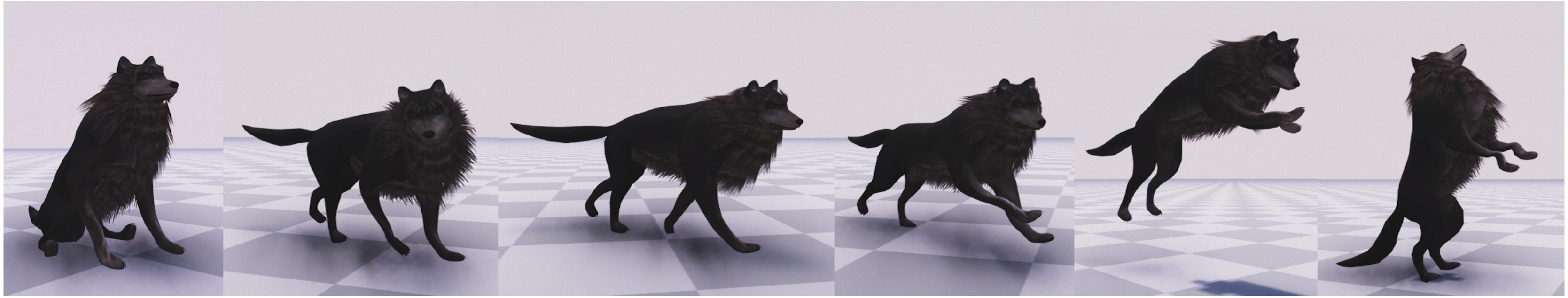
Mode-Adaptive Neural Networks for Quadruped Motion Control

He Zhang

he.zhang@ed.ac.uk

CGVU Group, Informatics School, University of Edinburgh

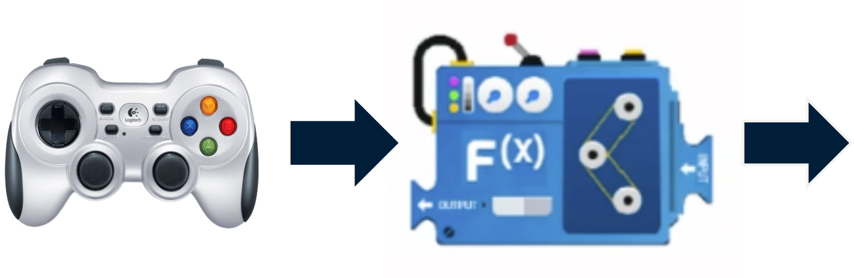


OUTLINE

- Research Background.
- Related Works.
- Mode-Adaptive Neural Networks.
- Discussion and Summary.

RESEARCH GOAL(1)

- Building interactive character controllers.
- Synthesizing realistic and smooth character motions in real-time.

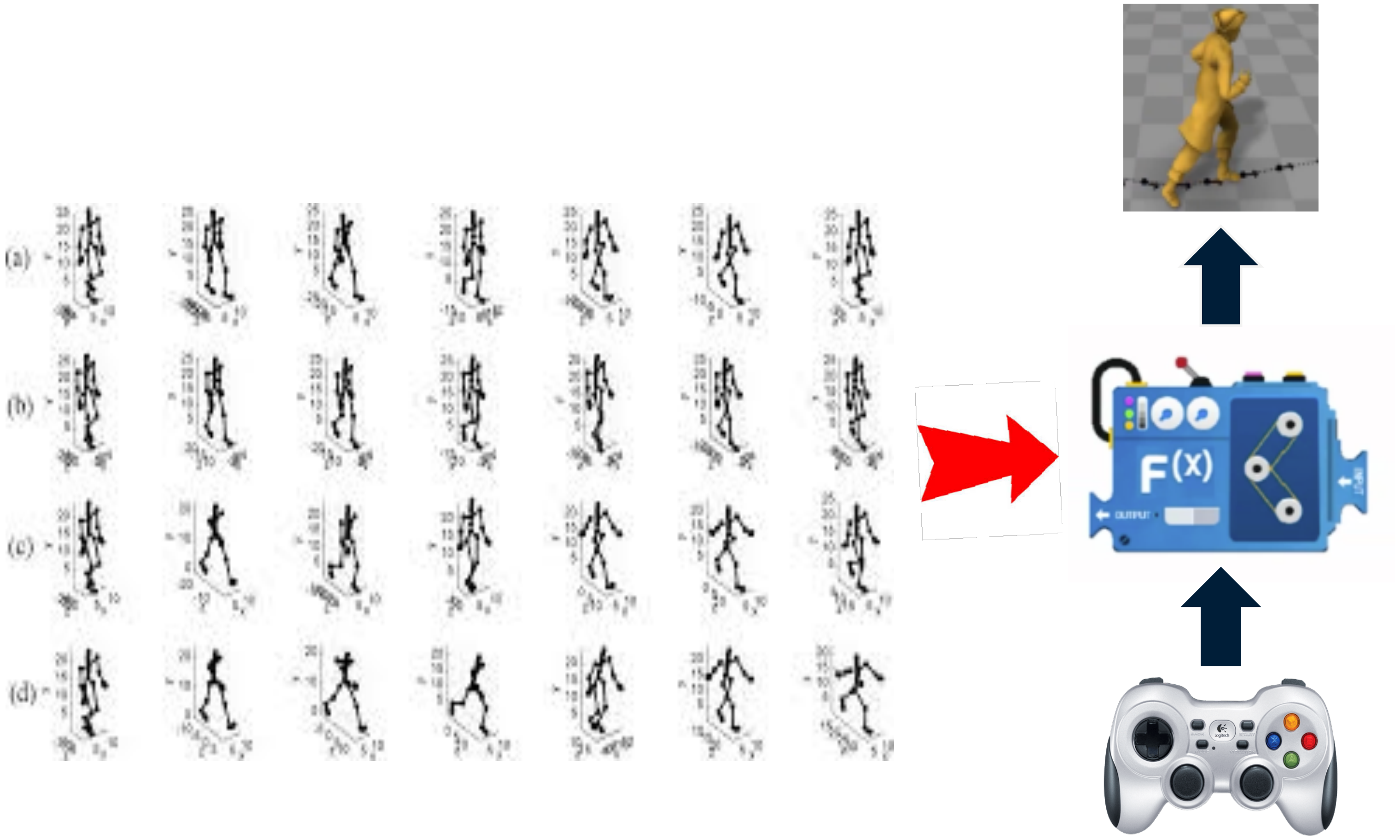


Control System

Example of character control
[Holden et al '17]

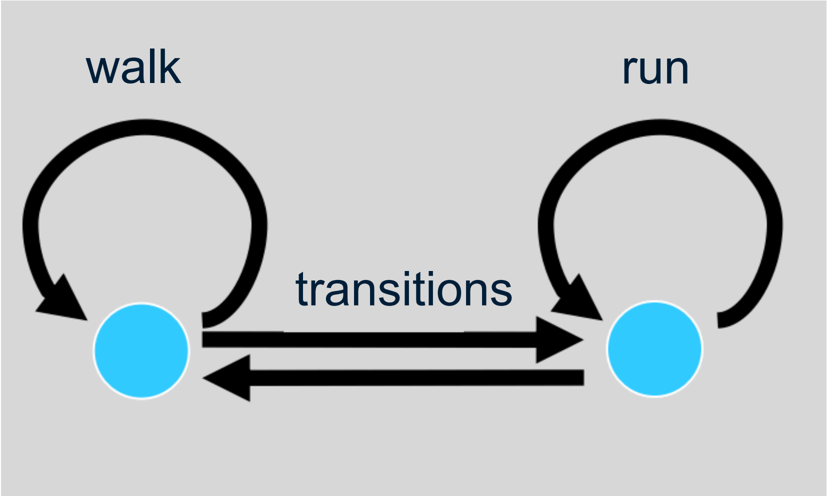
RESEARCH GOAL(2)

- Learn from a large data set:
  - Wide range of motions.
  - Small memory.
  - Fast in execution time.



RELATED WORKS(1)
DATA-DRIVEN CHARACTER CONTROLLERS

- Classic techniques:
  - Motion Graph [Kovar et al. 2002] [Lee et al. 2002] etc.
  - Motion Field [Lee et al. 2010]
  - Motion Matching [Clavet 2016]

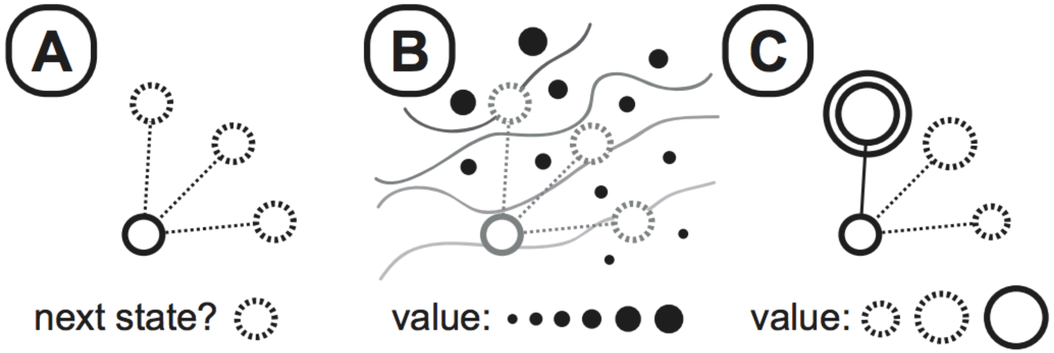


Structure of Motion Graph

  - Repeat motion clips, e.g. repeat walking cycle/ running cycle.
  - Interpolate to get the transitions, e.g. interpolate between walking and running to get transitions.

RELATED WORKS(1)
DATA-DRIVEN CHARACTER CONTROLLERS

- Classic techniques:
  - Motion Graph [Kovar et al. 2002] [Lee et al. 2002] etc.
  - Motion Field [Lee et al. 2010]
  - Motion Matching [Clavet 2016]



Structure of Motion Field

  - Search for K-Nearest poses for current pose from database.
  - Choose/blend from K-NN poses to get the next pose which satisfies user command best.
  - Using tricky structure for better searching, e.g. K-D trees.

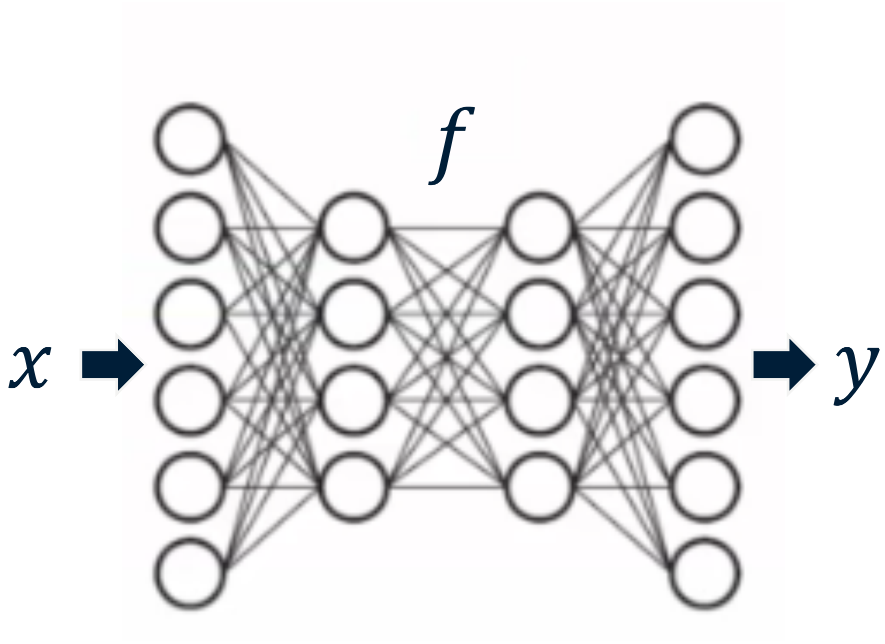
RELATED WORKS(1)
DATA-DRIVEN CHARACTER CONTROLLERS

- Classic techniques:
  - Motion Graph [Kovar et al. 2002] [Lee et al. 2002] etc.
  - Motion Field [Lee et al. 2010]
  - Motion Matching [Clavet 2016]

- Issues:
  - Require storing full motion database.
  - Require manual processing by artist, i.e. segmentation, labeling, mapping.
  - Require tricky structures (e.g.K-D trees)

RELATED WORKS(2)
DATA-DRIVEN CHARACTER CONTROLLERS

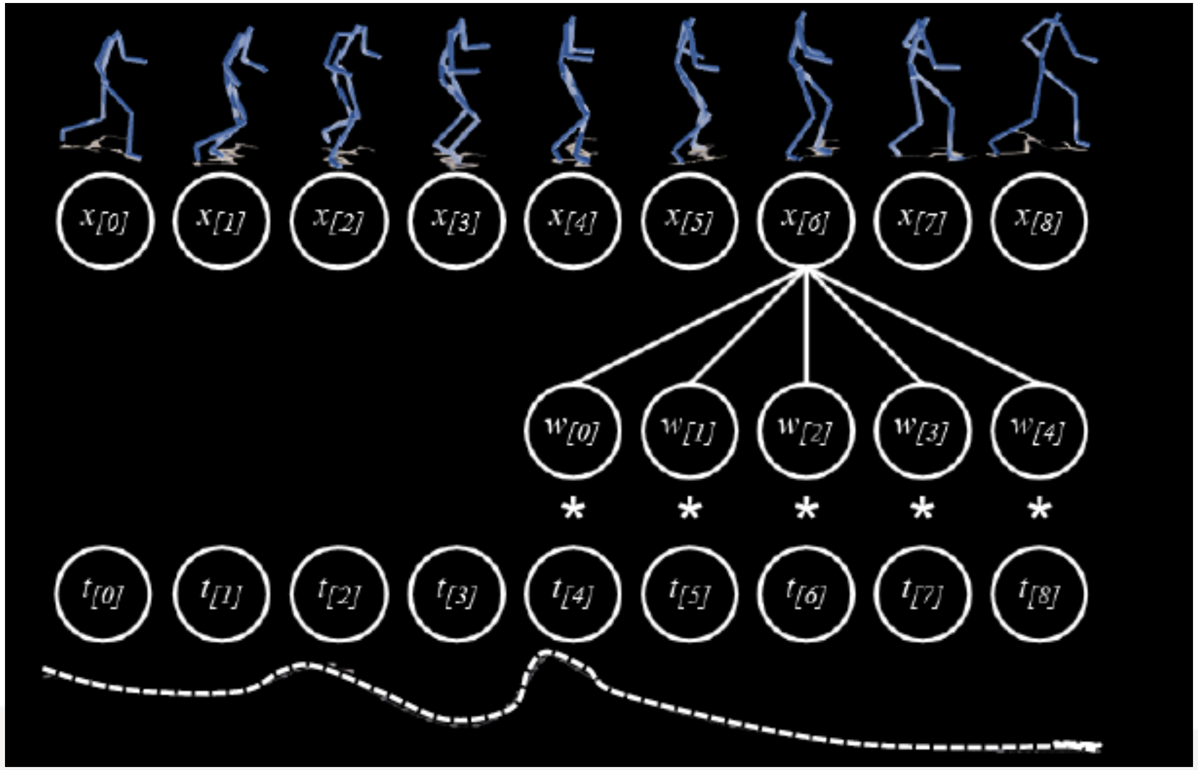
- Can Neural Networks Help?
  - Function Approximator (f)
- Advantage
  - Learn from large dataset.
  - Fast runtime / Low memory usage.



Example of Feed-Forward Neural Network

RELATED WORKS(2)
DATA-DRIVEN CHARACTER CONTROLLERS

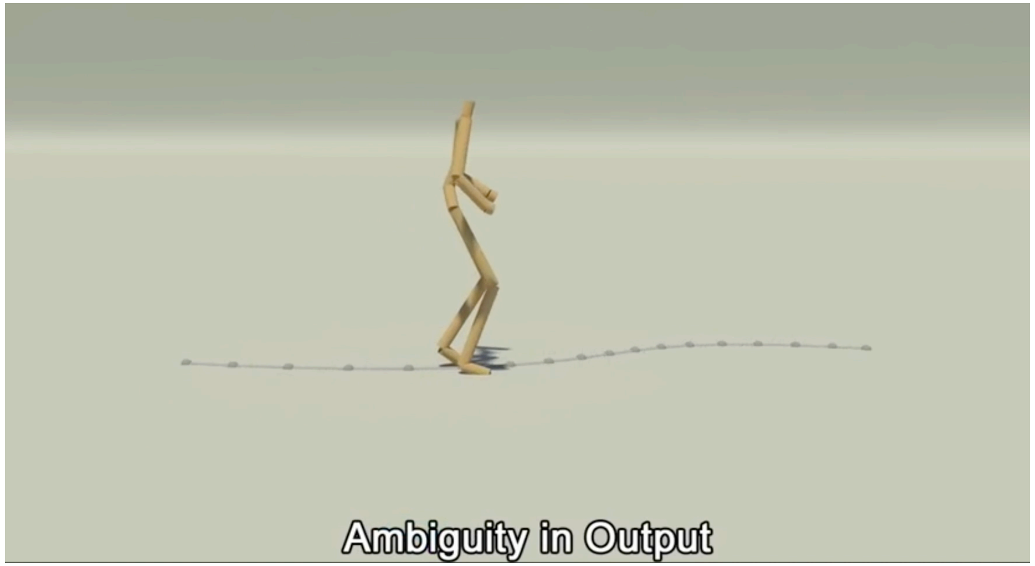
- Convolutional Neural Networks [Holden et al. 2016]
  - Learning a mapping from a user control signal to a motion.



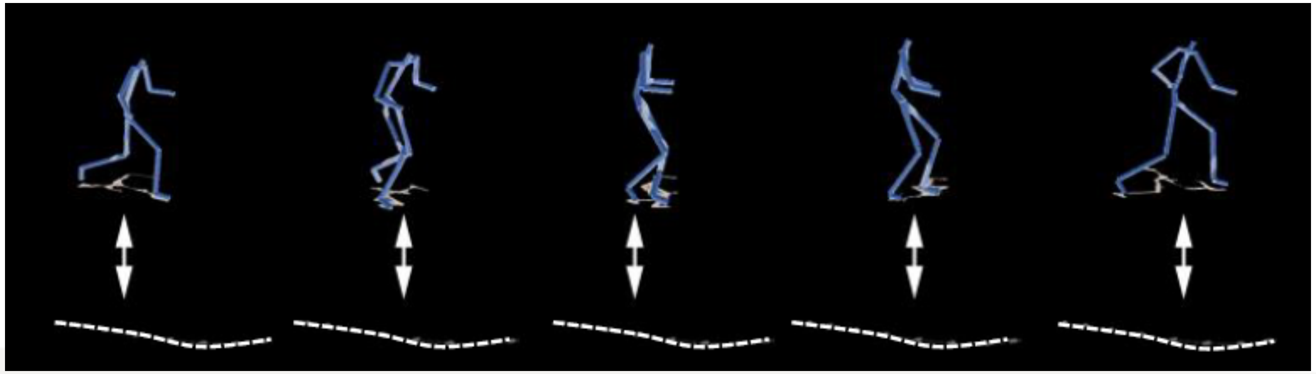
RELATED WORKS(2)
DATA-DRIVEN CHARACTER CONTROLLERS

- Convolutional Neural Networks [Holden et al. 2016]

- Issues
  - Ambiguous mapping between input and output.
  - Whole input trajectory must be given beforehand.
  - Muti-layer CNNs are still too slow.



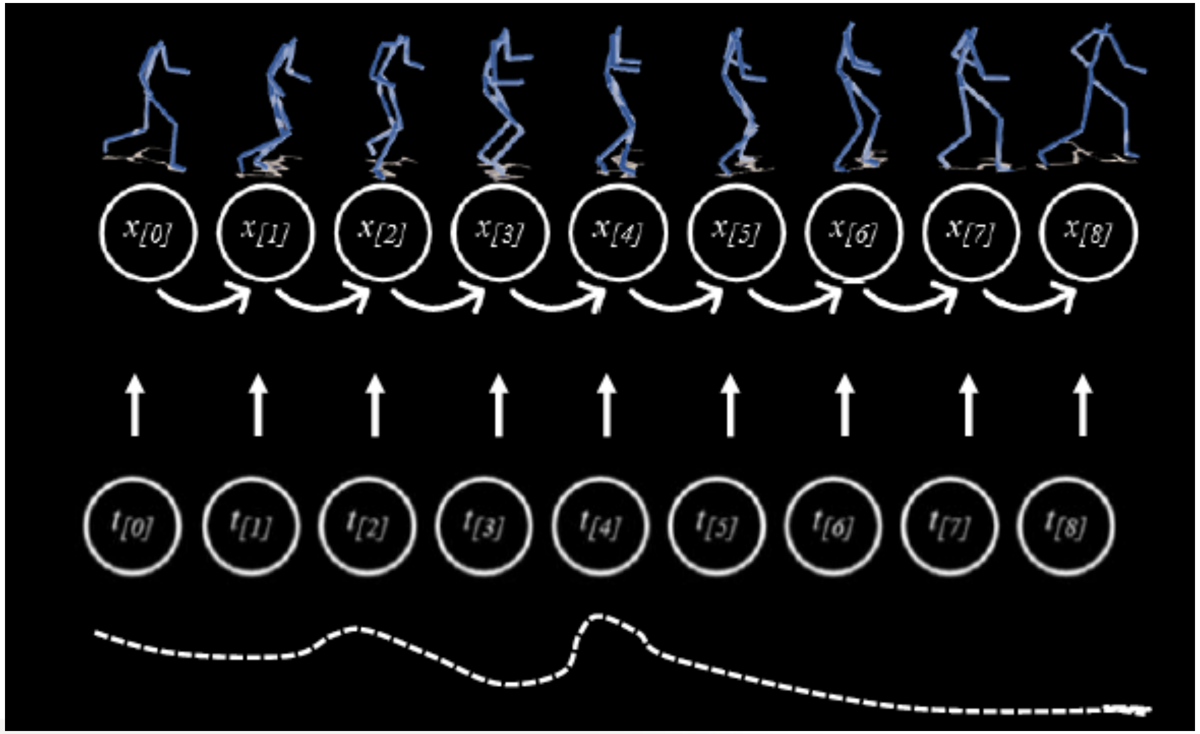
Issues of Floating caused by ambiguity



Same input trajectory can be mapped to different output

RELATED WORKS(2)
DATA-DRIVEN CHARACTER CONTROLLERS

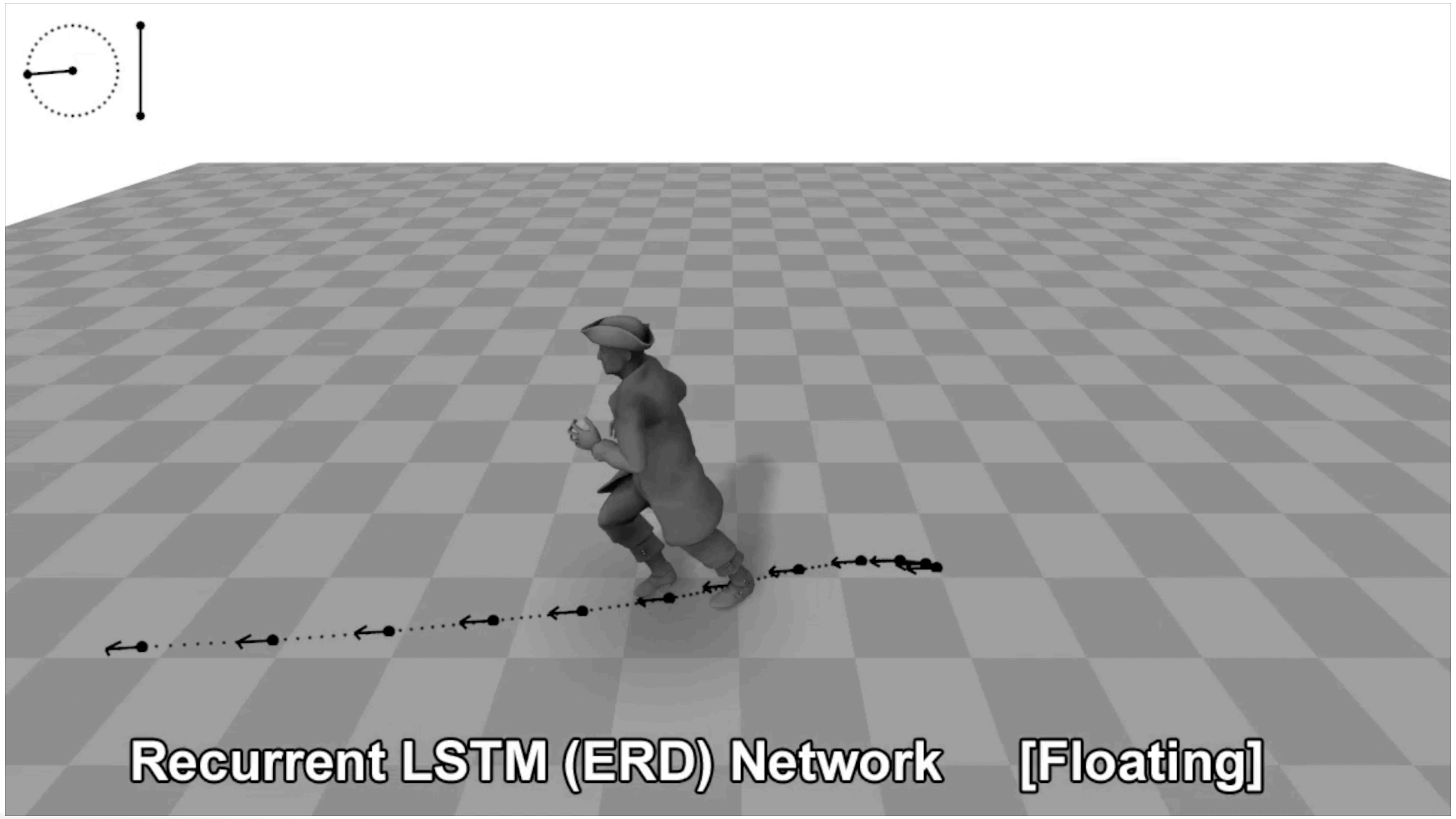
- Recurrent Neural Networks [Fragkiadaki et al. 2015]
  - Mapping from the previous frame(s) to next frame.



RELATED WORKS(2)
DATA-DRIVEN CHARACTER CONTROLLERS

- Recurrent Neural Networks [Fragkiadaki et al. 2015]

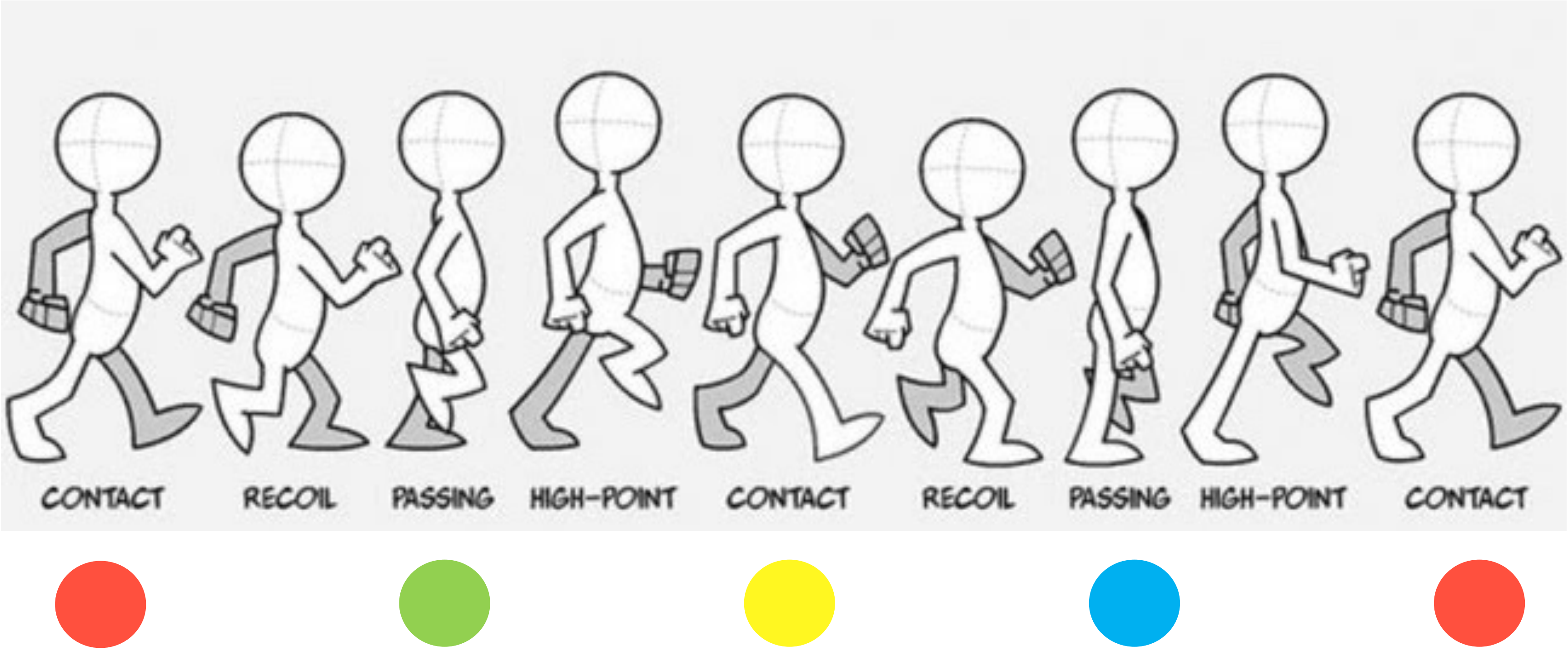
- Issues
  - Converge to average pose after ~10 seconds.
  - Difficult to avoid ”floating”.
  - Still has issues of ambiguity.



Issues of ‘floating’ still occurs in RNN model

RELATED WORKS(2)
DATA-DRIVEN CHARACTER CONTROLLERS

- Phase-functioned Neural Network [Holden et al. 2016]
  - Phase is introduced to segment the motion cycle.

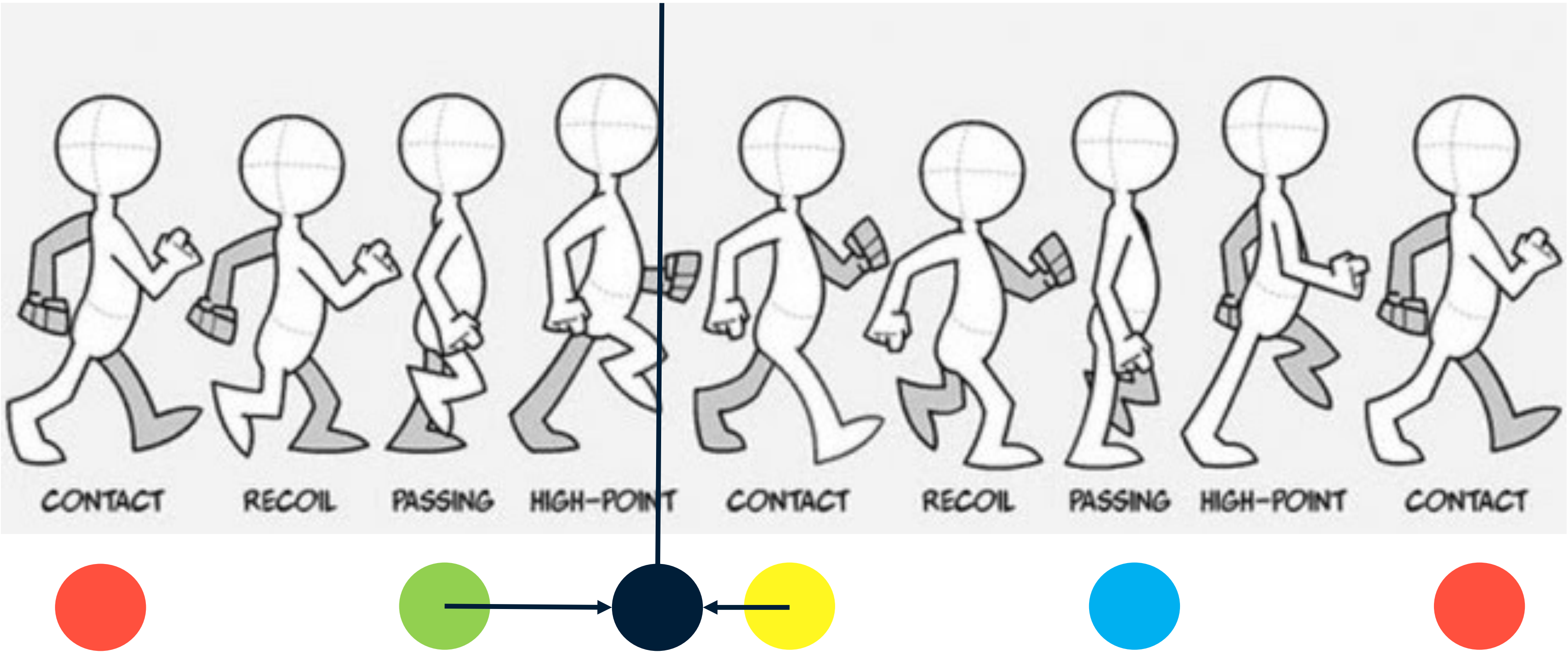


-4 control points

-4 neural networks

RELATED WORKS(2)
DATA-DRIVEN CHARACTER CONTROLLERS

- Phase-functioned Neural Network [Holden et al. 2016]
  - Phase is introduced to segment the motion.



-4 control points

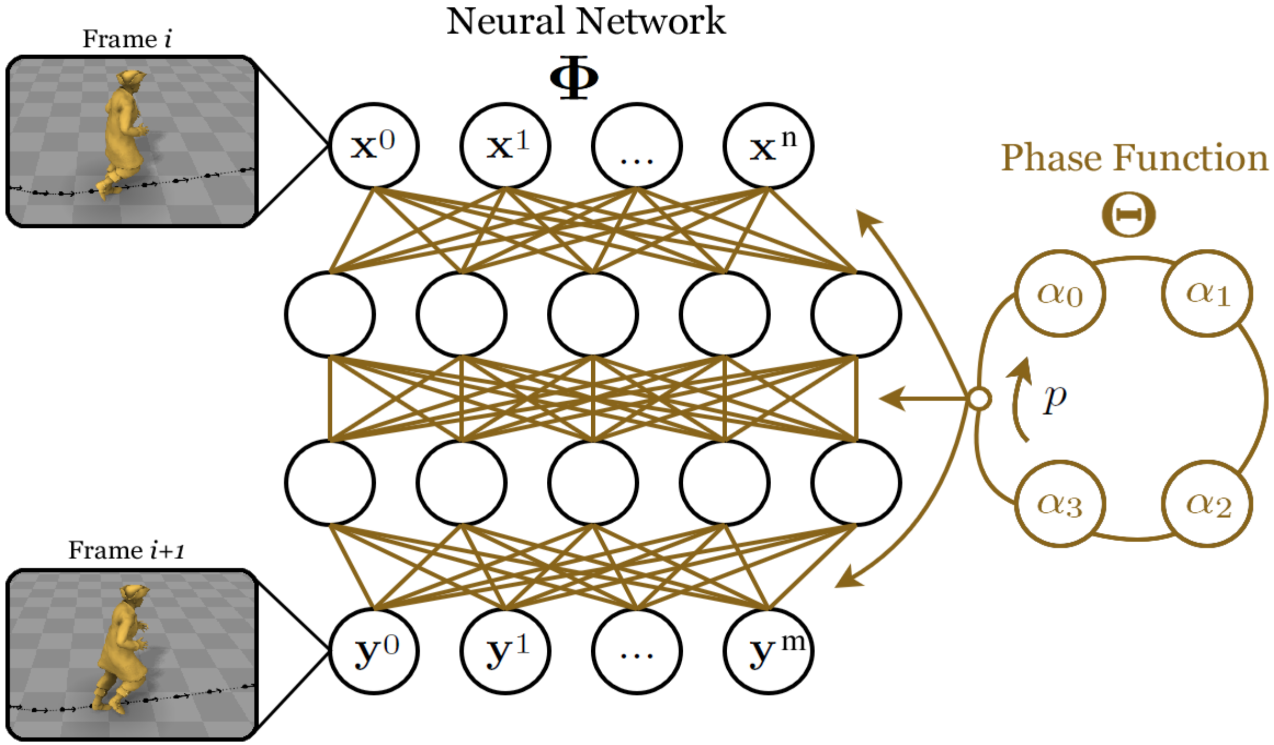
-4 neural networks

-current network weights

-linear blended by adjacent control points

RELATED WORKS(2)
DATA-DRIVEN CHARACTER CONTROLLERS

- Phase-functioned Neural Network [Holden et al. 2016]

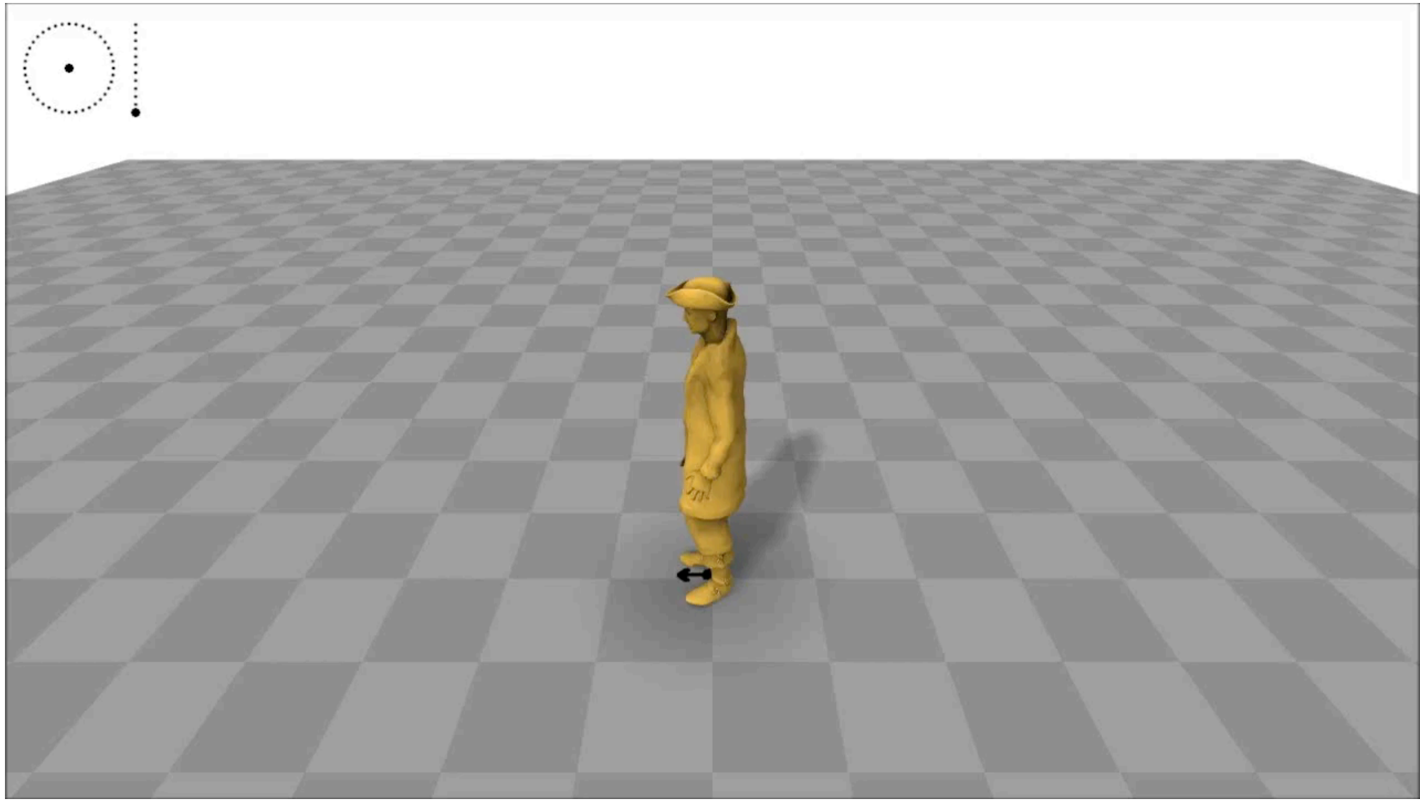


Model structure of PFNN

RELATED WORKS(2)
DATA-DRIVEN CHARACTER CONTROLLERS

- Phase-functioned Neural Network [Holden et al. 2016]

- Advantage of Phase
  - The pose of character is less ambiguous.
  - The space of poses is smaller and more convex.

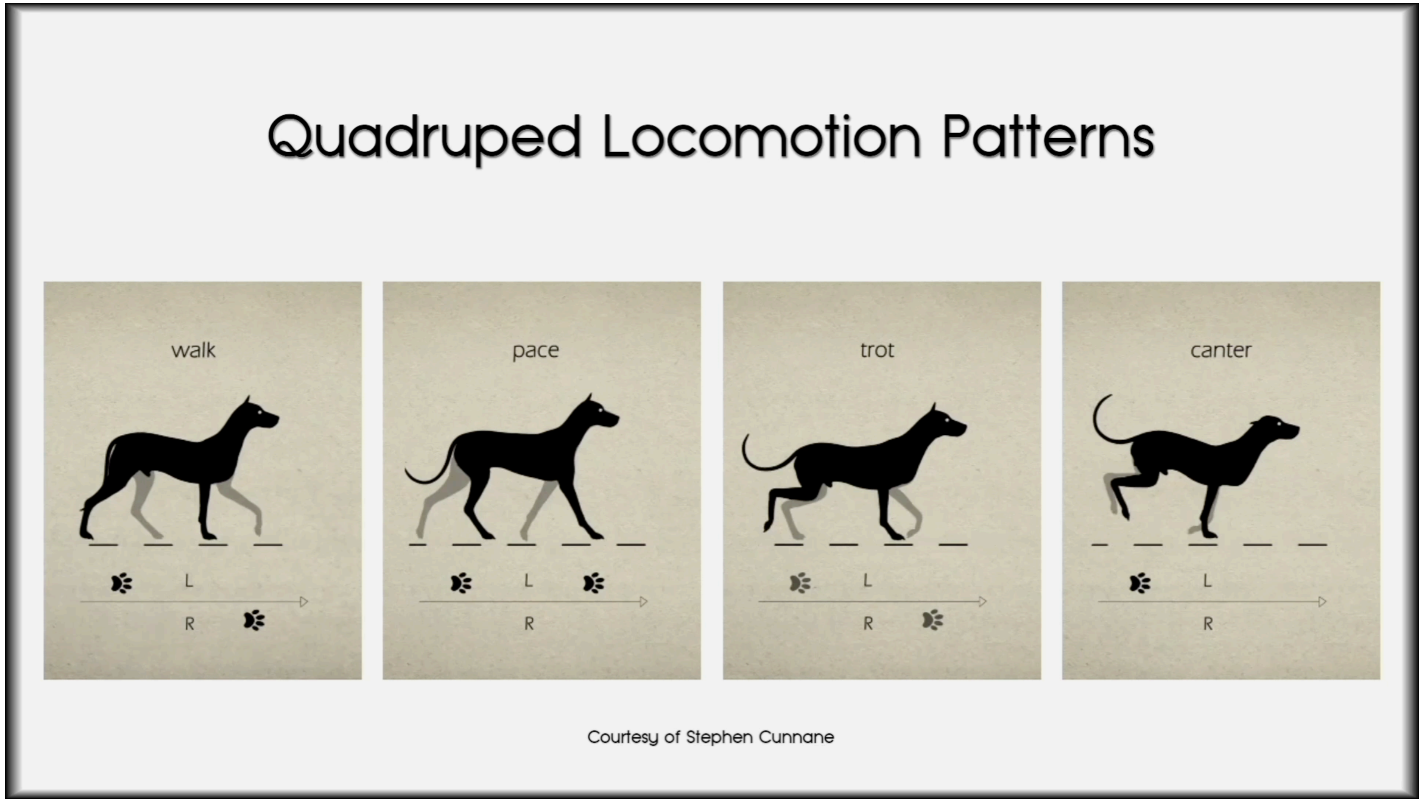


No floating issue in PFNN

RELATED WORKS(2)
DATA-DRIVEN CHARACTER CONTROLLERS

- Negative of PFNN
  - Require phase labels.
  - Cannot handle non-cyclic motions well.

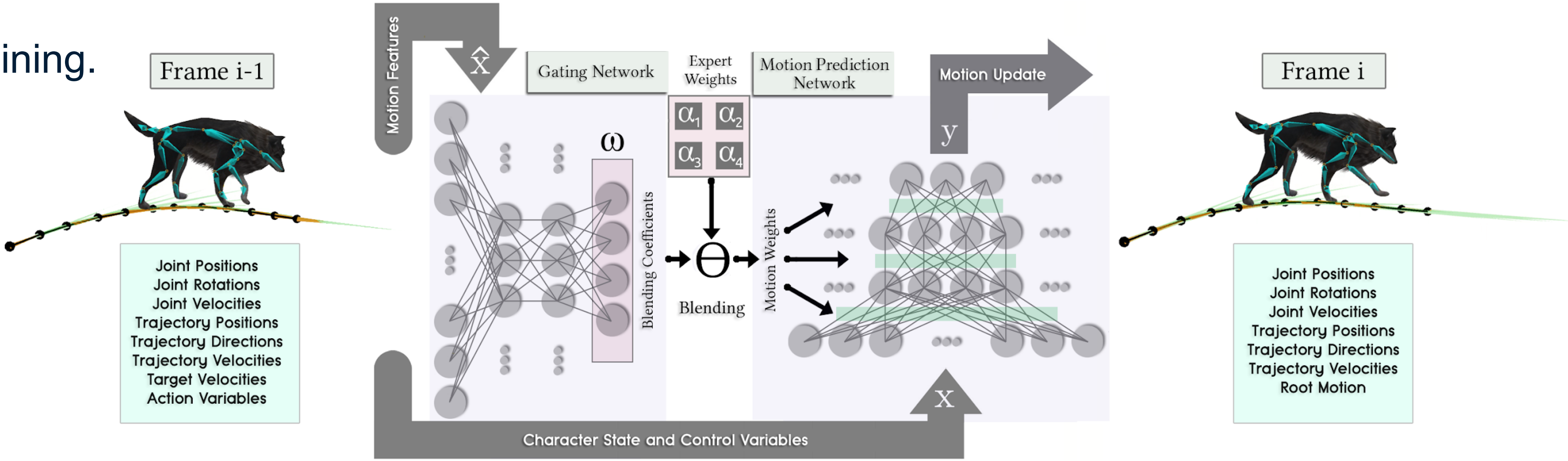
- Problems for quadruped motion capture data
  - Multi-modes and several actions.
  - Data are unstructured.
  - Non-cyclic motion, e.g. sitting, lying



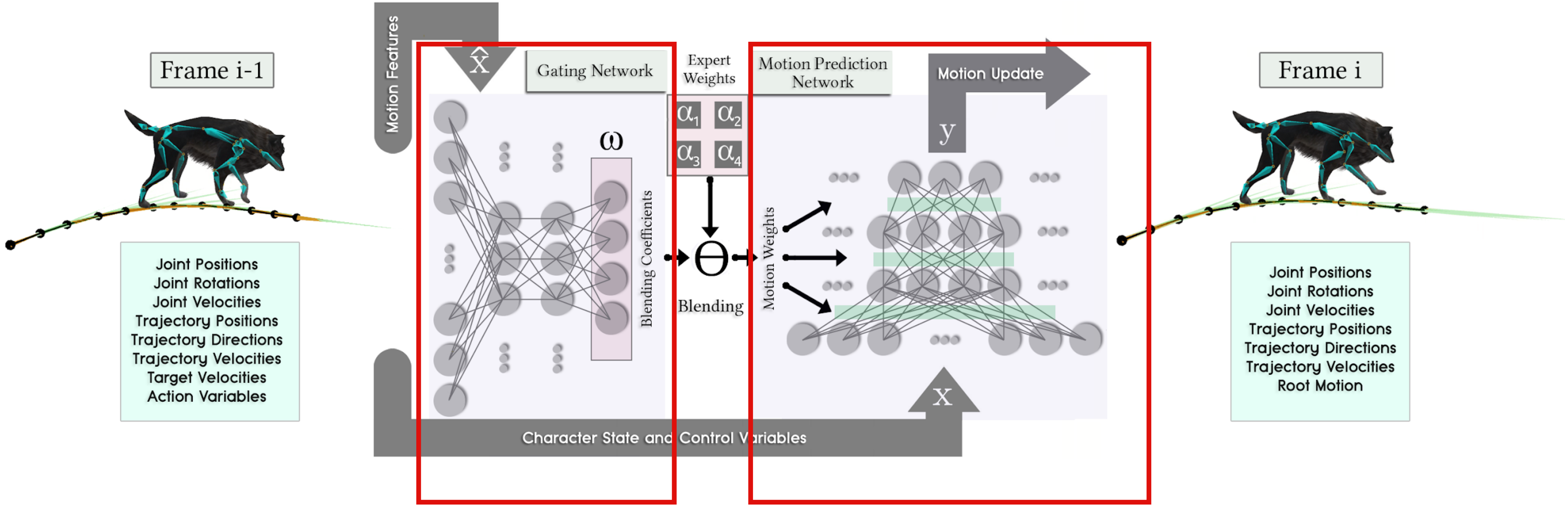
Quadruped motion capture data

MODE-ADAPTIVE NEURAL NETWORK OUTLINE

- Model Structure.
- Parameterization.
- Training.



MODE-ADAPTIVE NEURAL NETWORK MODEL STRUCTURE



Gating Network

Feed-Forward Network

2 hidden layers

32 hidden units per layer

elu, soft-max activation

Motion Prediction Network

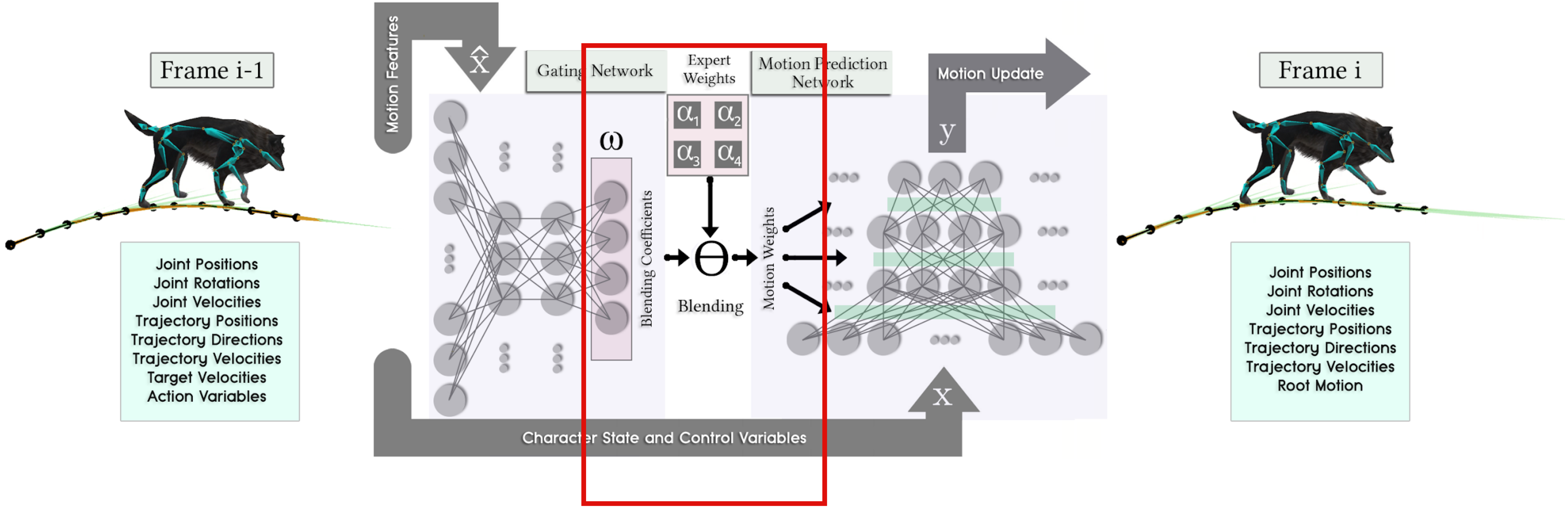
Feed-Forward Network

2 hidden layers

512 hidden units per layer

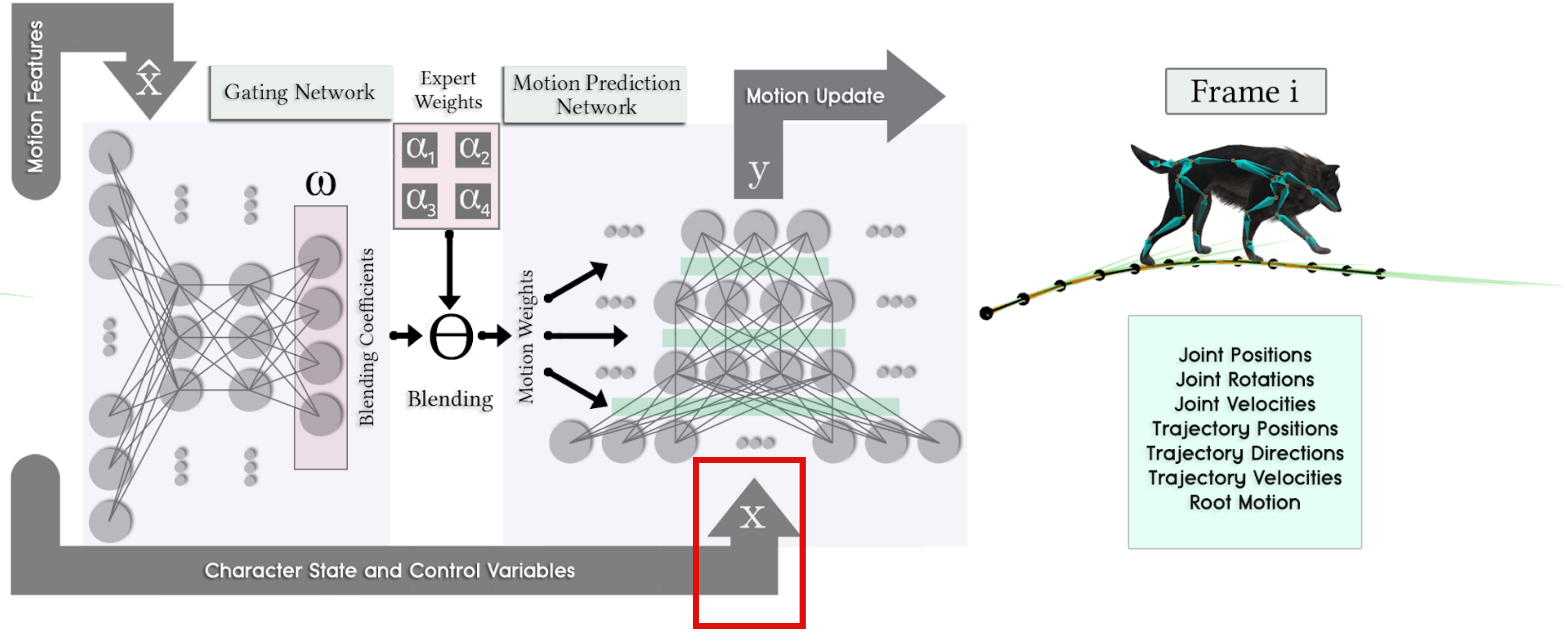
elu activation

MODE-ADAPTIVE NEURAL NETWORK MODEL STRUCTURE



Experts Blending: $\boldsymbol{\alpha} = \sum_{i=1}^{K} \omega_i \boldsymbol{\alpha}_i$

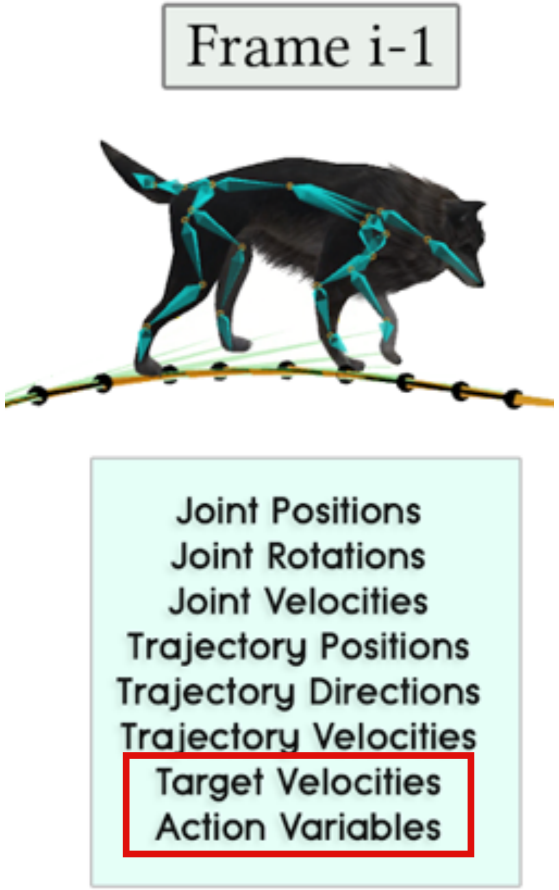
MODE-ADAPTIVE NEURAL NETWORK PARAMETERIZATION



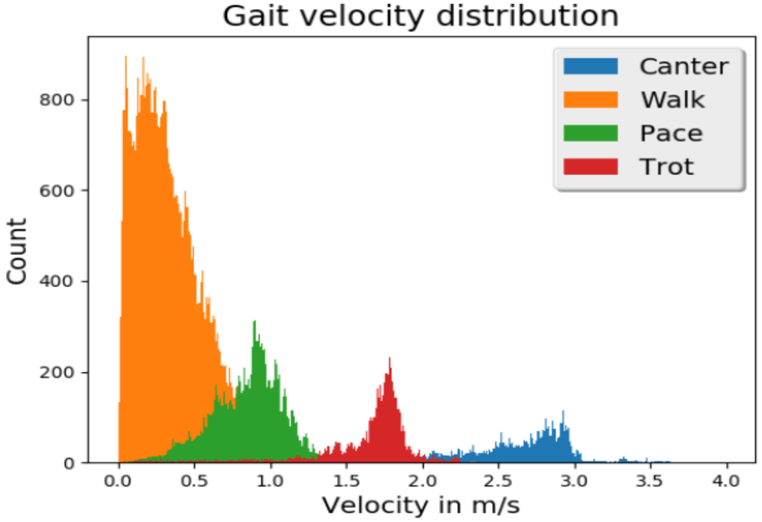
Input of System/Motion Prediction Network:

- Motion at previous frame.
- Trajectory at previous frame.

MODE-ADAPTIVE NEURAL NETWORK PARAMETERIZATION



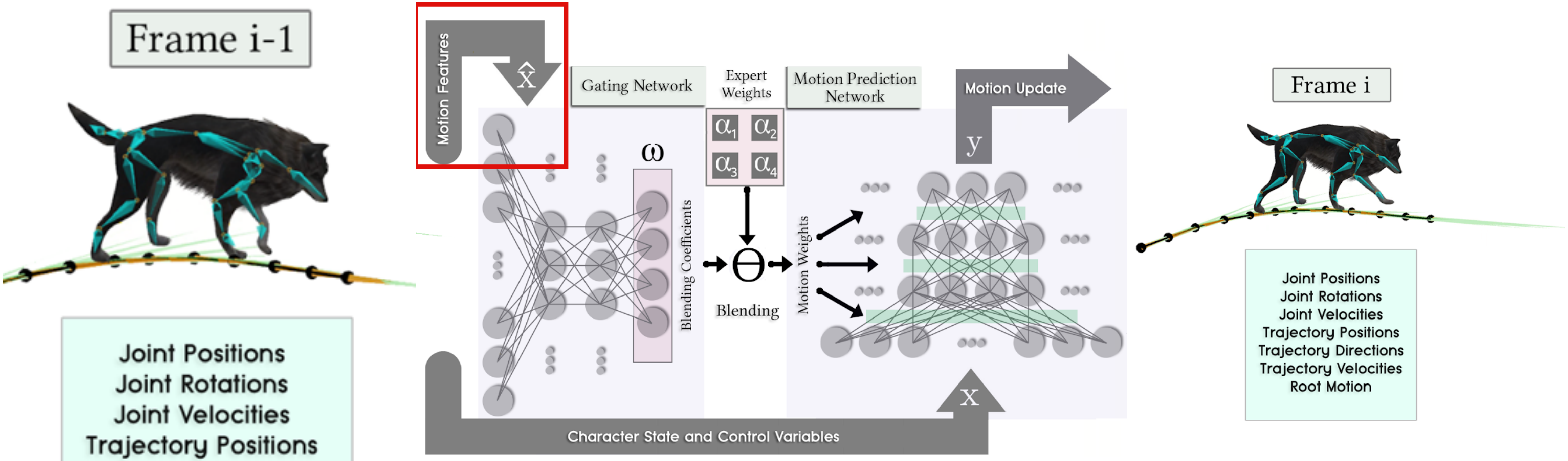
Motion type	time (sec)	frames	ratio (%)
idle	1433.70	86022	31.38
move	2190.62	131437	47.95
jump	35.50	2130	0.78
sit	528.70	31722	11.57
lie	307.63	18458	6.73
stand	72.07	4324	1.58



Action Control:

- 6 action signals which is labeled by one-hot vector.
- Target velocities control transitions between different gaits.

MODE-ADAPTIVE NEURAL NETWORK PARAMETERIZATION

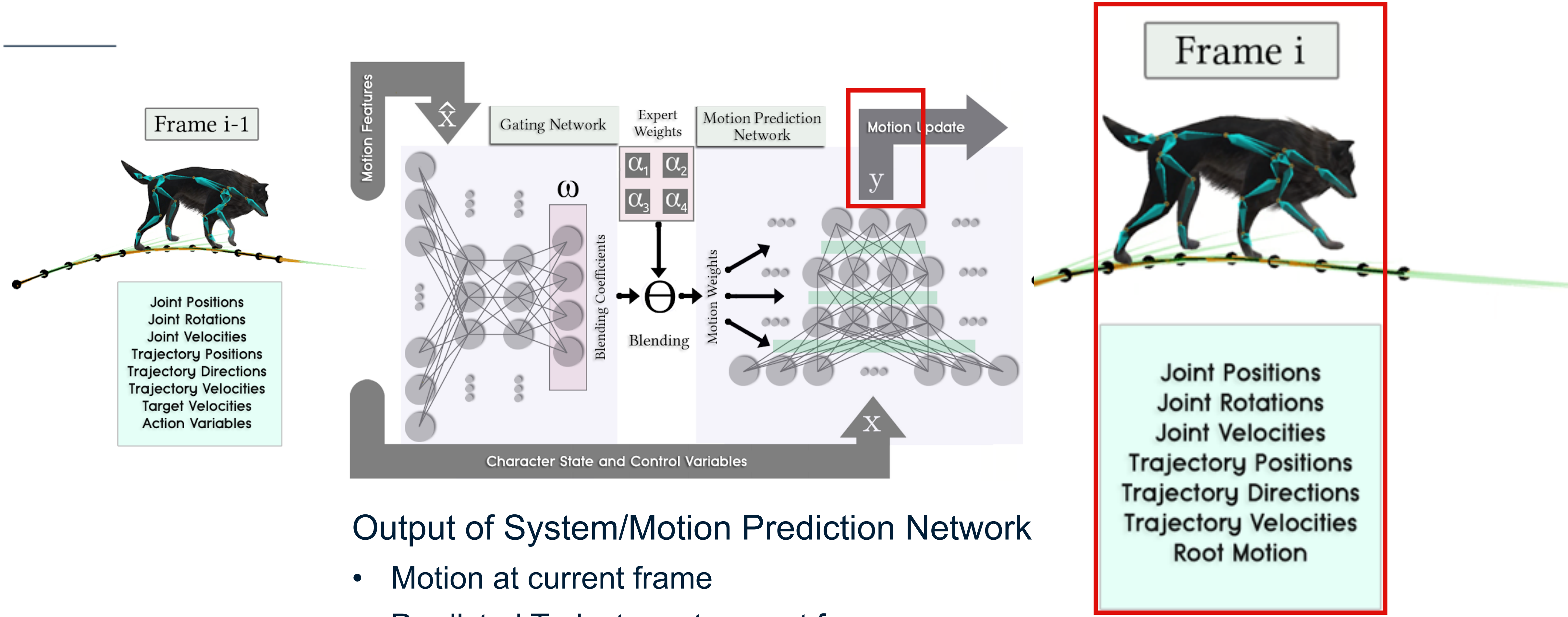


Joint Positions
Joint Rotations
Joint Velocities
Trajectory Positions
Trajectory Directions
Trajectory Velocities
Target Velocities
Action Variables

Input of Gating Network(Motion Features):

- Feet Joint Velocities at previous frame.
- Target Velocities at previous frame.
- Action Variables at previous frame.

MODE-ADAPTIVE NEURAL NETWORK PARAMETERIZATION



Output of System/Motion Prediction Network

- Motion at current frame
- Predicted Trajectory at current frame – for smooth transitions

MODE-ADAPTIVE NEURAL NETWORK TRANING

- Cost function:
  - Mean square error between the predicted error and the ground truth:

$$\text{Cost}(\mathrm{X}, \mathrm{Y}; \boldsymbol{\beta}, \mu) = \|\mathrm{Y} - \Theta(\mathrm{X}, \Omega(\hat{\mathrm{X}}; \mu); \boldsymbol{\beta})\|,$$

- Optimizer:
  - Stochastic gradient descent, AdamWR [Loshchilov and Hutter 2017]
- Training Time
  - 20/30 hours with 4/8 experts, respectively, using NVIDIA GeForce GTX 970 GPU

MODE-ADAPTIVE NEURAL NETWORK RESULT

- Compare with Standard NN and PFNN
  - Same number of layers and units.

MODE-ADAPTIVE NEURAL NETWORK RESULT

- What do the different experts learn?
  - Different modes corresponds to different combination of experts.
  - Some experts have learned features which are specifically responsible for certain motions/actions.

MODE-ADAPTIVE NEURAL NETWORK DISCUSSION

- Positive
  - No phase label needed
  - Can produce various high-quality locomotion modes
  - Can produce non-cyclic motions
- Negative
  - Training time
  - Artistic control
    - Difficult to edit the outcome

MODE-ADAPTIVE NEURAL NETWORK SUMMARY

- A novel time-series architecture to learn from a large unstructured quadruped motion capture dataset
- Allow the user to interactively control the velocity, direction and actions.
- End-to-end training without providing phase and gait labeling

- Project - https://github.com/ShikamaruZhang/AI4Animation

Q & A