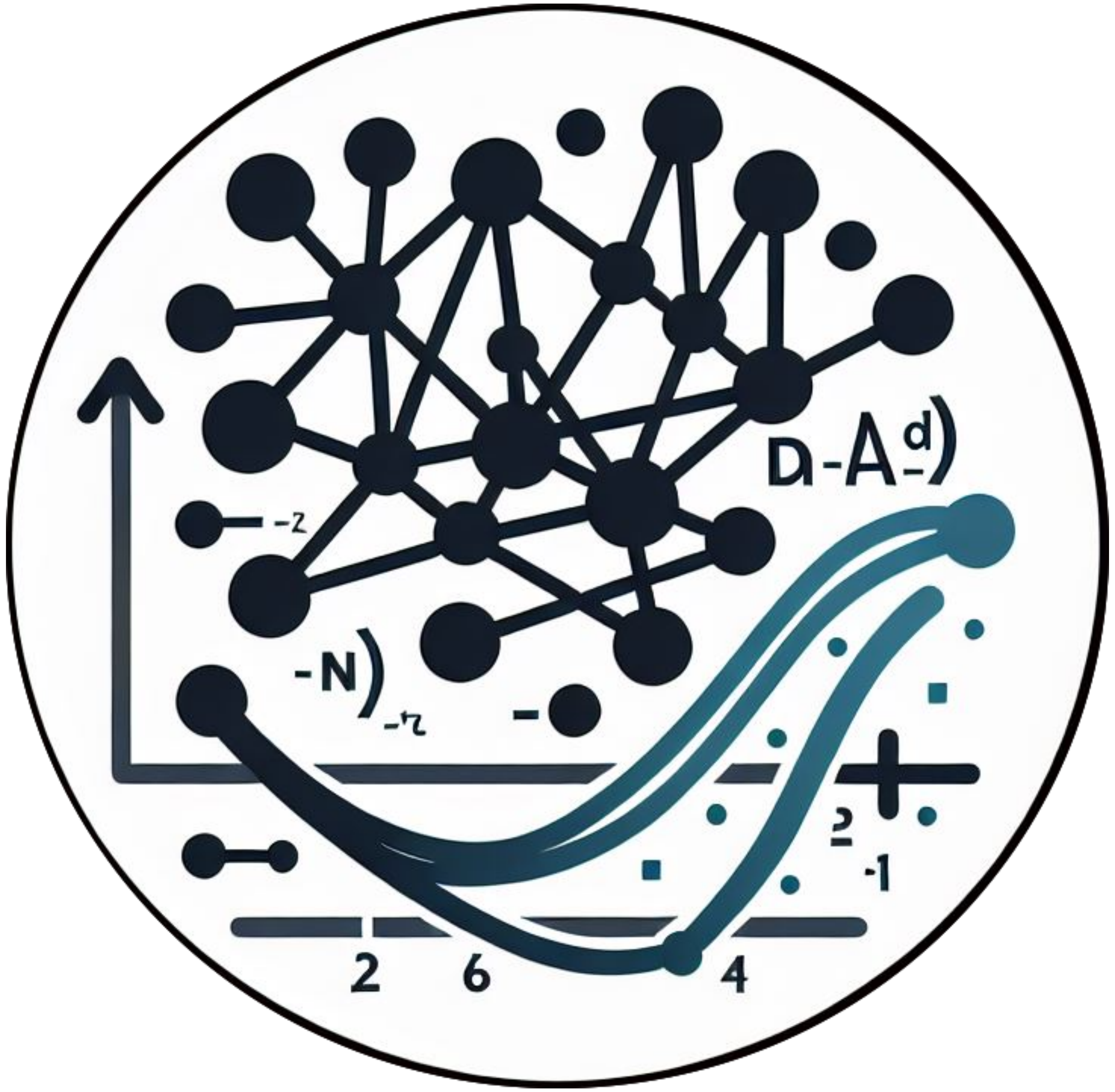
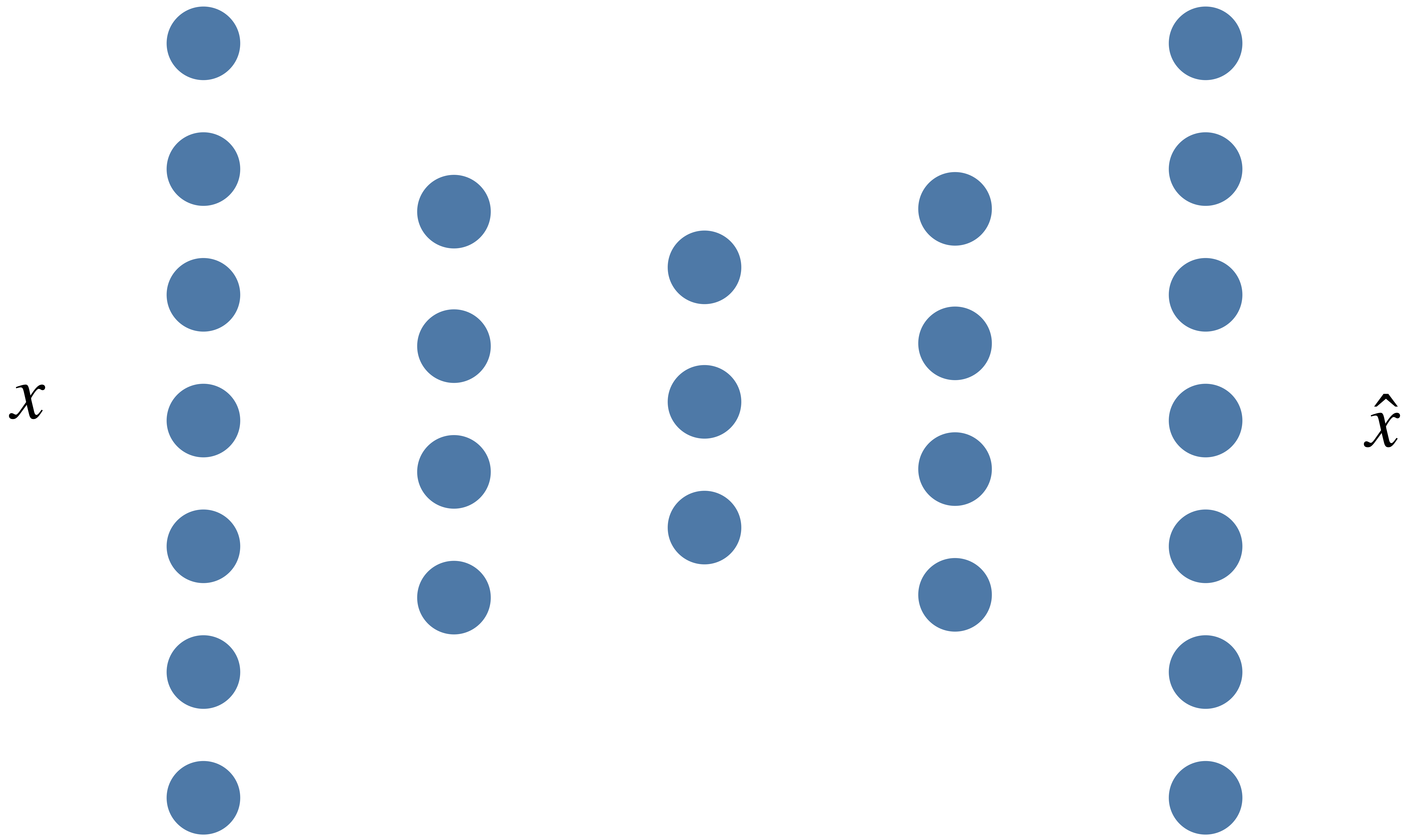
Autoencoders

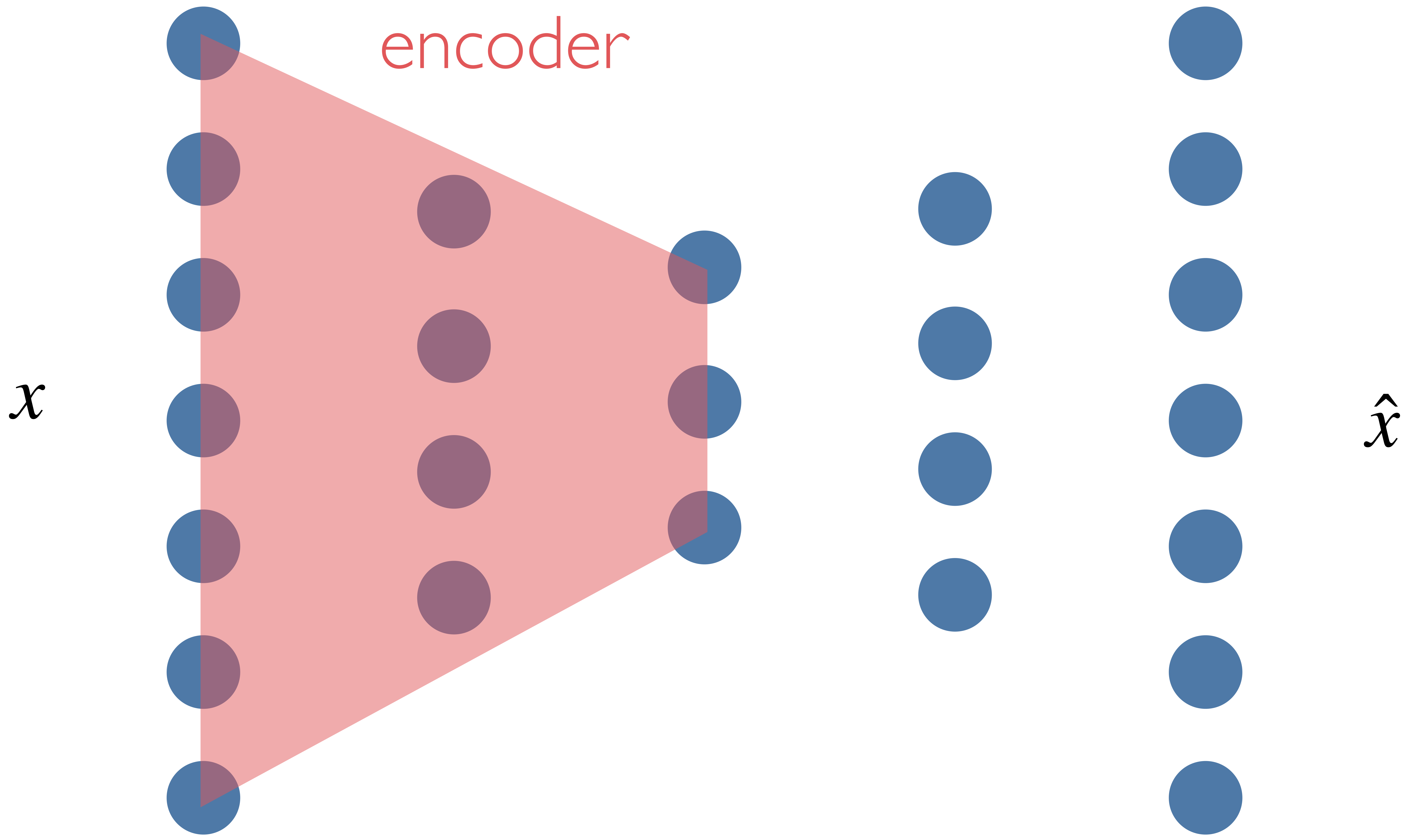


Deep Learning for Engineers
Andrew Ning
aning@byu.edu

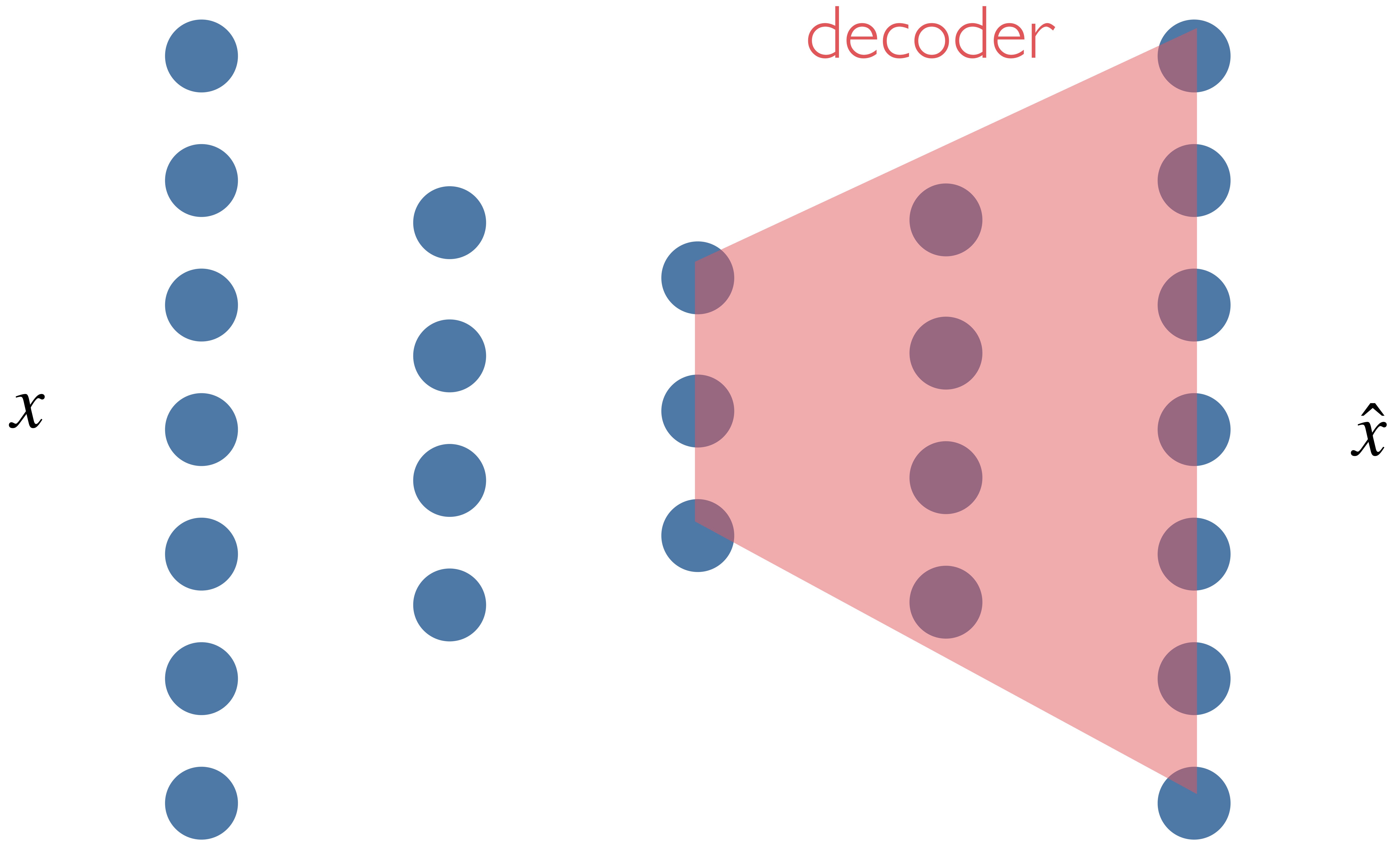
Autoencoder



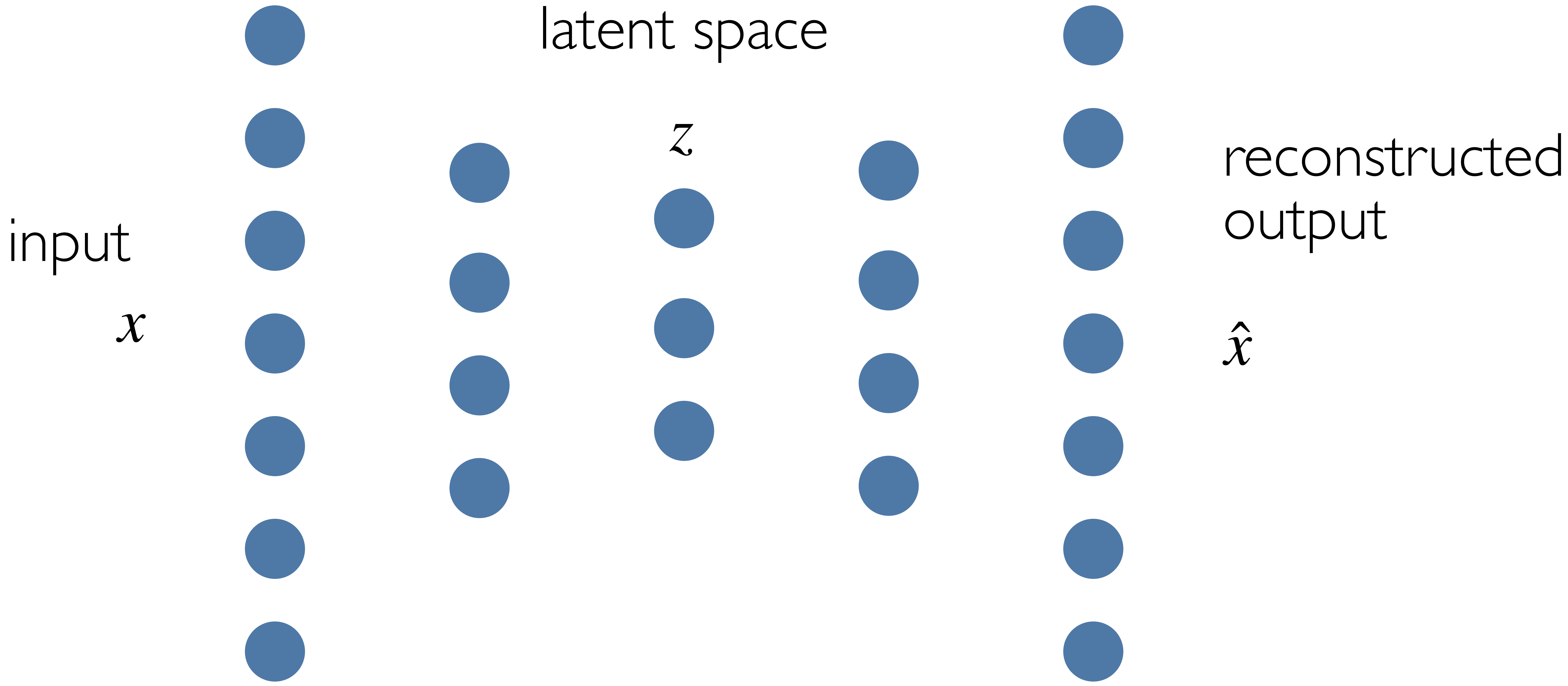
Autoencoder



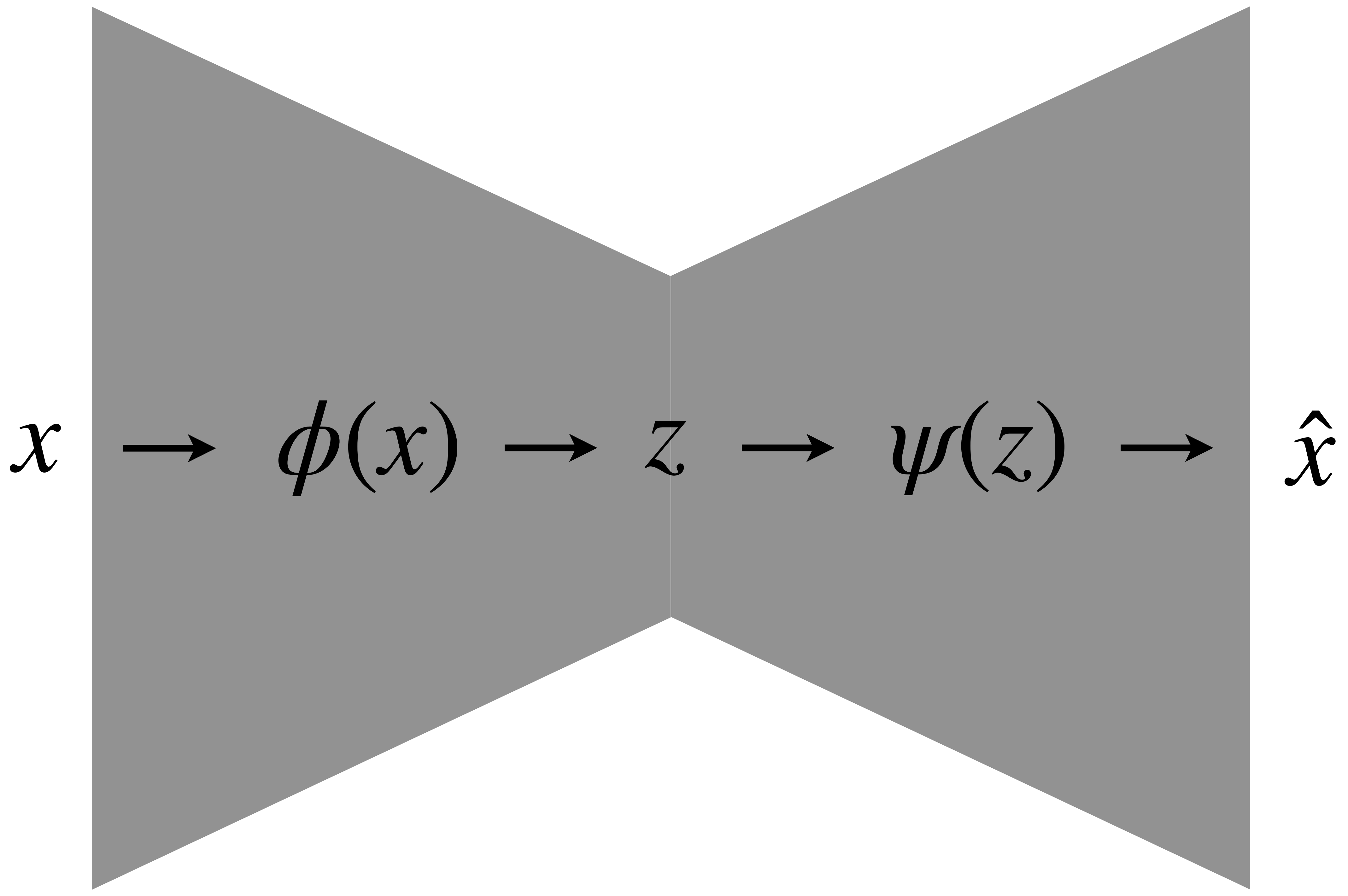
Autoencoder



Autoencoder



Autoencoder

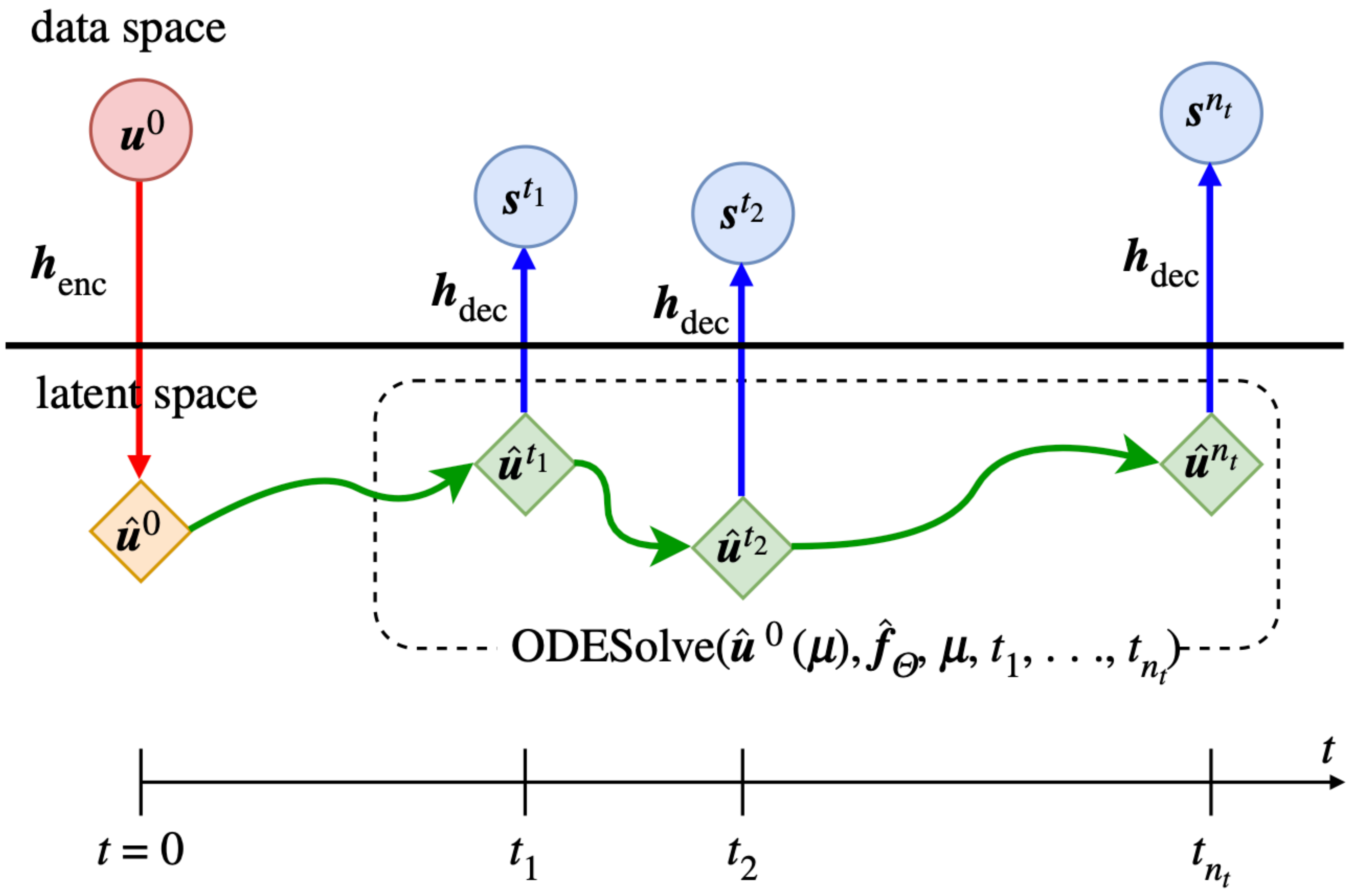


$$z = \phi(x) \qquad \hat{x} = \psi(z)$$

$$\hat{x} = \psi(\phi(x))$$

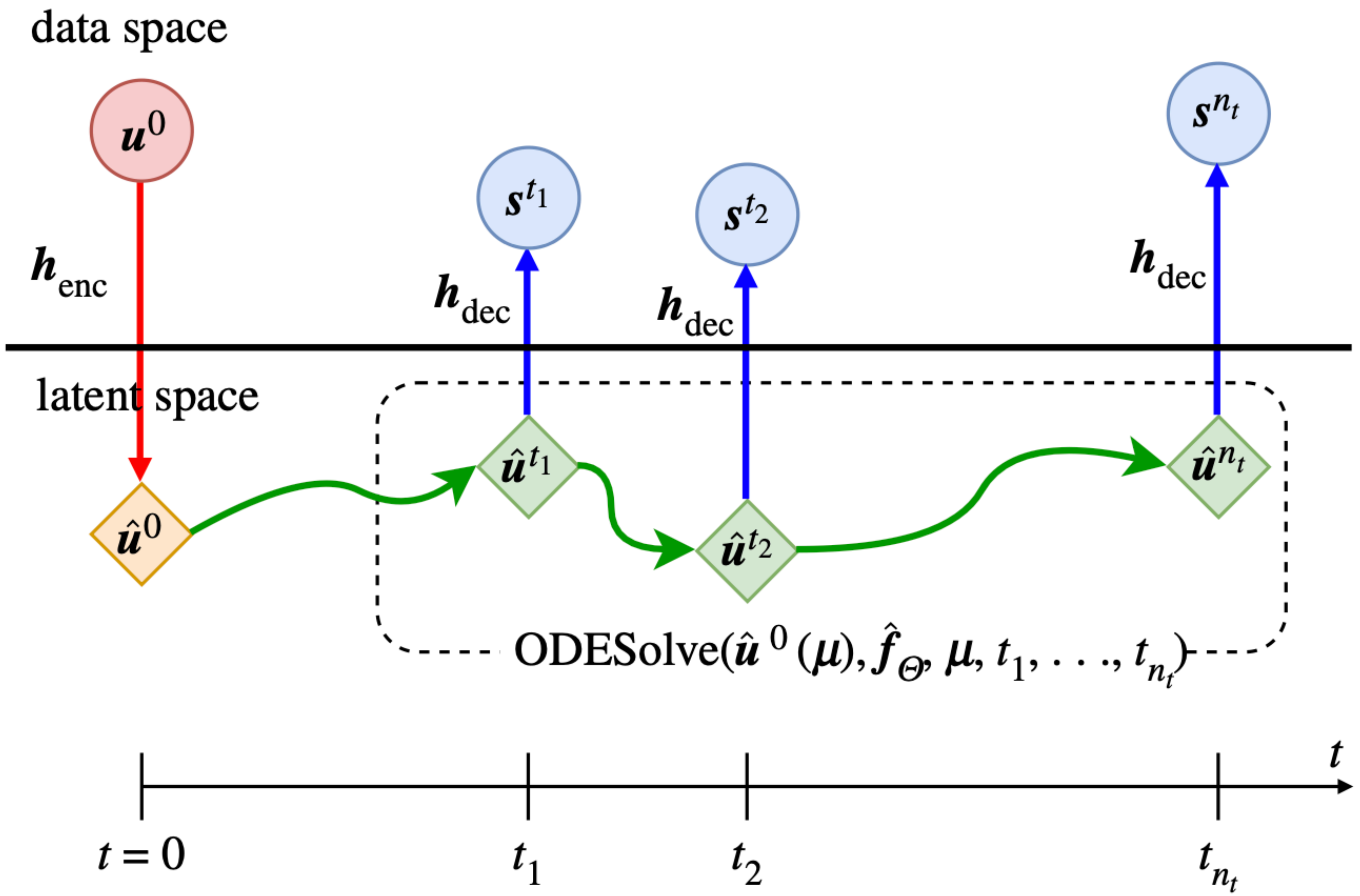
$$\phi, \psi = \underset{\theta}{\operatorname{argmin}} \|x - \psi(\phi(x))\|$$

Autoencoder with Neural ODE



Parameterized neural ordinary differential equations: applications to computational physics problems
Kookjin Lee and Eric J. Parish

Autoencoder with Neural ODE



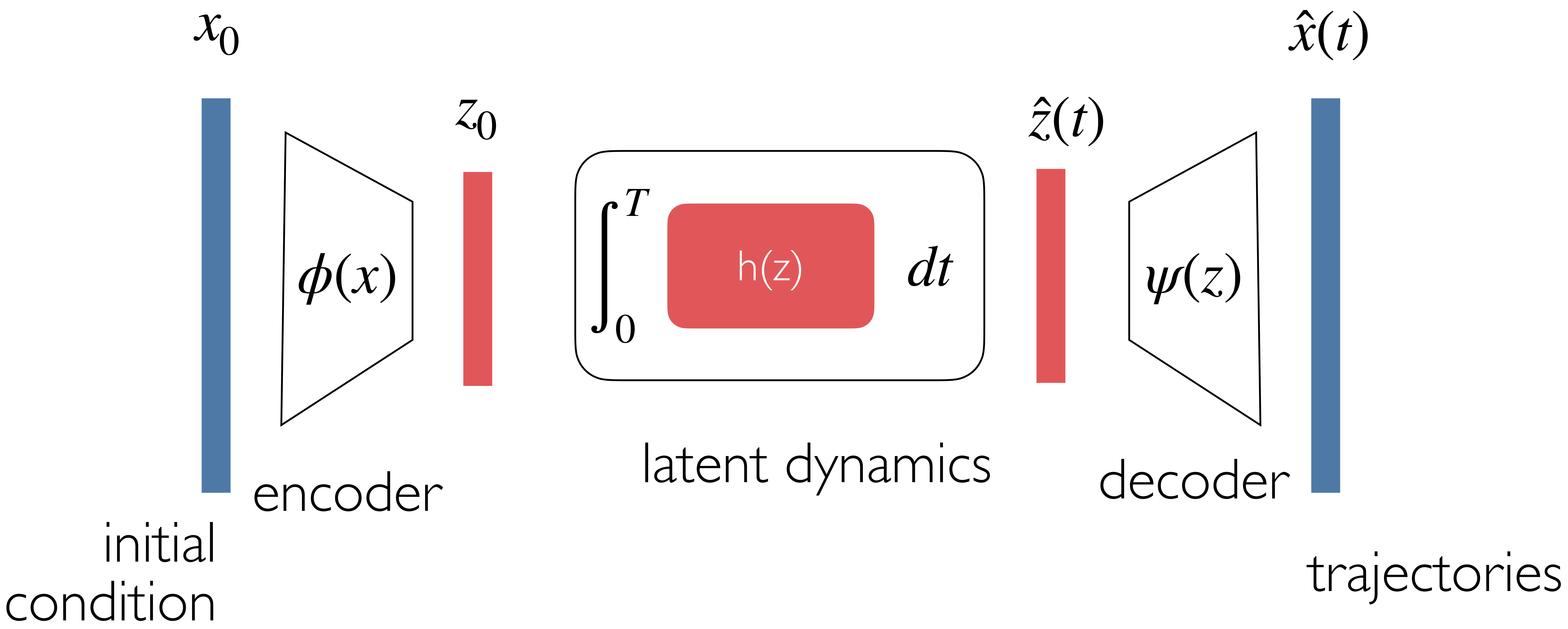
$$\hat{u}^0 = h_{enc}(u^0)$$

$$\hat{u} = \text{odesolve}(\hat{u}^0)$$

$$u = h_{dec}(\hat{u})$$

Parameterized neural ordinary differential equations: applications to computational physics problems
Kookjin Lee and Eric J. Parish

Neural ODE with autoencoder



Neural ODE with autoencoder

data loss		prediction loss		reconstruction loss
$L_{data} =$		$\|x - \psi(\text{odeint}(\phi(x_0)))\|^2$	$+$	$\|x - \psi(\phi(x))\|^2$

Code Outline

nbatches nt nx

```
z0 = encode(x[:, 0, :])
```

Code Outline (psuedocode)

```
z0 = encode(x[:, 0, :])
zhat = odeint(odefunc, z0)
```

Code Outline (psuedocode)

```
z0 = encode(x[:, 0, :])
zhat = odeint(odefunc, z0)
xhat = decode(zhat)
```

Code Outline (psuedocode)

```
z0 = encode(x[:, 0, :])
zhat = odeint(odefunc, z0)
xhat = decode(zhat)

loss_data = mse(xhat, x)
```

Code Outline (psuedocode)

```
z0 = encode(x[:, 0, :])
zhat = odeint(odefunc, z0)
xhat = decode(zhat)

loss_data = mse(xhat, x)
loss_recon = mse(x, decode(encode(x)))
```

Physics informed loss

www.nature.com/scientificreports

scientific reports

Check for updates

OPEN

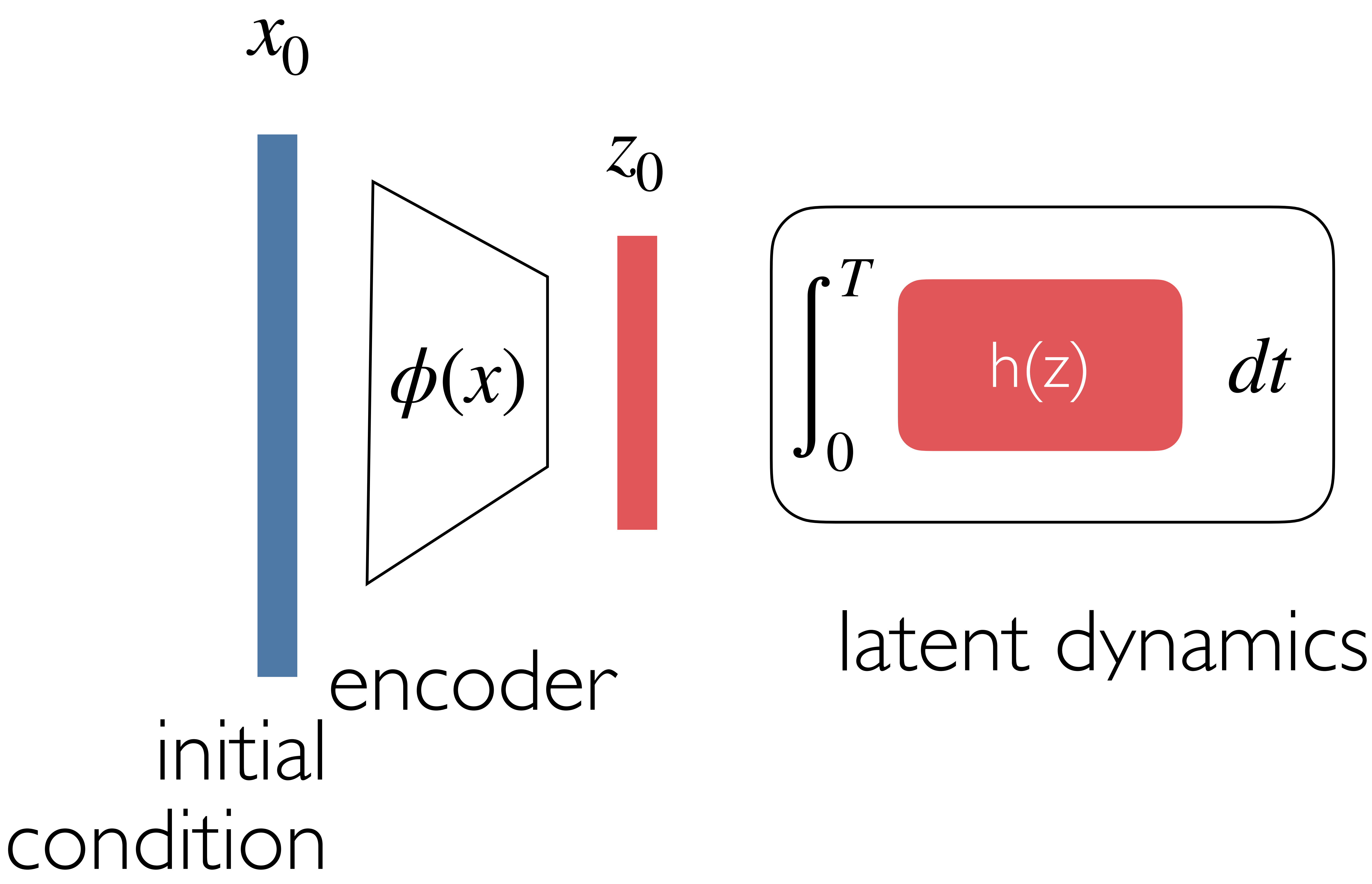
Physics-informed neural ODE (PINODE): embedding physics into models using collocation points

Aleksei Sholokhov[1], Yuying Liu[1], Hassan Mansour[2]✉ & Saleh Nabi[2]

Building reduced-order models (ROMs) is essential for efficient forecasting and control of complex dynamical systems. Recently, autoencoder-based methods for building such models have gained significant traction, but their demand for data limits their use when the data is scarce and expensive.

Physics informed loss

$$\frac{dx}{dt} = f(x) \qquad z = \phi(x)$$



Physics informed loss

$$\frac{dx}{dt} = f(x) \qquad z = \phi(x)$$

$$\frac{dz}{dt} = \frac{dz}{dx}\frac{dx}{dt} = \frac{d\phi}{dx}f(x)$$

Physics informed loss

$$\int_0^T h(z) \, dt$$

$$\frac{dx}{dt} = f(x) \qquad z = \phi(x)$$

$$\frac{dz}{dt} = h(z)$$

latent dynamics

$$\frac{dz}{dt} = \frac{dz}{dx}\frac{dx}{dt} = \frac{d\phi}{dx} f(x)$$

Physics informed loss

$$\frac{dx}{dt} = f(x) \qquad z = \phi(x)$$

$$\frac{dz}{dt} = h(z)$$

$$\frac{dz}{dt} = \frac{dz}{dx}\frac{dx}{dt} = \frac{d\phi}{dx} f(x)$$

$$\frac{dz}{dt} = h(z) = h(\phi(x))$$

Physics informed loss

$$\frac{dx}{dt} = f(x) \qquad z = \phi(x)$$

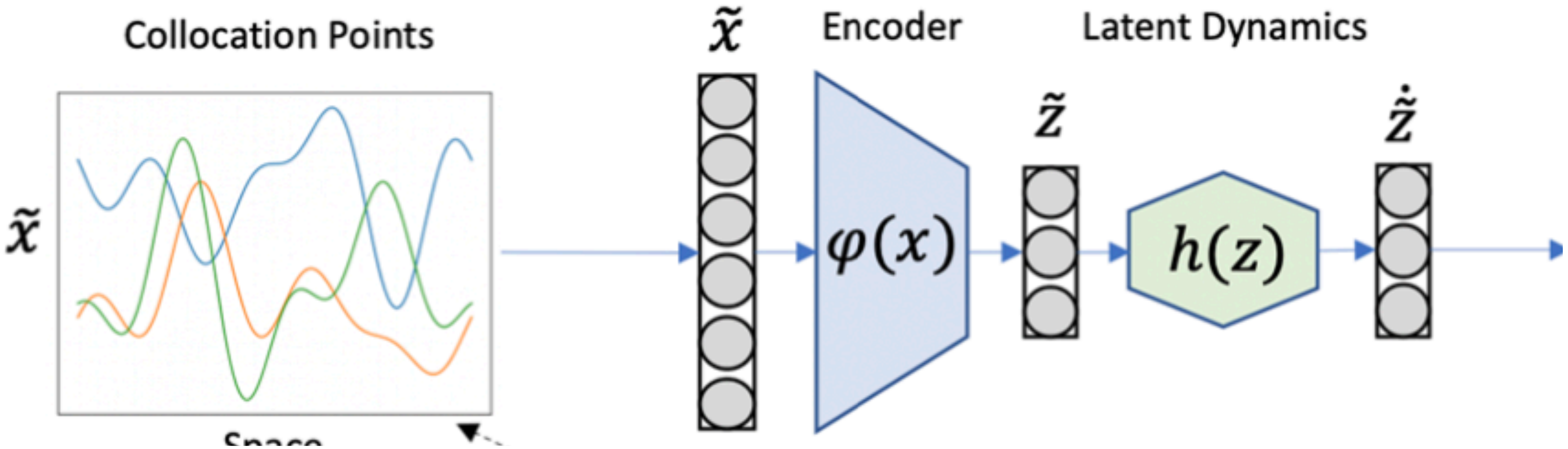
$$\frac{dz}{dt} = h(z)$$

$$\frac{dz}{dt} = \frac{dz}{dx}\frac{dx}{dt} = \frac{d\phi}{dx}f(x)$$

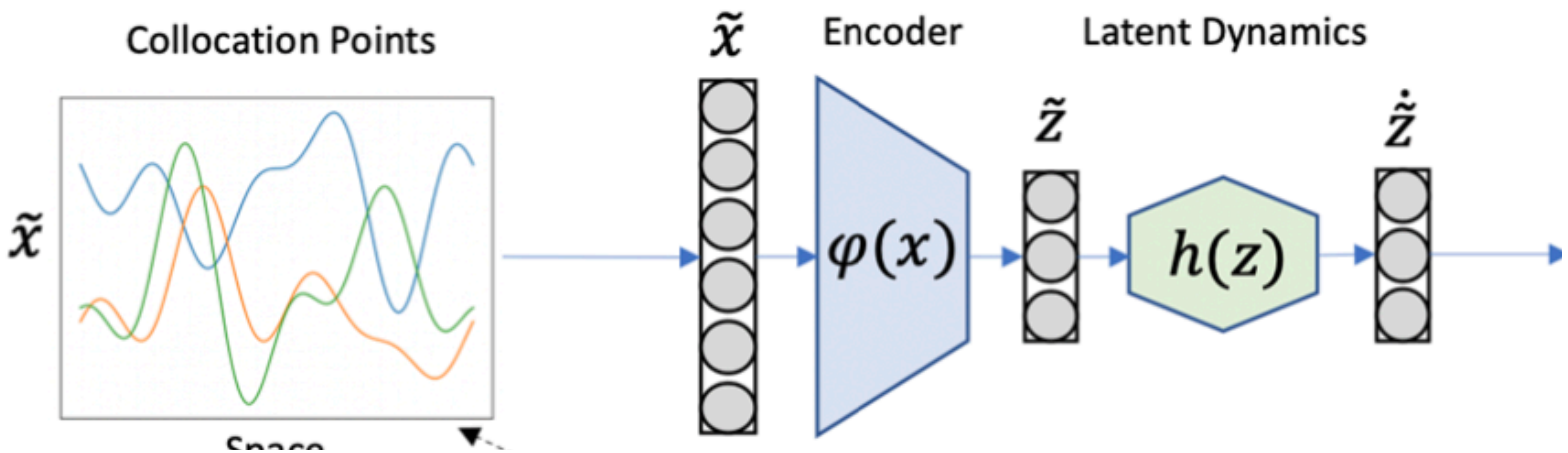
$$\frac{dz}{dt} = h(z) = h(\phi(x))$$

$$\frac{d\phi}{dx}f(x) = h(\phi(x))$$

Physics-informed neural ODE (PINODE): embedding physics into models using collocation points
Aleksei Sholokhov, Yuying Liu, Hassan Mansour, Saleh Nabi



Physics-informed neural ODE (PINODE): embedding physics into models using collocation points
Aleksei Sholokhov, Yuying Liu, Hassan Mansour, Saleh Nabi



physics loss | latent gradient loss | collocation reconstruction loss

$$L_{physics} = \quad \|\dot{\tilde{z}} - \frac{d\phi}{d\tilde{x}} f(\tilde{x})\|^2 \quad + \quad \|\tilde{x} - \psi(\phi(\tilde{x}))\|^2$$

```
z_col = encode(x_col)
```

```
z_col = encode(x_col)
zdot_col = odefunc(0.0, z_col)
```

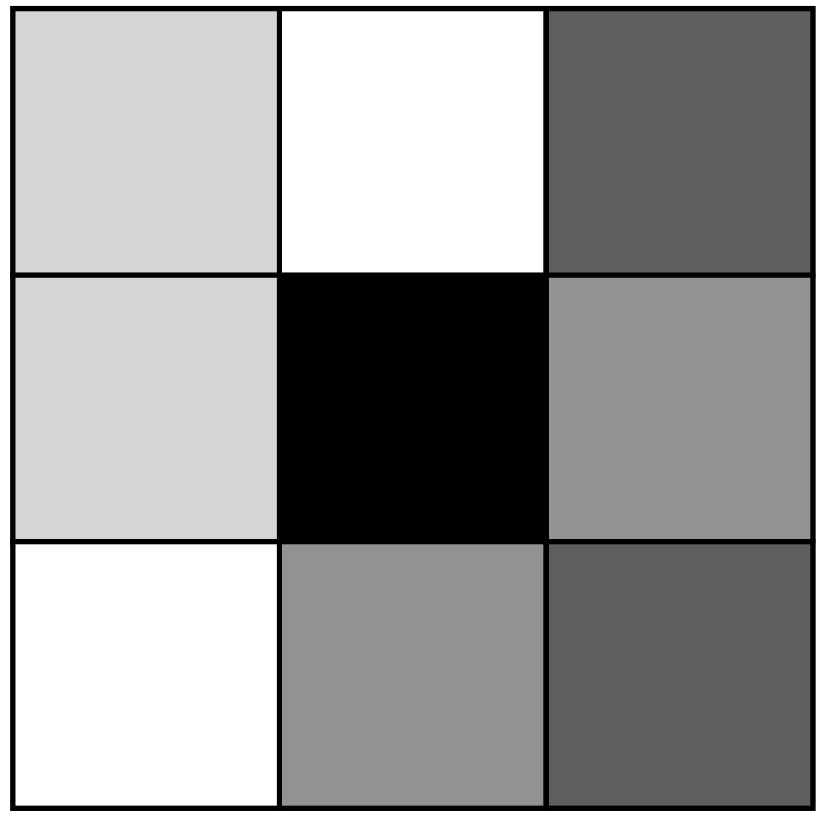
```
z_col = encode(x_col)
zdot_col = odefunc(0.0, z_col)

loss_physics = mse(zdot_col, dzdt)
loss_recon_2 = mse(x_col, decode(encode(x_col)))
```

Demo of derivatives

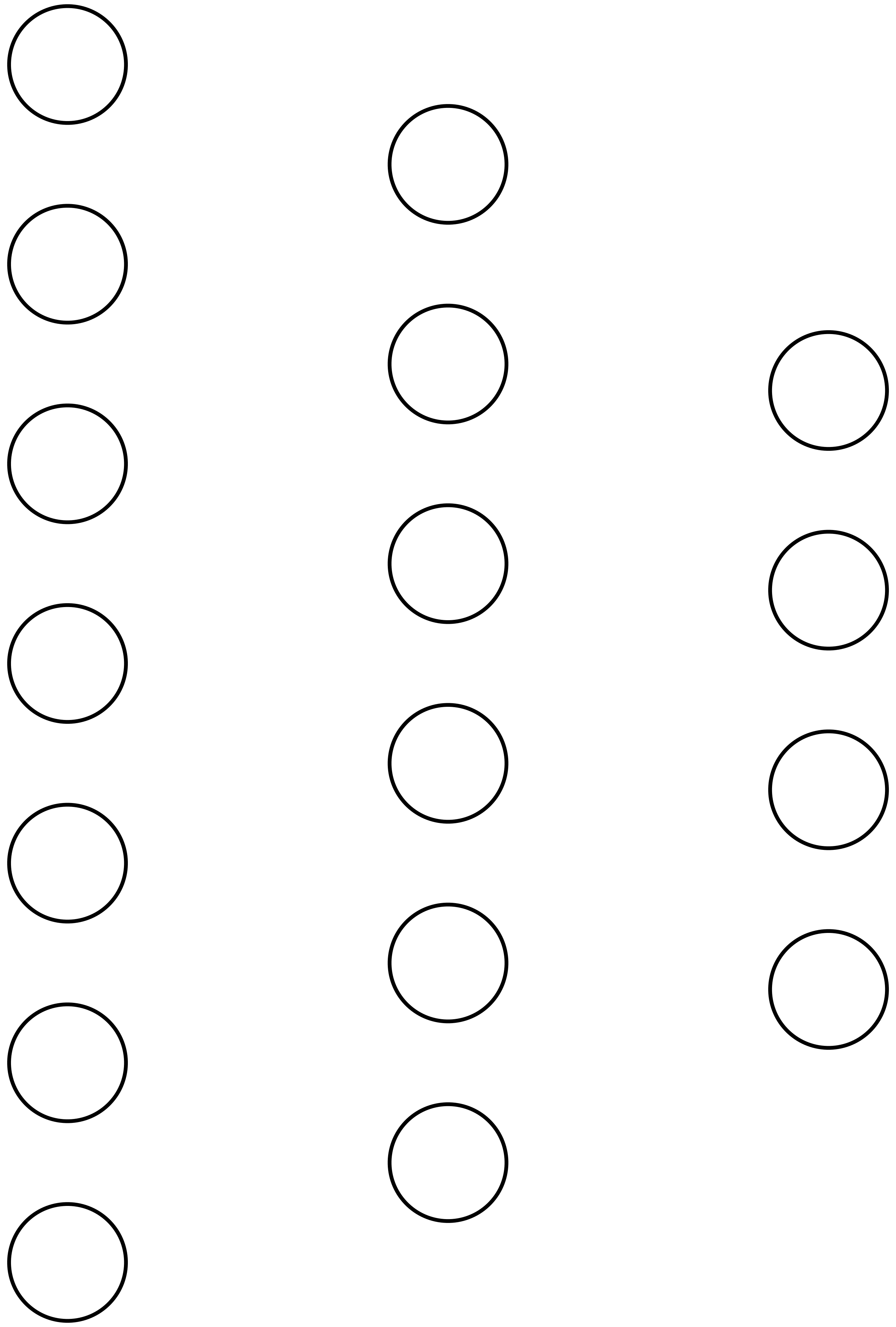
$$\frac{dz}{dt} = \frac{dz}{dx}\frac{dx}{dt} = \frac{d\phi}{dx}f(x)$$

Train PyTorch model on Google Colab GPUs



.75	1	.25
.75	0	.5
1	.5	.25

.75
1
.25
.75
0
.5
1
.5
.25



https://pytorch.org/tutorials/beginner/basics/quickstart_tutorial.html

Optional section:

a more advanced dive into reverse mode AD

Reverse Mode AD

$$\bar{\xi} = \frac{d\ell}{d\xi}$$

Reverse Mode AD

$$\bar{\xi} = \frac{d\ell}{d\xi}$$

$$\bar{x} = \frac{d\ell}{dx} = \frac{d\ell}{dy}\frac{dy}{dx}$$

Reverse Mode AD

$$\bar{\xi} = \frac{d\ell}{d\xi}$$

$$\bar{x} = \frac{d\ell}{dx} = \frac{d\ell}{dy}\frac{dy}{dx}$$

$$\bar{x}^T = \bar{y}^T J$$

Reverse Mode AD

$$\bar{\xi} = \frac{d\ell}{d\xi}$$

$$\bar{x} = \frac{d\ell}{dx} = \frac{d\ell}{dy}\frac{dy}{dx}$$

$$\bar{x}^T = \bar{y}^T J$$

$$\begin{bmatrix} \bar{y}_1 & \bar{y}_2 \end{bmatrix} \begin{bmatrix} \frac{dy_1}{dx_1} & \frac{dy_1}{dx_2} & \frac{dy_1}{dx_3} \\ \frac{dy_2}{dx_1} & \frac{dy_2}{dx_2} & \frac{dy_2}{dx_3} \end{bmatrix}$$

$$\begin{bmatrix} \bar{y}_1 & \bar{y}_2 \end{bmatrix} \begin{bmatrix} \frac{dy_1}{dx_1} & \frac{dy_1}{dx_2} & \frac{dy_1}{dx_3} \\ \frac{dy_2}{dx_1} & \frac{dy_2}{dx_2} & \frac{dy_2}{dx_3} \end{bmatrix}$$

$$\begin{bmatrix}\bar{y}_1 & \bar{y}_2\end{bmatrix}\begin{bmatrix}\frac{dy_1}{dx_1} & \frac{dy_1}{dx_2} & \frac{dy_1}{dx_3} \\ \frac{dy_2}{dx_1} & \frac{dy_2}{dx_2} & \frac{dy_2}{dx_3}\end{bmatrix}$$

$$\begin{bmatrix}\bar{y}\end{bmatrix}\begin{bmatrix}\frac{dy}{dx_1} & \frac{dy}{dx_2} & \frac{dy}{dx_3}\end{bmatrix}$$

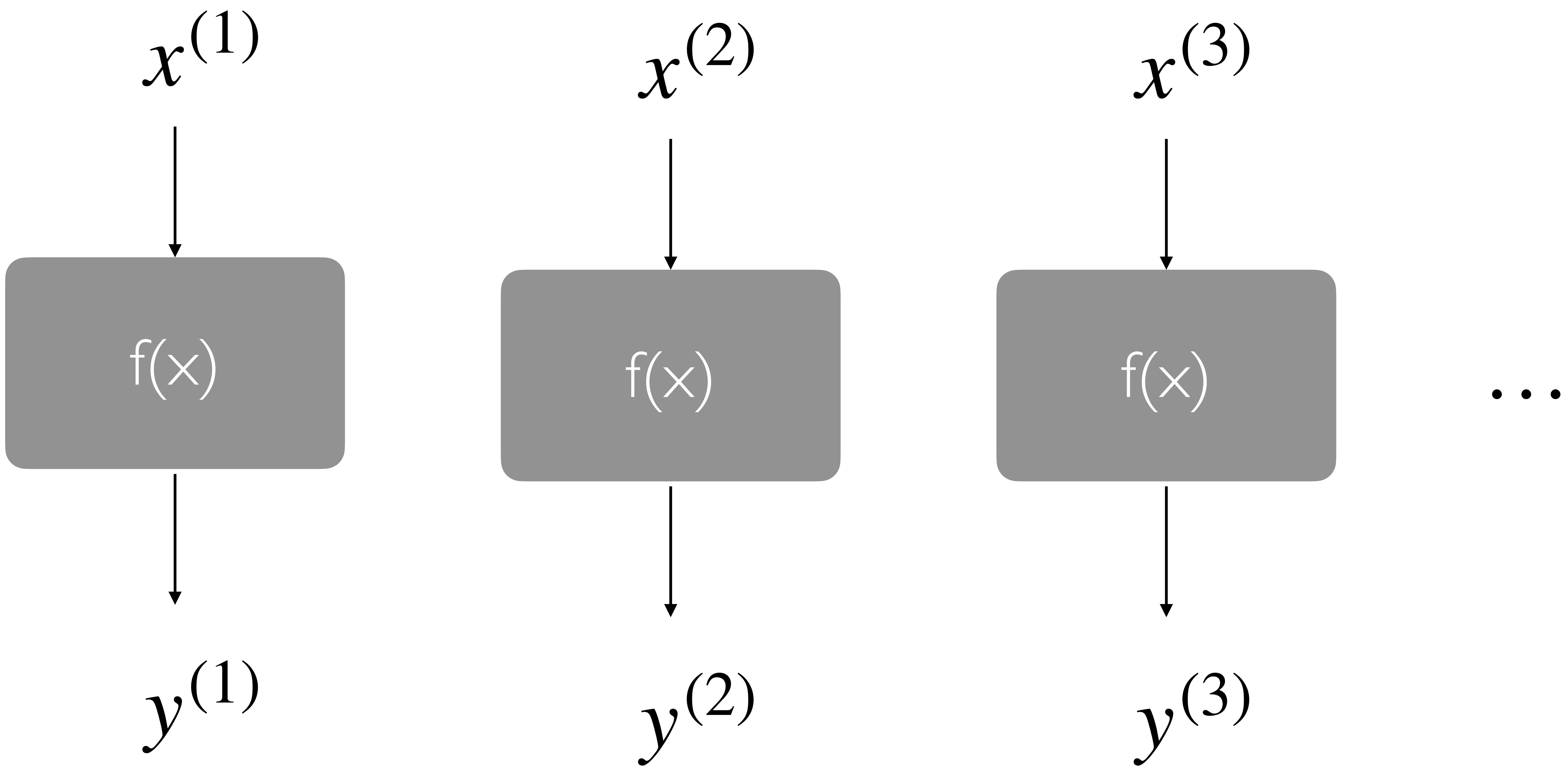
$$[\bar{y}_1 \quad \bar{y}_2] \begin{bmatrix} \frac{dy_1}{dx_1} & \frac{dy_1}{dx_2} & \frac{dy_1}{dx_3} \\ \frac{dy_2}{dx_1} & \frac{dy_2}{dx_2} & \frac{dy_2}{dx_3} \end{bmatrix}$$

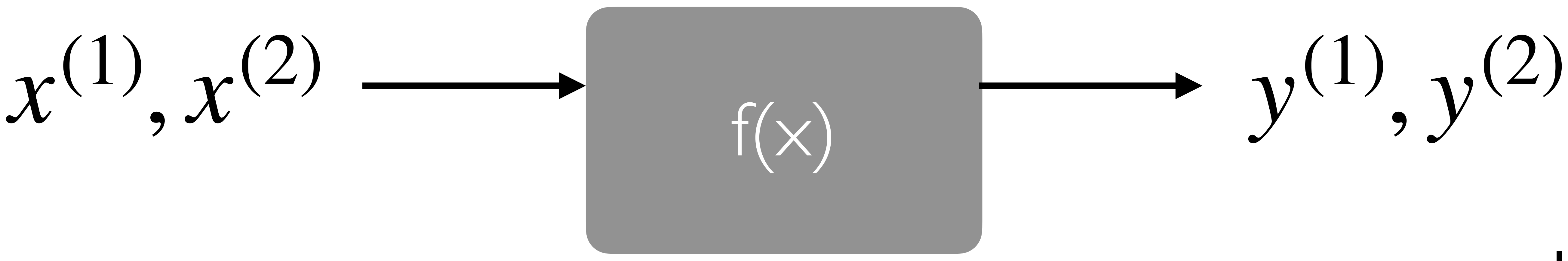
$$[\bar{y}] \begin{bmatrix} \frac{dy}{dx_1} & \frac{dy}{dx_2} & \frac{dy}{dx_3} \end{bmatrix}$$

$$[1 \quad 0] \begin{bmatrix} \frac{dy_1}{dx_1} & \frac{dy_1}{dx_2} & \frac{dy_1}{dx_3} \\ \frac{dy_2}{dx_1} & \frac{dy_2}{dx_2} & \frac{dy_2}{dx_3} \end{bmatrix}$$

$$[0 \quad 1] \begin{bmatrix} \frac{dy_1}{dx_1} & \frac{dy_1}{dx_2} & \frac{dy_1}{dx_3} \\ \frac{dy_2}{dx_1} & \frac{dy_2}{dx_2} & \frac{dy_2}{dx_3} \end{bmatrix}$$

Separate data points





let’s say y is a scalar in this case

$$\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{dy^{(1)}}{dx^{(1)}} & \frac{dy^{(1)}}{dx^{(2)}} & \frac{dy^{(1)}}{dx^{(3)}} \\ \frac{dy^{(2)}}{dx^{(1)}} & \frac{dy^{(2)}}{dx^{(2)}} & \frac{dy^{(2)}}{dx^{(3)}} \\ \frac{dy^{(3)}}{dx^{(1)}} & \frac{dy^{(3)}}{dx^{(2)}} & \frac{dy^{(3)}}{dx^{(3)}} \end{bmatrix}$$

let's say y is a scalar in this case

$$
\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}
\begin{bmatrix}
\frac{dy^{(1)}}{dx^{(1)}} & \cancel{\frac{dy^{(1)}}{dx^{(2)}}} & \cancel{\frac{dy^{(1)}}{dx^{(3)}}} \\
\cancel{\frac{dy^{(2)}}{dx^{(1)}}} & \frac{dy^{(2)}}{dx^{(2)}} & \cancel{\frac{dy^{(2)}}{dx^{(3)}}} \\
\cancel{\frac{dy^{(3)}}{dx^{(1)}}} & \cancel{\frac{dy^{(3)}}{dx^{(2)}}} & \frac{dy^{(3)}}{dx^{(3)}}
\end{bmatrix}
$$

$$
\begin{bmatrix} \frac{dy^{(1)}}{dx^{(1)}} & \frac{dy^{(2)}}{dx^{(2)}} & \frac{dy^{(3)}}{dx^{(3)}} \end{bmatrix}
$$

VJP

$$\bar{x}^T = \bar{y}^T J$$

VJP

$$\bar{x}^T = \bar{y}^T J$$

$$a^T = v^T J$$

VJP

$$\bar{x}^T = \bar{y}^T J$$

$$a^T = v^T J$$

$$a_1 = v_1 \frac{dy_1}{dx_1} + v_2 \frac{dy_2}{dx_1} + v_3 \frac{dy_3}{dx_1}$$

$$a_2 = v_1 \frac{dy_1}{dx_2} + v_2 \frac{dy_2}{dx_2} + v_3 \frac{dy_3}{dx_2}$$

VJP

$$\bar{x}^T = \bar{y}^T J$$

$$a^T = v^T J$$

$$a_1 = v_1 \frac{dy_1}{dx_1} + v_2 \cancel{\frac{dy_2}{dx_1}} + v_3 \cancel{\frac{dy_3}{dx_1}}$$

$$a_2 = v_1 \cancel{\frac{dy_1}{dx_2}} + v_2 \frac{dy_2}{dx_2} + v_3 \cancel{\frac{dy_3}{dx_2}}$$

$$a = \frac{d(v^T y)}{dx}$$

Jacobian

$$a = \frac{d(v^T y)}{dx}$$

JVP

$$\frac{dz}{dx}\frac{dx}{dt}$$

JVP

$$\frac{dz}{dx}\frac{dx}{dt}$$

$$\begin{bmatrix} \frac{dz_1}{dx_1} & \frac{dz_1}{dx_2} & \frac{dz_1}{dx_3} \\ \frac{dz_2}{dx_1} & \frac{dz_2}{dx_2} & \frac{dz_2}{dx_3} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

JVP

$$\frac{dz}{dx}\frac{dx}{dt}$$

$$\begin{bmatrix} \frac{dz_1}{dx_1} & \frac{dz_1}{dx_2} & \frac{dz_1}{dx_3} \\ \frac{dz_2}{dx_1} & \frac{dz_2}{dx_2} & \frac{dz_2}{dx_3} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

$$f_1\frac{dz_1}{dx_1} + f_2\frac{dz_1}{dx_2} + f_3\frac{dz_1}{dx_3}$$