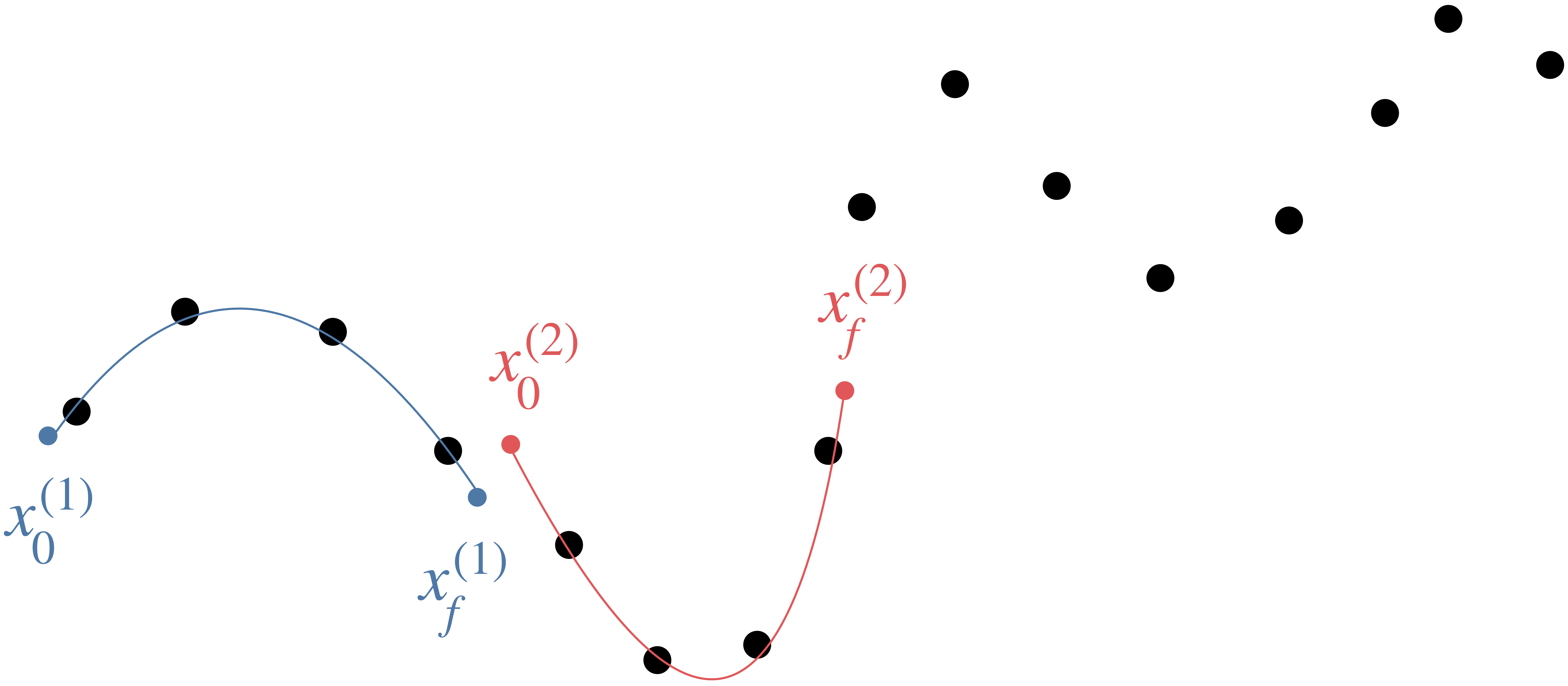
Neural ODE Code Walkthrough



torchdiffeq

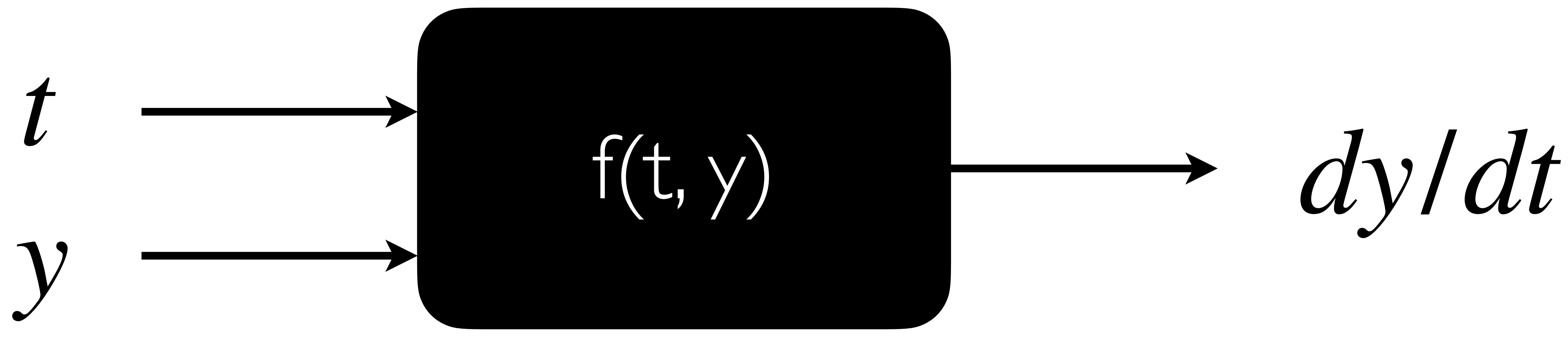
https://github.com/rtqichen/torchdiffeq

Load in the data, look at shapes, plot

data posted on our website

columns: t, y1, y2

Write a nn.Module for dydt = f(t, y)



let's use 3 layers, SiLU activation

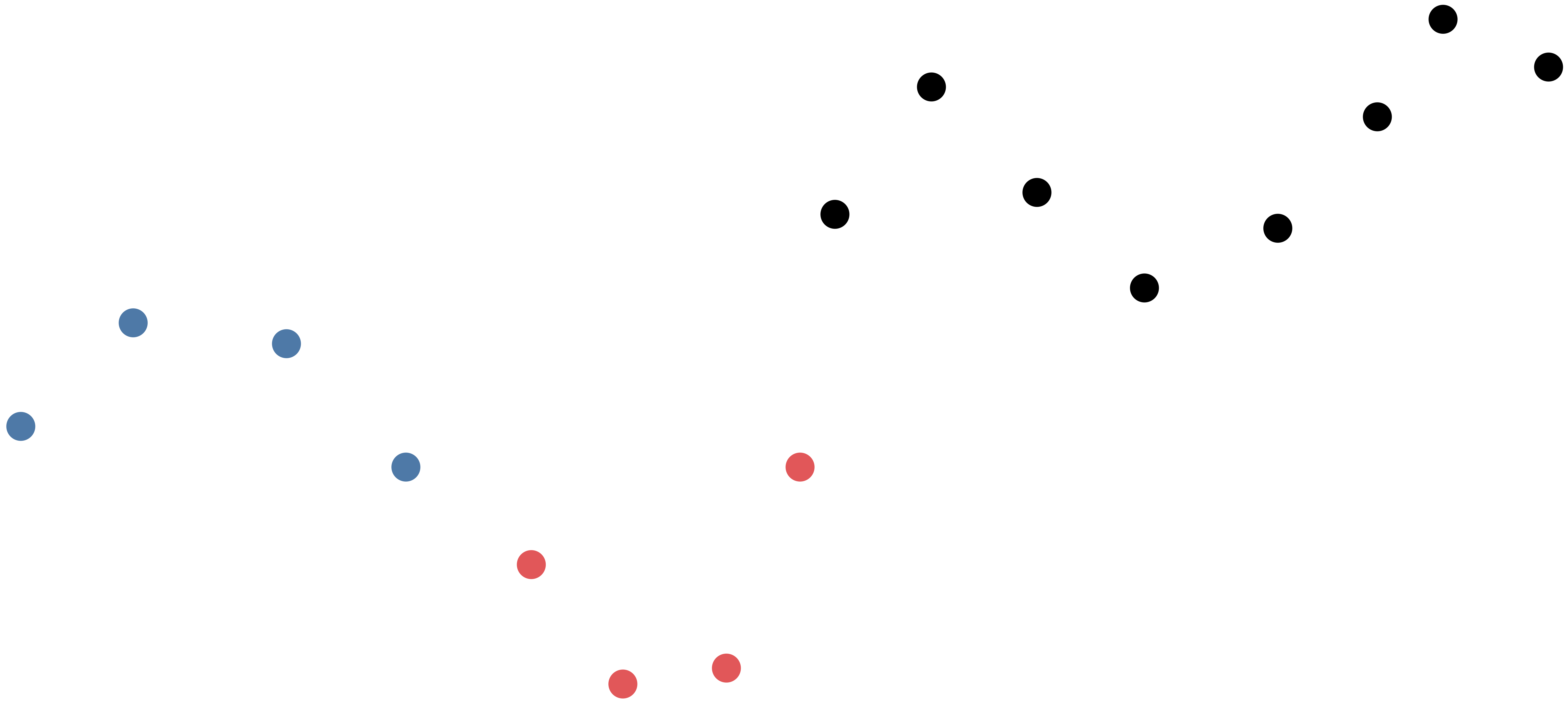
Use odeint to integrate

Write a train function

Optimize

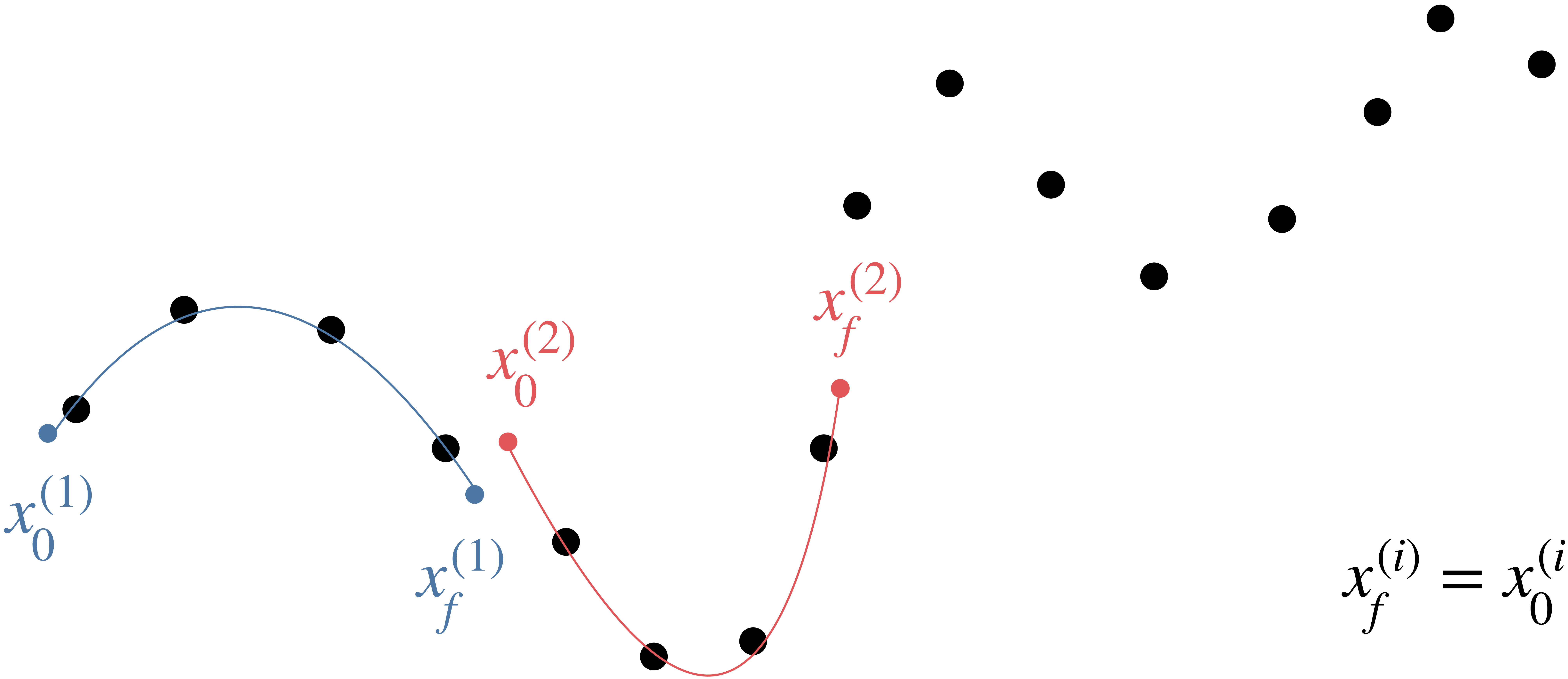
if struggling, try training on only a smaller fraction of the time steps at first, and then show it more time steps.

Incrementally add time sequence



https://sebastiancallh.github.io/post/neural-ode-weather-forecast/

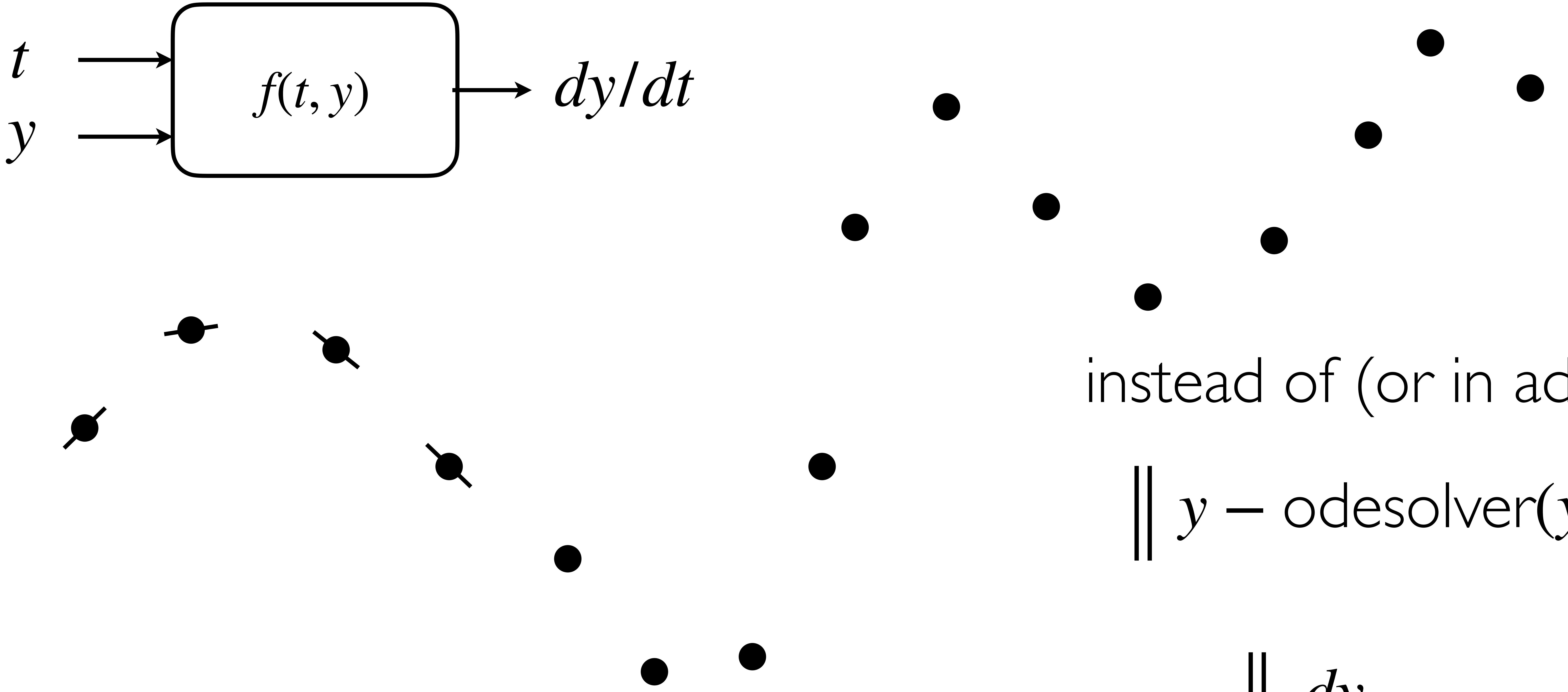
Multiple Shooting



$$x_f^{(i)} = x_0^{(i+1)}$$

Multiple shooting for training neural differential equations on time series
Evren Mert Turan, Johannes Jäschke

Neural ODE with derivatives (collocation)



instead of (or in addition to)

$$\left\| \, y - \text{odesolver}(y_0, t) \, \right\|_2^2$$

$$\left\| \, \frac{dy}{dt} - f(y, t) \, \right\|_2^2$$

Collocation based training of neural ordinary differential equations
Elisabeth Roesch, Christopher Rackauckas, Michael P H Stumpf

Don’t need to replace entire f(y, t) with a neural net

$$\dot{x} = \alpha x + u_1(x, y; \theta)$$

$$\dot{y} = -\theta_1 y + u_2(x, y; \theta)$$

example from Universal Differential Equations for Scientific Machine Learning
Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, Ali Ramadhan, Alan Edelman