NEURAL ORDINARY DIFFERENTIAL EQUATIONS

RECAP: ODE

$$\frac{d^2\mathbf{x}}{dt^2} = \frac{F(\dot{\mathbf{x}}, \mathbf{x}, t)}{m}$$

RECAP: ODE

$$\frac{d^2\mathbf{x}}{dt^2} = \frac{F(\dot{\mathbf{x}}, \mathbf{x}, t)}{m}$$

$$F(t, \mathbf{x}, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n-1)}) = \mathbf{x}^{(n)}$$

STOCHASTIC ODE

Reset

INITIAL VALUE PROBLEM

$$\begin{cases} \frac{dx}{dt} = f(x(t), t)) \\ x(t_0) = x_0 \end{cases}$$

INITIAL VALUE PROBLEM

$$\begin{cases} \frac{dx}{dt} = f(x(t), t)) \\ x(t_0) = x_0 \end{cases}$$

$$x(t) = x(t_0) + \int_{t_0}^{t} f(x(t), t)dt$$

QUIZZ

What is the difference between an ordinary differential equation (ODE) and a partial differential equation (PDE)?

Math time Solve the following ODE:

$$\frac{dx}{dt} = \sin(t) + 1/t$$

$$t_0 = \pi/2$$

$$x(t_0) = -\ln(3)$$

$$x(3\pi) = ?$$

HERE IS SOME GPT TO "HELP"

$$
\begin{aligned}
x(t) &= x(t_0) + \int_{t_0}^{t} f(x(t), t) dt \\
&= -\ln(3) + \int_{\pi/2}^{t} \sin(t) + 1/t dt \\
&= -\ln(3) + [-\cos(t) + \ln(t)]_{\pi/2}^{t} \\
&= -\ln(3) + [-\cos(3\pi) + \ln(3\pi)] - [-\cos(\pi/2) + \ln(\pi/2)]
\end{aligned}
$$

HERE IS SOME GPT TO "HELP"

$$
\begin{aligned}
x(t) = x(t_0) + \int_{t_0}^{t} f(x(t), t) dt \\
= -\ln(3) + \int_{\pi/2}^{t} \sin(t) + 1/t dt \\
= -\ln(3) + [-\cos(t) + \ln(t)]_{\pi/2}^{t} \\
= -\ln(3) + [-\cos(3\pi) + \ln(3\pi)] - [-\cos(\pi/2) + \ln(\pi/2)]
\end{aligned}
$$

Surprisingly enough, it's correct

NUMERICAL INTEGRATION OF ODE

EULER'S METHOD

Step

1 2 3 4

EULER'S METHOD

$$\tilde{x}(t+\tau) = \tilde{x}(t) + \tau f(\tilde{x}(t), t)$$

INSTABILITY

Initial angle

0 0.2π 0.4π 0.6π 0.8π π

Timestep

0.001 0.101 0.201 0.301 0.401 0.5 0.6 0.7 0.8 0.9 1

ERROR ESTIMATION

$$x(t+\tau) - \tilde{x}(t+\tau) = x(t+\tau) - x(t) - \tau x'(t)$$

ERROR ESTIMATION

$$x(t+\tau) - \tilde{x}(t+\tau) = x(t+\tau) - x(t) - \tau x'(t)$$

$$x(t+\tau) = x(t) + x'(t)\tau + \frac{1}{2}x''(t)\tau^2 + O(\tau^3)$$

ERROR ESTIMATION

$$x(t+\tau) - \tilde{x}(t+\tau) = x(t+\tau) - x(t) - \tau x'(t)$$

$$x(t+\tau) = x(t) + x'(t)\tau + \frac{1}{2}x''(t)\tau^2 + O(\tau^3)$$

$$x(t+\tau) - \tilde{x}(t+\tau) = \frac{1}{2}x''(t)\tau^2 + O(\tau^3)$$

ERROR ESTIMATION

$$x(t+\tau) - \tilde{x}(t+\tau) = x(t+\tau) - x(t) - \tau x'(t)$$

$$x(t+\tau) = x(t) + x'(t)\tau + \frac{1}{2}x''(t)\tau^2 + O(\tau^3)$$

$$x(t+\tau) - \tilde{x}(t+\tau) = \frac{1}{2}x''(t)\tau^2 + O(\tau^3)$$

$$\sum_{s=1}^{\frac{t-t_0}{\tau}} \left[\frac{1}{2}x''(t_0+s\tau)\tau^2 + O(\tau^3)\right] = C(t-t_0)\tau + O(\tau^2)$$

MIDPOINT METHOD

Step 1 2 3 4 5 6

MIDPOINT MATH

$$x_{n+1/2} = x_n + \frac{\tau}{2} f(x_n, t_n)$$

$$x_{n+1} = x_n + \tau f(x_{n+1/2}, t_n + 1/2\tau)$$

MIDPOINT MATH

$$x_{n+1/2} = x_n + \frac{\tau}{2} f(x_n, t_n)$$
$$x_{n+1} = x_n + \tau f(x_{n+1/2}, t_n + 1/2\tau)$$

What is the order of the method?

RUNGE-KUTTA METHOD

404 Not Found

- Code: NoSuchKey
- Message: The specified key does not exist.
- RequestID: 3749c53cb20e8f41

RUNGE-KUTTA METHOD

404 Not Found

- Code: NoSuchKey
- Message: The specified key does not exist.
- RequestID: dca70816525faed6

Sorry, I got lazy with the Copilot

RUNGE-KUTTA METHOD

RUNGE-KUTTA

Step

1 2 3 4 5 6 7 8 9 10

FÜR DAS PROTOKOLL

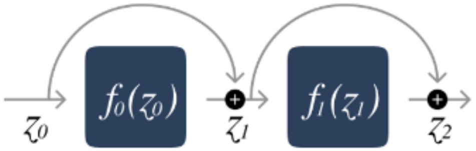
WHEN IS AN ODE INTEGRATION METHOD IS CALLED 3-RD ORDER?

- In uses a 3-rd order polynomial
- Its local truncation error is of the order $O(\tau^3)$, where τ is the time step
- Its total accumulated error is of the order $O(\tau^3)$, where τ is the time step
- It uses 3 evaluations of the derivative function $f(x(t), t)$

NEURAL ORDINARY DIFFERENTIAL EQUATIONS

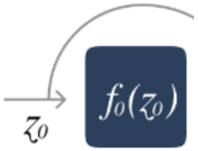
RESNET

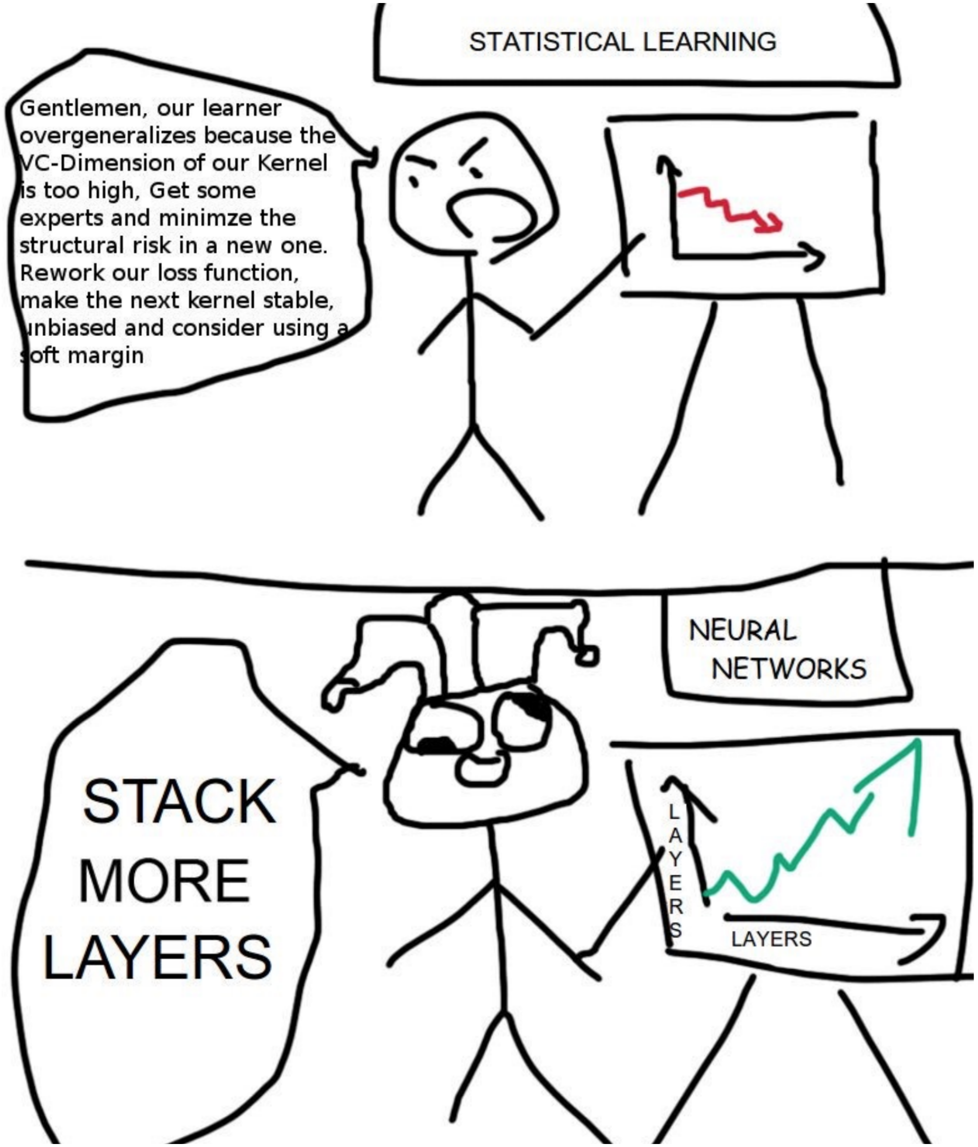
RESNET



$$z_{t+1} = z_t + f_t(z_t)$$

RESNET


$f_0(z_0)$
z_0


STATISTICAL LEARNING
Gentlemen, our learner overgeneralizes because the VC-Dimension of our Kernel is too high, Get some experts and minimze the structural risk in a new one. Rework our loss function, make the next kernel stable, unbiased and consider using a soft margin
NEURAL NETWORKS
STACK MORE LAYERS
LAYERS
LAYERS

RESNET

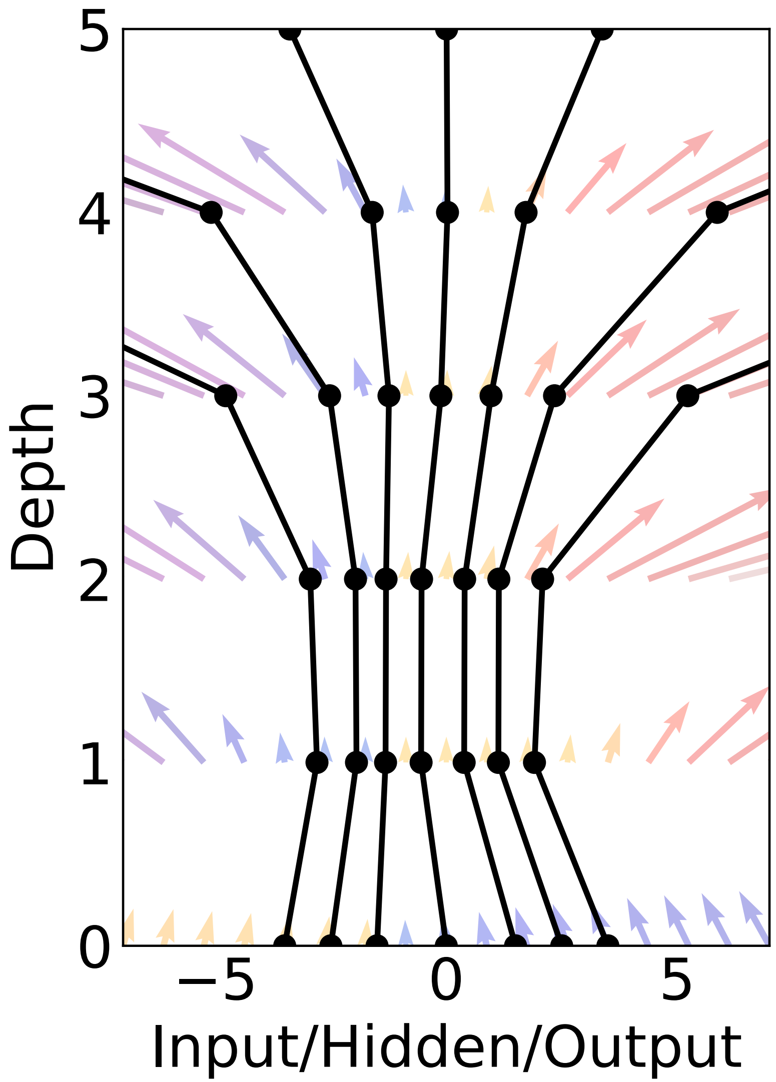
STATISTICAL LEARNING
Gentlemen, our learner overgeneralizes because the VC-Dimension of our Kernel is too high, Get some experts and minimze the structural risk in a new one. Rework our loss function, make the next kernel stable, unbiased and consider using a soft margin
TO THE MOON!
STACK MORE LAYERS
LAYERS
LAYERS

REUSE WEIGHTS BETWEEN LAYERS

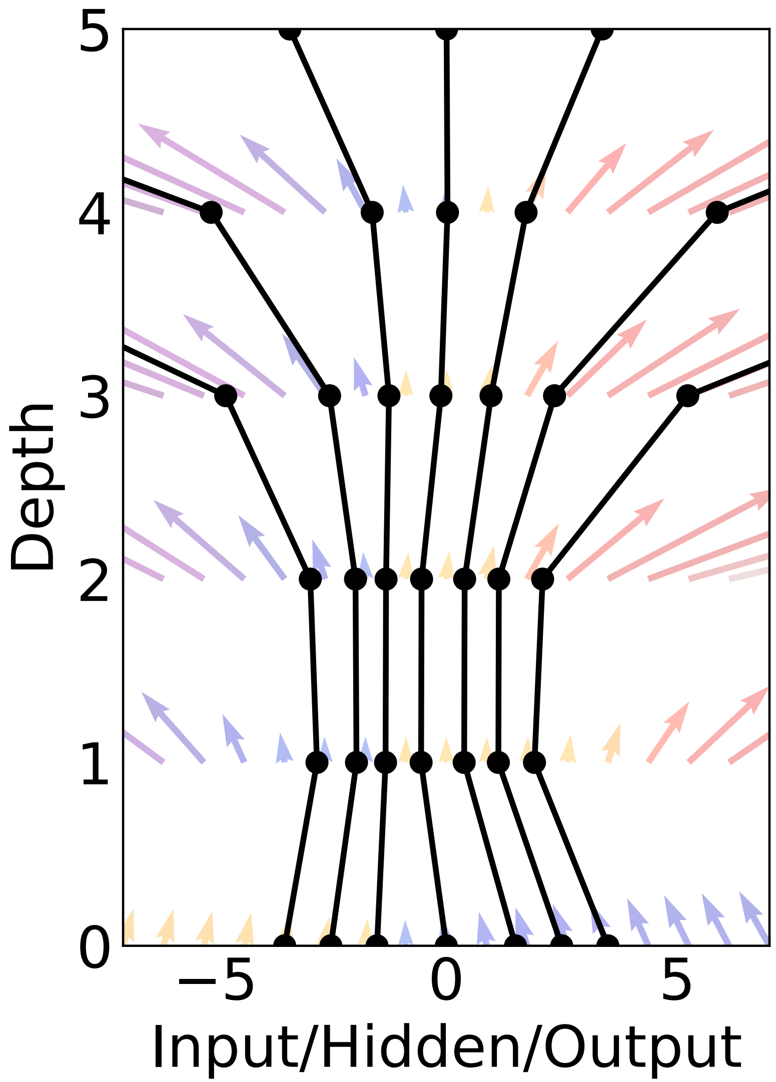
$$\mathbf{z}_{l+1} = \mathbf{z}_l + f_\theta(\mathbf{z}_l, l)$$

LAYER INDEX → TIME

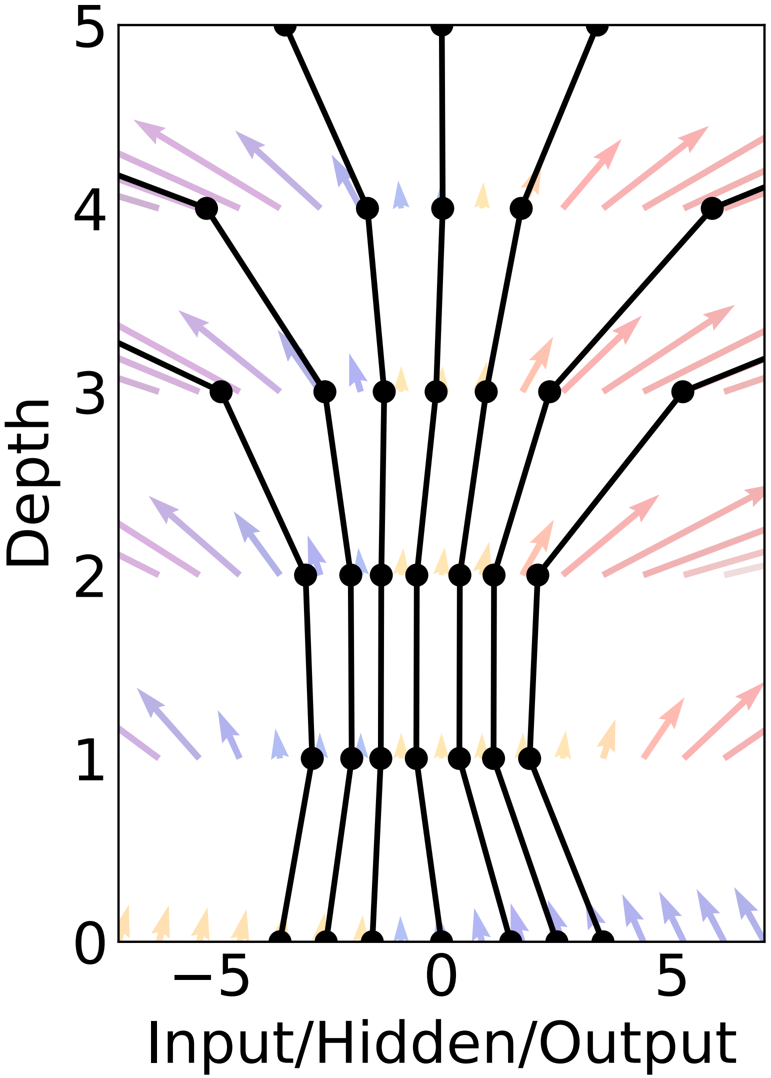
ODE NETWORK



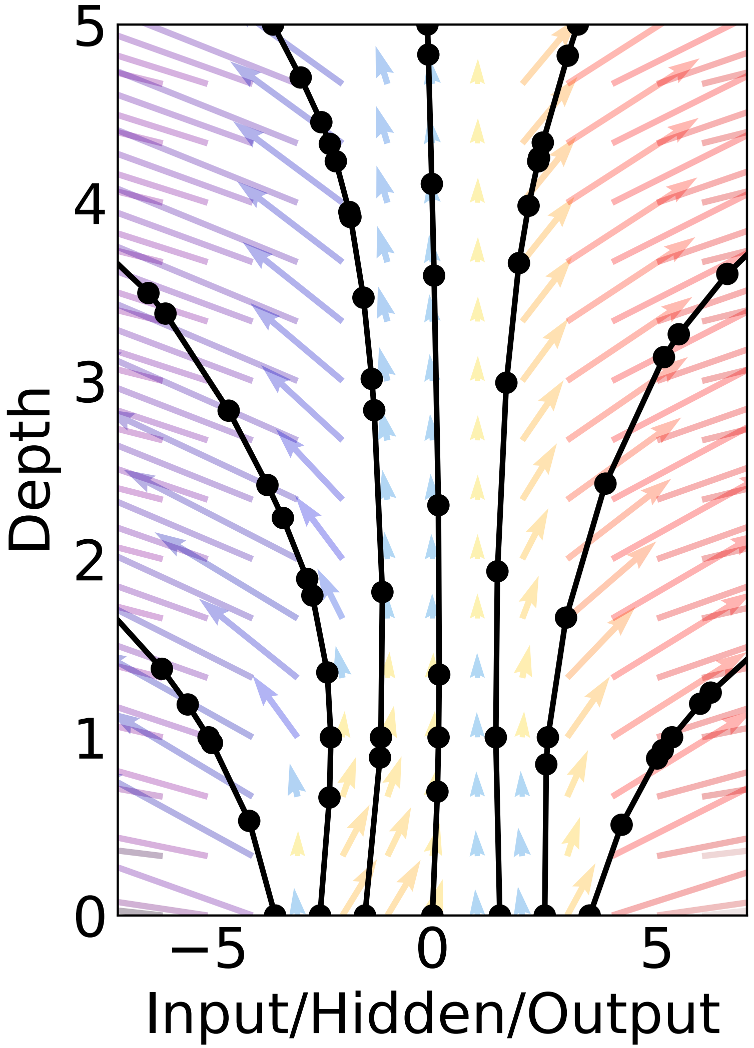
ODE NETWORK



ODE NETWORK



Depth
Input/Hidden/Output
0
1
2
3
4
5
−5
0
5



Depth
Input/Hidden/Output
0
1
2
3
4
5
−5
0
5

NEURALODE DEFINITION

TRAINING NEURALODE

OPTIMISATION PROBLEM

$$\arg\min_{\theta} \mathbb{E}_{\mathbf{x}\in X,\mathbf{y}\in Y} L\left(\mathbf{y}, \mathrm{ODESOLVE}(\mathbf{x}, f, t_0, t_1, \theta)\right)$$

OPTIMISATION PROBLEM

$$\arg\min_{\theta} \mathbb{E}_{\mathbf{x}\in X,\mathbf{y}\in Y} L\left(\mathbf{y}, \mathrm{ODESOLVE}(\mathbf{x}, f, t_0, t_1, \theta)\right)$$

Can we backprop?

OPTIMISATION PROBLEM

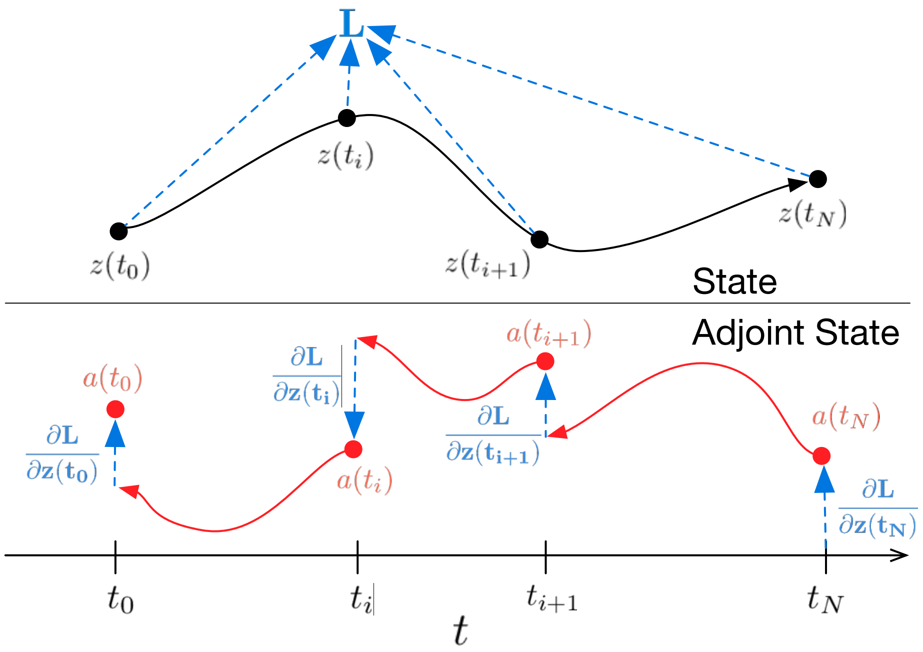
$$\arg\min_{\theta} \mathbb{E}_{\mathbf{x}\in X,\mathbf{y}\in Y} L\left(\mathbf{y}, \mathrm{ODESOLVE}(\mathbf{x}, f, t_0, t_1, \theta)\right)$$

Can we backprop?

Yes, but for $O\left(N_{\mathrm{steps}}\right)$ memory

MEET ADJOINT

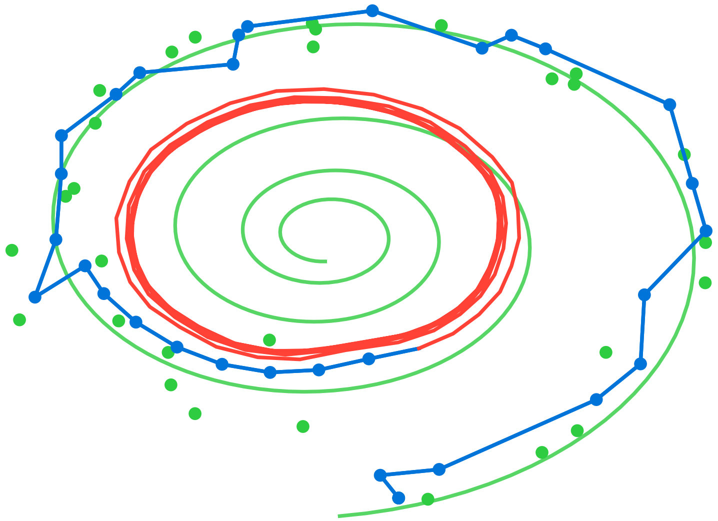
ADJOINT WITH TRAJECTORY LOSS

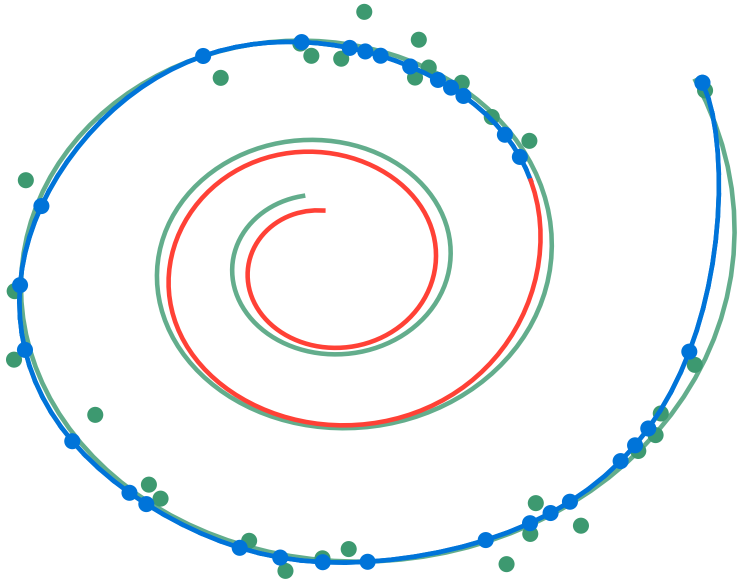


NEURALODE VS RNN

For time series

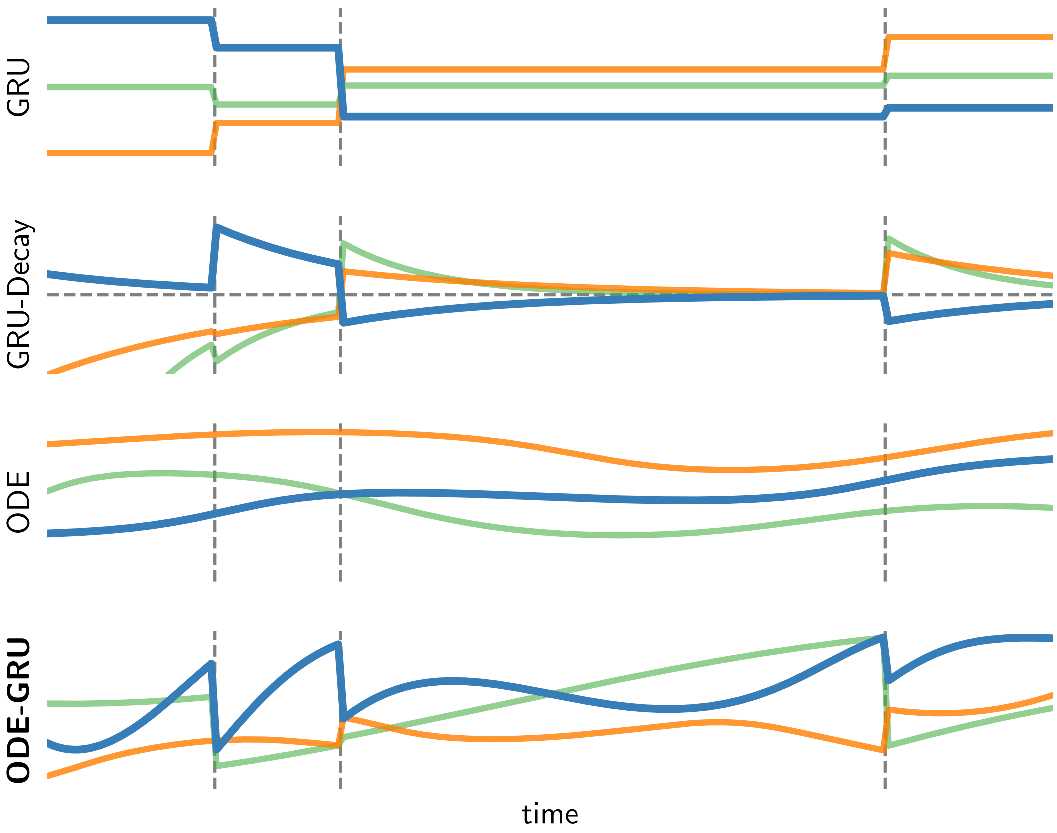
- Native support of irregular time intervals
- Smooth trajectory





NEURAL ODE IN A GENERATIVE MODEL

TIME SERIES



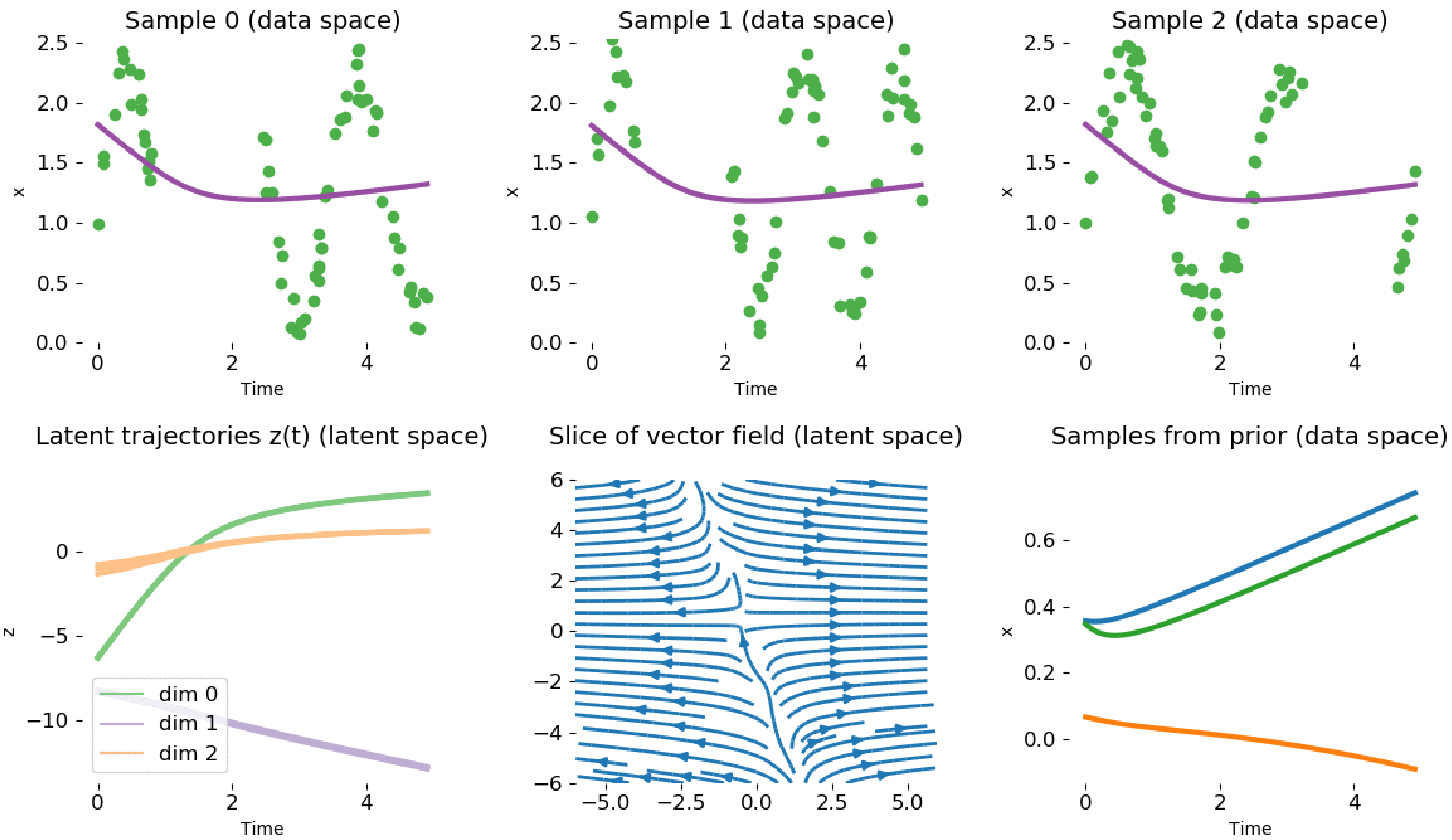
Hidden space trajectories

ODE-RNN

Algorithm 1 The ODE-RNN. The only difference, highlighted in blue, from standard RNNs is that the pre-activations h' evolve according to an ODE between observations, instead of being fixed.

Input: Data points and their timestamps $\{(x_i, t_i)\}_{i=1..N}$
$h_0 = \mathbf{0}$
for i in 1, 2, …, N **do**
  $h'_i = \mathrm{ODESolve}(f_\theta, h_{i-1}, (t_{i-1}, t_i))$ ▷ Solve ODE to get state at t_i
  $h_i = \mathrm{RNNCell}(h'_i, x_i)$ ▷ Update hidden state given current observation x_i
end for
$o_i = \mathrm{OutputNN}(h_i)$ for all $i = 1..N$
Return: $\{o_i\}_{i=1..N}; h_N$

NOISY DATA



Sample 0 (data space)
Sample 1 (data space)
Sample 2 (data space)
2.5
2.0
1.5
1.0
0.5
0.0
x
0
2
4
Time
Latent trajectories z(t) (latent space)
0
−5
−10
z
dim 0
dim 1
dim 2
Slice of vector field (latent space)
6
4
2
0
−2
−4
−6
−5.0
−2.5
0.0
2.5
5.0
Samples from prior (data space)
0.6
0.4
0.2
0.0

LATENT ODE AKA ODE-VAE

$$z_0 \sim p(z_0)$$

$$z_0, z_1, \ldots, z_N = \mathrm{ODESolve}(f_\theta, z_0, (t_0, t_1, \ldots, t_N))$$

$$x_i \overset{\text{indep.}}{\sim} p(x_i | z_i); i = 0, 1, \ldots, N$$

STEP 1: ESTIMATE POSTERIOR FOR Z_0

$$q(z_0 | \{x_i, t_i\}_{i=0}^{N}) = \mathcal{N}(\mu_{z_0}, \sigma_{z_0})$$

$$\mu_{z_0}, \sigma_{z_0} = g_\phi(\text{ODE-RNN}_\phi(\{x_i, t_i\}_{i=0}^{N}))$$

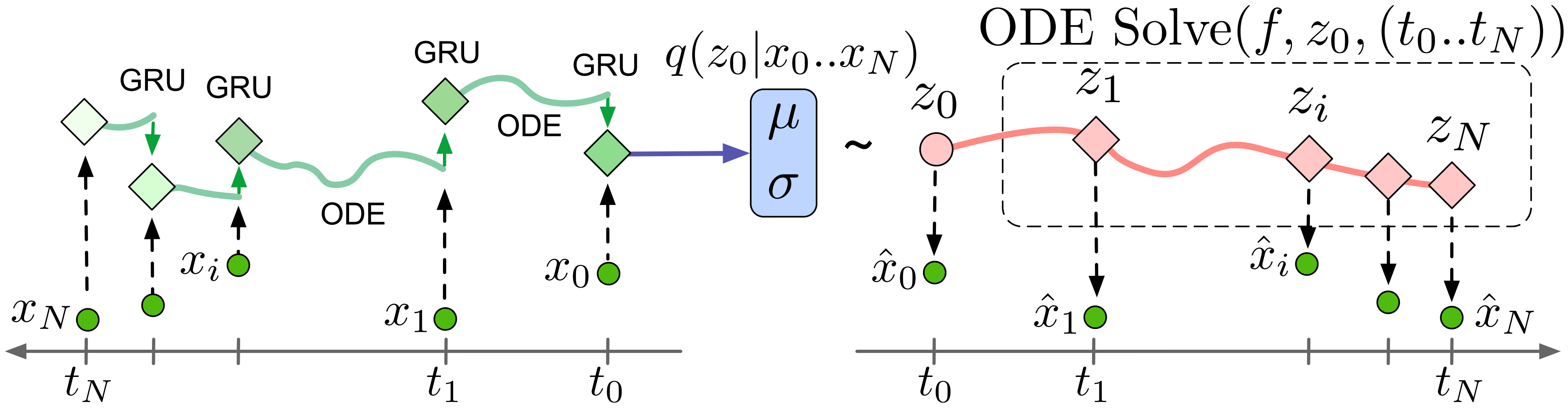
STEP 2: LATENT DYNAMICS

$$\text{ODESolve}(f_\theta, z_0, (t_0, t_1, \ldots, t_N))$$

STEP 3: PREDICT

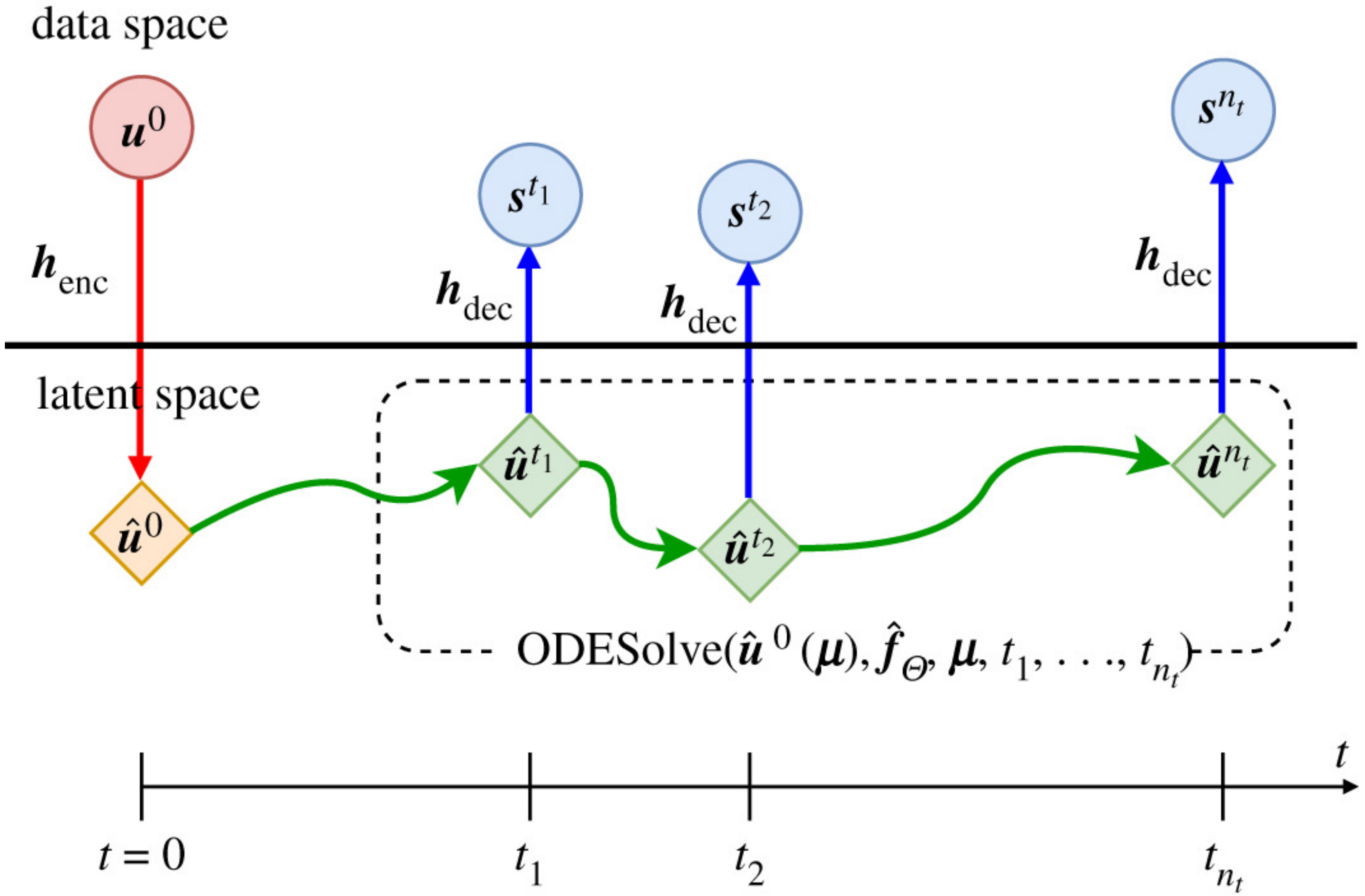
Sample $p(x_i|z_i)$

LATENT ODE



GRU
GRU
GRU
GRU
ODE
ODE
$q(z_0|x_0..x_N)$
μ
σ
~
ODE Solve$(f, z_0, (t_0..t_N))$
z_0
z_1
z_i
z_N
x_N
x_i
x_1
x_0
$\hat{x}_0$
$\hat{x}_1$
$\hat{x}_i$
$\hat{x}_N$
t_N
t_1
t_0
t_0
t_1
t_N

PHYSICS: INITIAL STATE IS FULL STATE



LAST TOUCH: PARAMETERS

$$\frac{d\mathbf{z}}{dt} = f_\theta(\mathbf{z}(t), t, \mu)$$

SUMMARY

SUMMARY

- Neural ODEs are a natural model for modeling processes continuous in time

SUMMARY

- Neural ODEs are a natural model for modeling processes continuous in time
- In physics, a lot of processes also are subjects to conservation laws - see the next lecture

SUMMARY

- Neural ODEs are a natural model for modeling processes continuous in time
- In physics, a lot of processes also are subjects to conservation laws - see the next lecture
- NeuralODE can be used as a general-purpose supervised model, but is not necessarily superior in this role

REFERENCES

- Chen, Ricky TQ, et al. "Neural ordinary differential equations." Advances in neural information
- Rubanova, Yulia, Ricky TQ Chen, and David K. Duvenaud. "Latent ordinary differential equatio
- Lee, Kookjin, and Eric J. Parish. "Parameterized neural ordinary differential equations: Applica
- Interactive visualisations by Oleg Danilov
- Our textbook on NeuralODE

QUESTIONS?