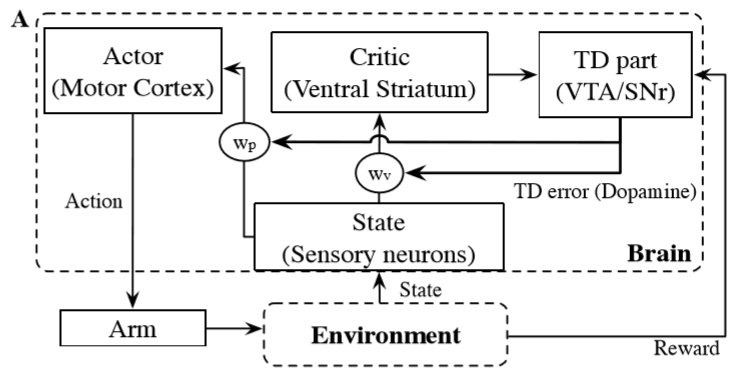
# Towards an Integrative Spiking Neuron Model of Motor Control – from Cortex & Basal Ganglia to Muscles & Sensory Feedback

Yufei Wu and Aldo Faisal

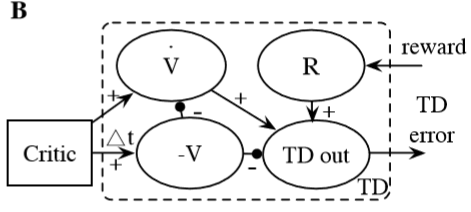
## Overview:

* Simulated reward-based learning in sensorimotor control
* Actor-Critic model
* Asynchronous?

## Network Structure:



* State s = (x, y, dx/dt, dy/dt)
  + Sensory input (7920, Poisson)
  + Population code – neuron for each state
  + Neuron FR determined by dist. to current state
* Actor
  + Motor output (256, LIF)
  + Population code – neuron for each unit vector [0,2pi]
  + Competition – excitatory to neighbours, inhibitory to others
  + Overall pop vector = sum of directions weighted by FRs
* Critic
  + Estimates value of s (200, Poisson) – why 200?
  + Is value encoded in critic activity or synaptic weight matrix wv?
* TD (temporal difference)
  + *Calculates reward prediction error* (800, Poisson) – why 800?



## Reinforcement Learning

* Policy
  + wp i.e. synaptic weights from state to action
  + wp (7920 x 256)
* Value estimation
  + wv i.e. synaptic weights from state to critic
  + wv (7920 x 200) – why is this the value estimation? Surely value should be 7920 or 7920 x 256?
* Learning Rule
  + Update Policy and Value estimation
  + Dopaminergic LTP
  + , n=lr, eTD=TD error, g(Fpre, Fpost)= LTP param?
* Reward
  + Squared dist. Between wrist and target
  + Larger reward if exact target area
  + Cost = Tused/Dmin – system learns to maximise reward and minimise cost – how is cost fed to system?

## General Questions

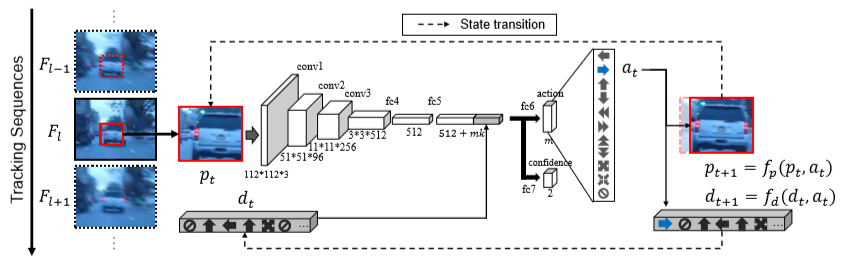
* In my case what is the state?
  + If each pixel is a state and predicted pupil position is the action, then state-action mapping is not one-to-one? i.e. single pixel firing corresponds many possible pupil locations
  + Would need temporal correlation between states to predict action
  + Alternatively, could have output of subsequent feature extractor layers as state
  + How to capture temporal dynamics in state?
* What is output (action)?
  + Pupil position?
  + Would prefer not to have windowing
* Model uses many Poisson neurons
  + Value represented by mean firing rate of critic neurons
  + Does firing rate cause problem?

# Action-Decision Networks for Visual Tracking with Deep Reinforcement Learning

## Overview

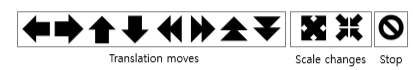
* Object tracker (centre & bounding box)
* Uses *actions* to pursue target
* CNN – combined learning algorithm of SL and RL
* SL – learn appearance of obj
* RL – learn tracking/action dynamics
* 3-15 fps
* RL enables use of partially labelled data

## Network Structure

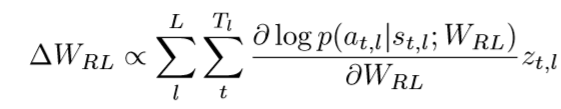


## Reinforcement Learning

* Frame operated on repeatedly until final state reached for each frame -> reward signal
* Action
  + 11 -dim

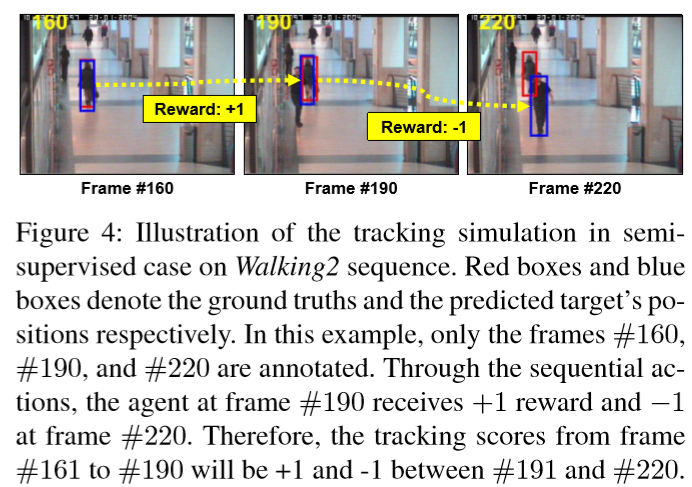


* State s = image patch & prev. actions
  + Tuple (pt, dt)
  + pt = image patch, located by 4D vector bt = [x, y, w, h]
  + dt = prev actions over k timesteps (*needed for system to have memory*)
* State transition function
  + Determined by patch transition function fp and action dynamics function fd
  + fp moves patch by discrete amount
* Reward
  + Either -1 or 1, depends if overlap sufficient
  + Overlap = IoU (Intersection over union)
  + Compared with ground truth
* Learning Rule
  + Policy gradient
  + Stochastic gradient ascent – weights tuned based on tracking score (reward)



## Results

* 88% and 85.15% precision (with 20 pixel centre error threshold)
* Only 4% of frames required >5 actions to localise
  + Far more efficient search than many existing methods
* *Failed to follow abrupt movement/rapid scale changes!*
  + *Saccades are abrupt!*
* Can use partially labelled data

**