GAMES 105
Fundamentals of Character Animation

Lecture 06+

Learning-based Character Animation (cont.)

Libin Liu

School of Intelligence Science and Technology
Peking University

GAMES105 课程交流

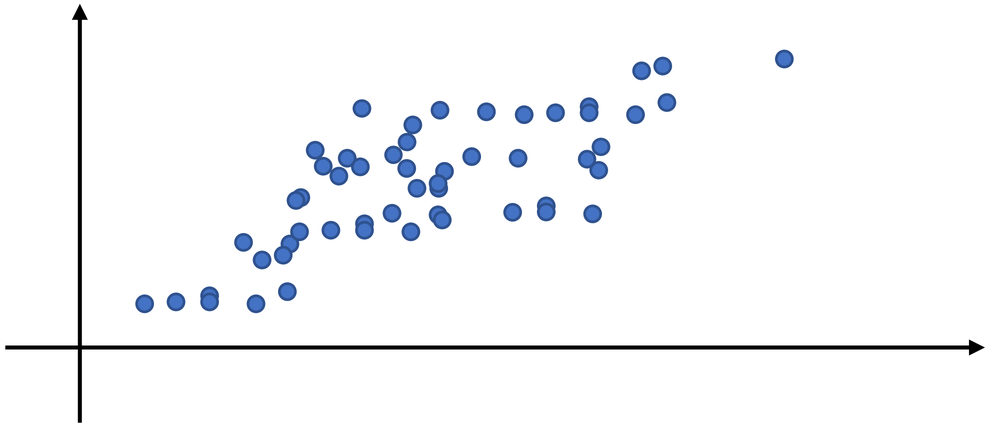
VCL @ PKU

Outline

- Learning-based Character Animation (cont.)
  - Motion Models
  - Autoregressive models: PFNN
  - Generative models

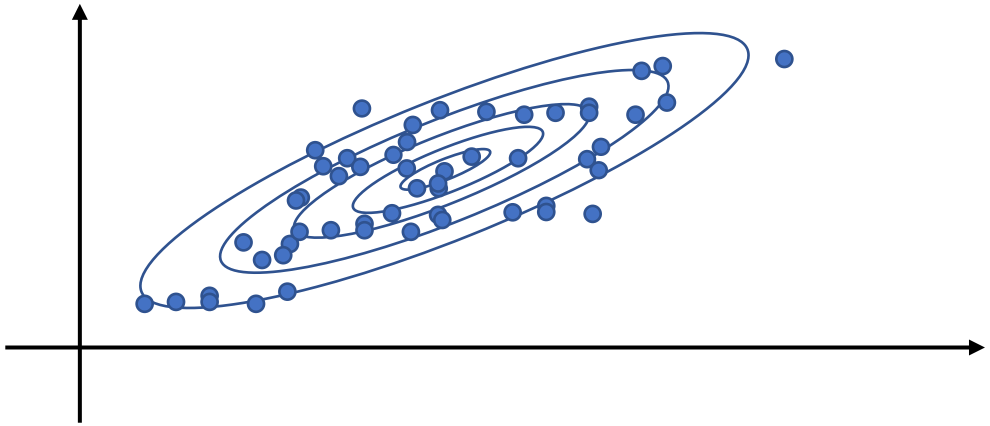
Learning Motion Models

Given a set of example motions $\{\boldsymbol{x}_i\} \sim p(\boldsymbol{x})$



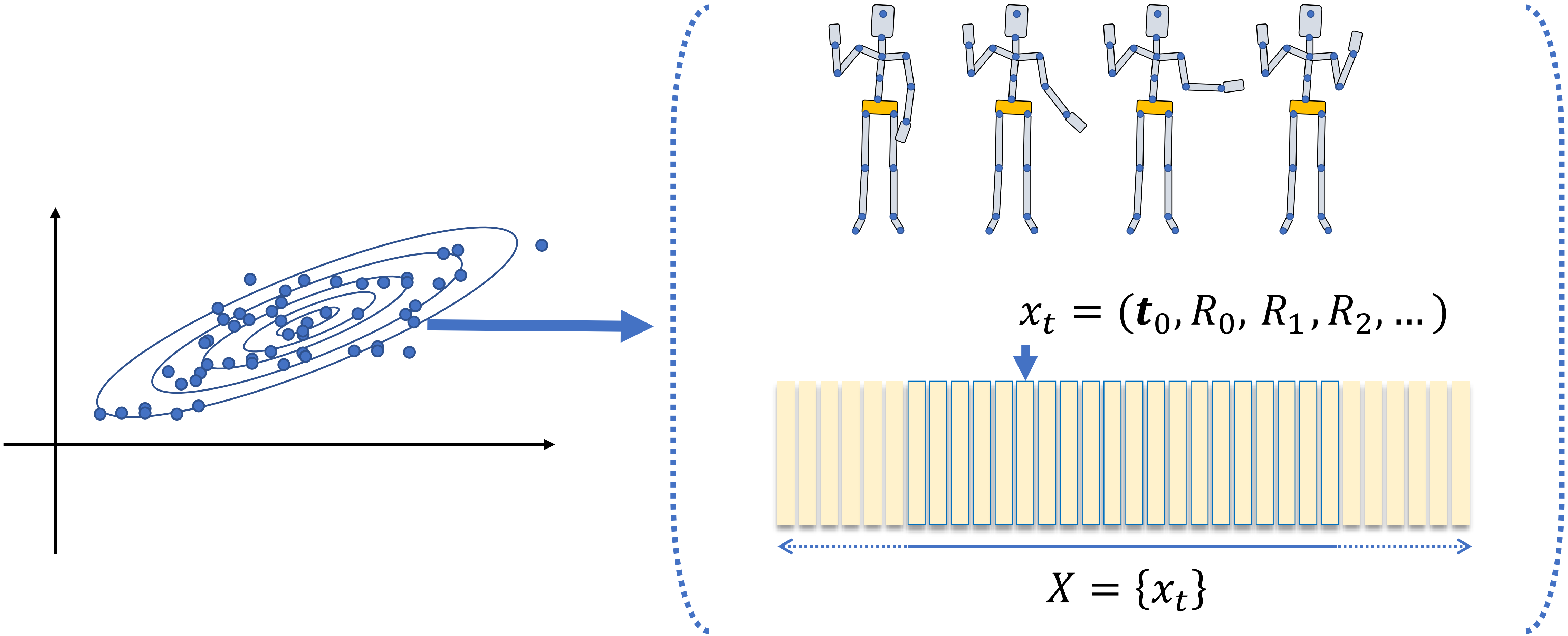
Learning Motion Models

Given a set of example motions $\{\boldsymbol{x}_i\} \sim p(\boldsymbol{x})$

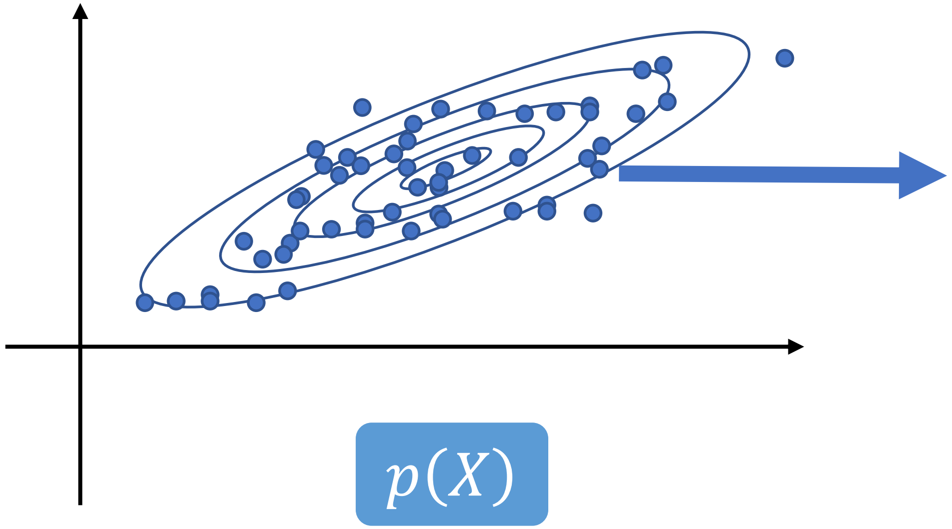


$p(\boldsymbol{x})$: probability that $\boldsymbol{x}$ is a natural motion

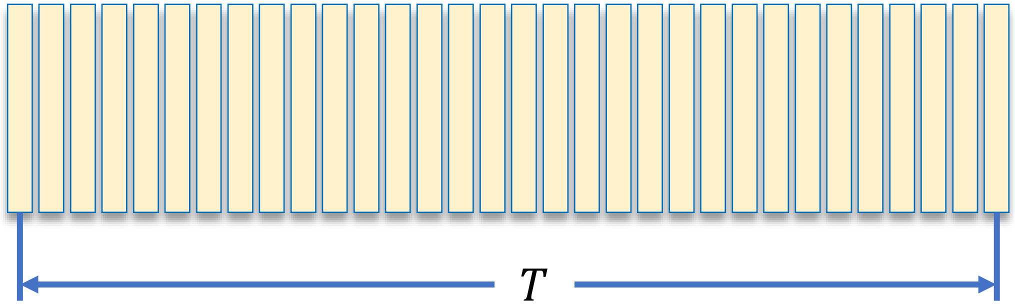
Learning Motion Models



Learning Motion Models

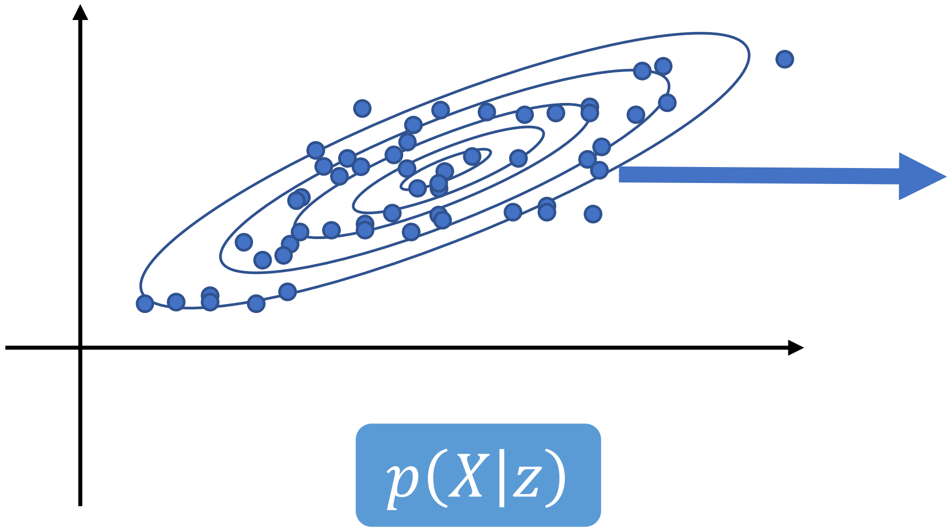


$$X = \{x_1, \dots, x_T\}$$



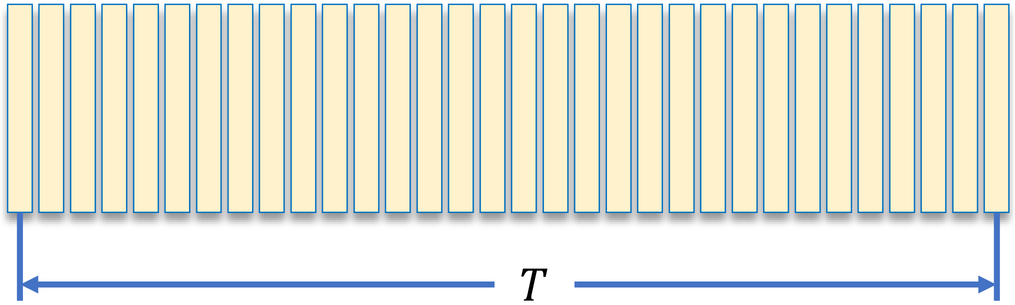
$$p(X) = p(x_1, \dots, x_T)$$

Learning Motion Models



Conditional distribution

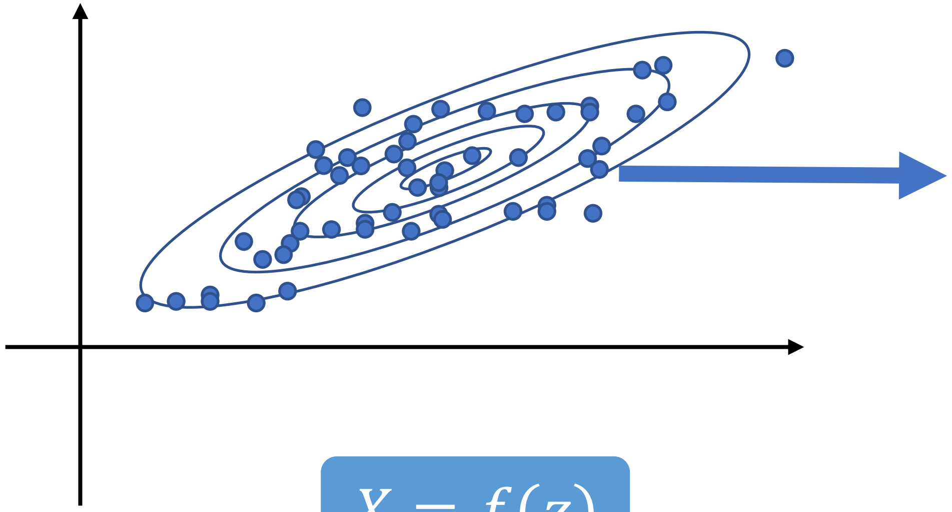
$$X = \{x_1, \dots, x_T\}$$



$$p(X|z) = p(x_1, \dots, x_T|z)$$

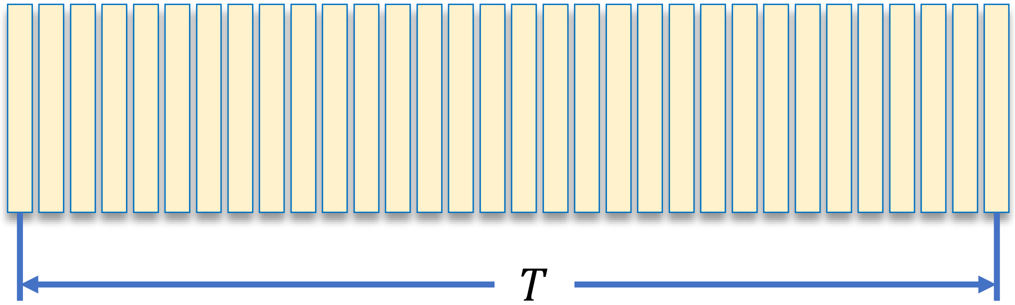
z: control parameters
latent variables
……

Learning Motion Models



$X = f(z)$

$$X = \{x_1, \ldots, x_T\}$$

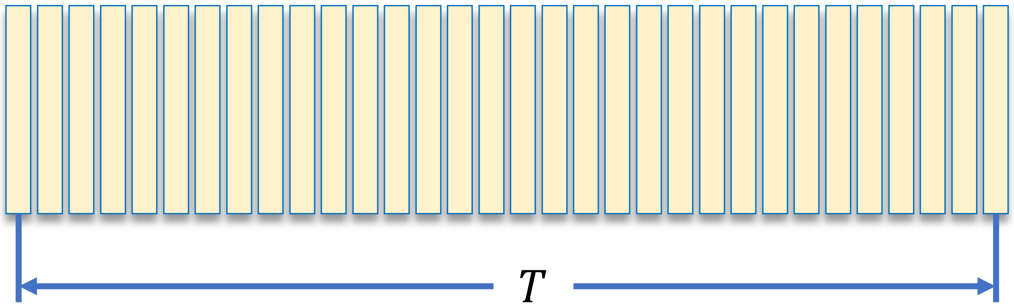


$$(x_1, \ldots, x_T) = f(z)$$

z: control parameters
latent variables
……

Two Perspectives on a Motion Sequence

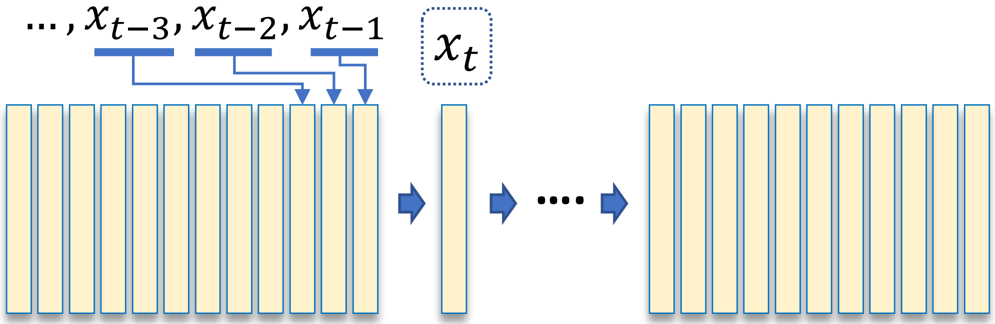
$$X = \{x_1, \dots, x_T\}$$



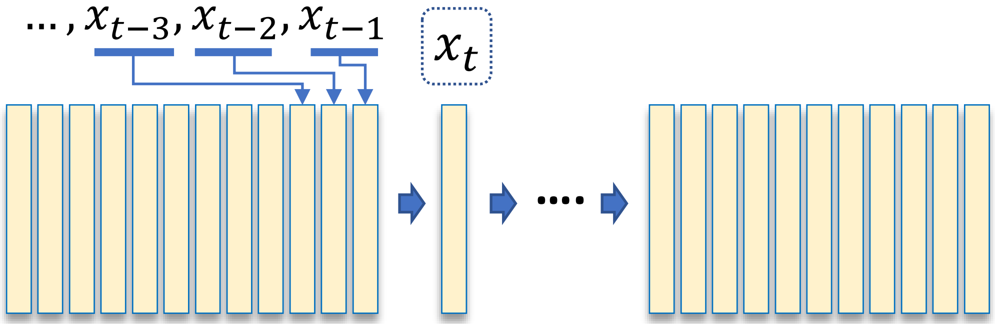
$$p(X|z)$$

$$X = f(z)$$

Two Perspectives on a Motion Sequence

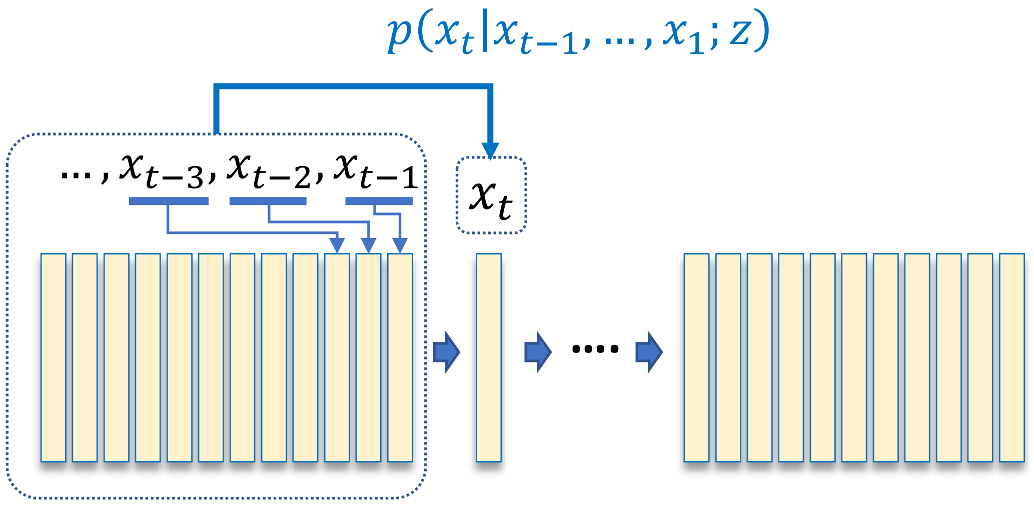


Two Perspectives on a Motion Sequence



$$p(X|z) = p(x_1, \dots, x_T|z)$$

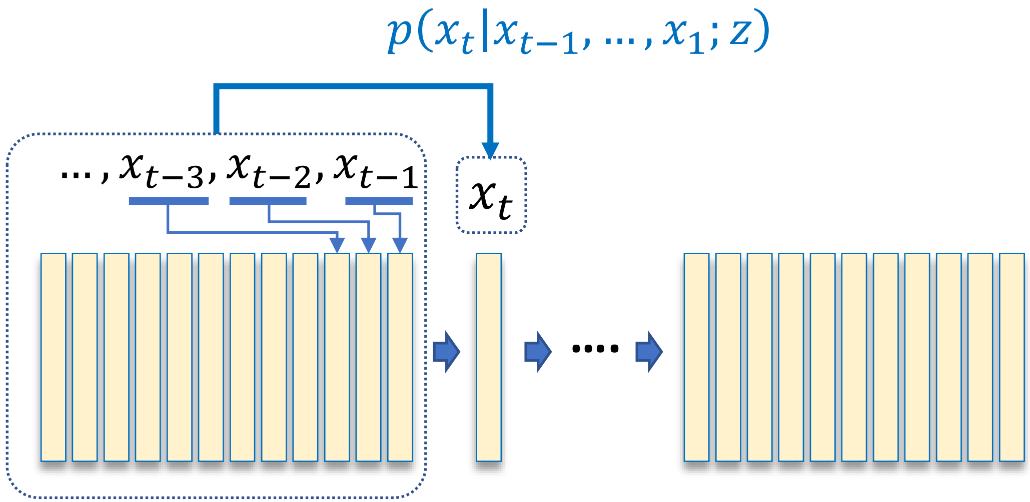
Two Perspectives on a Motion Sequence



$$p(X|z) = p(x_1,\dots,x_T|z)$$

$$= p(x_1)\prod_t p(x_t|x_{t-1},\dots,x_1;z)$$

Two Perspectives on a Motion Sequence



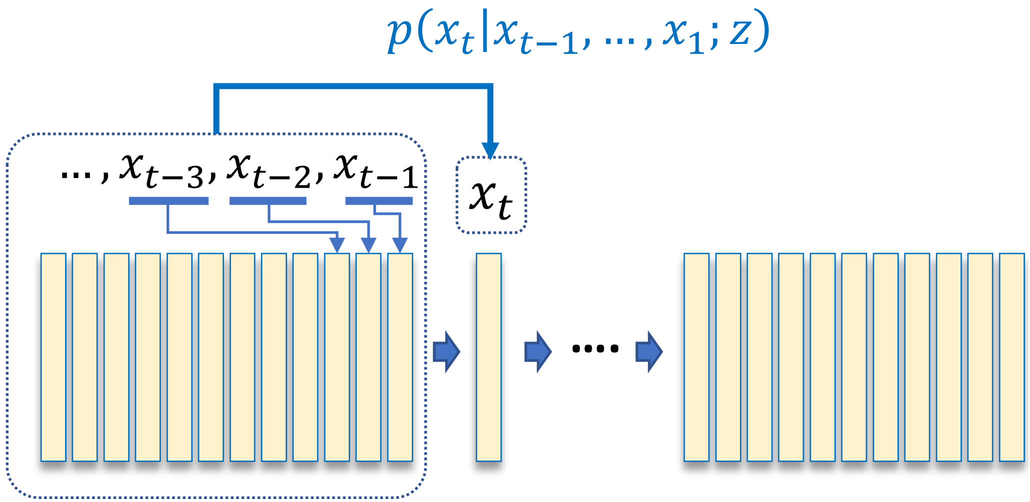
$$p(X|z) = p(x_1, \dots, x_T|z)$$

$$= p(x_1) \prod_t p(x_t|x_{t-1}, \dots, x_1; z)$$

*The chain rule of conditional probabilities:

$$p(x_1, x_2) = p(x_2|x_1)p(x_1)$$

Two Perspectives on a Motion Sequence



$$p(X|z) = p(x_1,\dots,x_T|z)$$

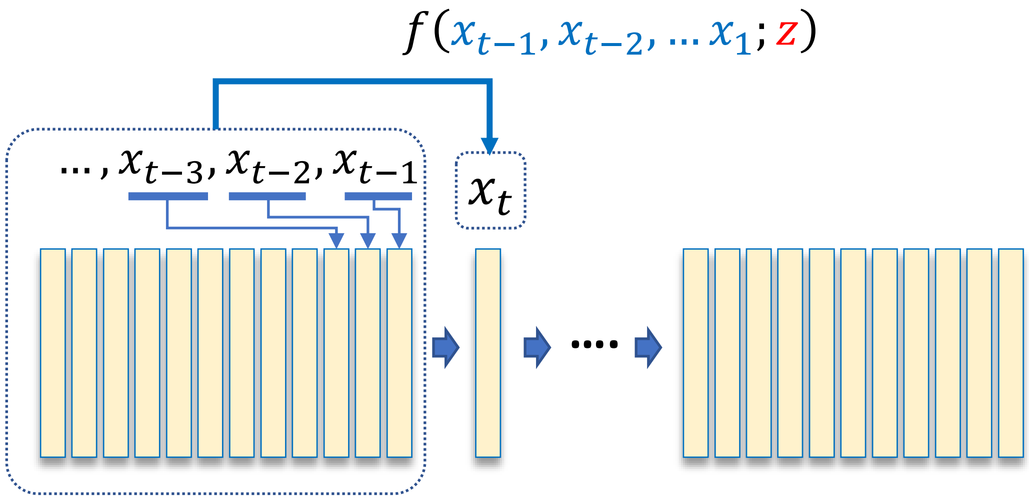
$$= p(x_1)\prod_t p(x_t|x_{t-1},\dots,x_1;z)$$

*The chain rule of conditional probabilities:

$$p(x_1,x_2) = p(x_2|x_1)p(x_1)$$

$$p(x_1,x_2,x_3) = p(x_2,x_3|x_1)p(x_1)$$
$$= p(x_3|x_2,x_1)p(x_2|x_1)p(x_1)$$

Two Perspectives on a Motion Sequence

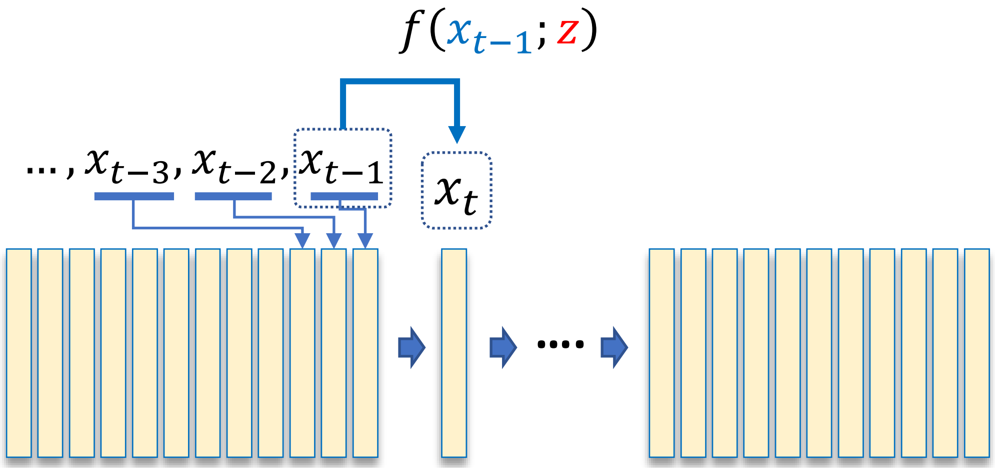


$$p(X|z) = p(x_1, \dots, x_T|z)$$

$$= p(x_1) \prod_t p(x_t | x_{t-1}, \dots, x_1; z)$$

$$x_t = f(x_{t-1}, x_{t-2}, \dots x_1; z)$$

Two Perspectives on a Motion Sequence



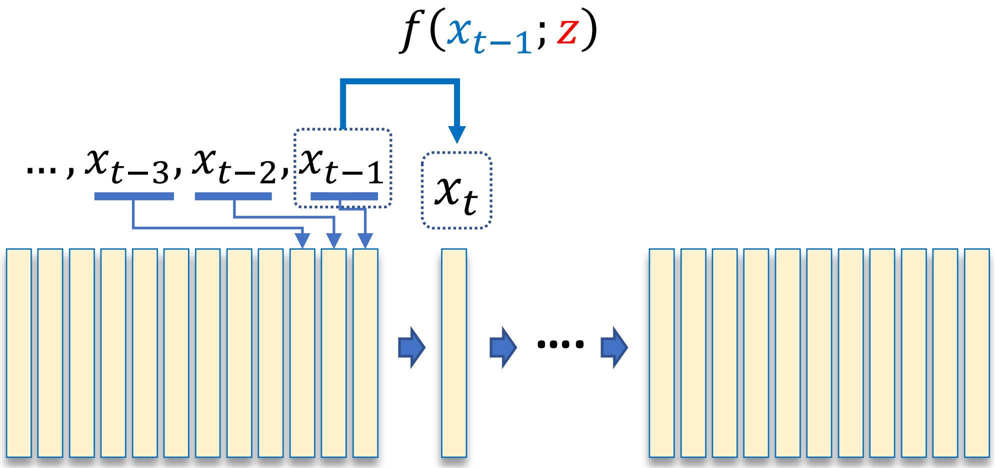
$$p(X|z) = p(x_1, \ldots, x_T|z)$$

$$= p(x_1) \prod_t p(x_t|x_{t-1}, \ldots, x_1; z)$$

$$x_t = f(x_{t-1}, x_{t-2}, \ldots, x_1; z)$$

Markov Property

Two Perspectives on a Motion Sequence



$$p(X|z) = p(x_1, \dots, x_T|z)$$

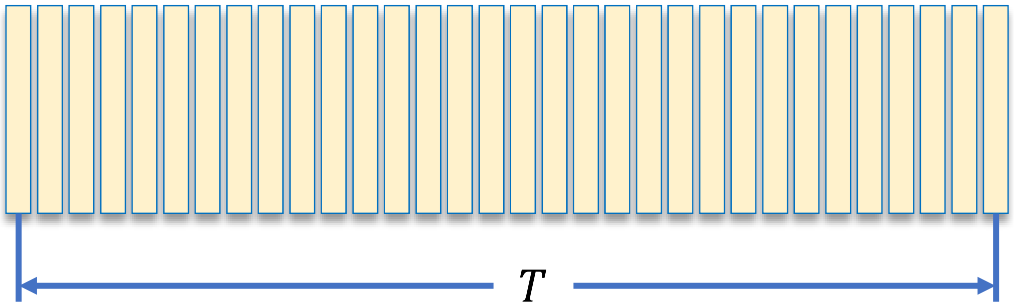
$$= p(x_1) \prod_t p(x_t|x_{t-1}; z)$$

$$x_t = f(x_{t-1}; z)$$

Markov Property

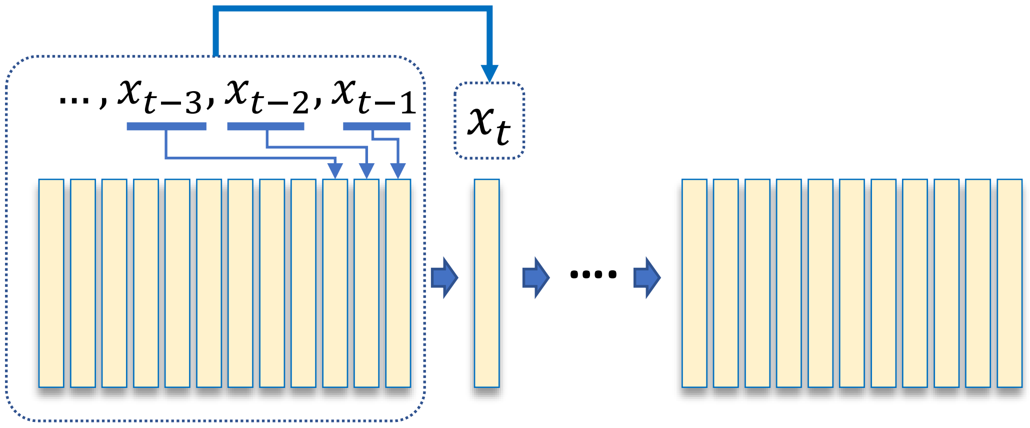
Two Perspectives on a Motion Sequence

$$X = \{x_1, \dots, x_T\}$$



$$p(X|z)$$

$$X = f(z)$$



$$p(x_t|x_{t-1}, \dots, x_1; z)$$

$$x_t = f(x_{t-1}, x_{t-2}, \dots; z)$$

$$x_t = f(x_{t-1}; z)$$

Markov Property

Autoregressive Model

$$x_t = f(x_{t-1}, x_{t-2}, \dots; z)$$

$$x_t = f(x_{t-1}; z)$$

Autoregressive Model

$$x_t = f(x_{t-1}, x_{t-2}, \ldots; z)$$

$$x_t = f(x_{t-1}; z)$$

Autoregressive Model

$$x_t = f(x_{t-1}; z)$$

z: control parameters
latent variables
……

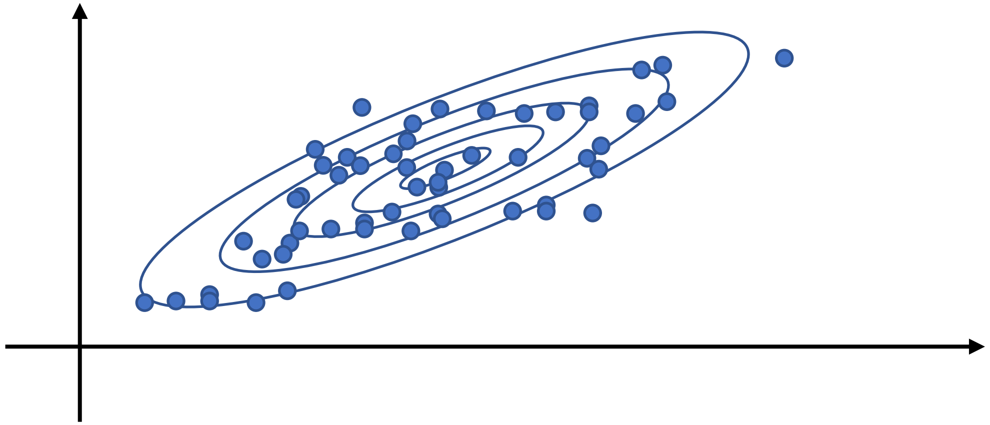
Autoregressive Model

$$x_t = f(x_{t-1})$$

Autoregressive Model

$$x_t = f(x_{t-1})$$

Given a set of example motions $\{\boldsymbol{X}_i\} \sim p(\boldsymbol{X})$

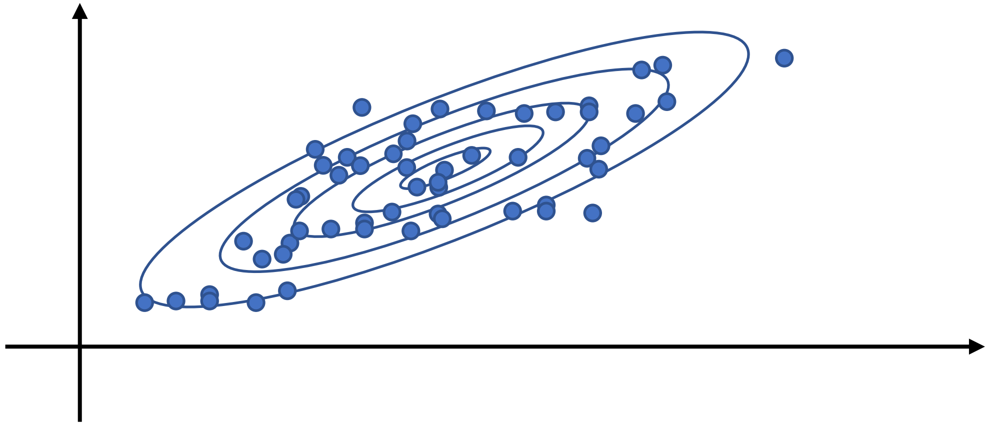


Autoregressive Model

$$x_t = f(x_{t-1})$$

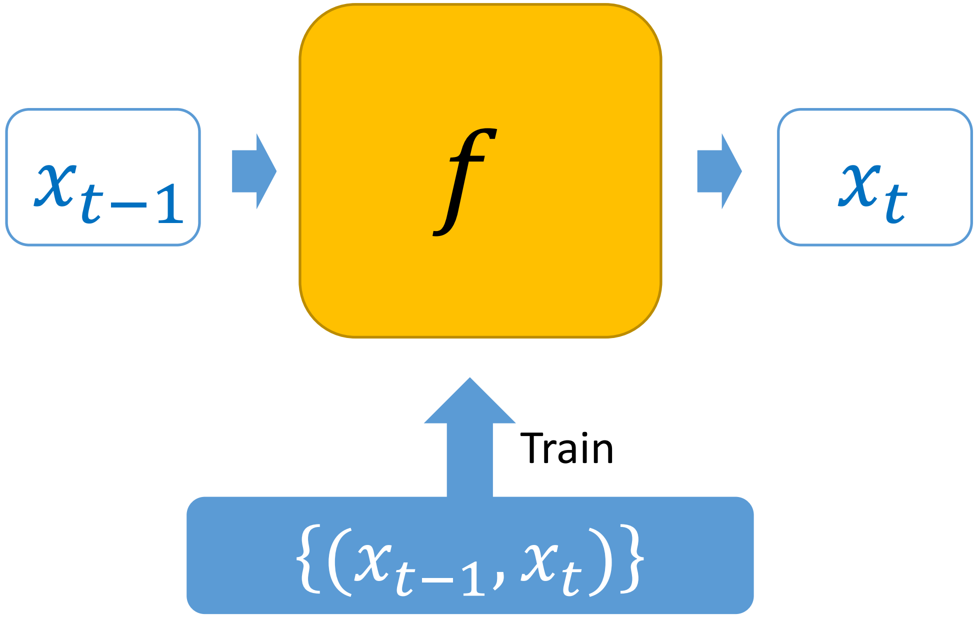
Given a set of example transitions

$$\{(x_{t-1}, x_t)\} \sim p(\boldsymbol{X})$$



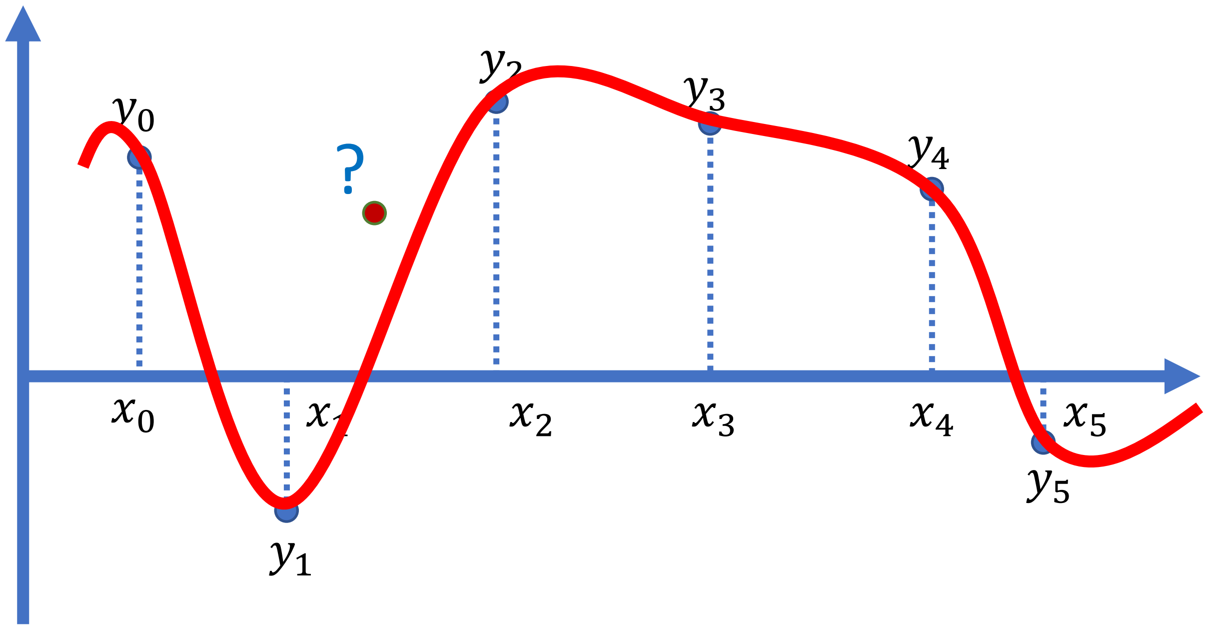
Learning Motion Models

$$x_t = f(x_{t-1})$$

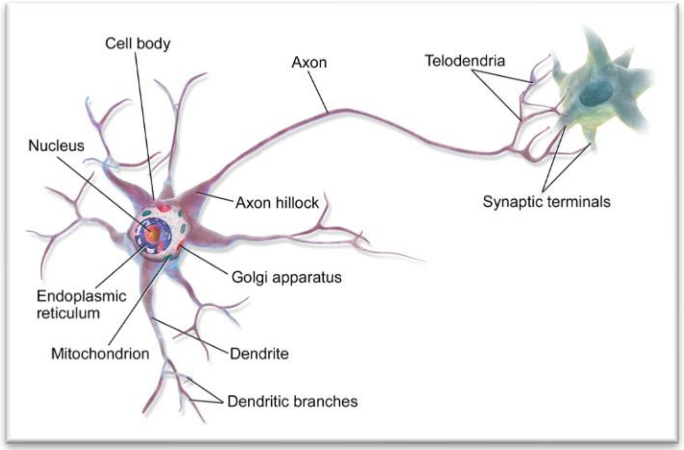


Interpolation

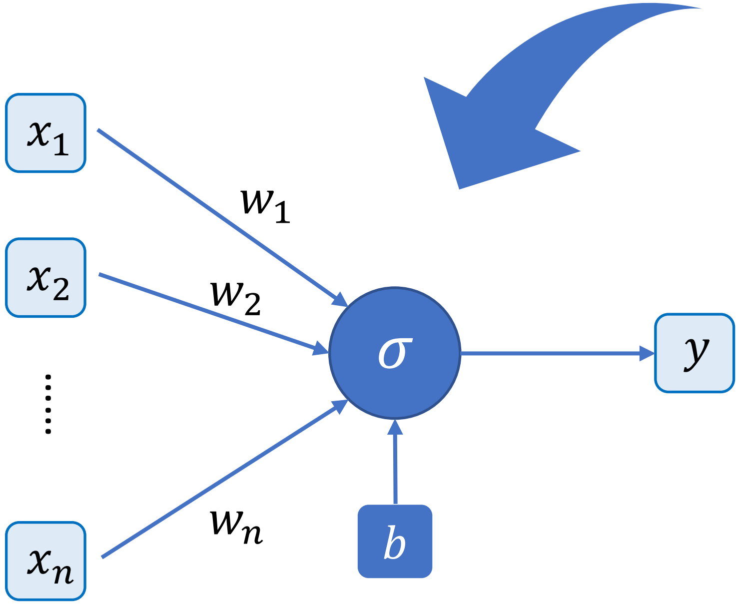
$$f(x_i) = y_i \quad \Rightarrow \quad f(x) = ?$$



Neural Networks

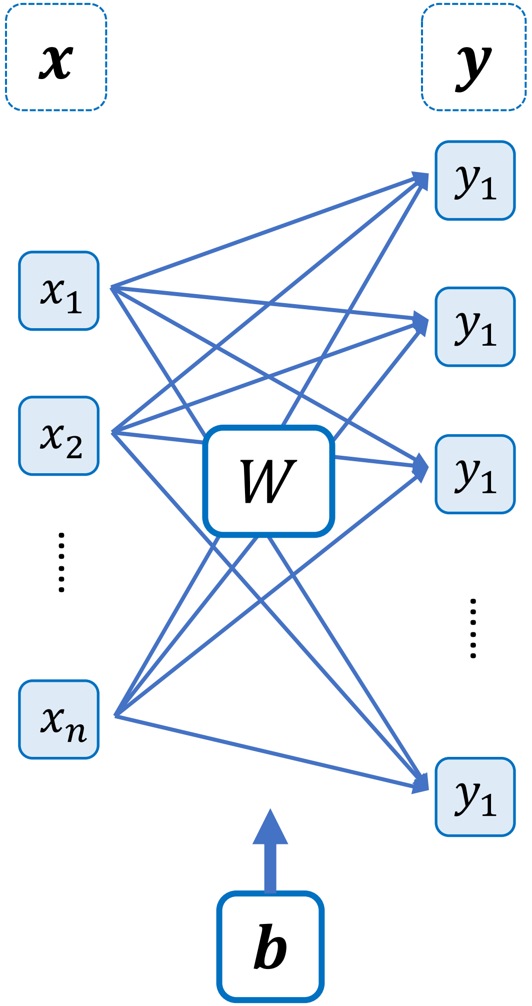


A Multipolar Neuron



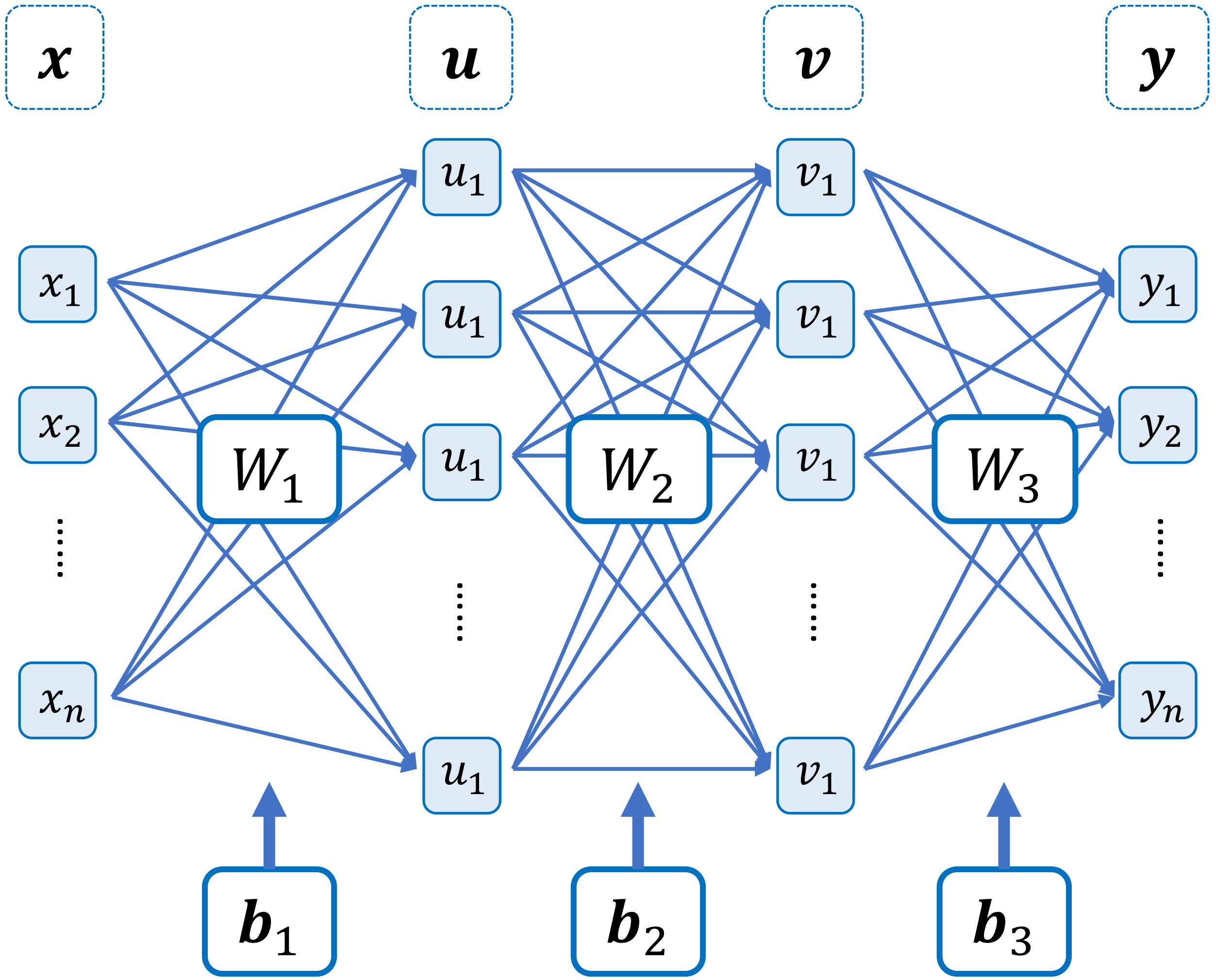
$$y = \sigma\left(\sum_i w_i x_i + b\right)$$

Neural Networks

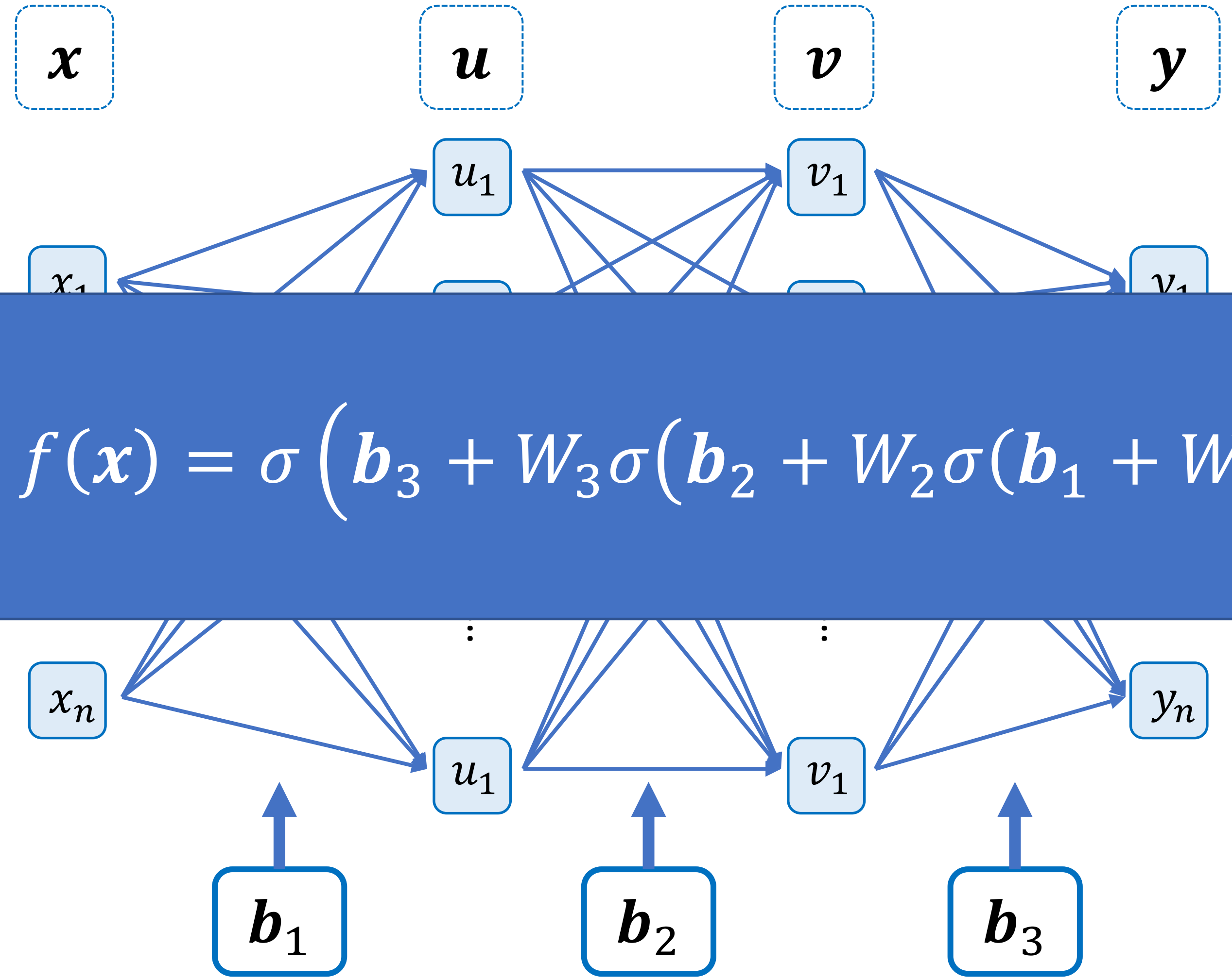


$$\boldsymbol{y} = \sigma(W\boldsymbol{x} + \boldsymbol{b})$$

Neural Networks

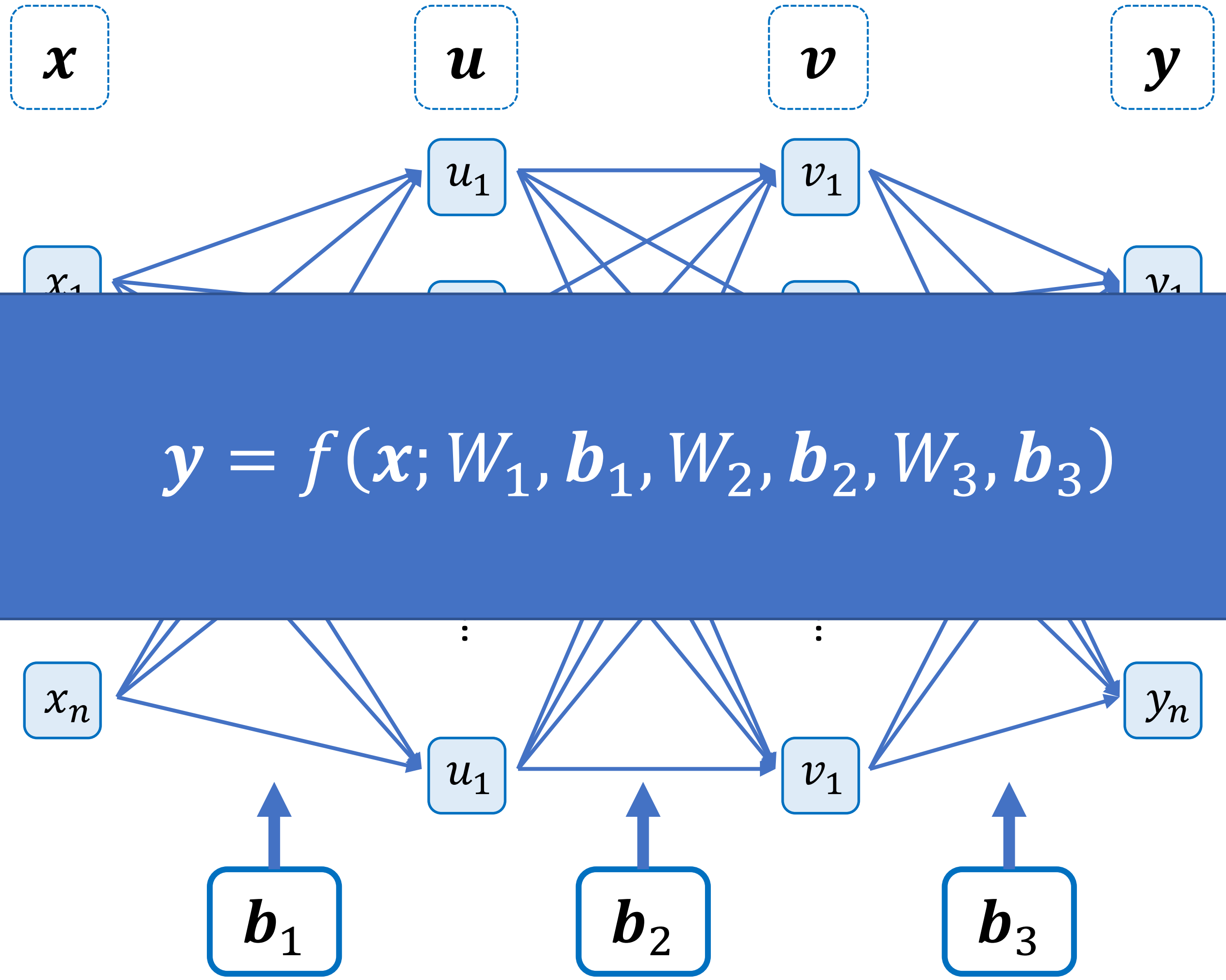


Neural Networks



$$\boldsymbol{y} = f(\boldsymbol{x}) = \sigma\left(\boldsymbol{b}_3 + W_3\sigma\left(\boldsymbol{b}_2 + W_2\sigma(\boldsymbol{b}_1 + W_1\boldsymbol{x})\right)\right)$$

Neural Networks

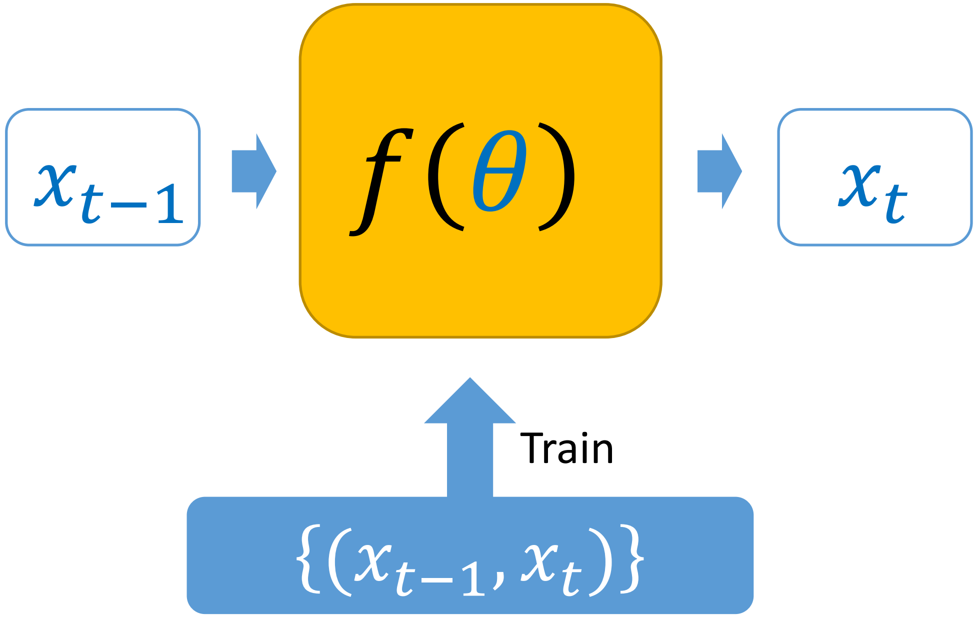


$$\boldsymbol{y} = f(\boldsymbol{x}; W_1, \boldsymbol{b}_1, W_2, \boldsymbol{b}_2, W_3, \boldsymbol{b}_3)$$

Learning Motion Models

$$x_t = f(x_{t-1}; \theta)$$

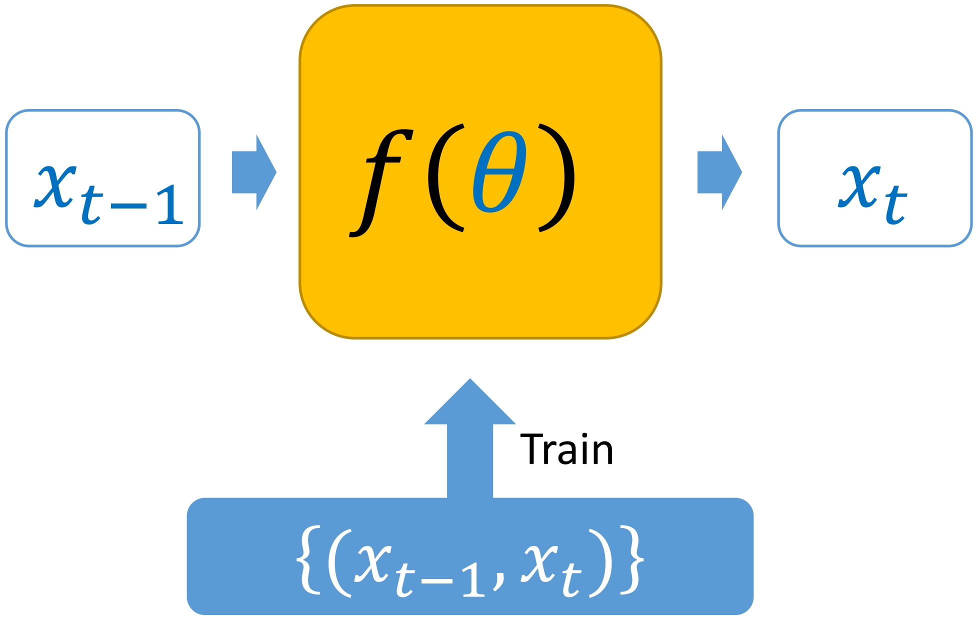
$$\theta = (W_1, \boldsymbol{b}_1, W_2, \boldsymbol{b}_2, \dots)$$



Learning Motion Models

$$x_t = f(x_{t-1}; \theta)$$

$$\theta = (W_1, \boldsymbol{b}_1, W_2, \boldsymbol{b}_2, \dots)$$



Given a set of example transitions

$$\{(x_{t-1}, x_t)\} \sim p(\boldsymbol{X})$$

Find $\theta = (W_1, \boldsymbol{b}_1, W_2, \boldsymbol{b}_2, \dots)$ that minimizes

$$F(\theta) = \sum_{(x_{t-1}, x_t)} \| f(x_{t-1}; \theta) - x_t \|$$

Learning Motion Models

Stochastic Gradient Descent:

For a batch of random sample $\left\{\left(x_{t-1}^{(i)}, x_t^{(i)}\right)\right\} \sim \{(x_{t-1}, x_t)\}$

Compute the approximate gradient

$$\nabla_\theta F(\theta) \approx \sum_i \nabla_\theta \left(\left\| f\left(x_{t-1}^{(i)}; \theta\right) - x_t^{(i)} \right\| \right)$$

update $\theta = (W_1, \boldsymbol{b}_1, W_2, \boldsymbol{b}_2, \ldots)$ as

$$\theta \leftarrow \theta - \alpha \nabla_\theta F(\theta)$$

Learning Motion Models

Stochastic Gradient Descent:

For a batch of random sample $\left\{\left(x_{t-1}^{(i)}, x_t^{(i)}\right)\right\} \sim \{(x_{t-1}, x_t)\}$

Compute the approximate gradient

$$\nabla_\theta F(\theta) \approx \sum_i \nabla_\theta \left(\left\| f\left(x_{t-1}^{(i)}; \theta\right) - x_t^{(i)} \right\| \right)$$

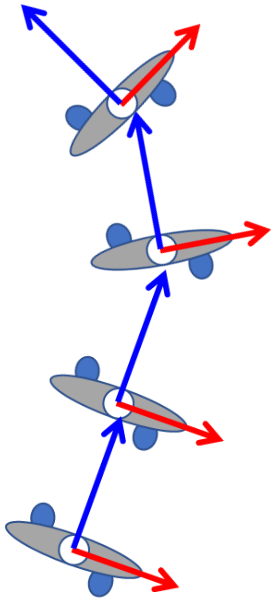
using backpropagation

update $\theta = (W_1, \boldsymbol{b}_1, W_2, \boldsymbol{b}_2, \ldots)$ as

$$\theta \leftarrow \theta - \alpha \nabla_\theta F(\theta)$$

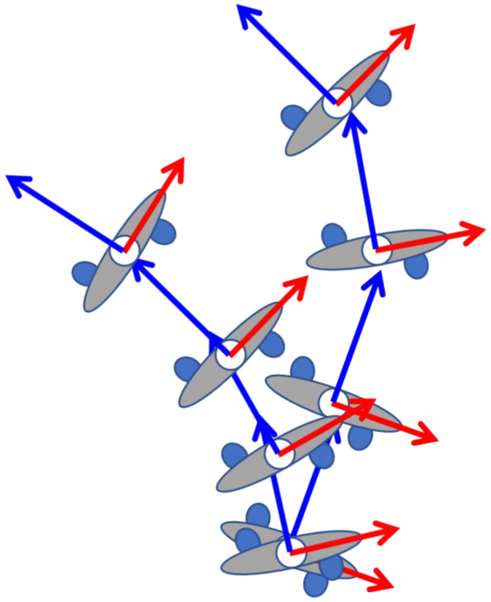
Ambiguity Issue

$$x_t = f(x_{t-1})$$



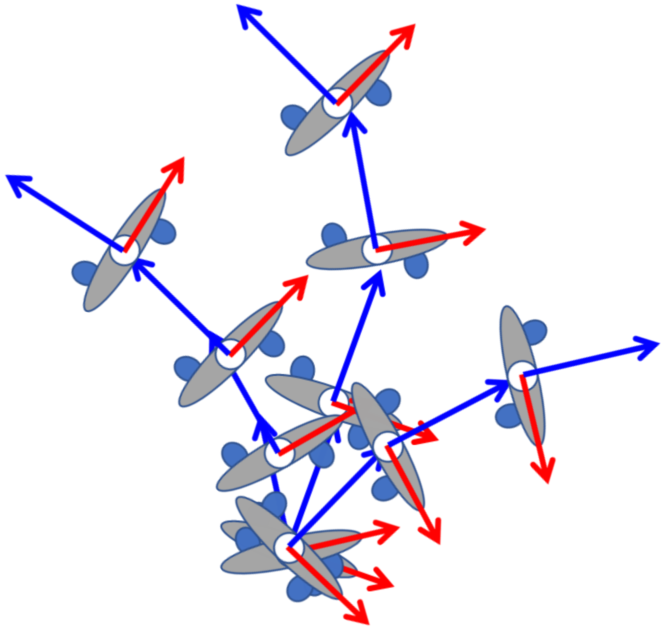
Ambiguity Issue

$$x_t = f(x_{t-1})$$



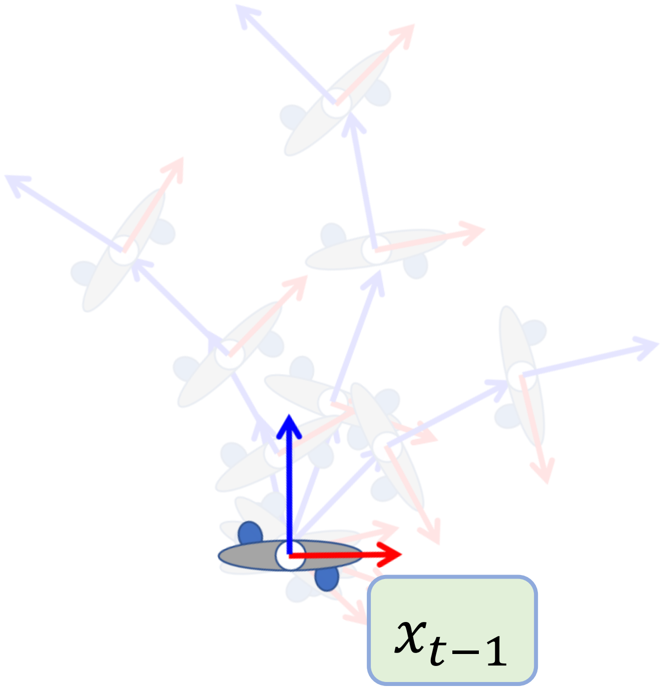
Ambiguity Issue

$$x_t = f(x_{t-1})$$



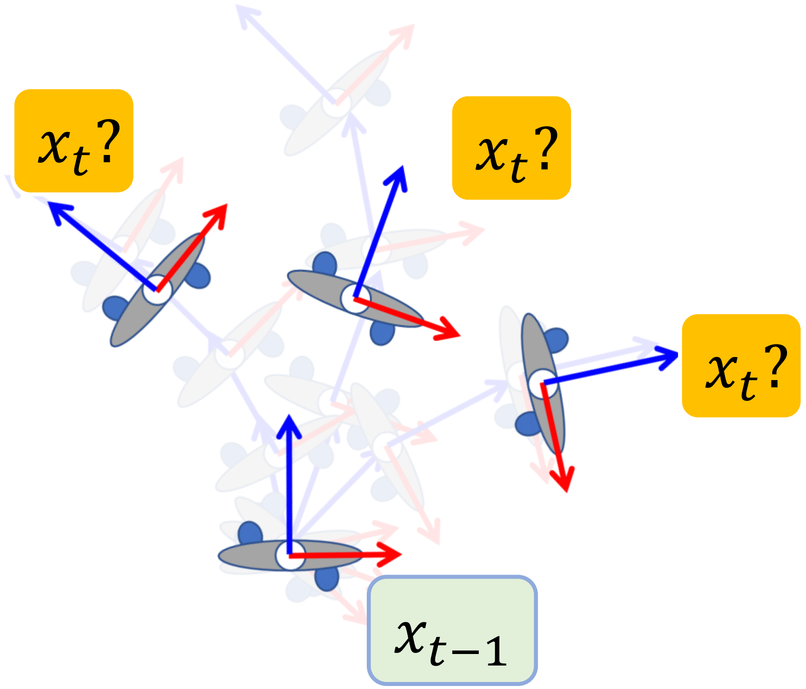
Ambiguity Issue

$$x_t = f(x_{t-1})$$



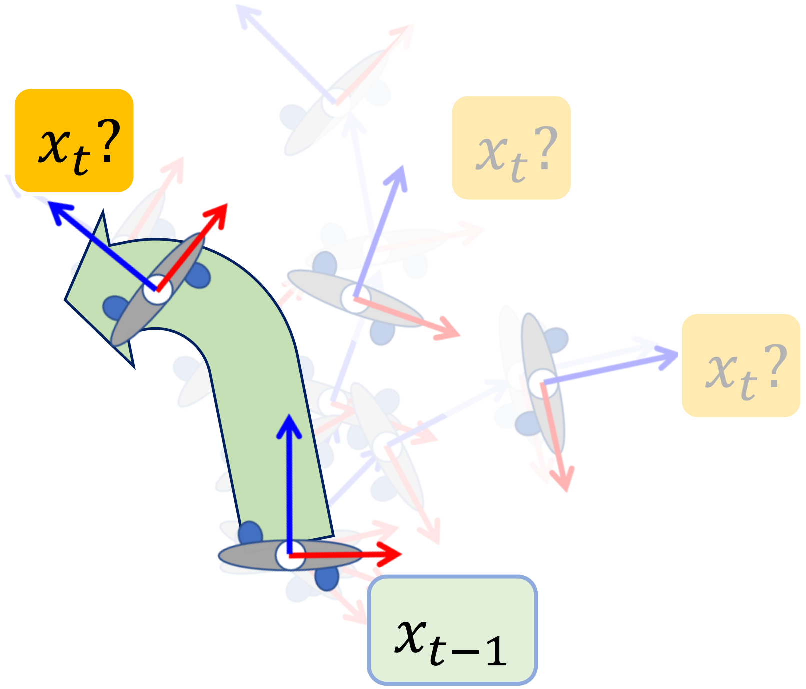
Ambiguity Issue

$$x_t = f(x_{t-1})$$

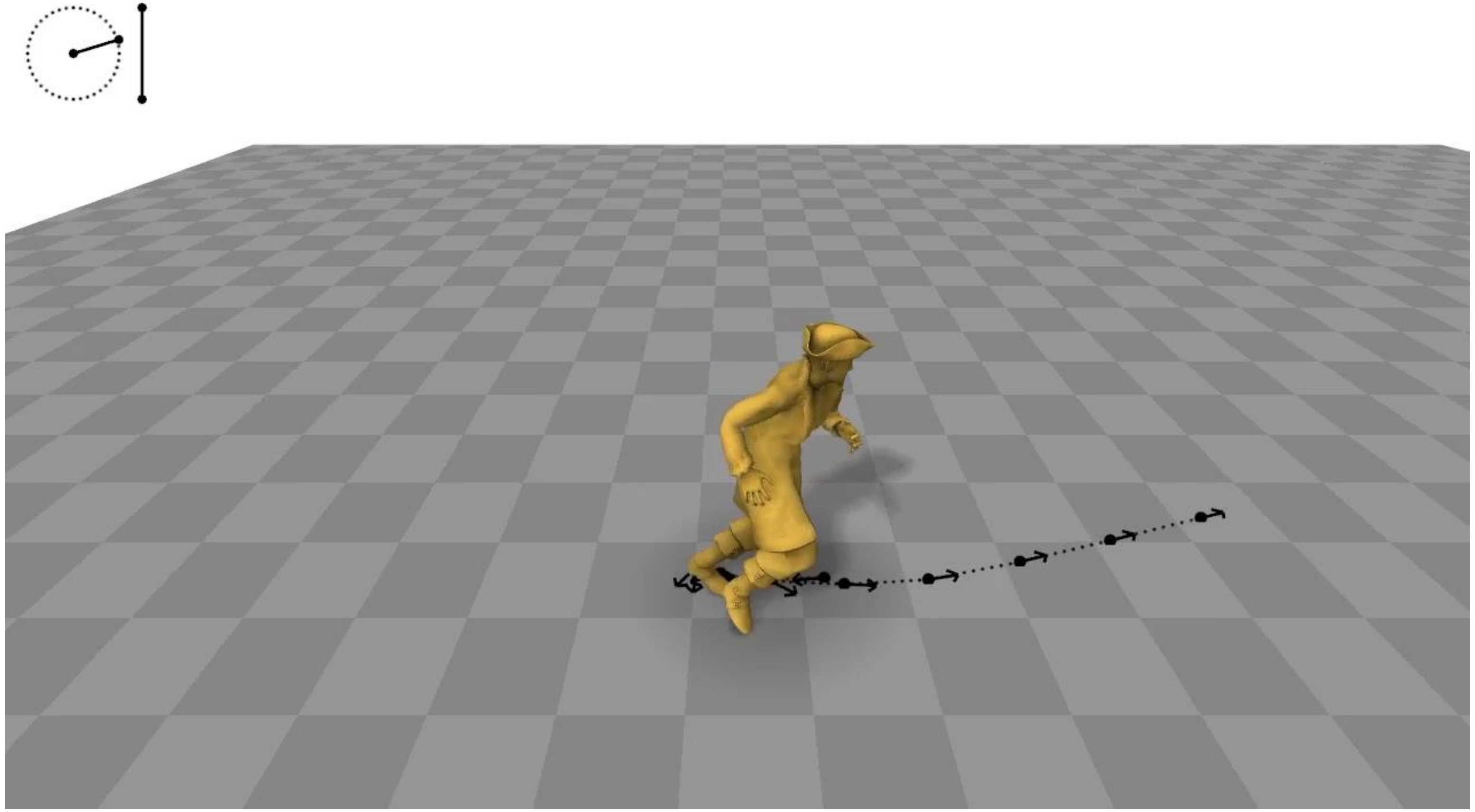


Hidden Variables

$$x_t = f(x_{t-1}; z)$$



PFNN: Phase-Functioned Neural Networks



Phase-Functioned Neural Networks for Character Control

DANIEL HOLDEN, University of Edinburgh
TAKU KOMURA, University of Edinburgh
JUN SAITO, Method Studios

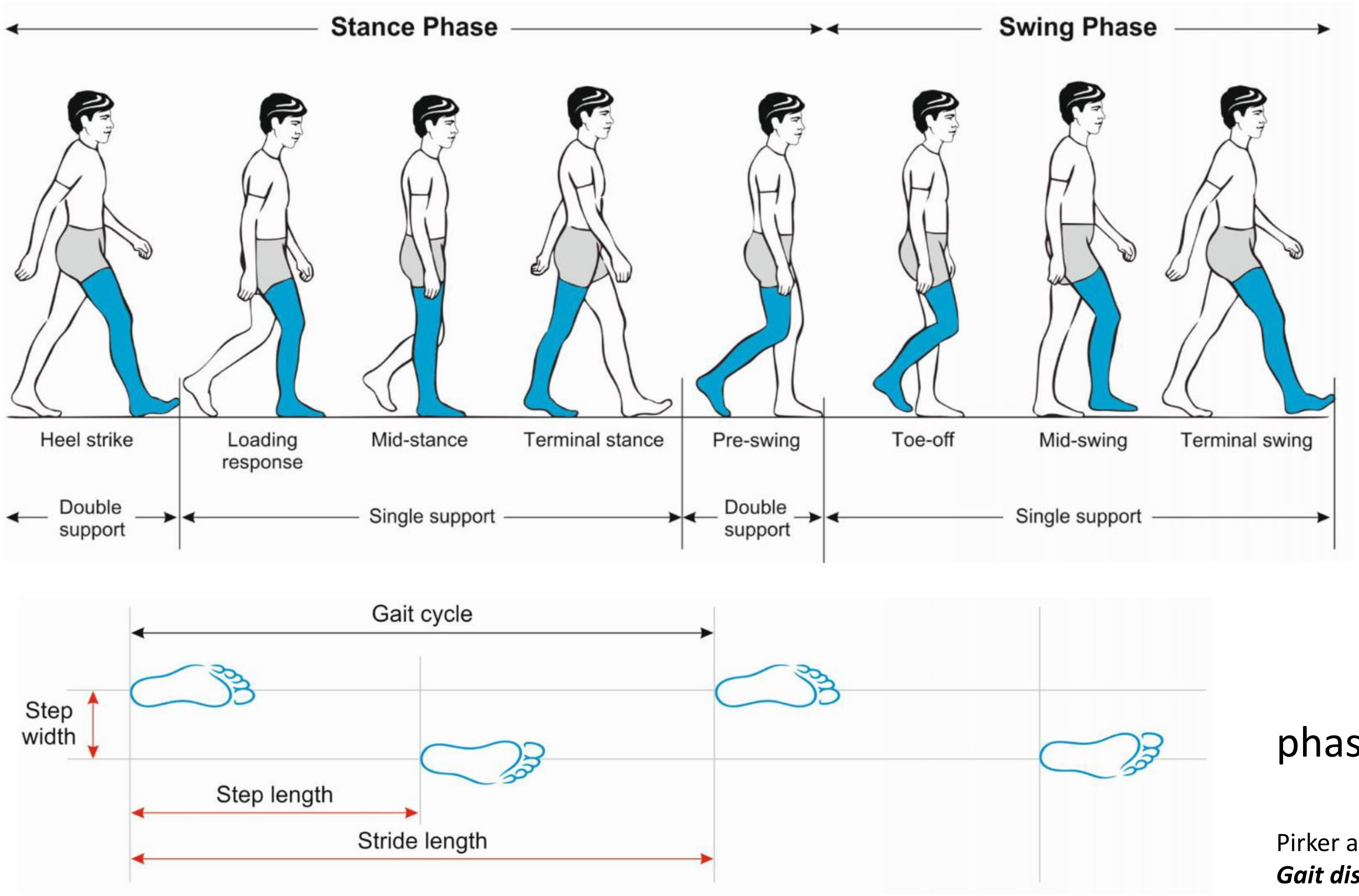
*SIGGRAPH 2017

PFNN: Phase-Functioned Neural Networks

$$x_t = f(x_{t-1}; z_t)$$

$z_t \rightarrow$ control parameters
phase parameter

PFNN: Phase-Functioned Neural Networks

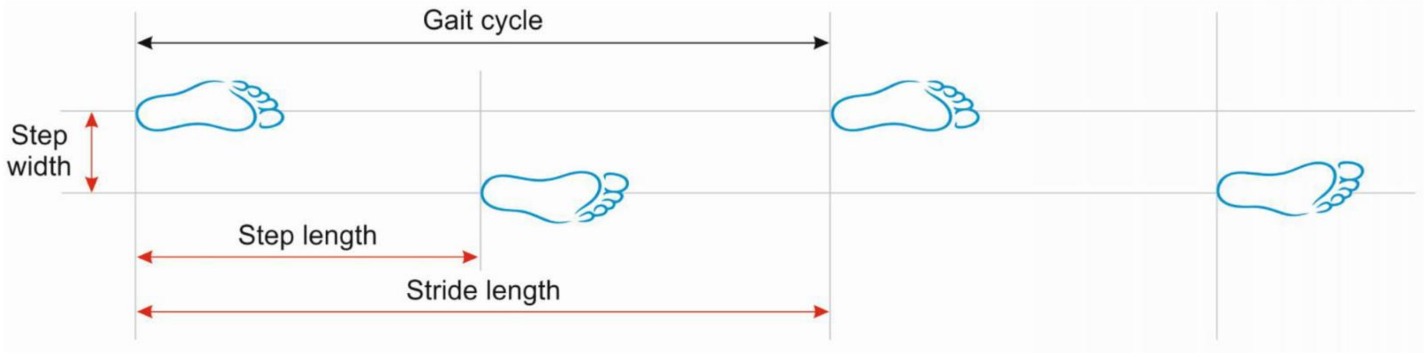


phases of a walking gait cycle

Pirker and Katzenschlager 2017.
Gait disorders in adults and the elderly.

PFNN: Phase-Functioned Neural Networks

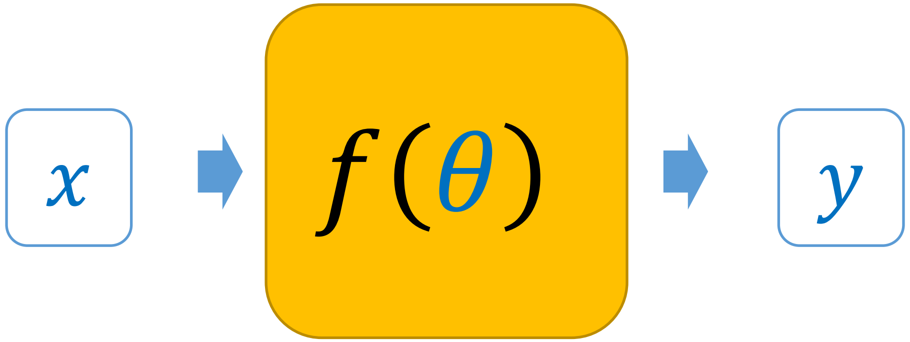
$$x_t = f(x_{t-1}; z_t) \qquad z_t = (c_t, \phi_t)$$



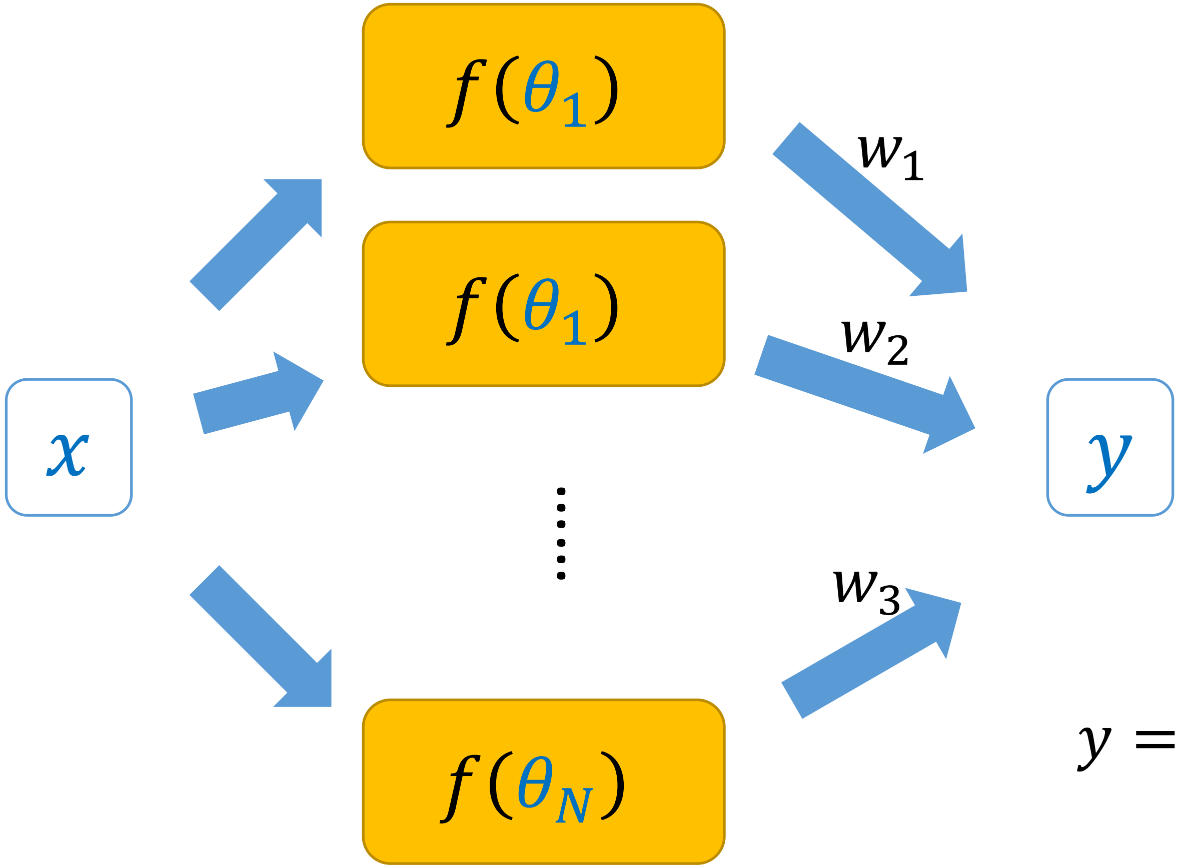
$\phi =$ 0.0 → 0.5 → 1.0

0.0 → 0.5 → 1 →

Mixture of Experts

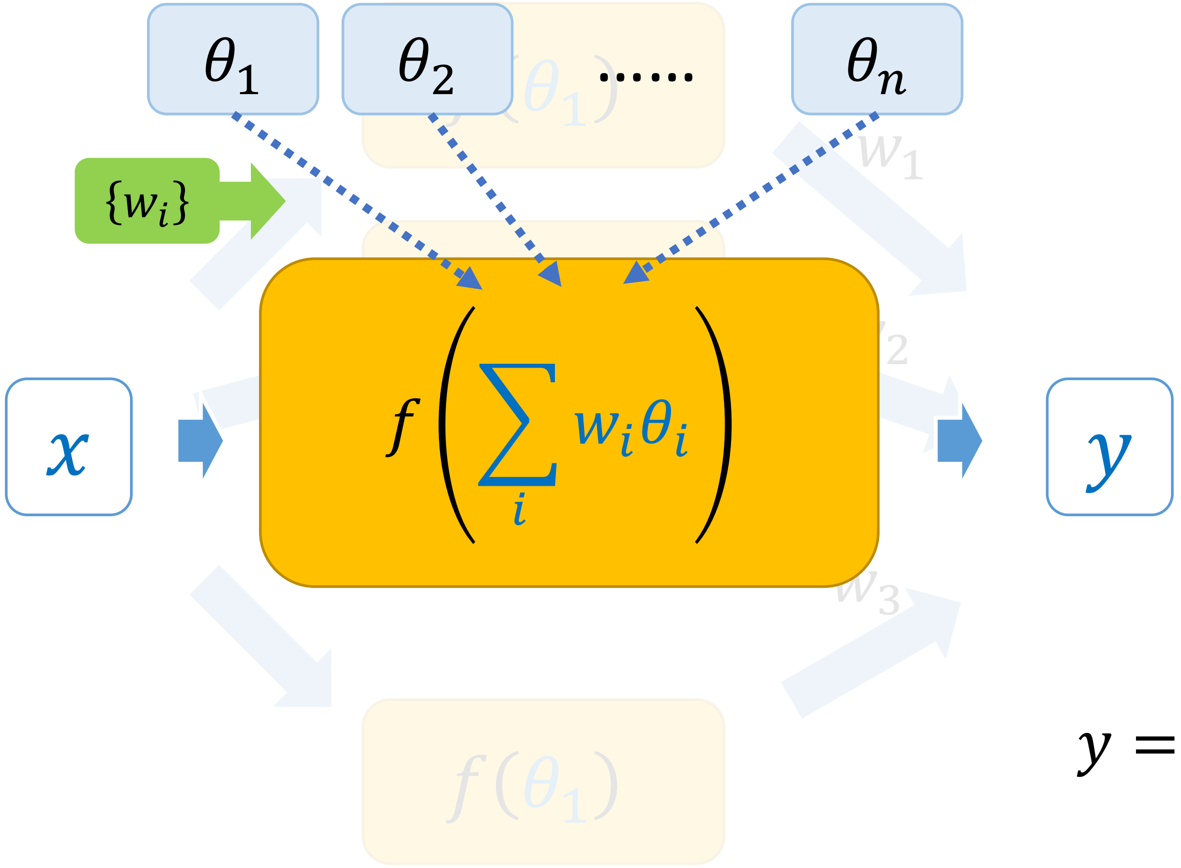


Mixture of Experts



$$y = \sum_i w_i f(x; \theta_i)$$

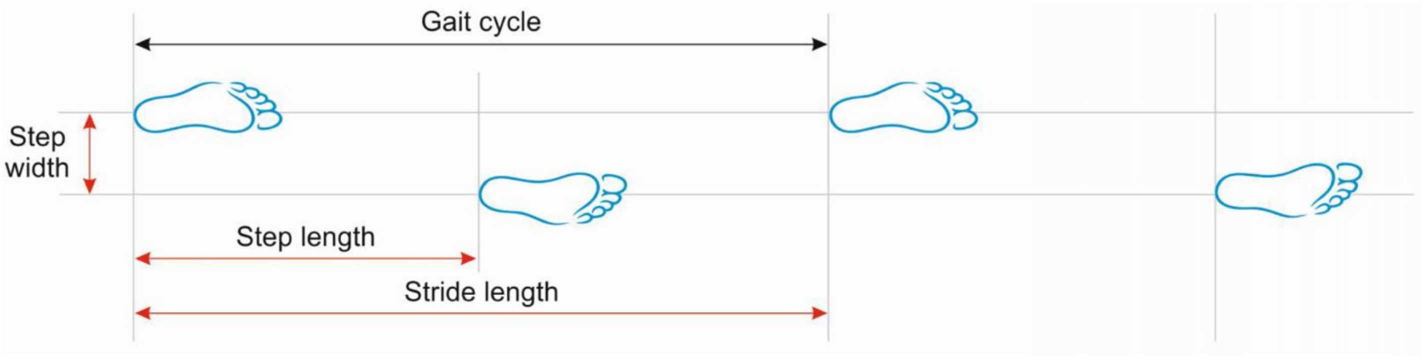
Weighted-Blended Mixture of Experts



$$y = f\left(x; \sum_i w_i\theta_i\right)$$

PFNN: Phase-Functioned Neural Networks

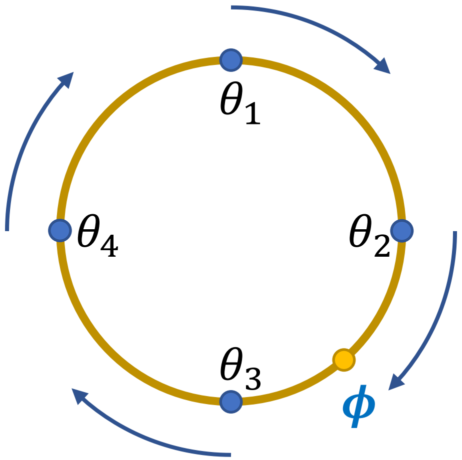
$$x_t = f\left(x_{t-1}; c_t, \theta_t = \sum_i w_i(\phi_t)\theta_i\right)$$



$\phi = 0.0 \dashrightarrow 0.5 \dashrightarrow$ 1.0 / 0.0 $\dashrightarrow 0.5 \dashrightarrow 1 \dashrightarrow$

PFNN: Phase-Functioned Neural Networks

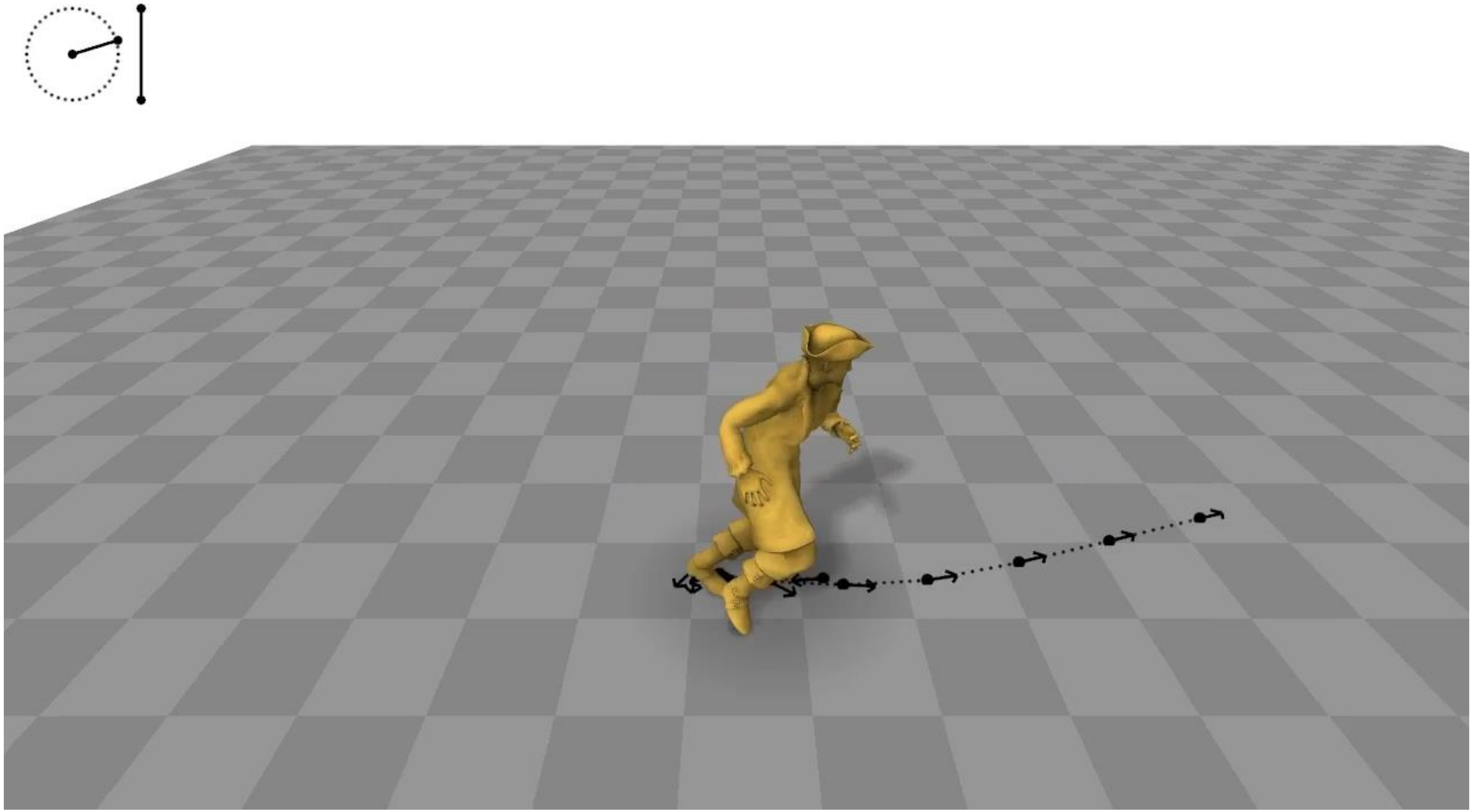
$$x_t = f\left(x_{t-1}; c_t, \theta_t = \sum_i w_i(\phi_t)\theta_i\right)$$



Cubic Catmull-Rom Spline:

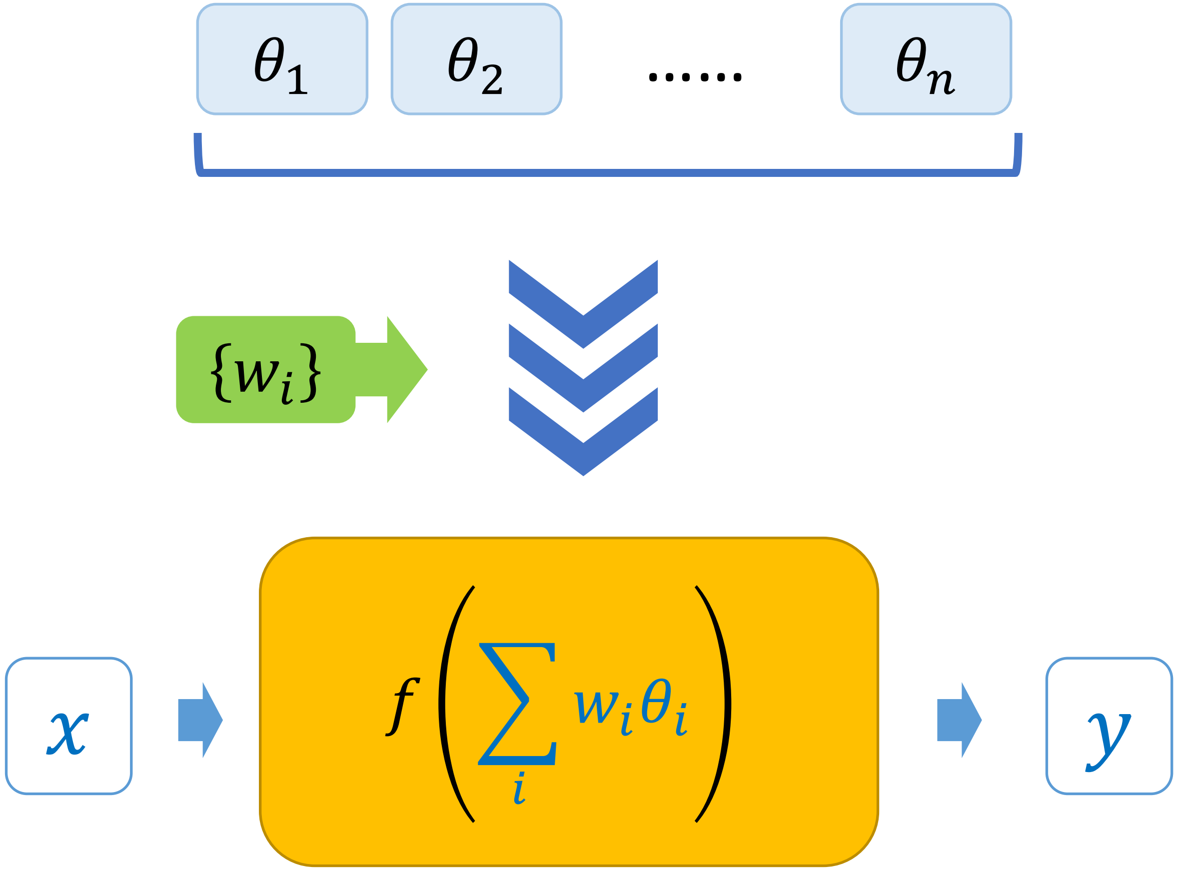
$$\begin{aligned}\theta_t &= \theta_2 \\ &+ \phi\left(-\frac{1}{2}\theta_1 + \frac{1}{2}\theta_3\right) \\ &+ \phi^2\left(\theta_1 - \frac{5}{2}\theta_2 + 2\theta_3 - \frac{1}{2}\theta_4\right) \\ &+ \phi^3\left(-\frac{1}{2}\theta_1 + \frac{3}{2}\theta_2 - \frac{3}{2}\theta_3 + \frac{1}{2}\theta_4\right)\end{aligned}$$

PFNN: Phase-Functioned Neural Networks

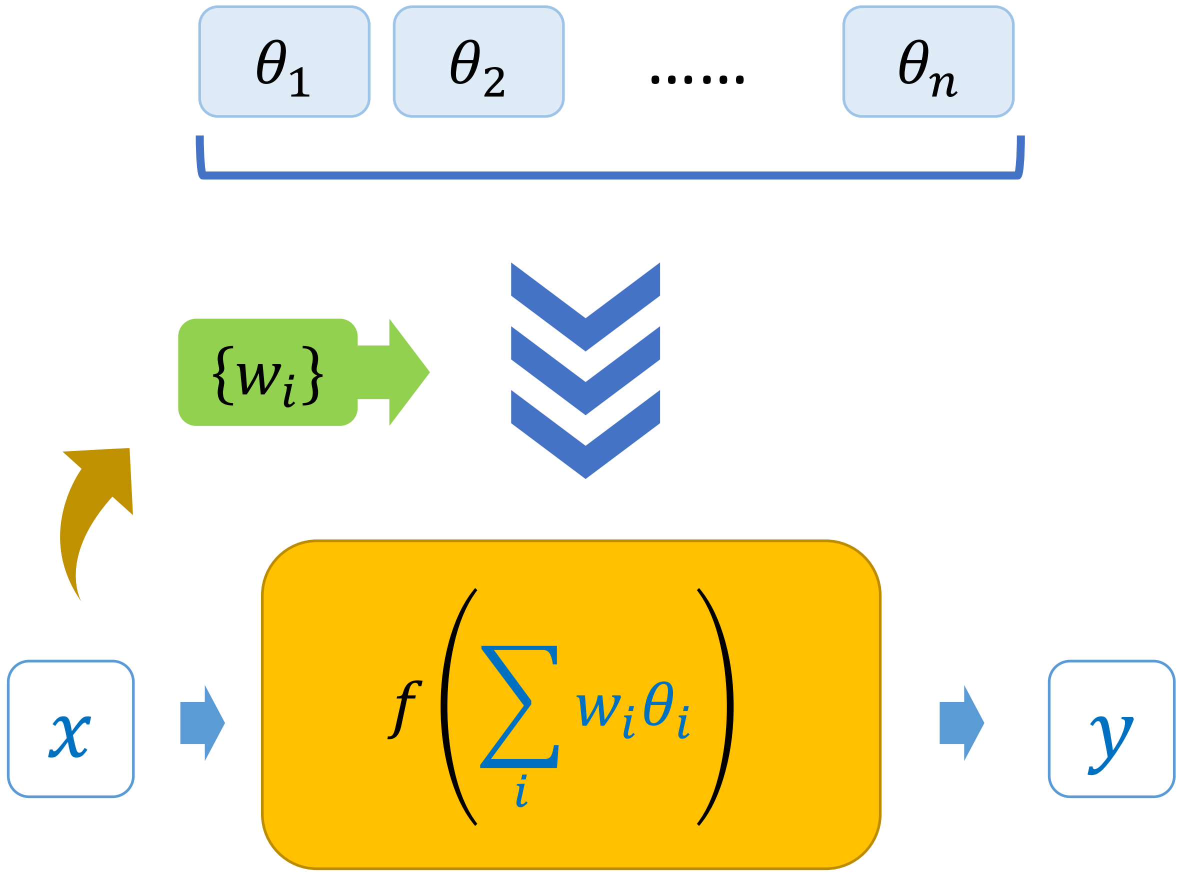


[Holden et al. 2017]

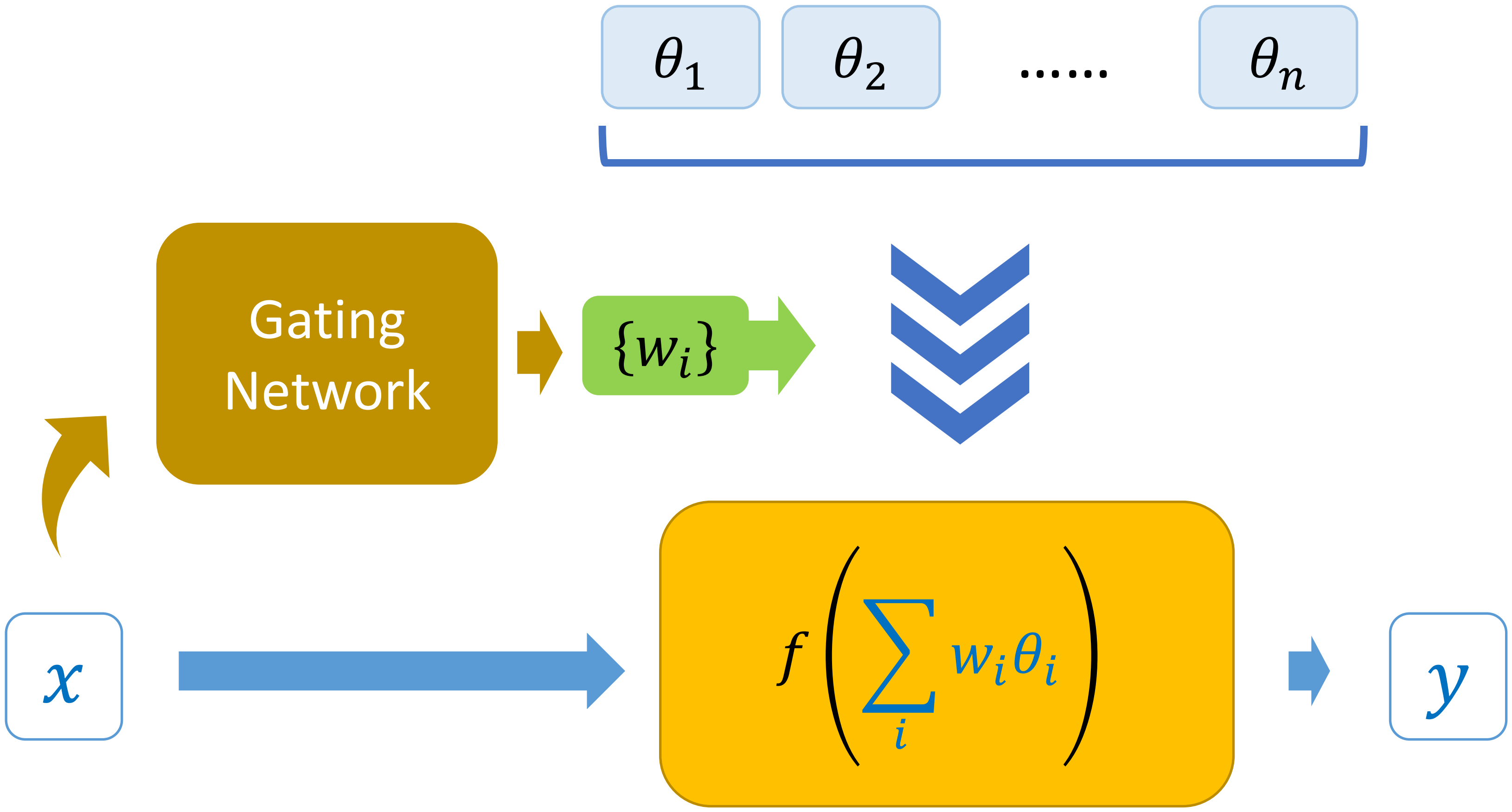
Advanced Phase Functions



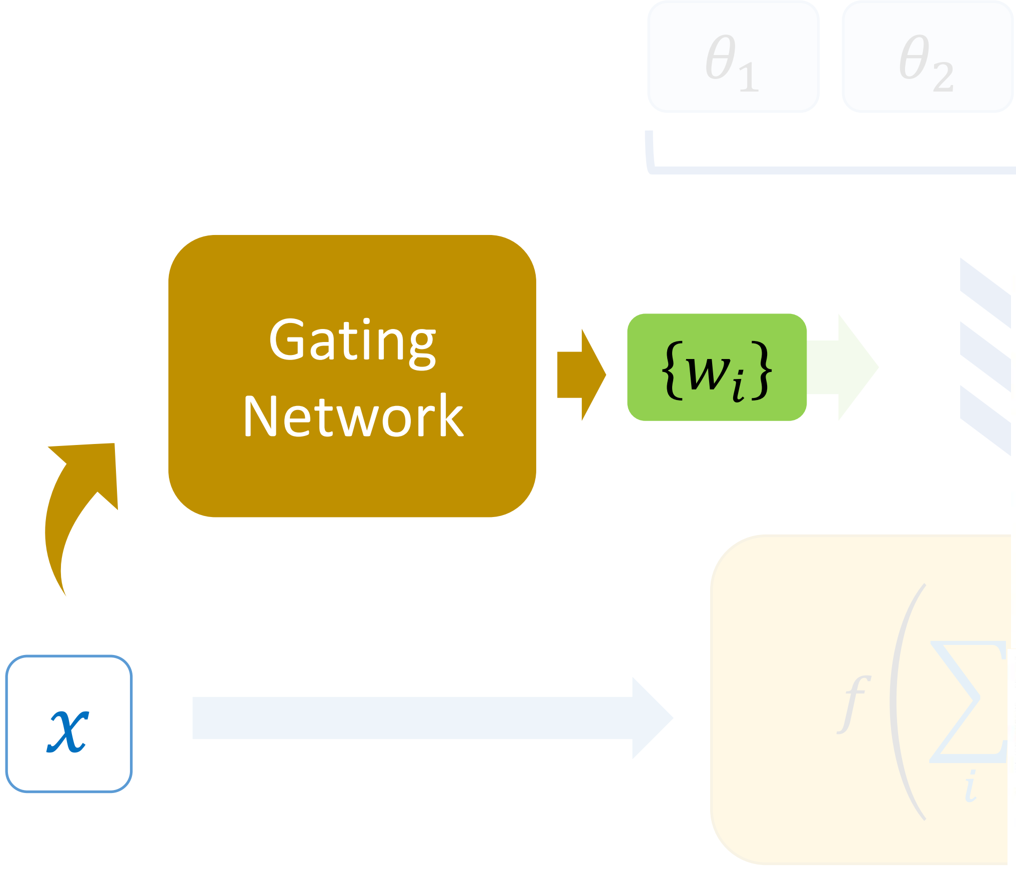
Advanced Phase Functions

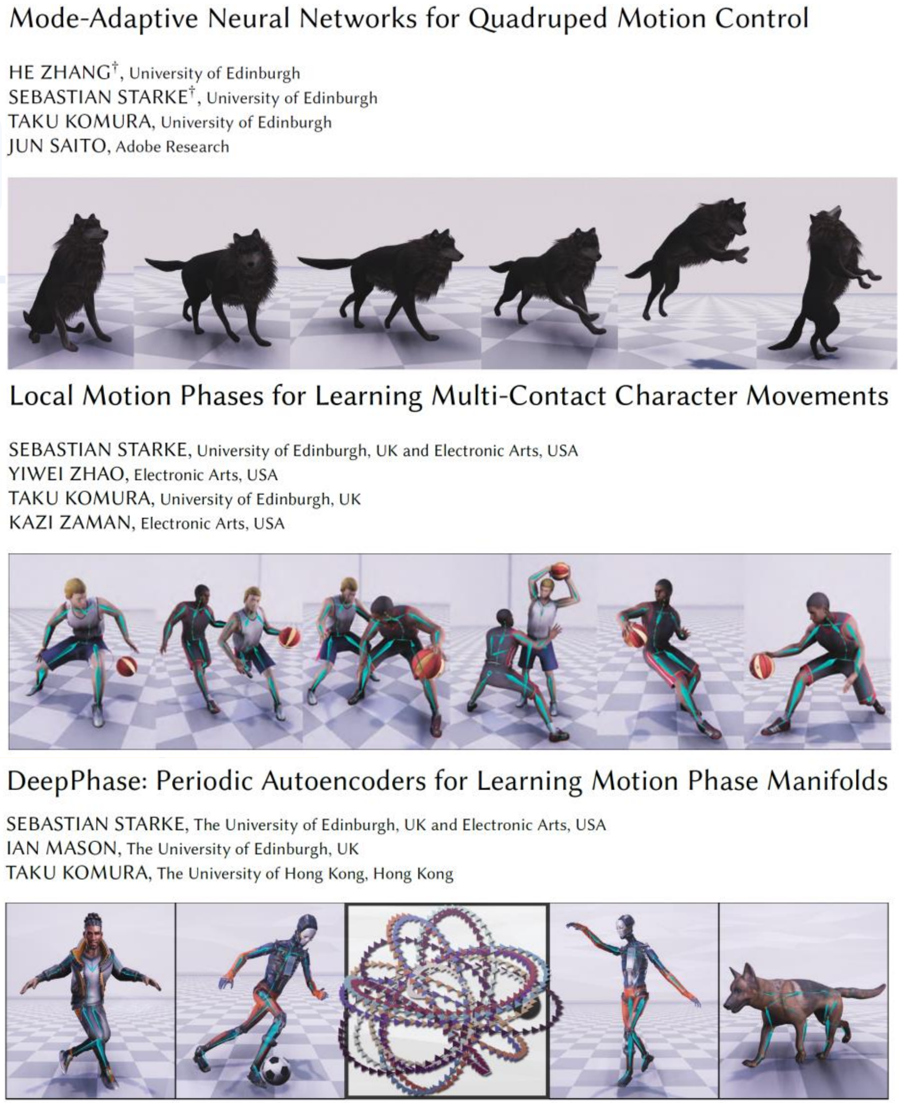


Advanced Phase Functions



Advanced Phase Functions





Mode-Adaptive Neural Networks for Quadruped Motion Control

HE ZHANG[†], University of Edinburgh
SEBASTIAN STARKE[†], University of Edinburgh
TAKU KOMURA, University of Edinburgh
JUN SAITO, Adobe Research

Local Motion Phases for Learning Multi-Contact Character Movements

SEBASTIAN STARKE, University of Edinburgh, UK and Electronic Arts, USA
YIWEI ZHAO, Electronic Arts, USA
TAKU KOMURA, University of Edinburgh, UK
KAZI ZAMAN, Electronic Arts, USA

DeepPhase: Periodic Autoencoders for Learning Motion Phase Manifolds

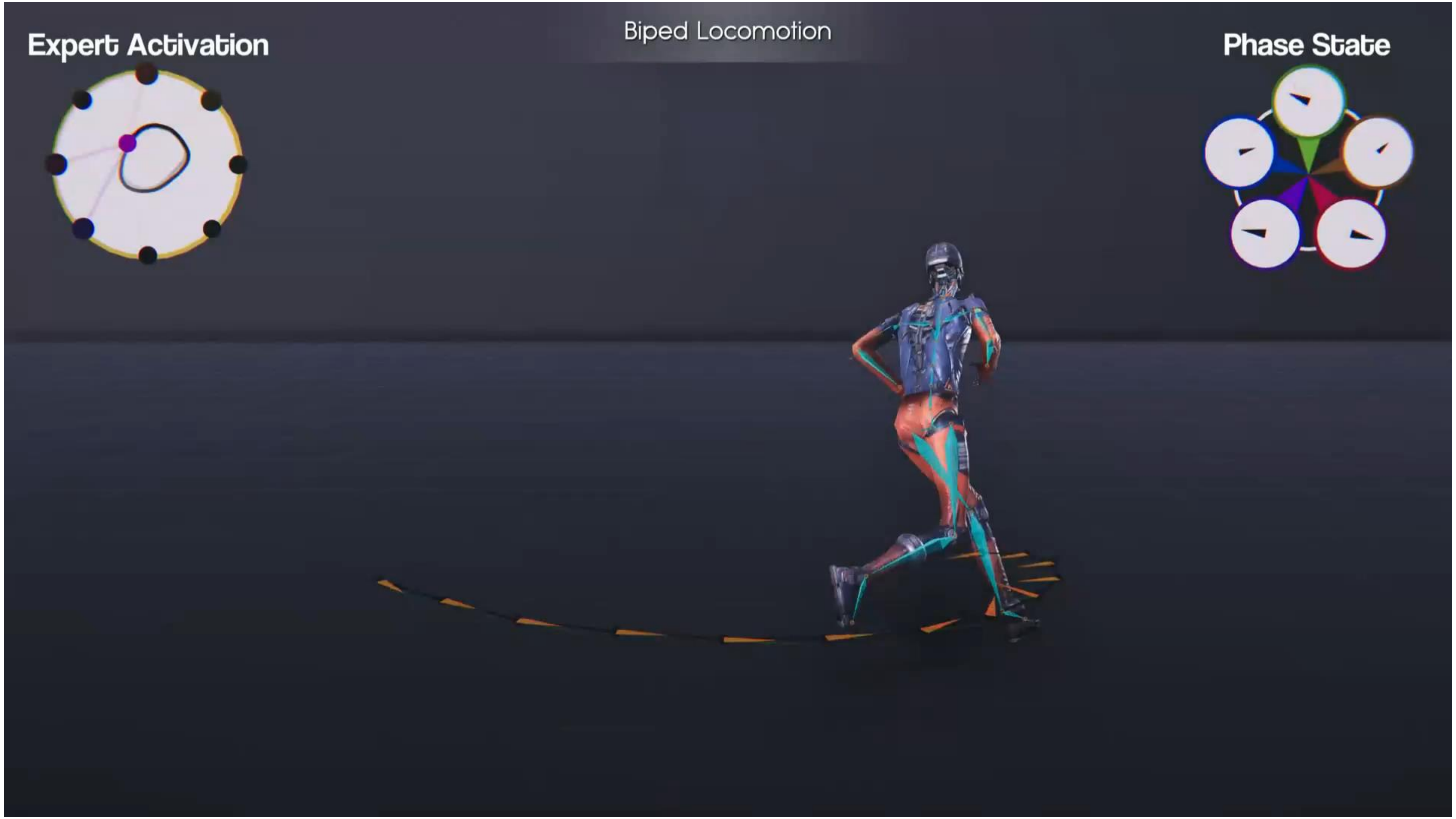
SEBASTIAN STARKE, The University of Edinburgh, UK and Electronic Arts, USA
IAN MASON, The University of Edinburgh, UK
TAKU KOMURA, The University of Hong Kong, Hong Kong

*SIGGRAPH 2018

*SIGGRAPH 2020

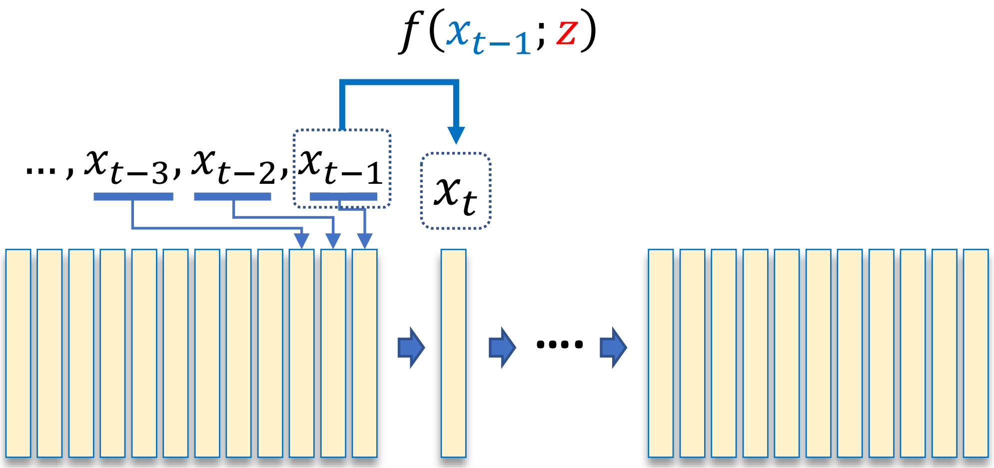
*SIGGRAPH 2022

Advanced Phase Functions



[Starke et al. 2022]

Generative Models

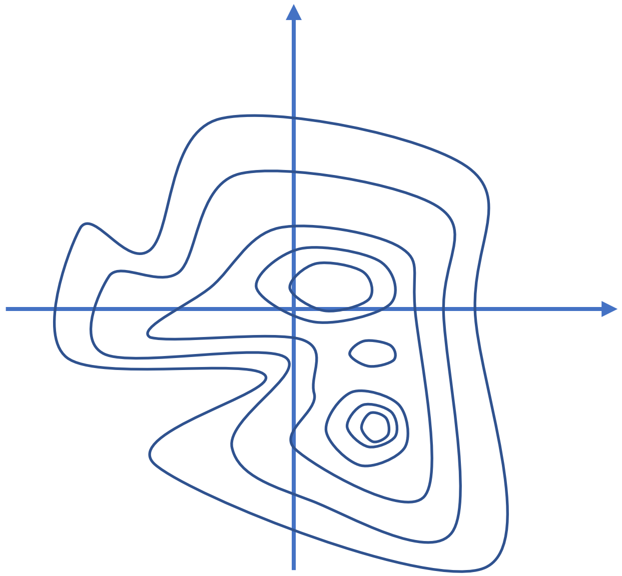


$$p(X|z) = p(x_1, \dots, x_T|z)$$

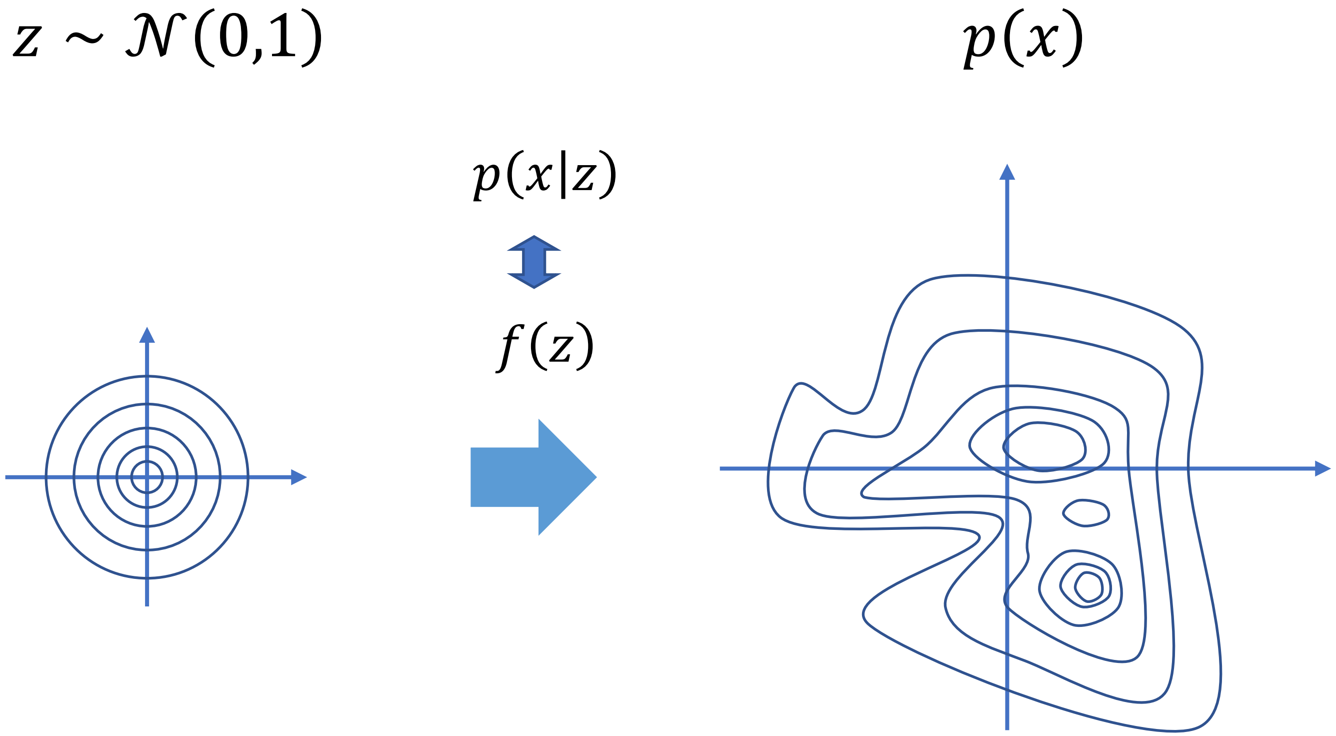
$$= p(x_1) \prod_t p(x_t|x_{t-1}; z)$$

Generative Models

$$p(x)$$

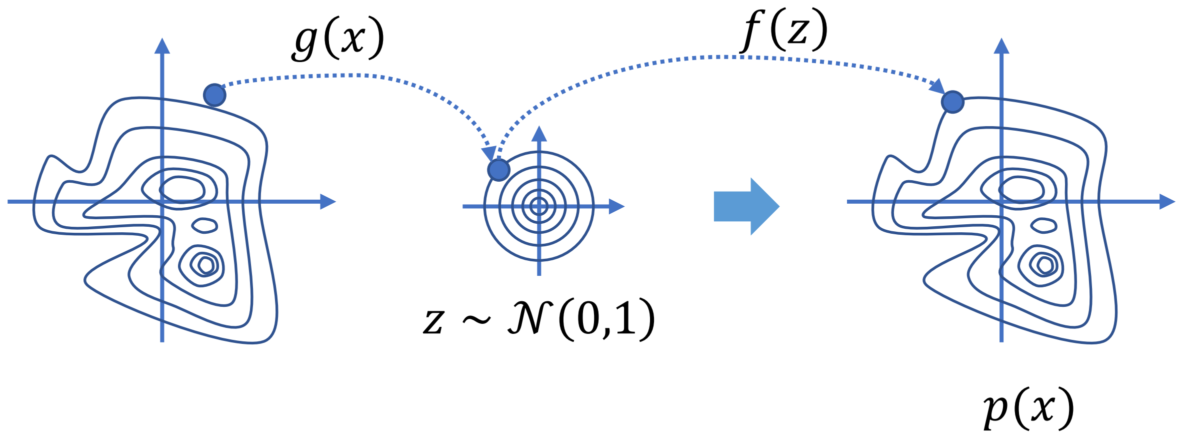


Generative Models

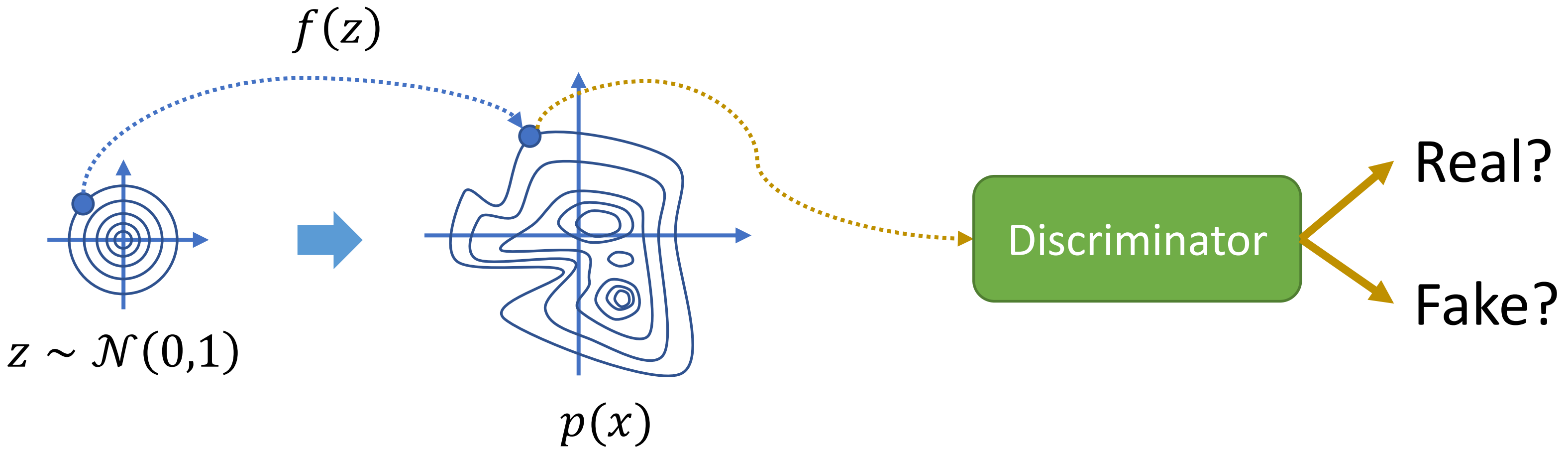


Generative Models

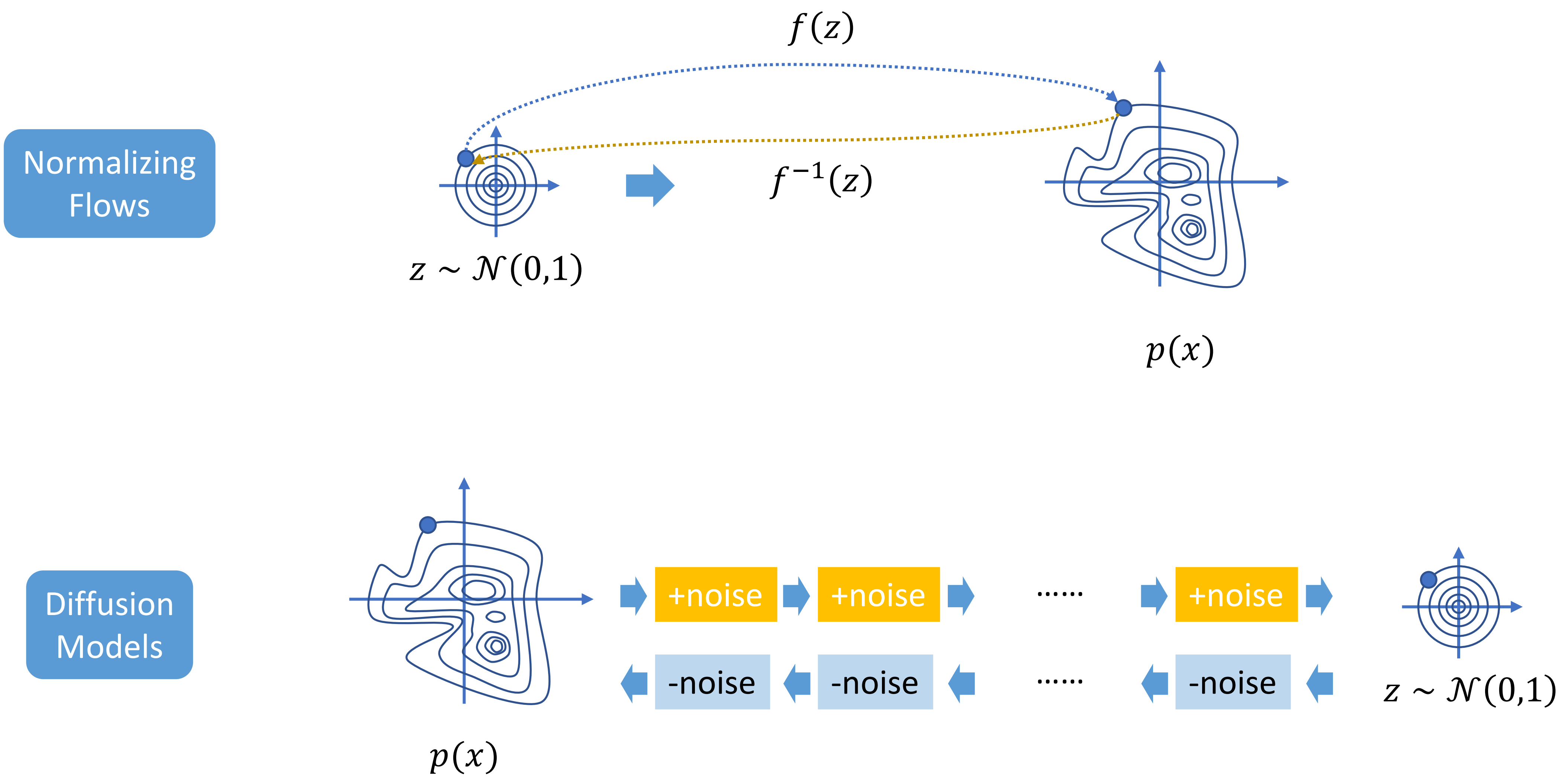
Variational Autoencoders



Generative Adversarial Network



Generative Models

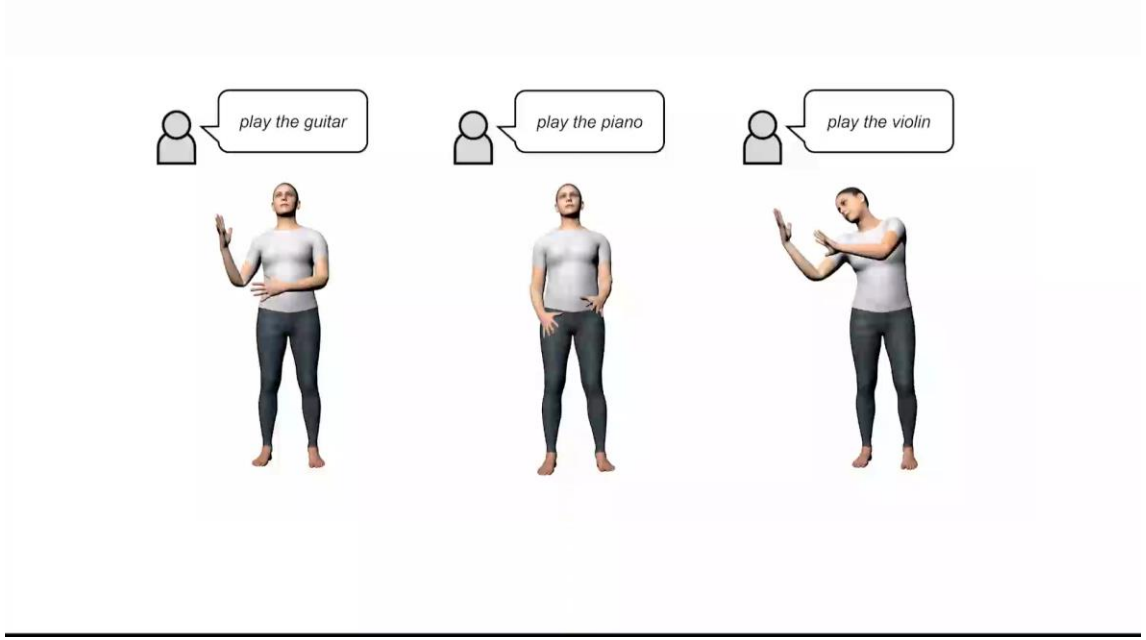


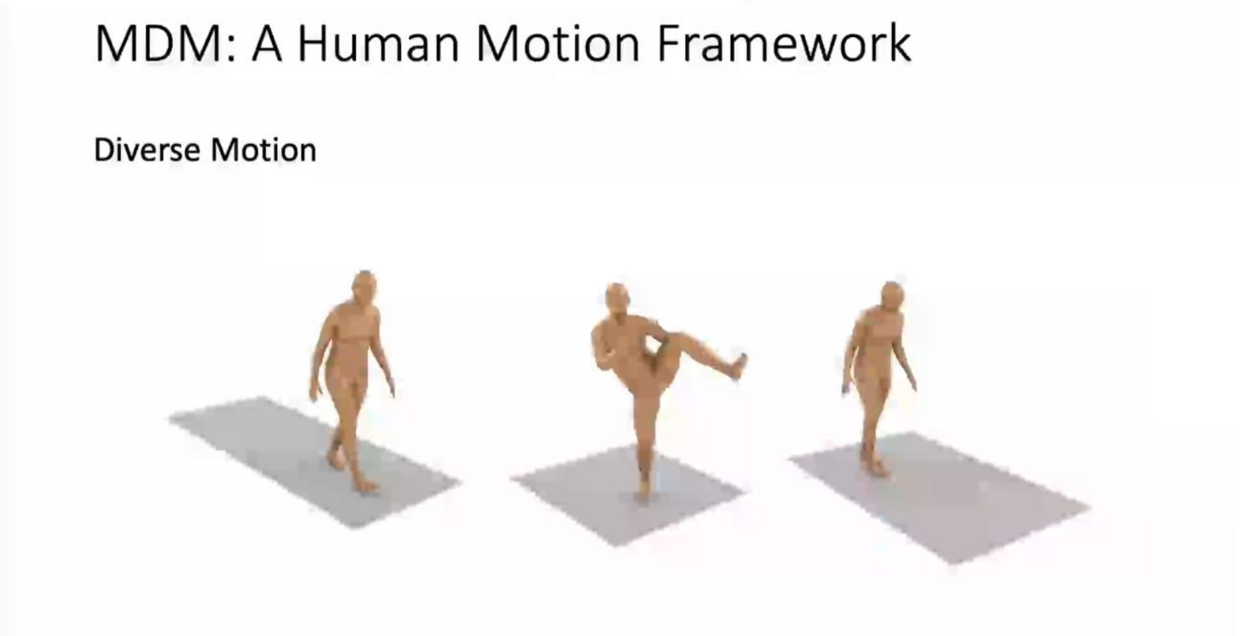
Generative Models

[Ling et al. 2021 Character Controllers Using Motion **VAEs**]

[Henter et al. 2020, MoGlow: Probabilistic and Controllable Motion Synthesis Using **Normalising Flows**]

Generative Models





[Zhang et al. 2022, arXiv, MotionDiffuse: Text-Driven Human Motion Generation with **Diffusion Model**]

[Tevet et al. 2022, arXiv, MDM: Human Motion **Diffusion Model**]

Outline

- Learning-based Character Animation (cont.)
  - Motion Models
  - Autoregressive models: PFNN
  - Generative models

Questions?