Beneficial roles for chaotic variability in learning systems

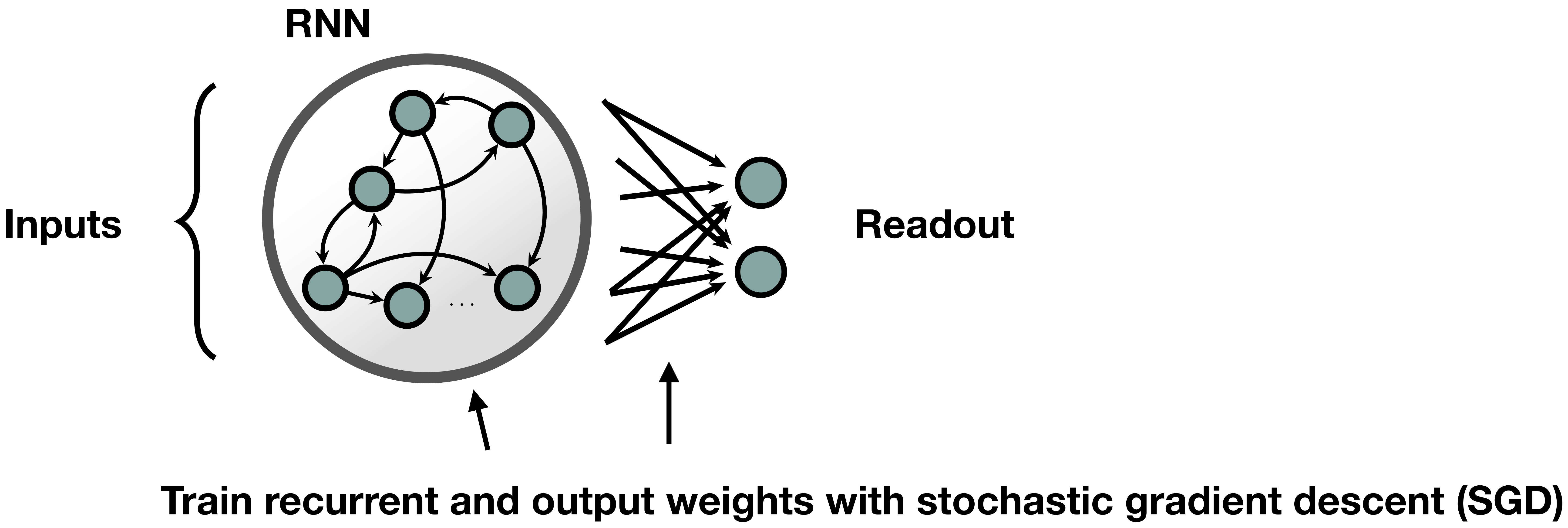
Matthew Farrell[1-2], Stefano Recanatesi[1], Timothy Moore[1], Guillaume Lajoie[3-4], and Eric Shea-Brown[1-2]

[1] *Computational Neuroscience Center, University of Washington*
[2] *Dept. of Applied Mathematics, University of Washington*
[3] *Mila—Québec AI Institute*
[4] *Dept. of Mathematics and Statistics, Université de Montréal*

Abstract. Neural responses are highly variable, even under identical task conditions. Significant efforts are being directed toward explaining how the brain copes with and may even leverage such variability to help learn the task and environment. Here we explore the issue in a recurrent neural network model that is trained to classify inputs. We find two potential beneficial roles for chaotic variability in these systems: (1) chaos can accelerate the flexible relearning of a task after it is modified; and (2) chaos can lift the network representation of data into a higher-dimensional space, which allows the network to classify inputs embedded in low-dimensional spaces.

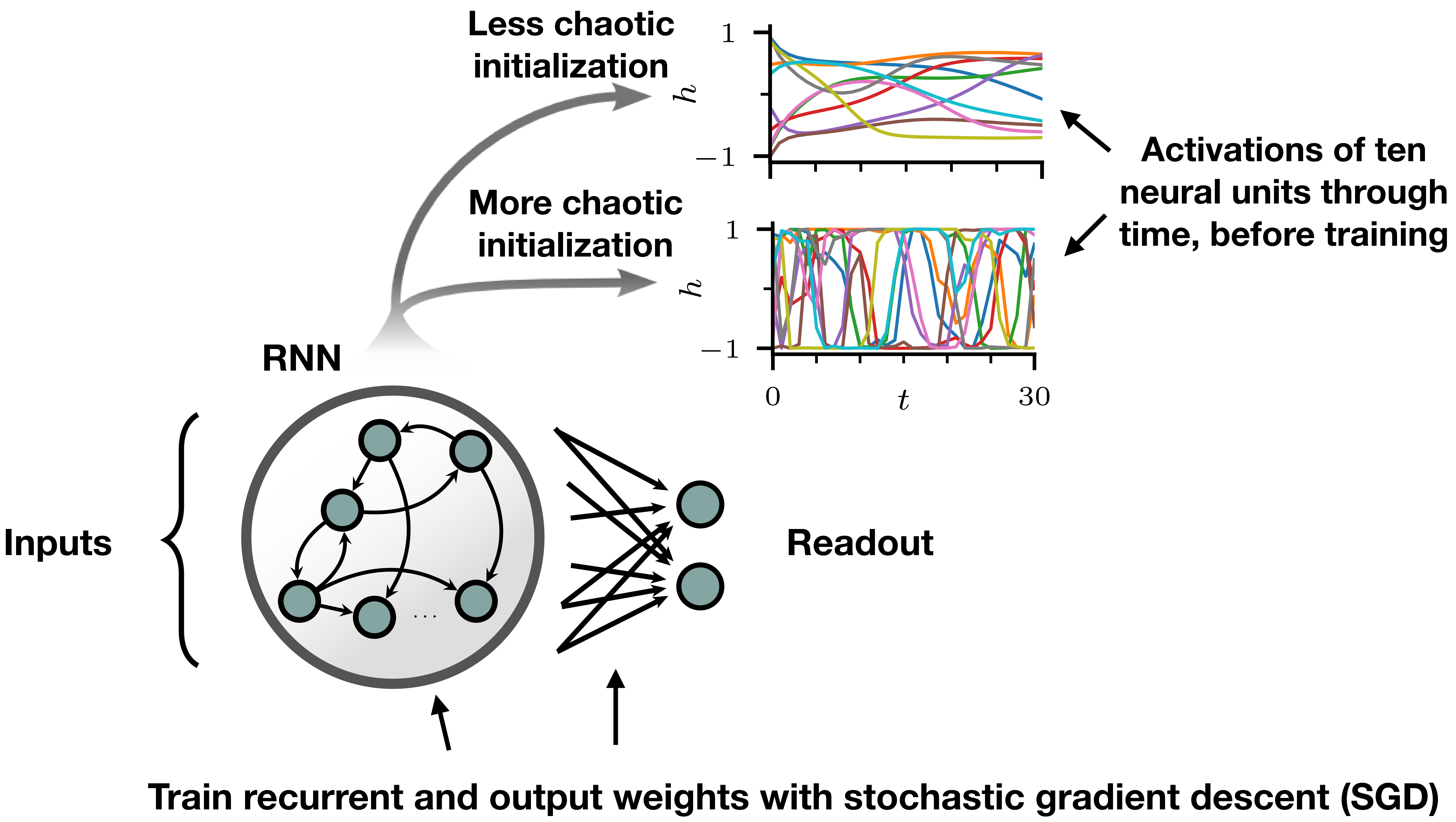
Goal: investigate how the initial dynamics of a recurrent neural network (RNN) influence the solutions learned by the network when trained to do a task.

We characterize these dynamics by how chaotic the trajectories are.



Goal: investigate how the initial dynamics of a recurrent neural network (RNN) influence the solutions learned by the network when trained to do a task.

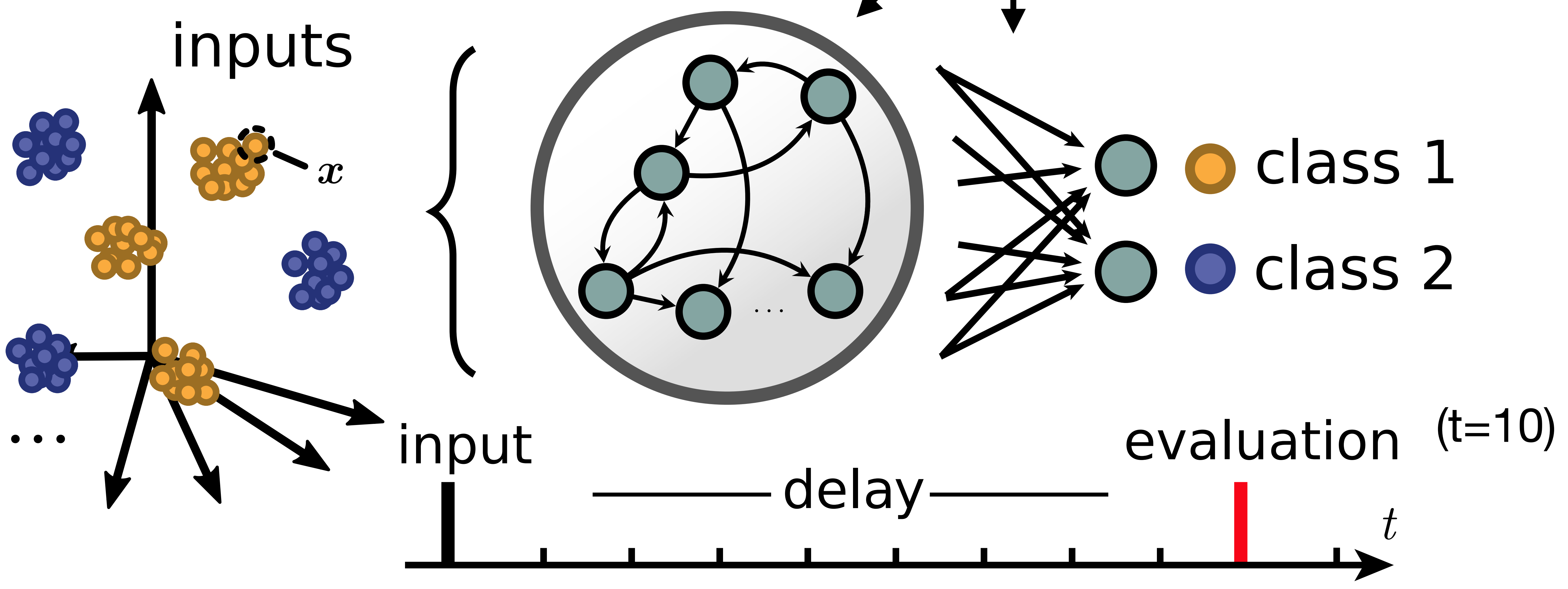
We characterize these dynamics by how chaotic the trajectories are.



Let’s choose a task to train the network on

Delayed classification task

Train with SGD, using Categorical Cross Entropy loss



The error between the network output and the class label is used to update the weights.

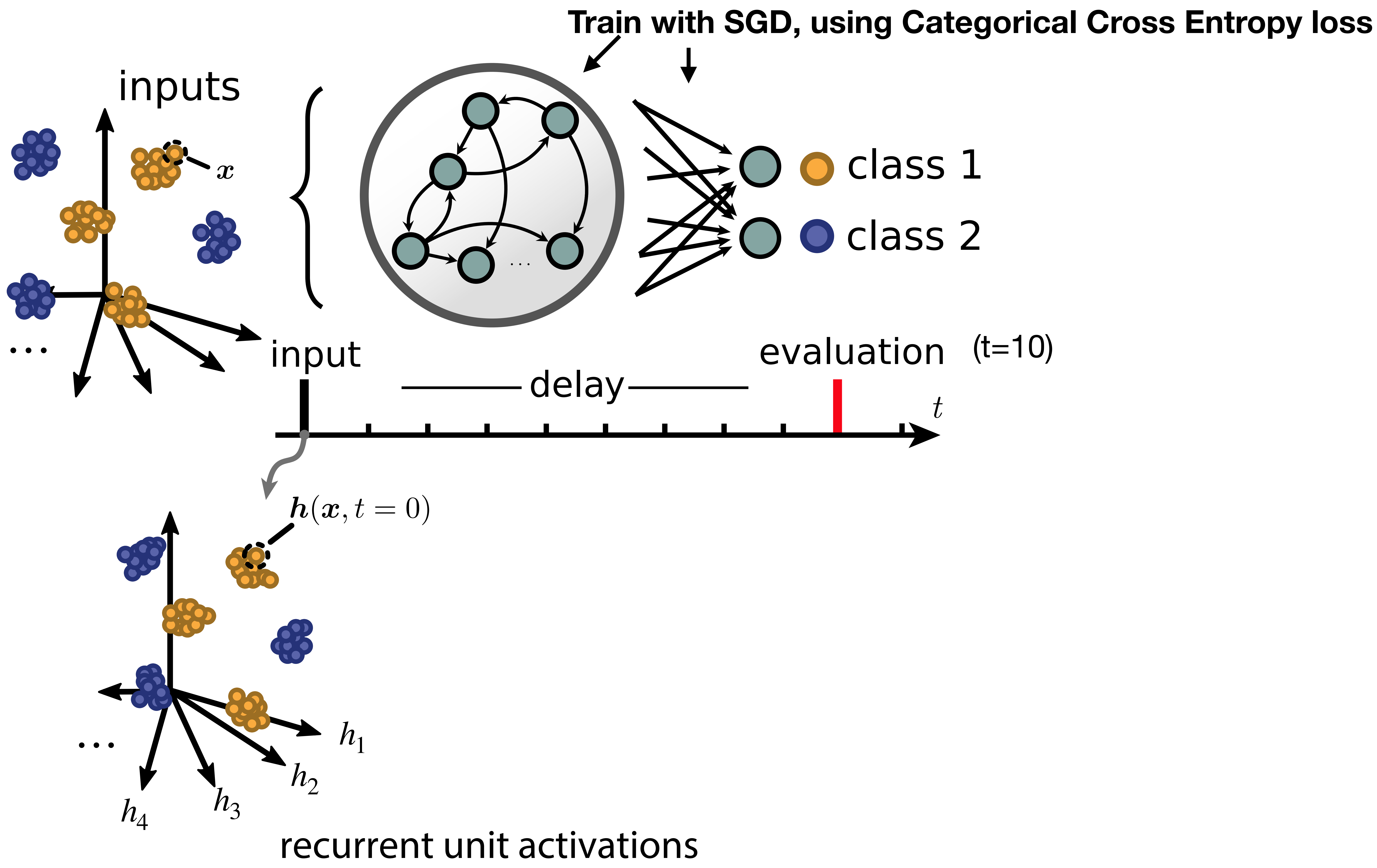
inputs are gaussian balls randomly distributed in space

The input is only shown to the network at time $t = 0$. After this the network evolves autonomously

The network state is read out and used to classify the input at time $t = 10$

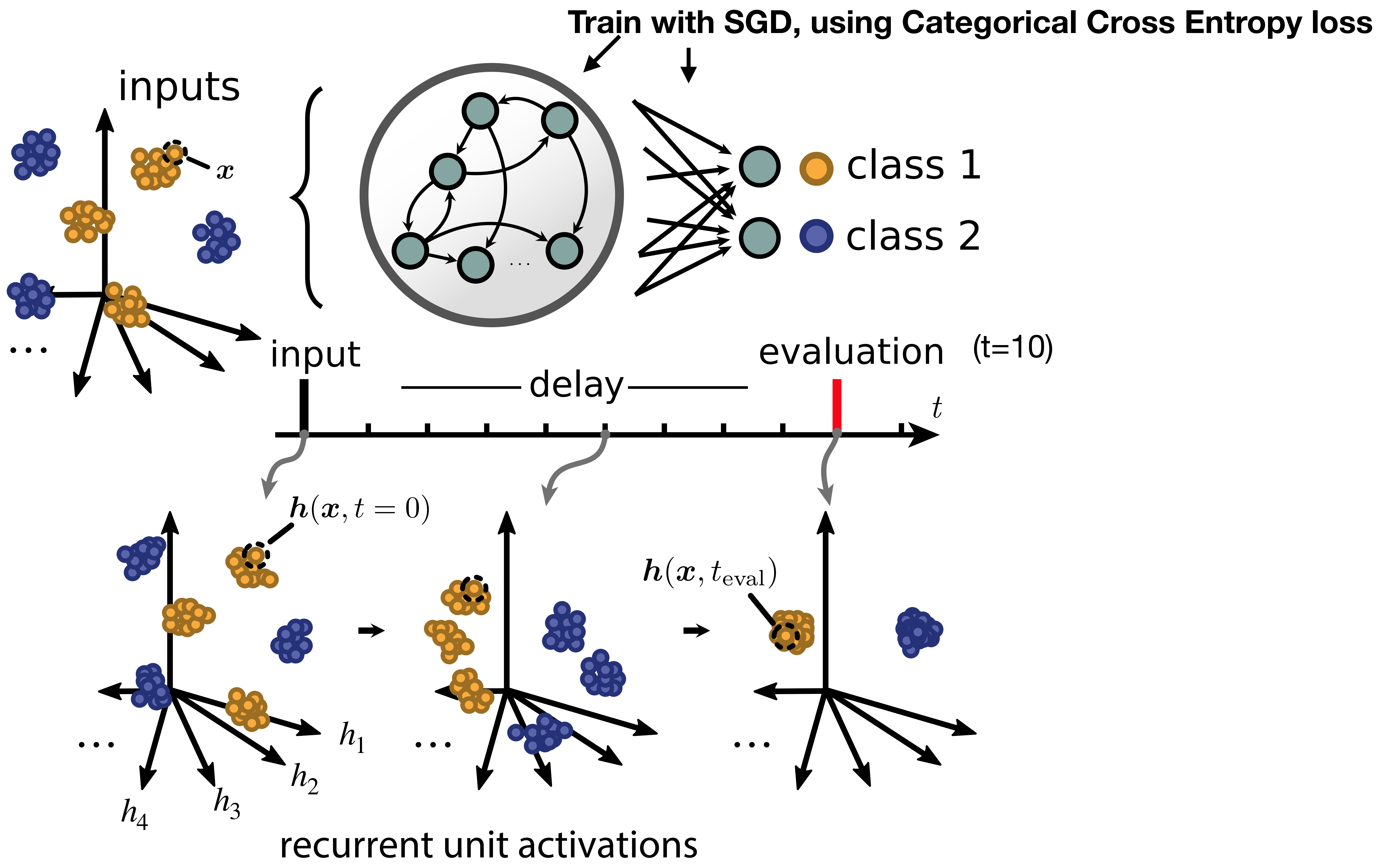
Let's choose a task to train the network on

Delayed classification task

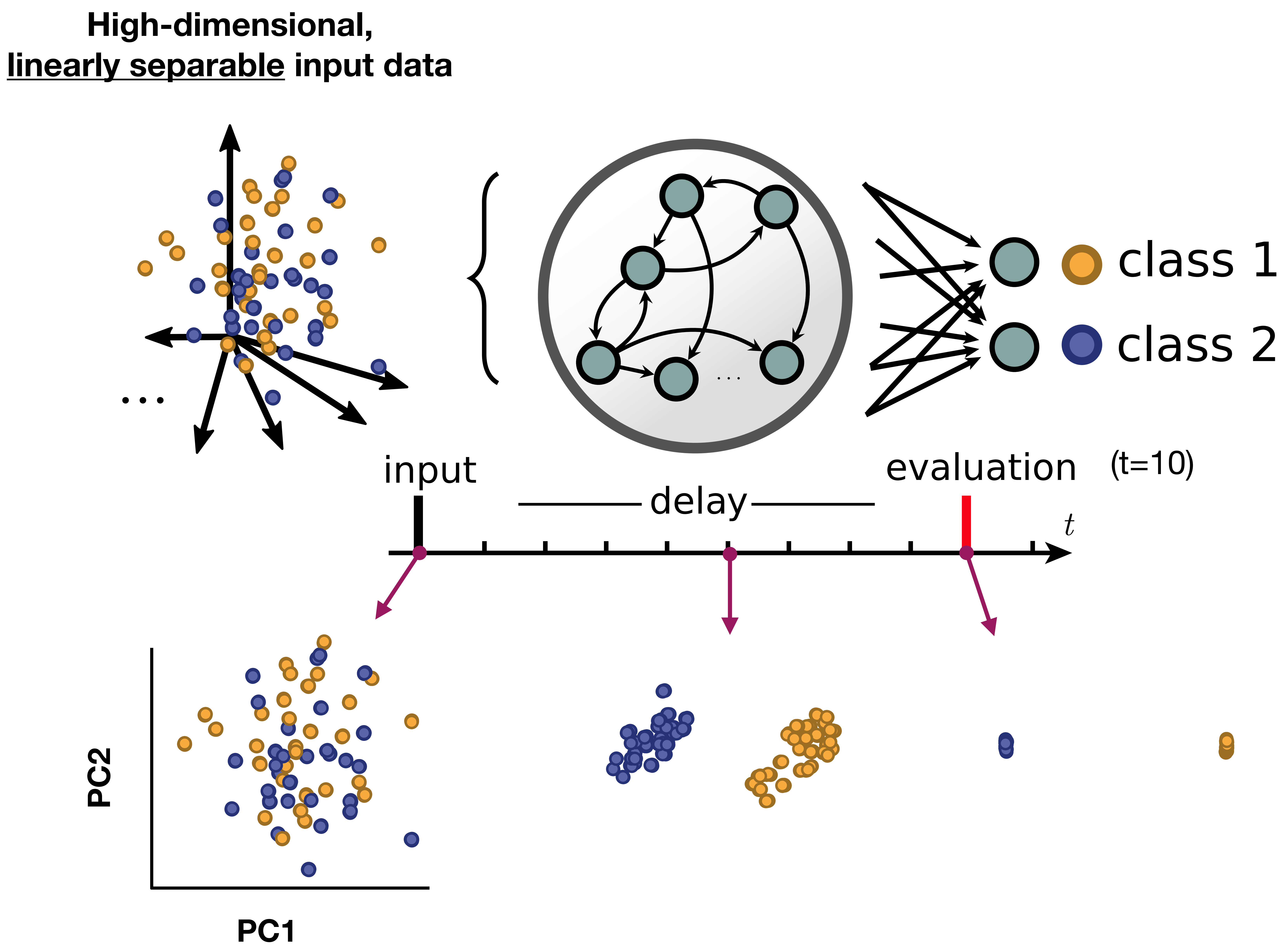


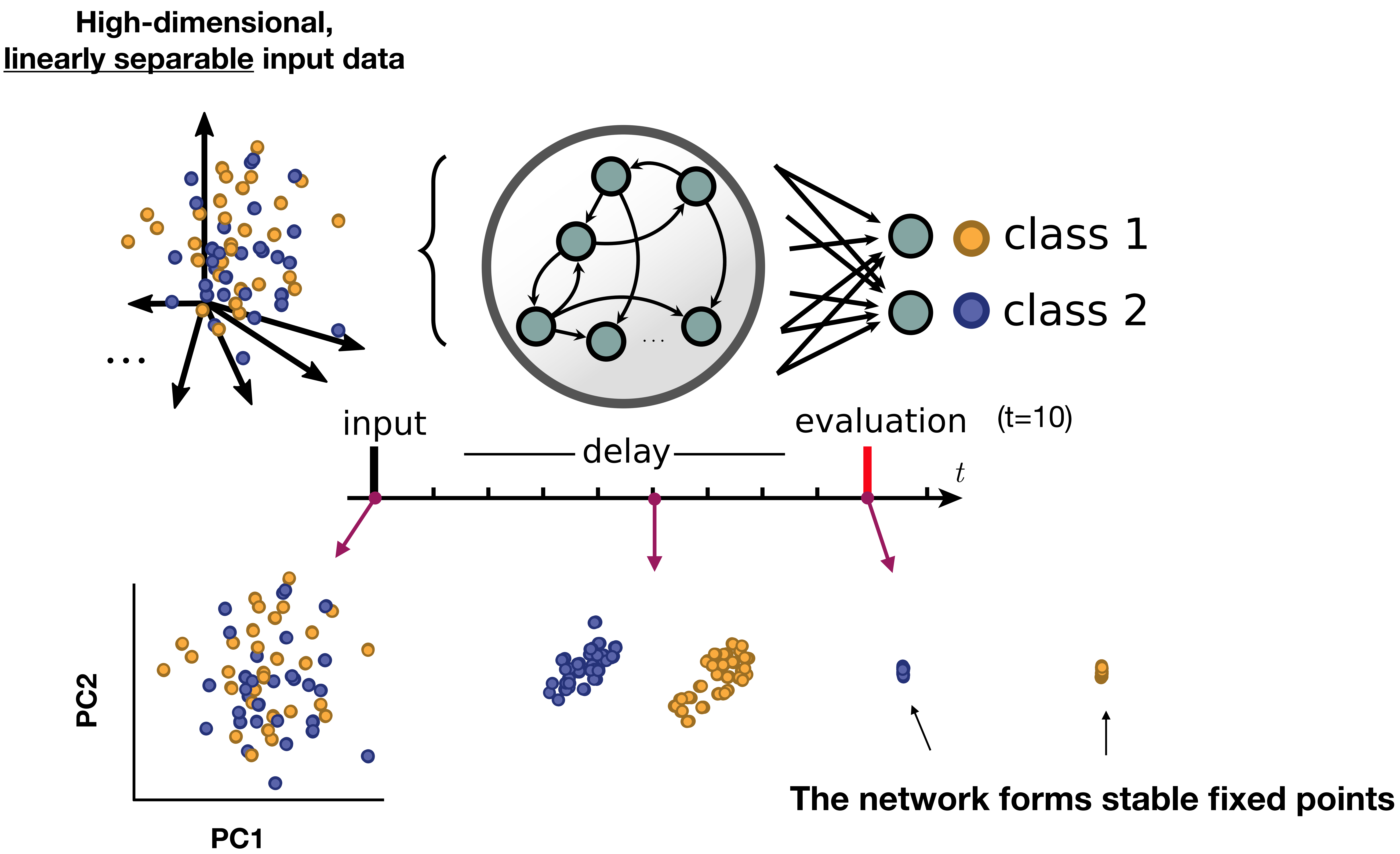
These inputs induce a response in the network. Here $\boldsymbol{h}(x, t = 0)$ is the response of the network to input vector $\boldsymbol{x}$ at time $t = 0$.

Let's choose a task to train the network on



This response evolves through time. This is only a schematic though. Let's see what happens when we actually train the network.

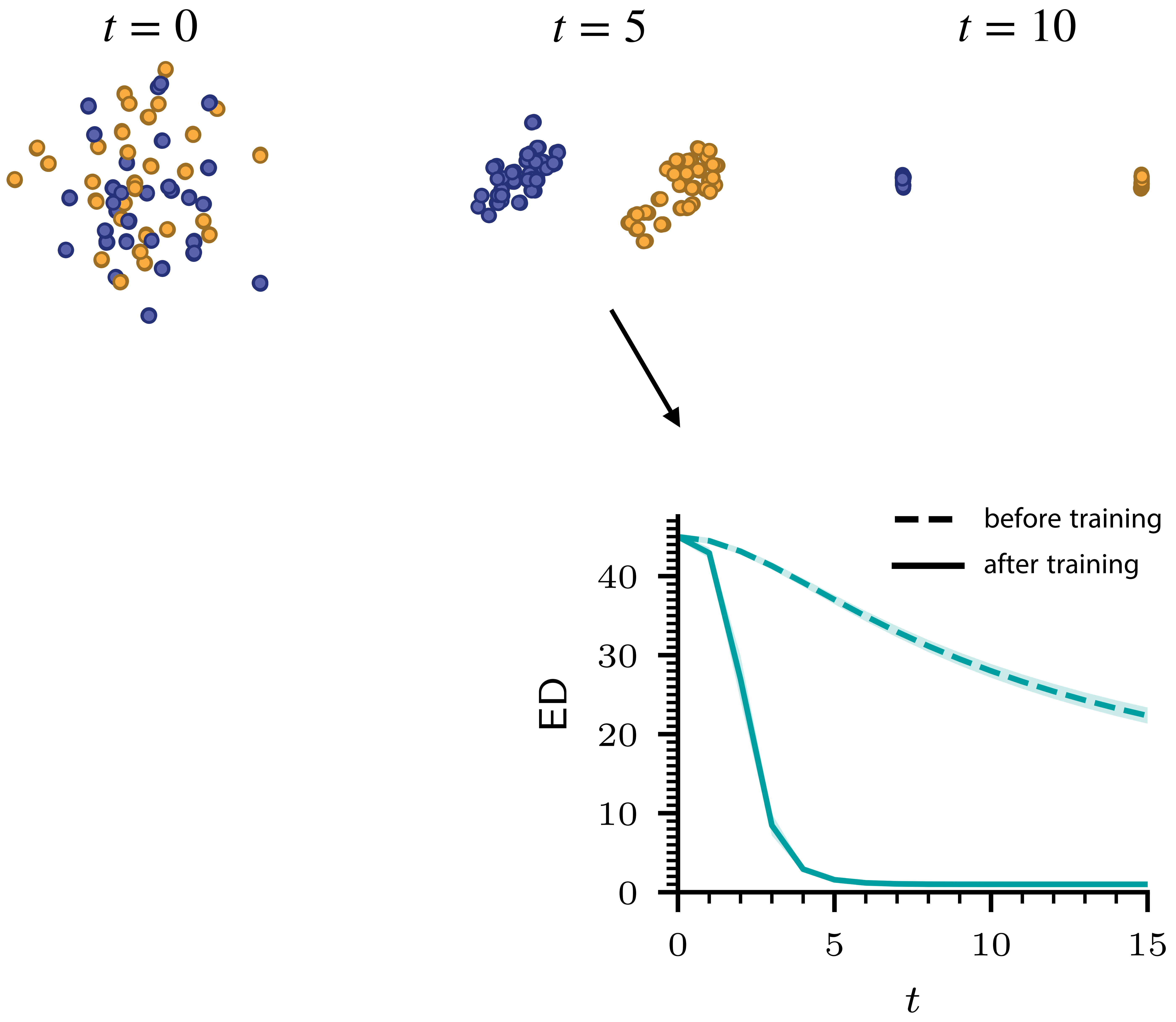

High-dimensional,
linearly separable input data
class 1
class 2
input
delay
evaluation
(t=10)
t
PC2
PC1


High-dimensional,
linearly separable input data
...
class 1
class 2
input
delay
evaluation
(t=10)
t
PC2
PC1
The network forms stable fixed points

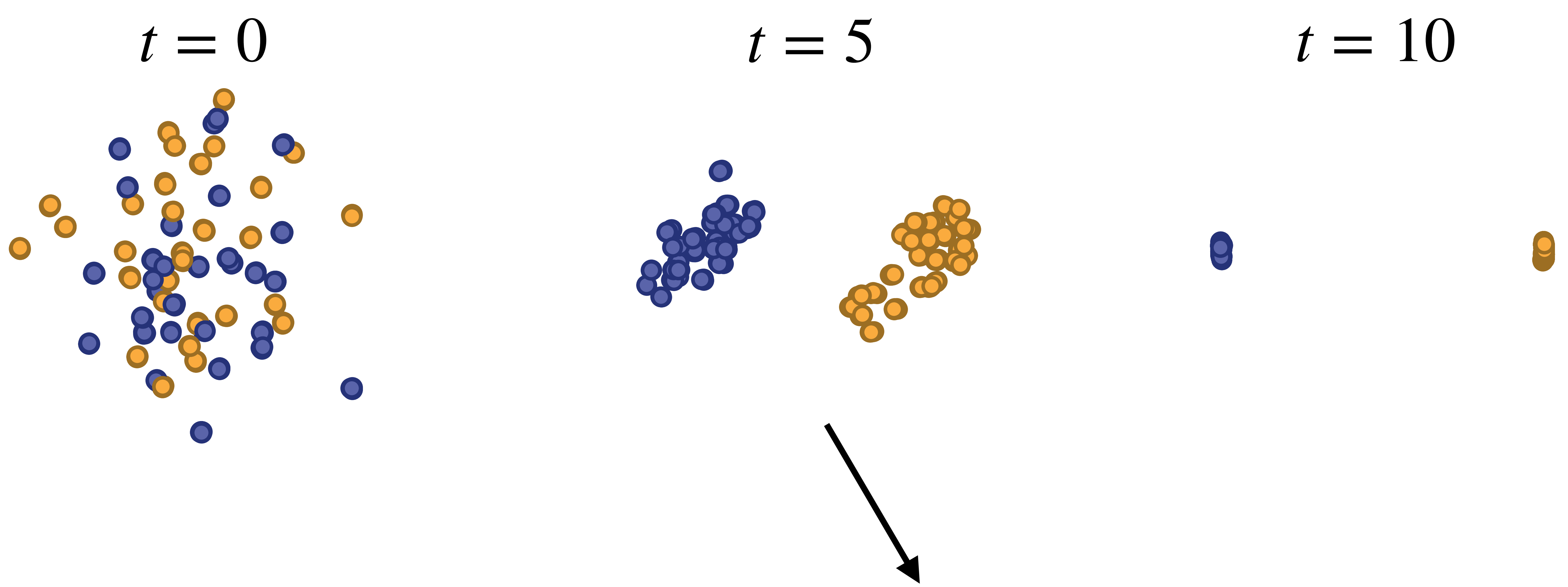
How do we characterize the geometry of these?

Effective dimensionality (ED) is one approach

Rajan et al., NeurIPS (2010)

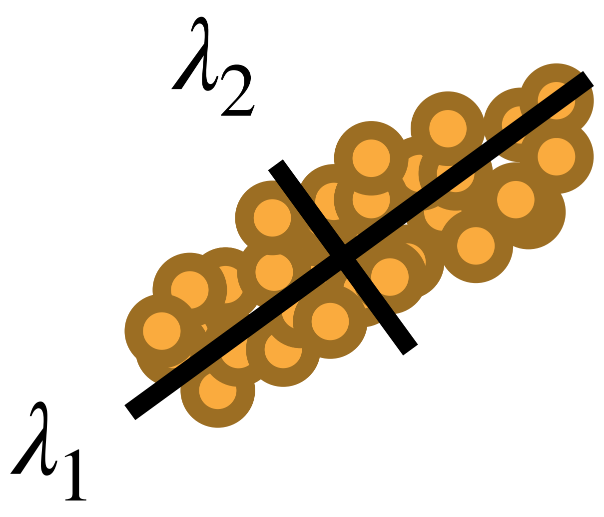


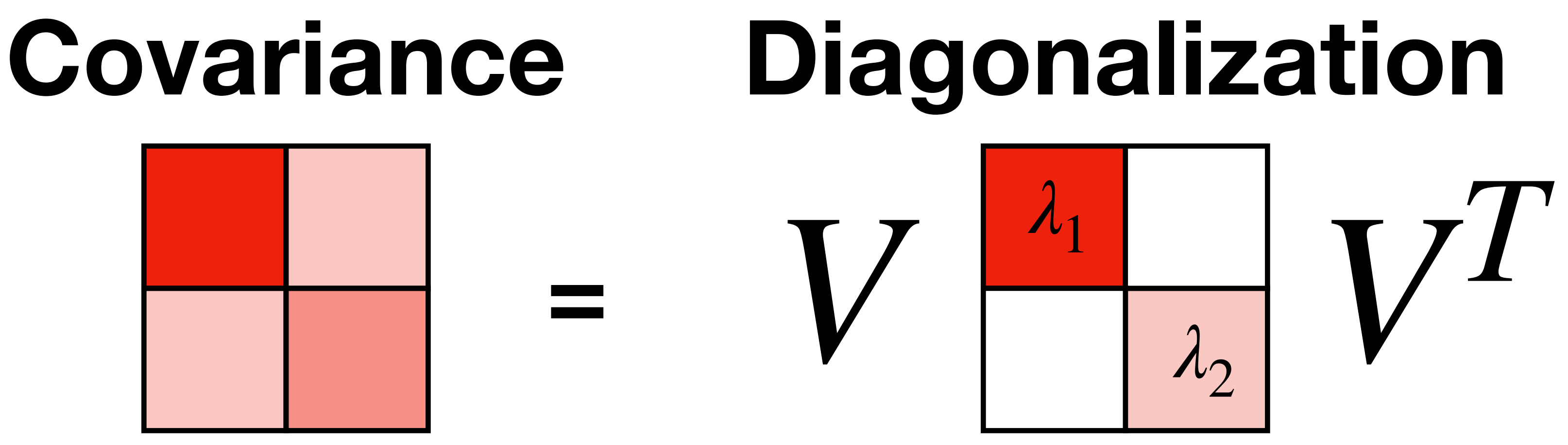
How do we characterize the geometry of these?



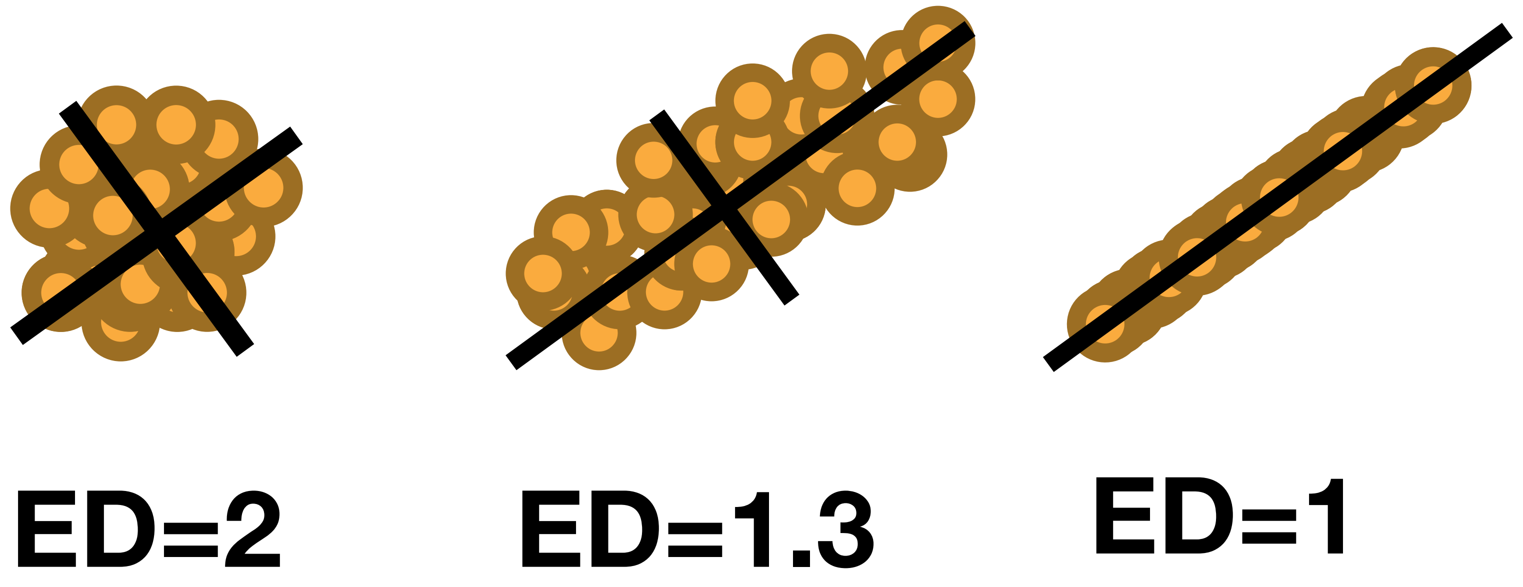
Effective dimensionality (ED) is one approach

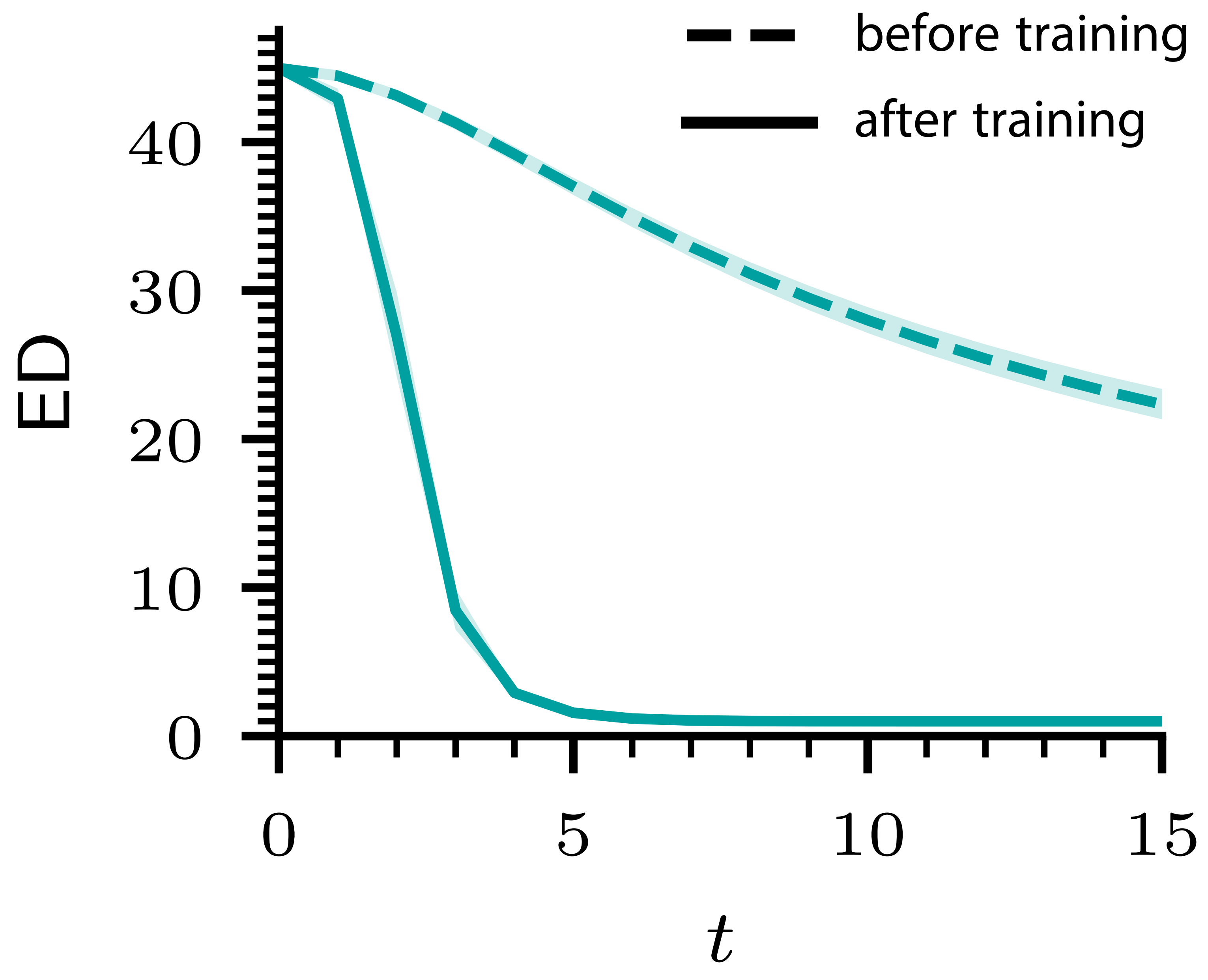
Rajan et al., NeurIPS (2010)



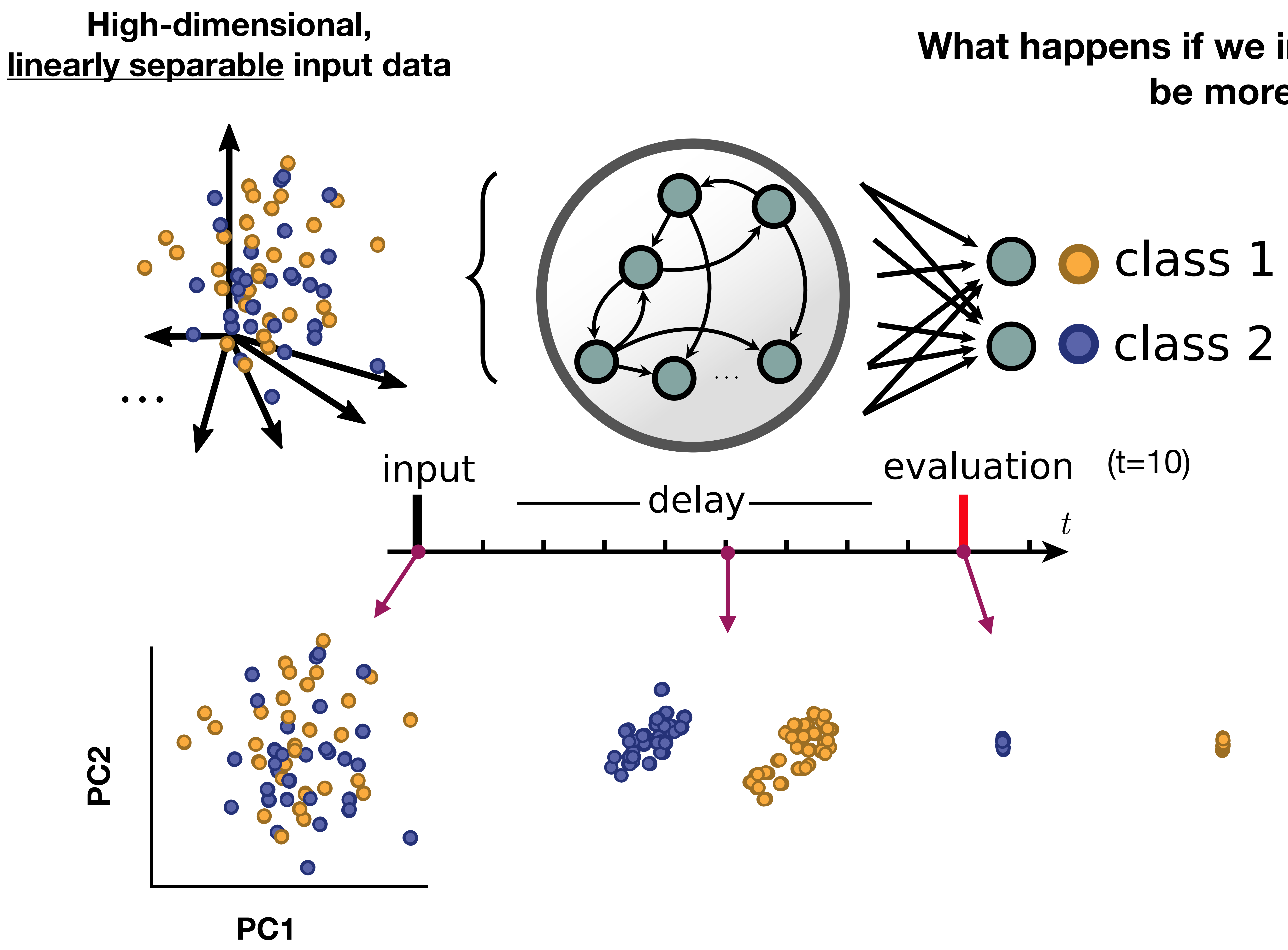


$$\textbf{ED} = \frac{\left(\sum_{k=1}^{d} \lambda_k\right)^2}{\sum_{k=1}^{d} \lambda_k^2}$$

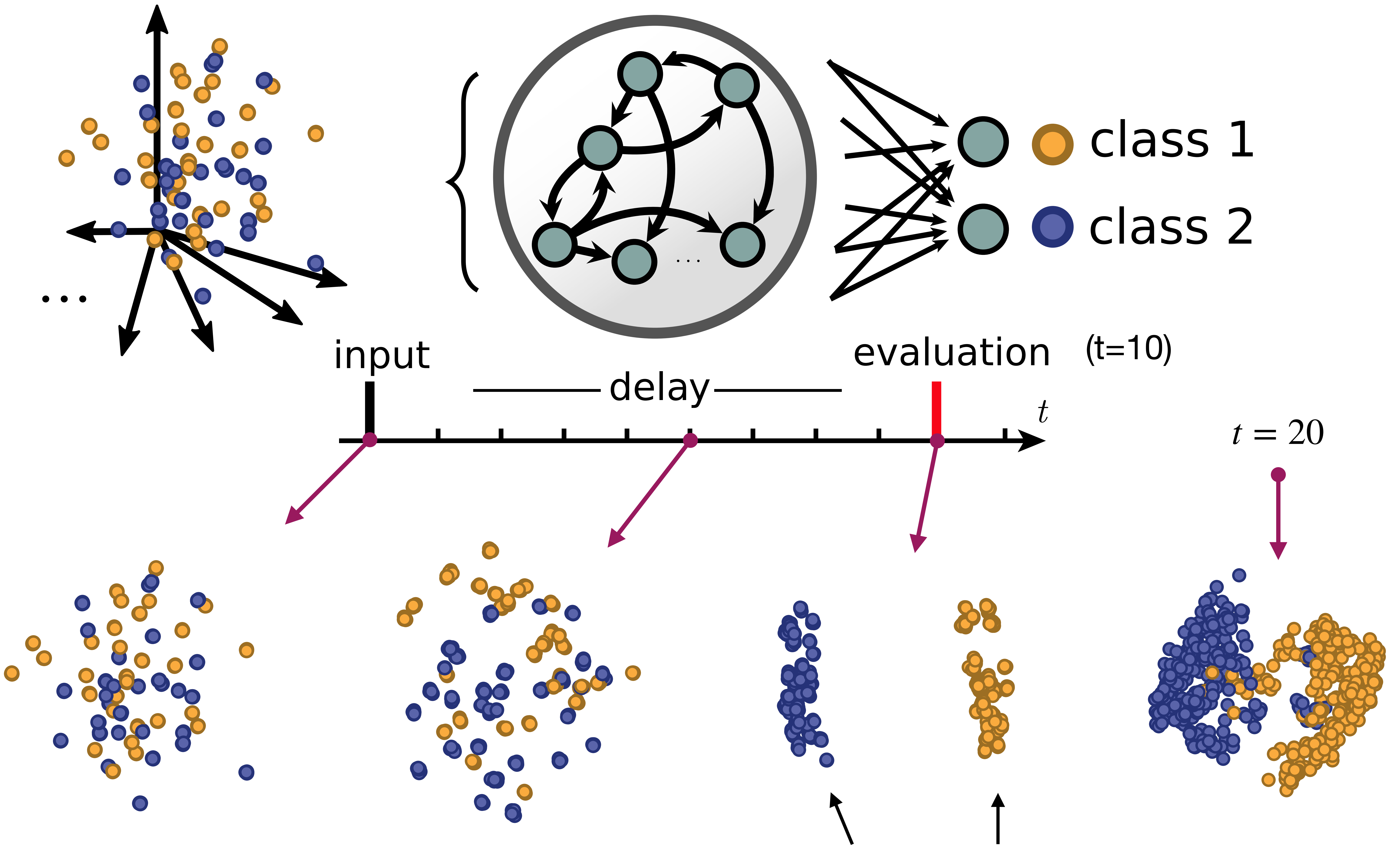




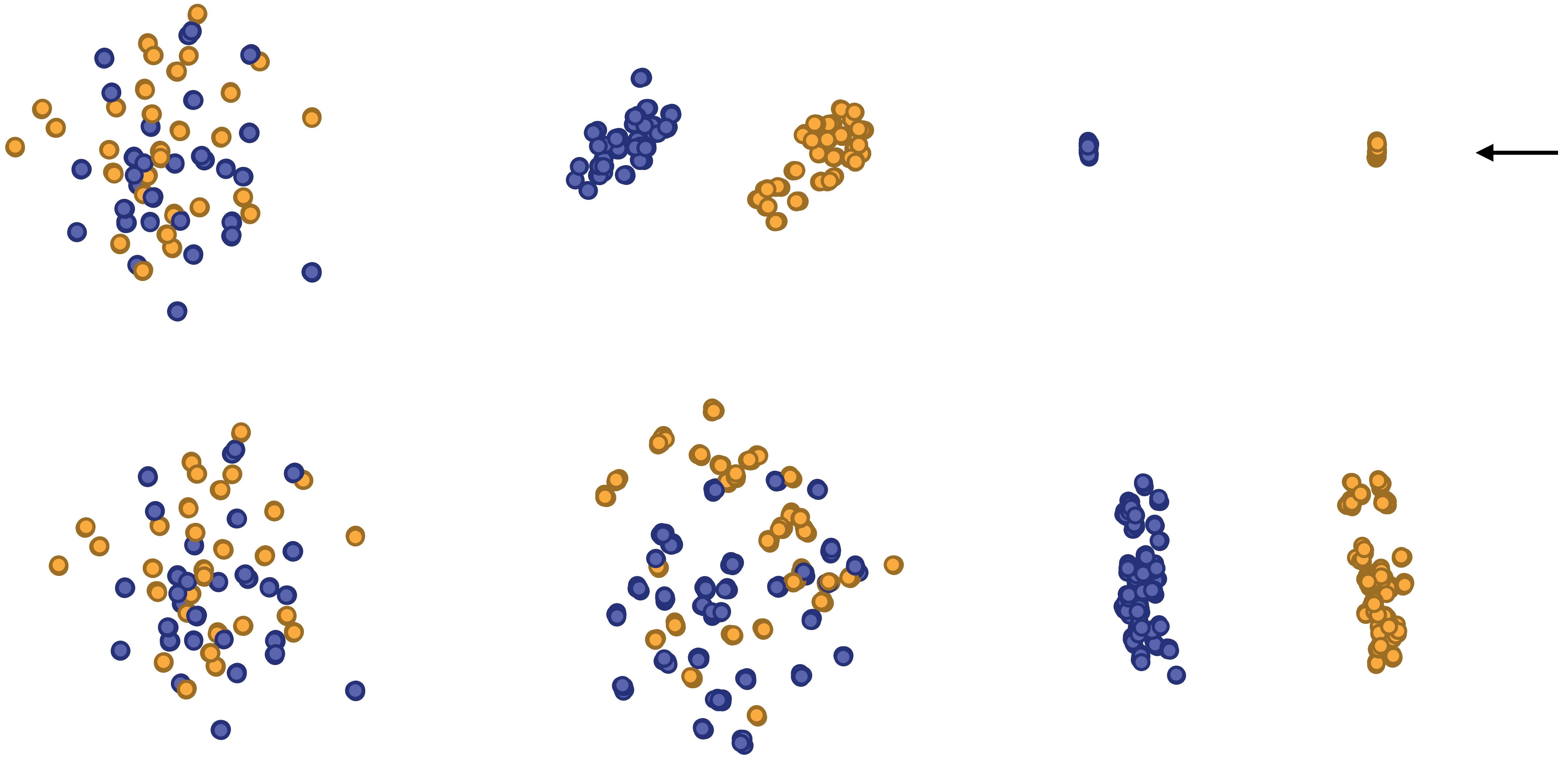
What happens if we initialize the network to be more chaotic?



High-dimensional,
linearly separable input data



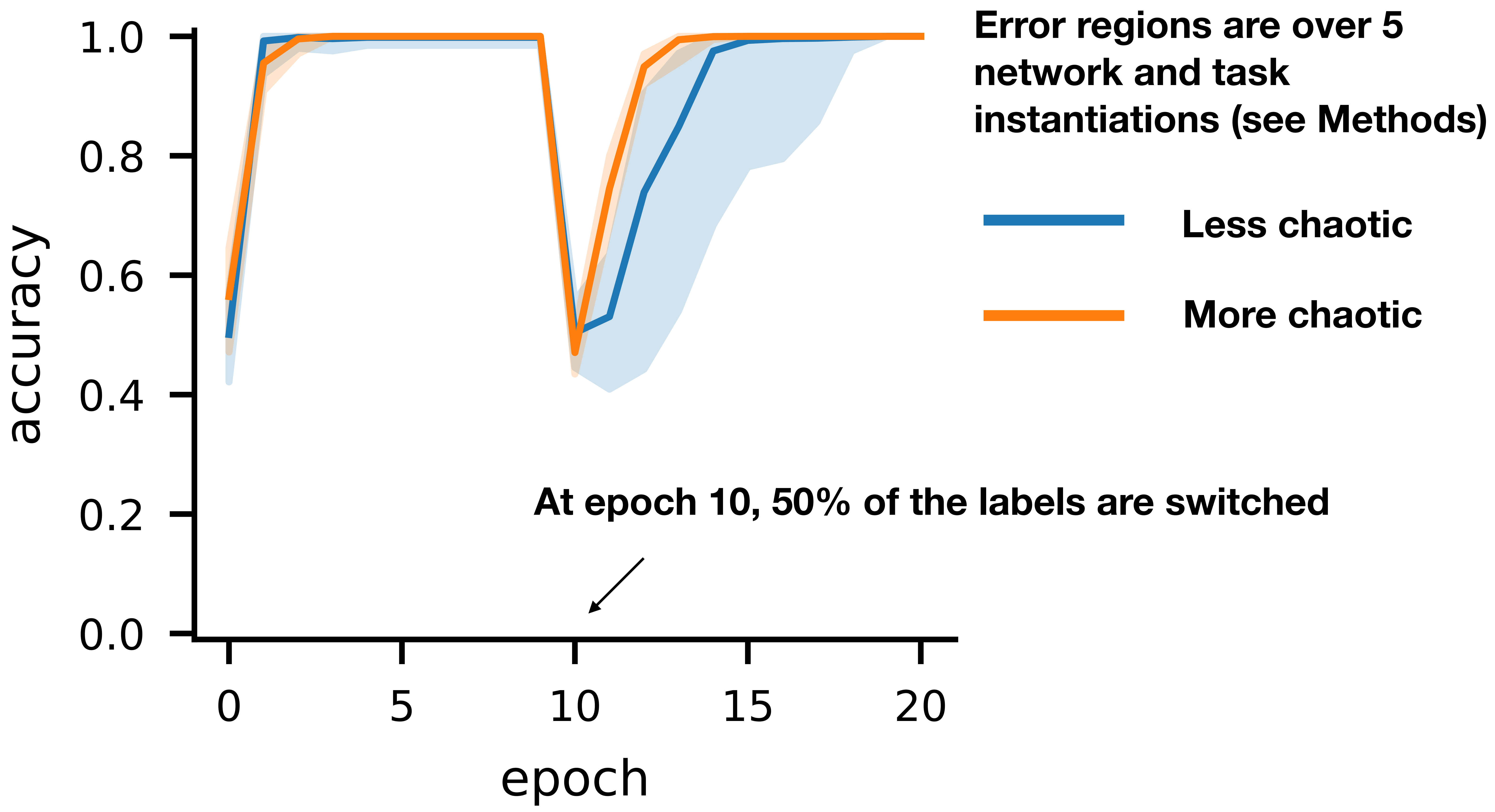
The network forms low-dimensional chaotic attractors



If the labels are changed, this representation may be problematic due to tight overlap between points belonging to different classes

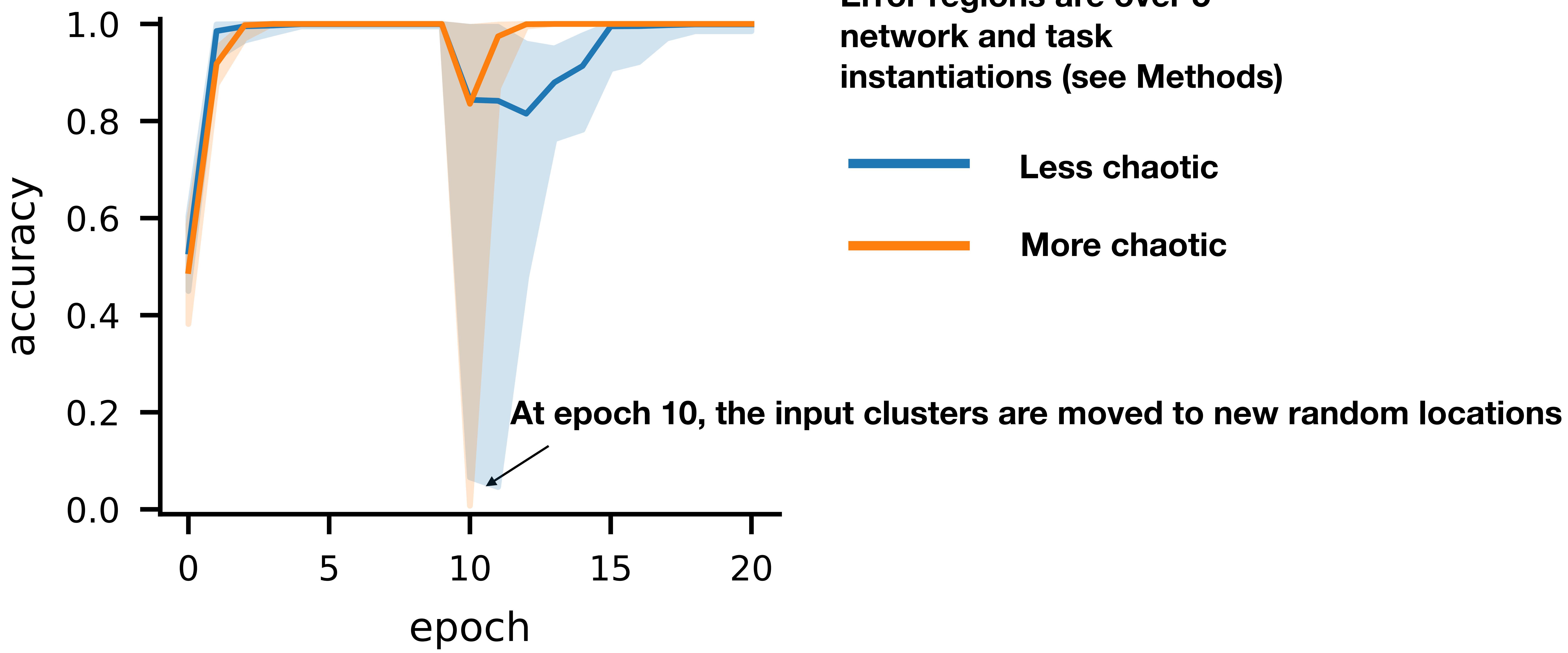
Both networks have perfect classification accuracy at the end of training.
What happens if the task is changed in the middle of training?

Switching task labels



The more chaotic network is better at recovering after the label switch.

Switching cluster locations



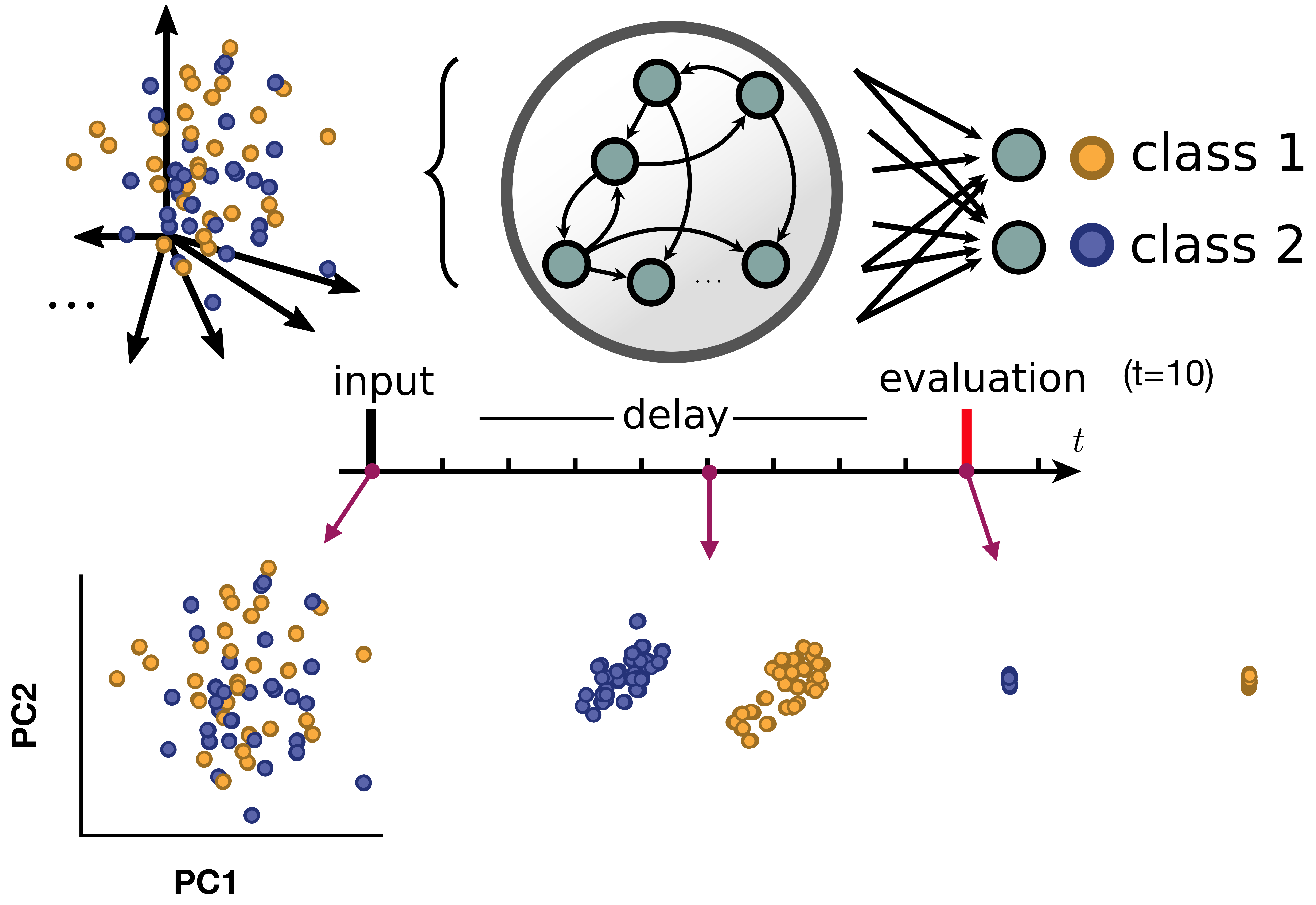
Error regions are over 5 network and task instantiations (see Methods)

Less chaotic

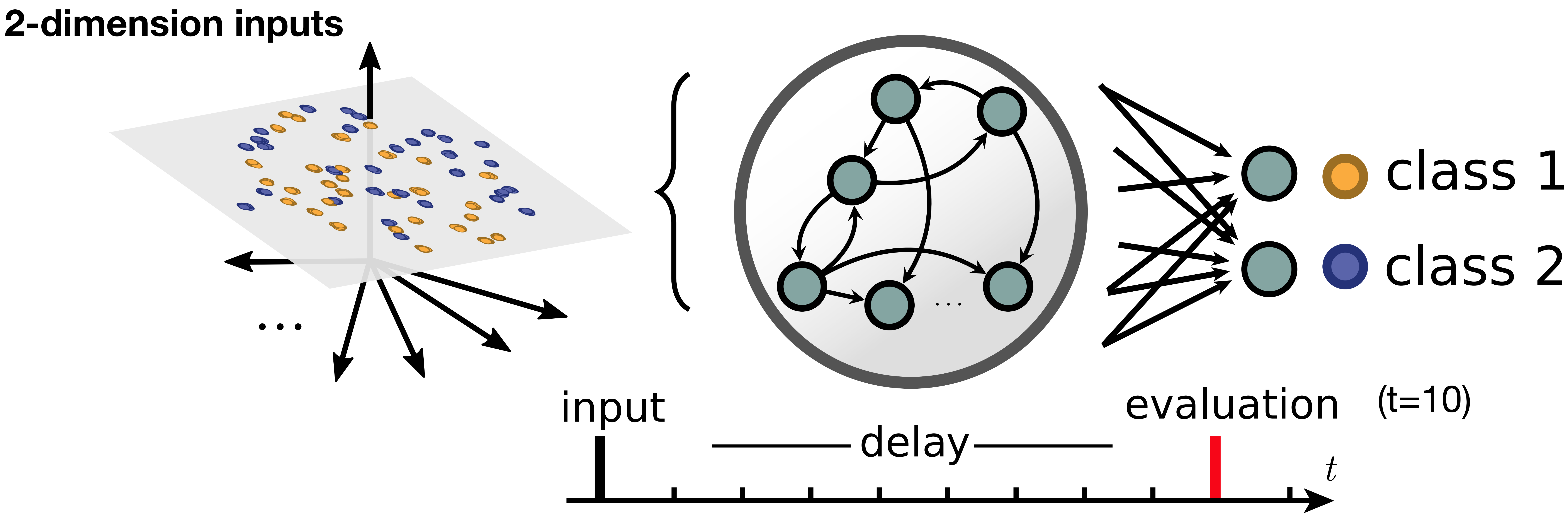
More chaotic

The more chaotic network is better at recovering after the label switch.

- **Before we had high-dimensional, <u>linearly separable</u> input data.**
- **Now we'll look at the case of very low-dimensional input data**



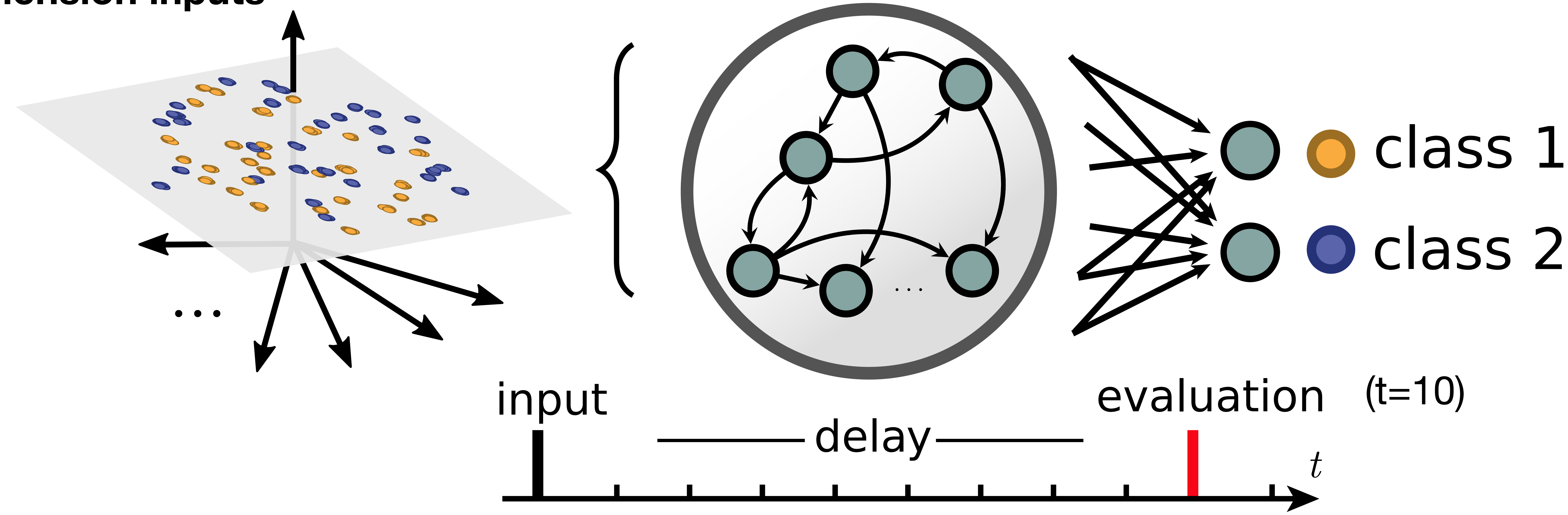
- Before we had high-dimensional, linearly separable input data.
- Now we'll look at the case of very low-dimensional input data



- In this case, it is useful to expand dimensionality to help separate out the classes.
- Chaotic dynamics is one mechanism for this expansion (Legenstein and Maass, *Neural Networks* (2007)).

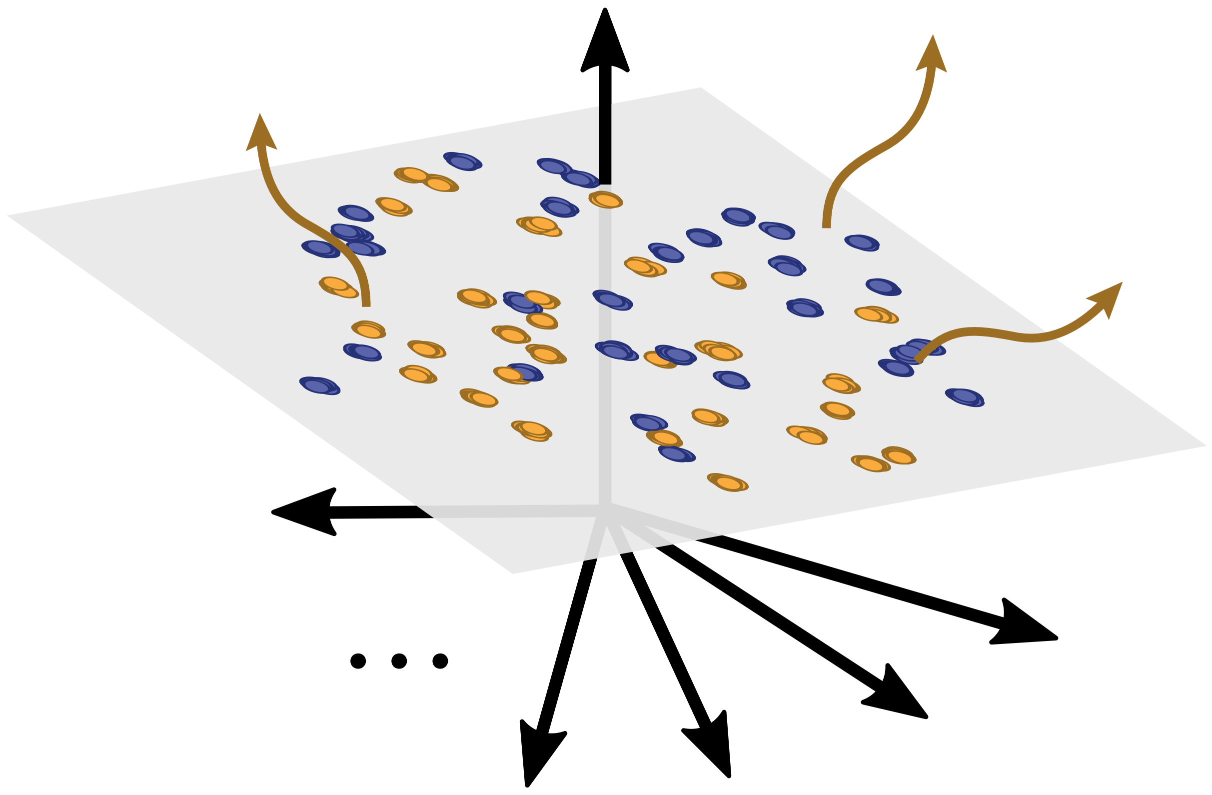
- Before we had high-dimensional, <u>linearly separable</u> input data.
- Now we'll look at the case of very low-dimensional input data

2-dimension inputs



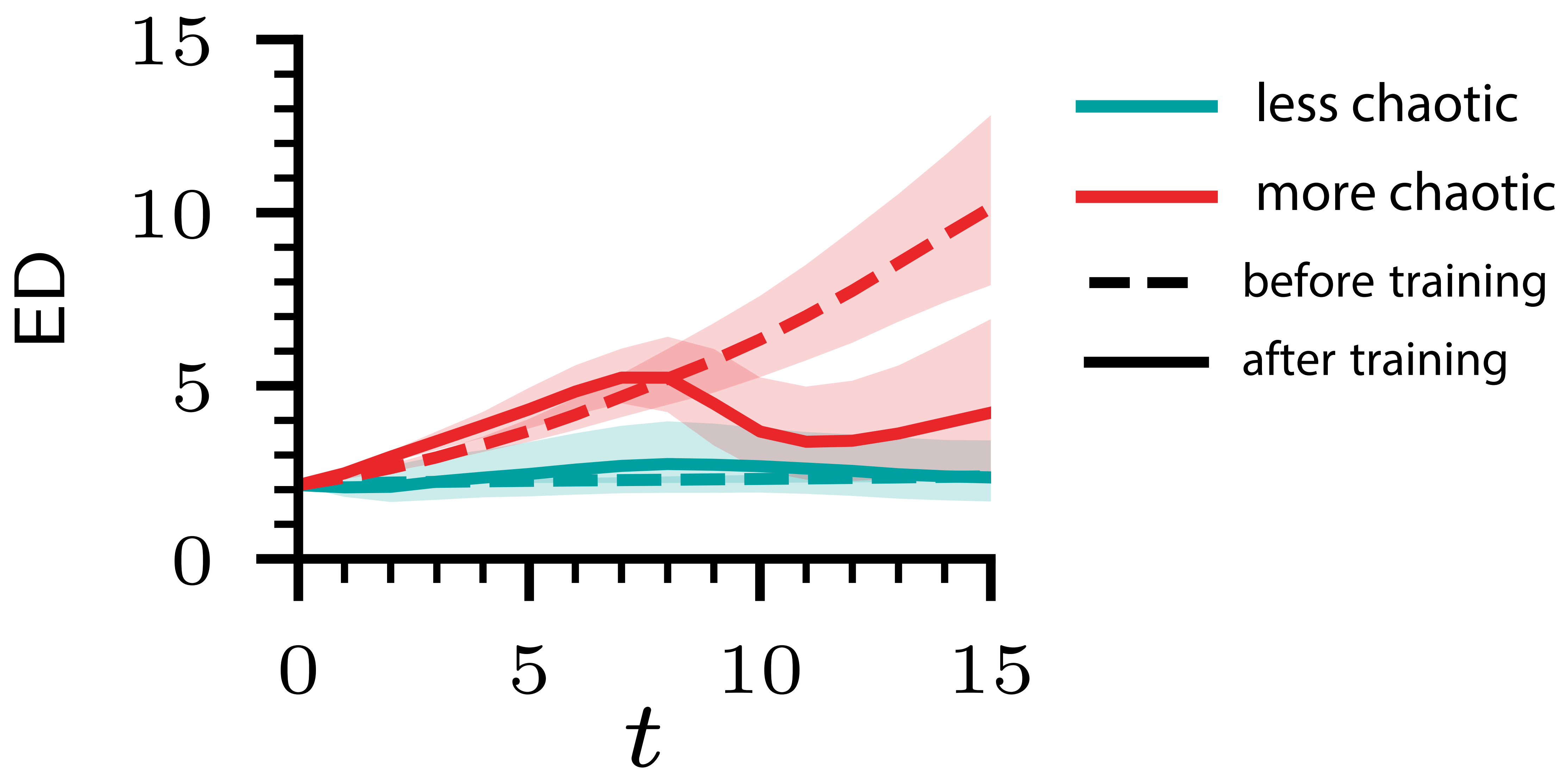
- **In this case, it is useful to expand dimensionality to help separate out the classes.**
- **Chaotic dynamics is one mechanism for this expansion (Legenstein and Maass, *Neural Networks* (2007)).**

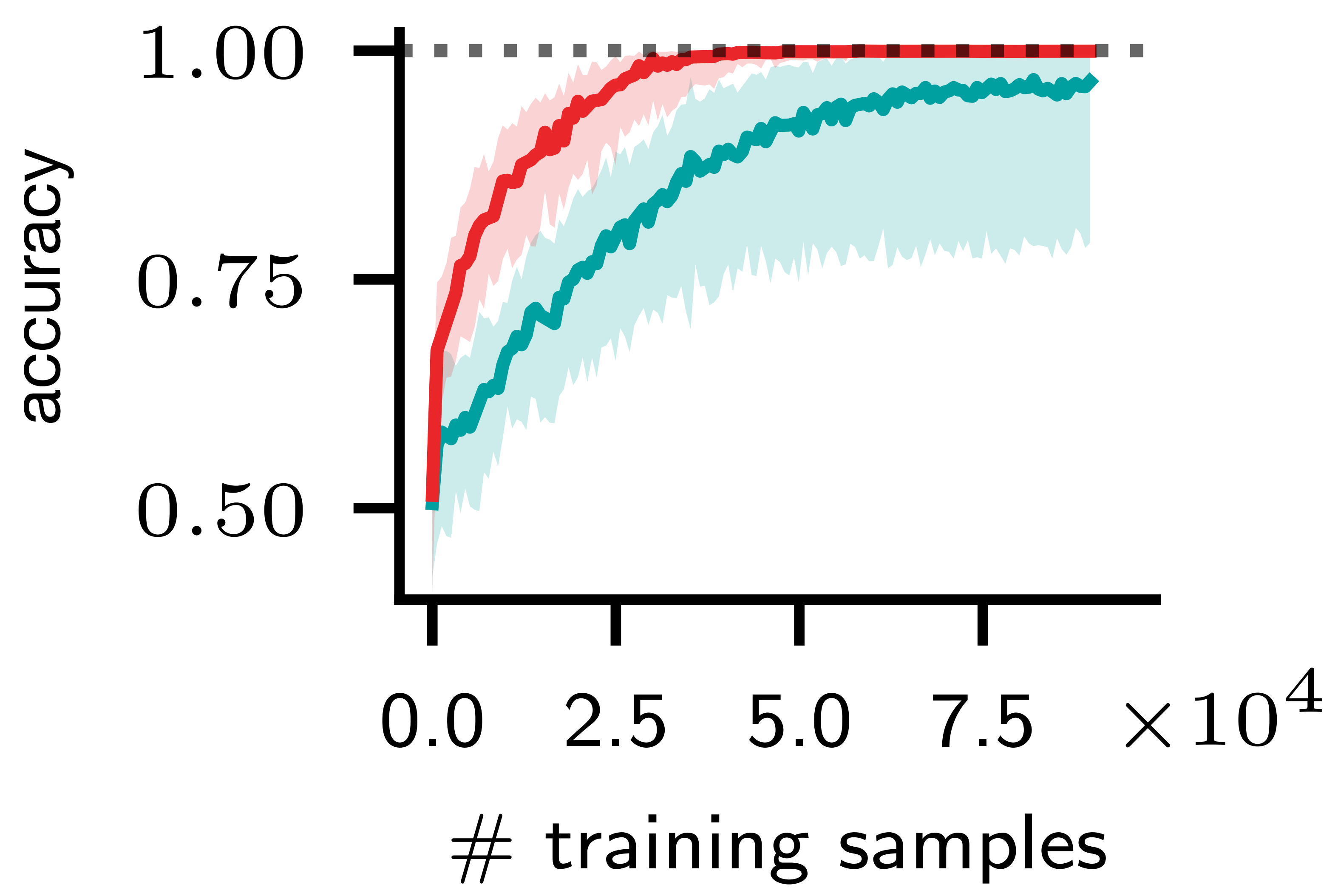
Chaotic divergence



- **The more chaotic network expands dimensionality before training as well as after training.**
- **The less chaotic network is not able to learn this expansion.**

- **This allows the more chaotic network to do better at the task.**





Conclusions

- Chaotic dynamics can have utility in recurrent neural network models:
  - Leads to solutions that can more quickly adapt to changing data.
  - Expands dimensionality, which can help when classifying low-dimensional data.
- This suggests beneficial attributes of the variability seen in biology, which may in part be generated by chaotic dynamics.

Methods

- Network equations

$$h_t = \tanh(Wh_{t-1} + x_t + b) \quad \textbf{recurrent unit activations}$$

$$\hat{o}_t = Rh_t + b' \quad \textbf{readout}$$

- Chaos was modulated by increasing the variance of the connections between neurons at initialization: $W = (1-\epsilon)I + (g\epsilon/\sqrt{N})J$ where $\epsilon = .01$ and J is a matrix with standard normal entries. For the less chaotic network $g = 5$, and for the more chaotic network $g = 250$.
- Error regions on plots are found by fitting a gamma distribution to the data, and shading the region containing 75% probability mass of the distribution. For the accuracy plots this distribution was reflected around the y-axis and shifted by +1 (since accuracy is bounded above by 1), and for the dimensionality plots the distribution was shifted by +1 (since the effective dimensionality is bounded from below by 1).