J}$ or $\delta\mathbf{x}$, but we do have access to $\mathbf{x}_1(t), \mathbf{x}_2(t)...\mathbf{x}_K(t)$ where $\mathbf{x}_i(t)$ is the N dimensional vector of neural response at time t in the trial.

Contraction Analysis

Look at 'on manifold' contraction. Differential dynamics.

$$\delta\dot{\mathbf{x}} = \mathbf{J}\delta\mathbf{x} \tag{1}$$

We don't really have access to $\mathbf{J}$ or $\delta\mathbf{x}$, but we do have access to $\mathbf{x}_1(t), \mathbf{x}_2(t)...\mathbf{x}_K(t)$ where $\mathbf{x}_i(t)$ is the N dimensional vector of neural response at time t in the trial.
We can approximate the differential dynamics of this system by taking pairwise differences of neural responses:

$$\delta\mathbf{x}_{ij} \approx \mathbf{x}_i(t) - \mathbf{x}_j(t) \tag{2}$$

Contraction Analysis

Look at 'on manifold' contraction. Differential dynamics.

$$\delta\dot{\mathbf{x}} = \mathbf{J}\delta\mathbf{x} \tag{1}$$

We don't really have access to $\mathbf{J}$ or $\delta\mathbf{x}$, but we do have access to $\mathbf{x}_1(t), \mathbf{x}_2(t)...\mathbf{x}_K(t)$ where $\mathbf{x}_i(t)$ is the N dimensional vector of neural response at time t in the trial.
We can approximate the differential dynamics of this system by taking pairwise differences of neural responses:

$$\delta\mathbf{x}_{ij} \approx \mathbf{x}_i(t) - \mathbf{x}_j(t) \tag{2}$$

A system is said to be contracting if there exists some (potentially state-dependent and time-varying) symmetric matrix $\mathbf{M}$ such that:

$$\mathbf{M} \succ 0 \tag{3}$$

$$\frac{d}{dt}\delta\mathbf{x}^T\mathbf{M}\delta\mathbf{x} \leq -\lambda\delta\mathbf{x}^T\mathbf{M}\delta\mathbf{x} \tag{4}$$

Or, equivalently:

$$\frac{d}{dt}\delta\mathbf{x}^T\mathbf{M}\delta\mathbf{x} + \lambda\delta\mathbf{x}^T\mathbf{M}\delta\mathbf{x} \leq 0 \tag{5}$$

We can use the above equation to begin constructing a loss function to use in learning **M**. Namely, we can define the loss **L** along a single trajectory at time t as

$$\mathcal{L}_{dd,t} = \text{ReLU}(\frac{d}{dt}\delta\mathbf{x}^T\mathbf{M}\delta\mathbf{x} + \lambda\delta\mathbf{x}^T\mathbf{M}\delta\mathbf{x}) \tag{6}$$

We can use the above equation to begin constructing a loss function to use in learning **M**. Namely, we can define the loss **L** along a single trajectory at time t as

$$\mathcal{L}_{dd,t} = \text{ReLU}(\frac{d}{dt}\delta\mathbf{x}^T\mathbf{M}\delta\mathbf{x} + \lambda\delta\mathbf{x}^T\mathbf{M}\delta\mathbf{x}) \tag{6}$$

and the total differential dynamics (dd) loss for the trial as the average loss across time.

We can use the above equation to begin constructing a loss function to use in learning **M**. Namely, we can define the loss **L** along a single trajectory at time t as

$$\mathcal{L}_{dd,t} = \text{ReLU}(\frac{d}{dt}\delta\mathbf{x}^T\mathbf{M}\delta\mathbf{x} + \lambda\delta\mathbf{x}^T\mathbf{M}\delta\mathbf{x}) \tag{6}$$

and the total differential dynamics (dd) loss for the trial as the average loss across time.

$$\mathcal{L}_{dd} = \frac{1}{T}\sum_{t=0}^{T}\mathcal{L}_{dd,t} \tag{7}$$

We also include a regularizing 'logdet' term which ensures the positive-definiteness of **M** by tending to $+\infty$ as **M** becomes singular:

$$\mathcal{L}_M = -\log(\det(\mathbf{M})) \tag{8}$$

The total loss is the weighted sum of these two individual loss terms.

$$\mathcal{L} = \mathcal{L}_{dd} + \lambda_M \mathcal{L}_M \tag{9}$$

Learning M

Rather than learn an $N \times N$ symmetric, positive definite matrix directly, we construct $\mathbf{M}$ in terms of it's Cholesky decomposition:

$$\mathbf{L}_t = \text{LSTM}(\mathbf{x}_t) \tag{10}$$

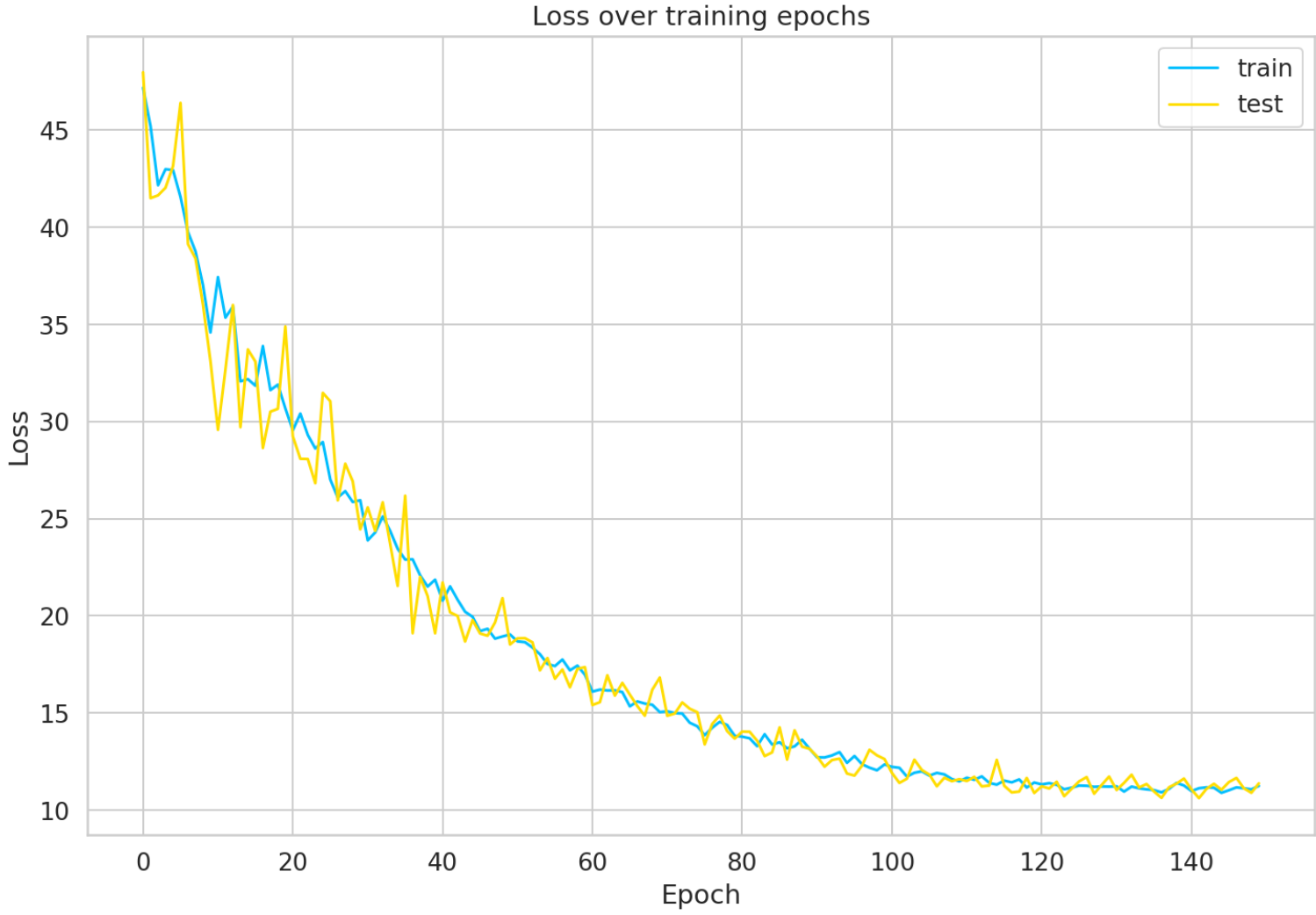
$$\mathbf{M}_t = \mathbf{L}_t^T \mathbf{L}_t \tag{11}$$

This reduces the dimensionality of the target from N^2 to $\frac{N(N+1)}{2}$, the number of non-zero elements of a triangular matrix.

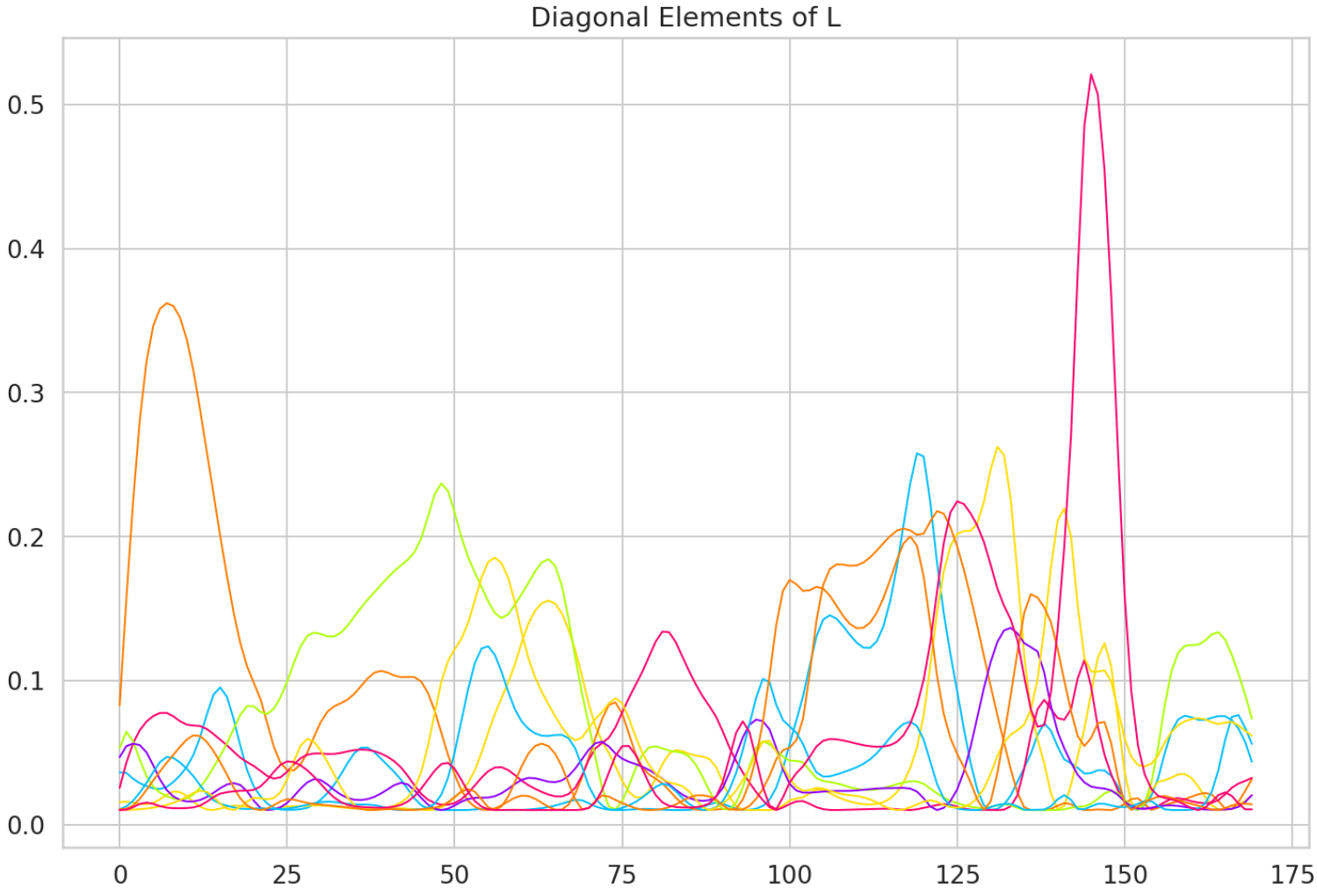
This representation also allows us to efficiently constrain $\mathbf{M}$ to be positive definite, because the logdet can be computed directly from the diagonal elements of the Cholesky decomposition:

$$\begin{aligned} \log(\det(\mathbf{M})) = \\ \log(\det(\mathbf{L}^T\mathbf{L}) = \\ \log(\det(\mathbf{L})^2) = \\ 2\log(\det(\mathbf{L})) = \\ 2\sum_i \log(L_{ii}) \end{aligned} \tag{12}$$

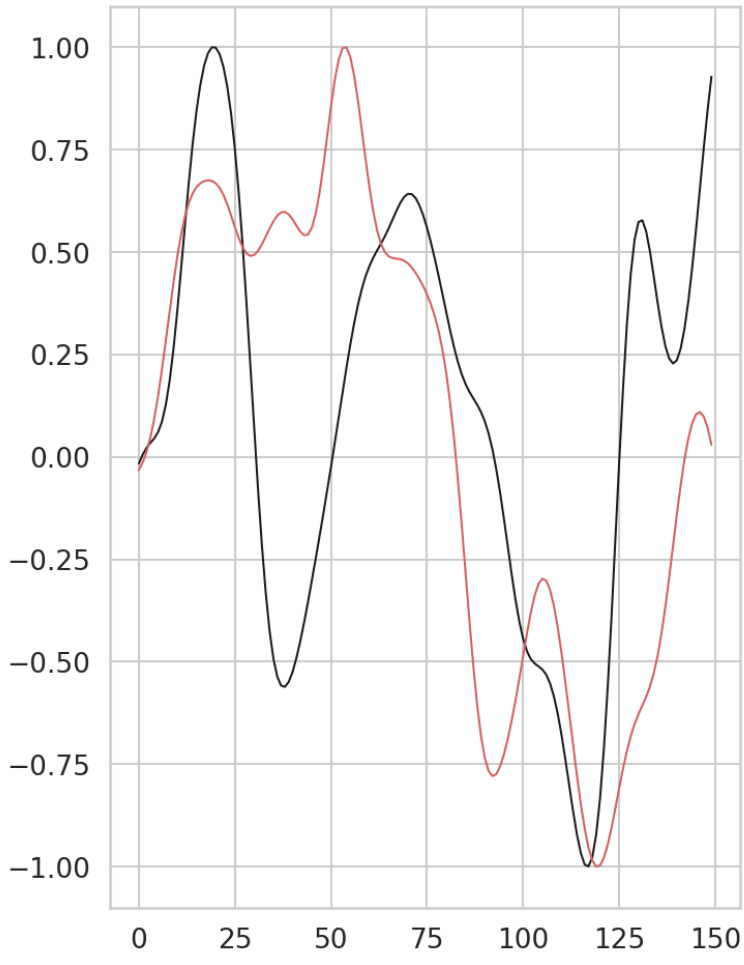
Does it work?

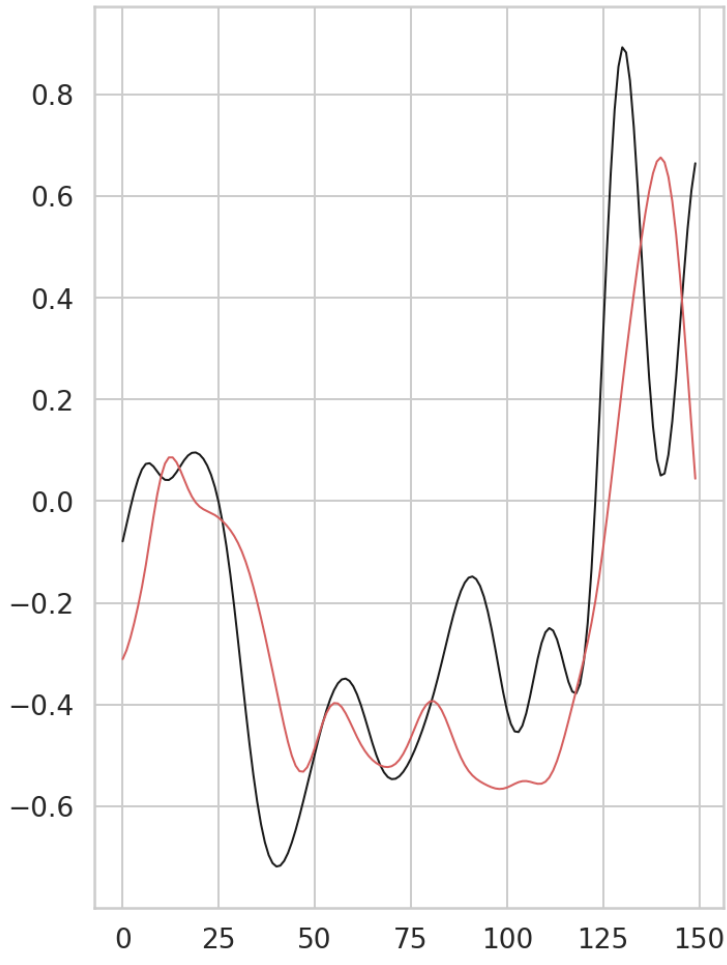

Loss over training epochs
train
test
Loss
Epoch
45
40
35
30
25
20
15
10
0
20
40
60
80
100
120
140

State-dependent, time-varying metrics. Using full 'power' of contraction analysis.

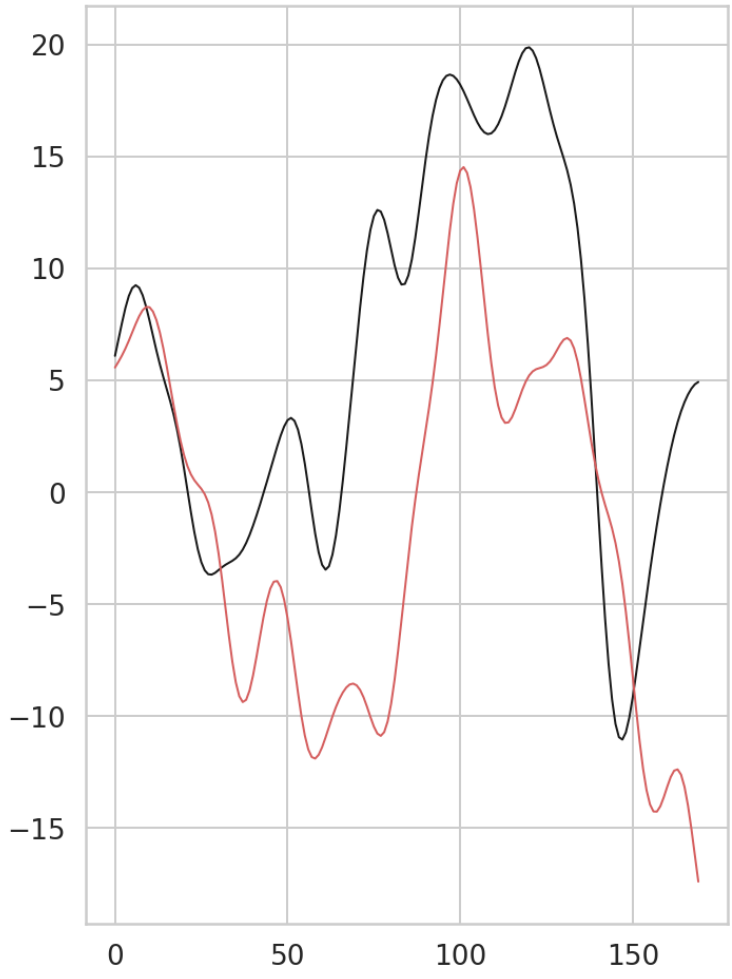


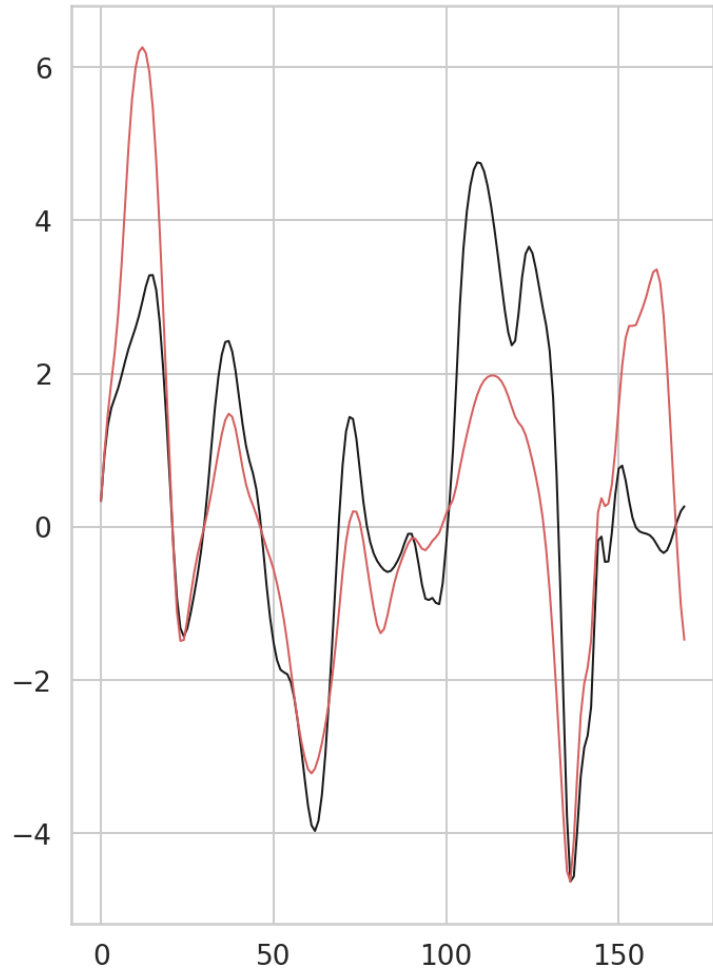
So far, slight visible improvements.



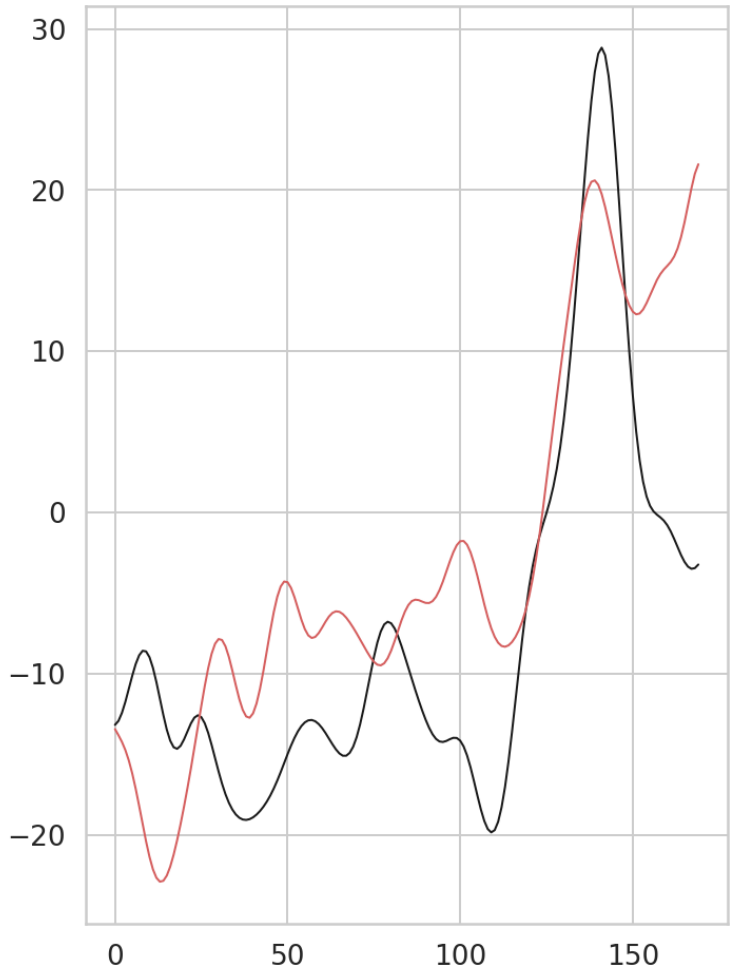


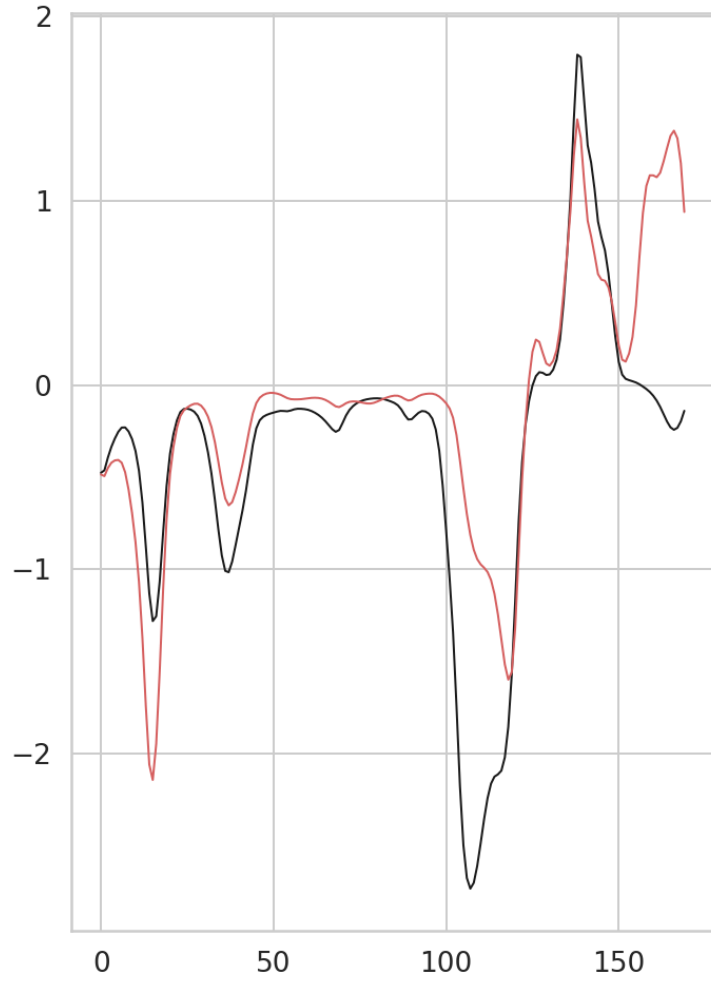
So far, slight visible improvements.



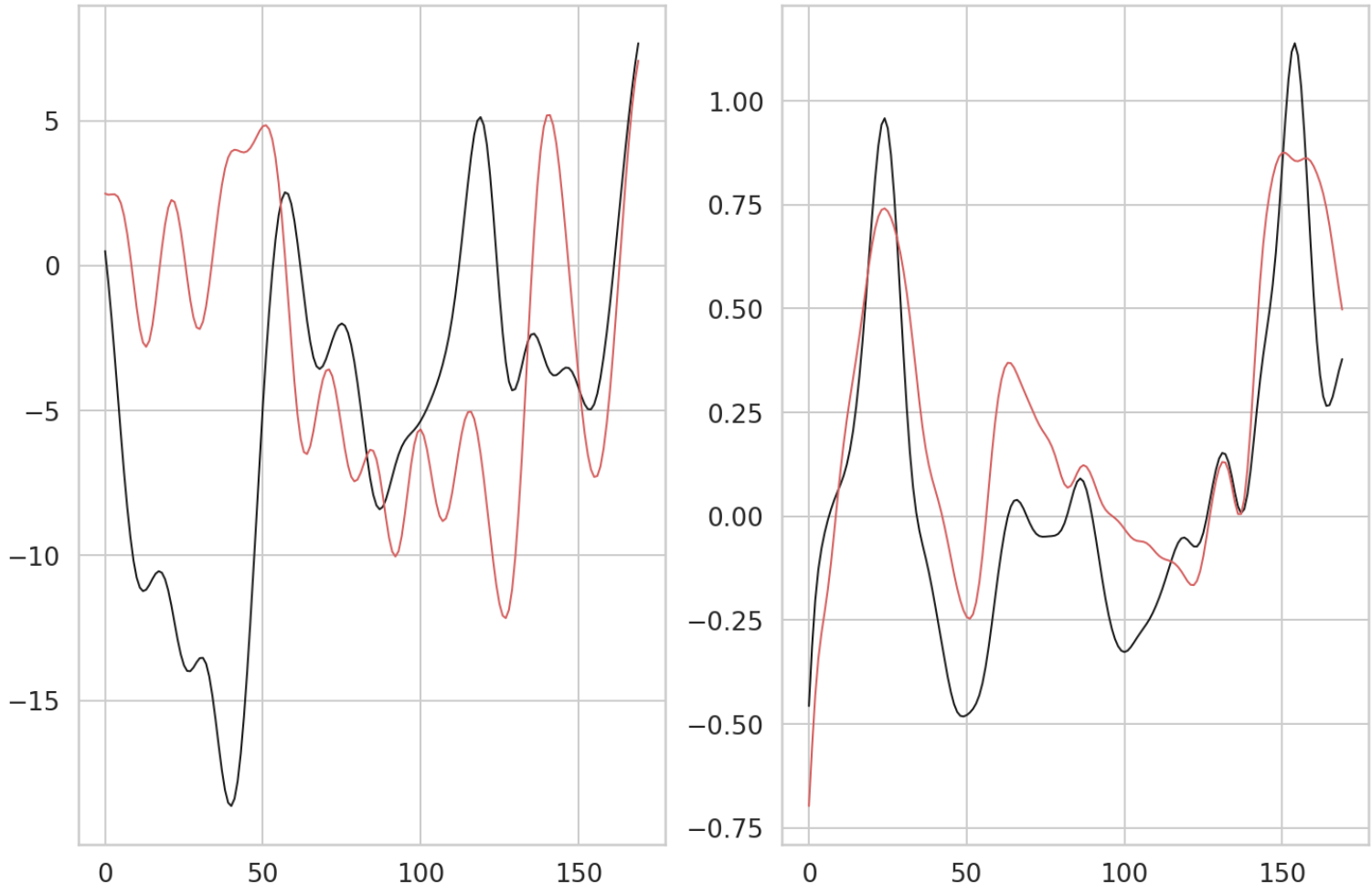


So far, slight visible improvements.

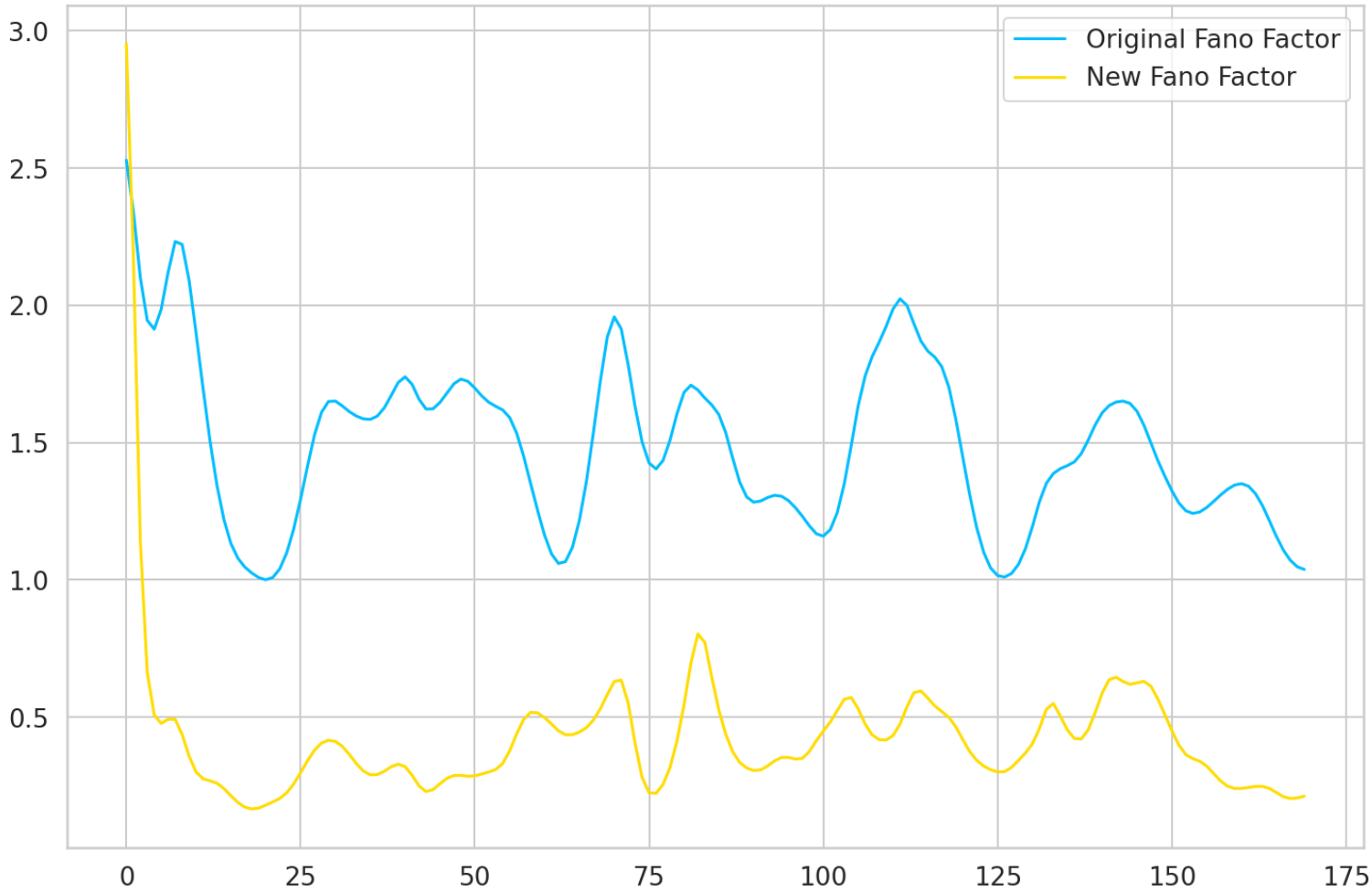




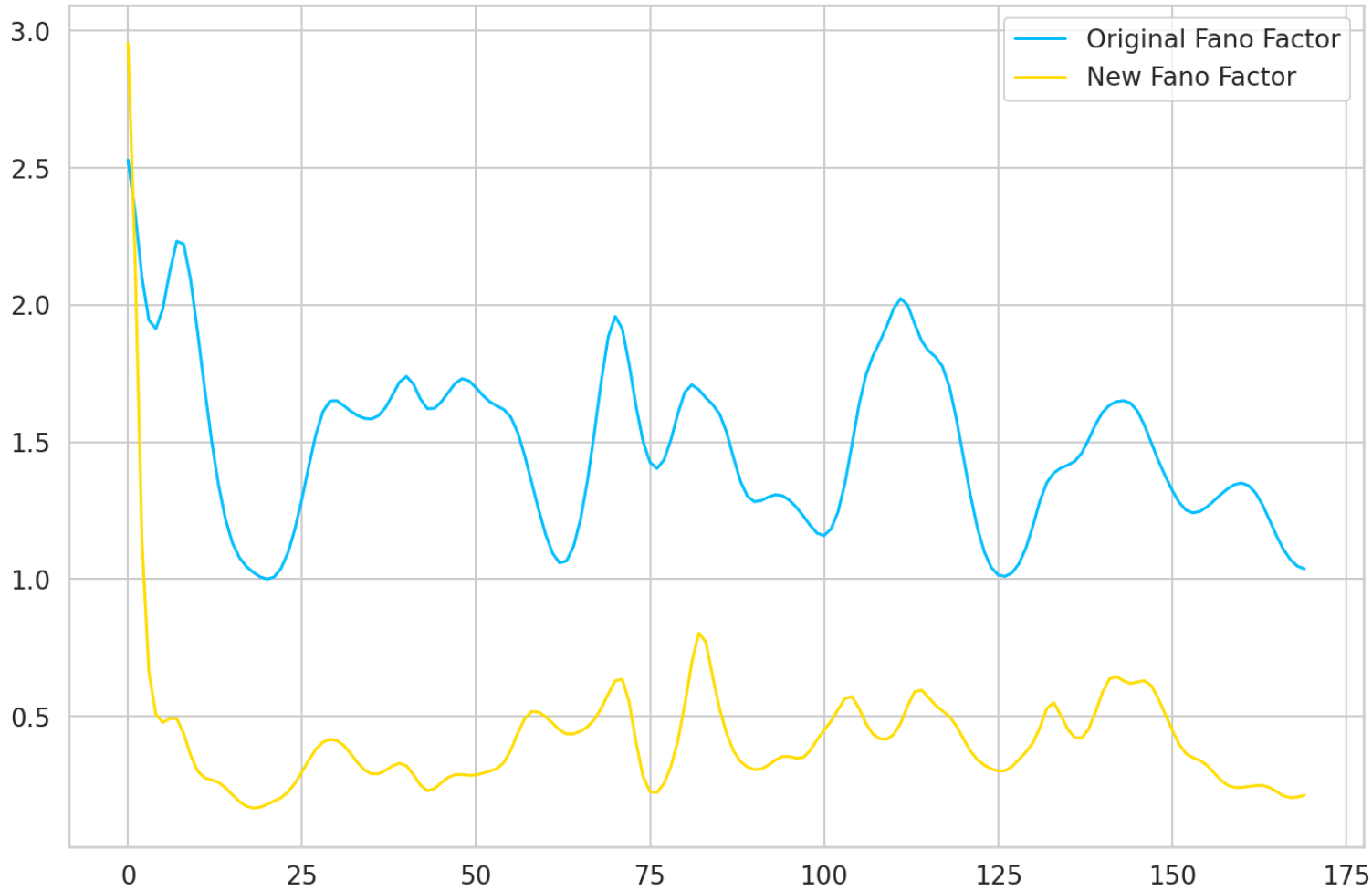
So far, slight visible improvements.



Fano factor is improved along trajectories in new metric (ratio of noise to mean signal amplitude).



Fano factor is improved along trajectories in new metric (ratio of noise to mean signal amplitude).



Noise can never be fully eliminated in new metric, because the system is inherently noisy.

- On-manifold neural dynamics are more stable than raw neural data.
- Deep learning can be used to learn contraction metrics directly from neural data. This reveals non-obvious, trial to trial stability in the neural signal.