# Reproducing and reducing a model of memory and learning in Drosophila

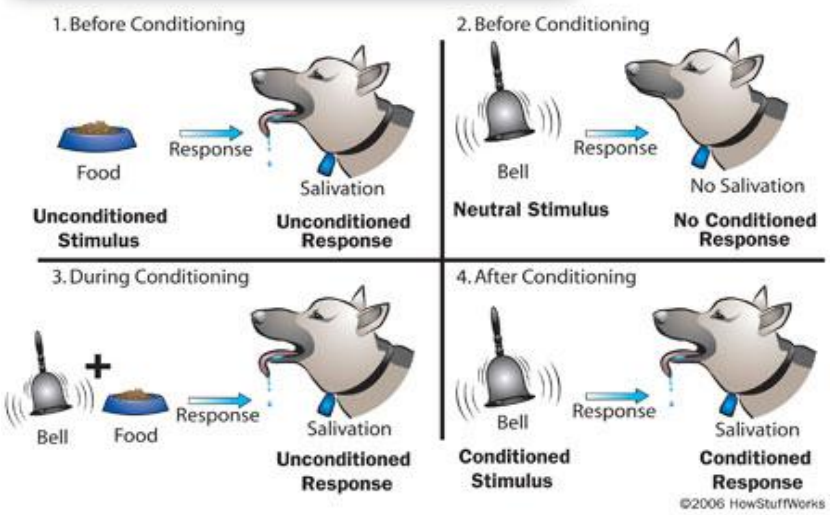
Lab Rotation Report

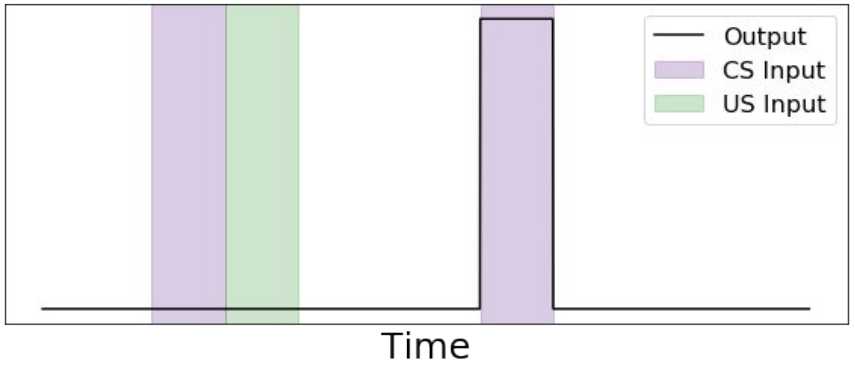
Marshall Mykietyshyn
BCCN, Technische Universität Berlin

# L. Jiang (2020) Task

## Classical conditioning task

- Inputs are odours (appetitive or aversive)
- Unconditioned stimuli (US) used to condition neutral stimuli (CS)
- US signaled by reinforcement
- Novelty: dopamine-gated plasticity

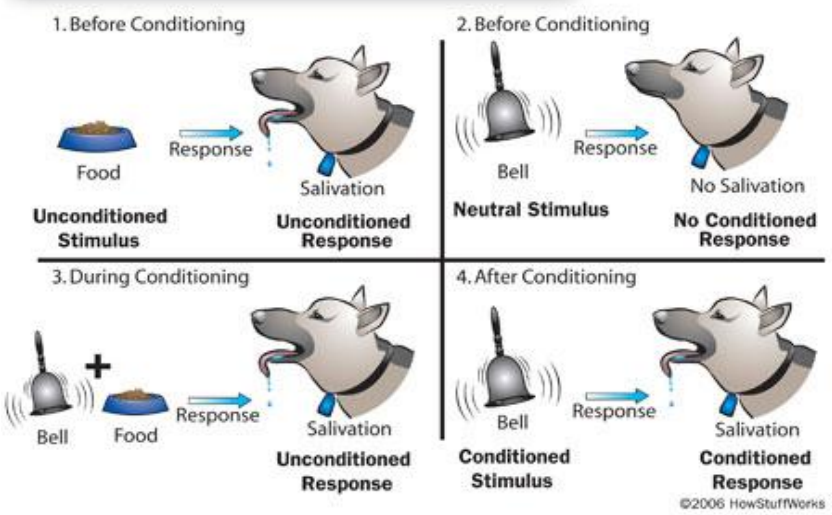


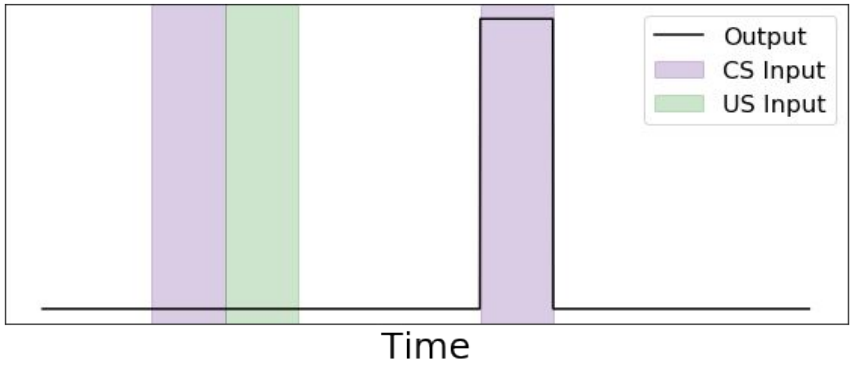



# L. Jiang (2020) Task

## Classical conditioning task

- Task trials = series of intervals
- Stimulus presentation times are randomized (within a window)
- Stimulus presentation lengths are constant
- Neuron activities reset after intervals

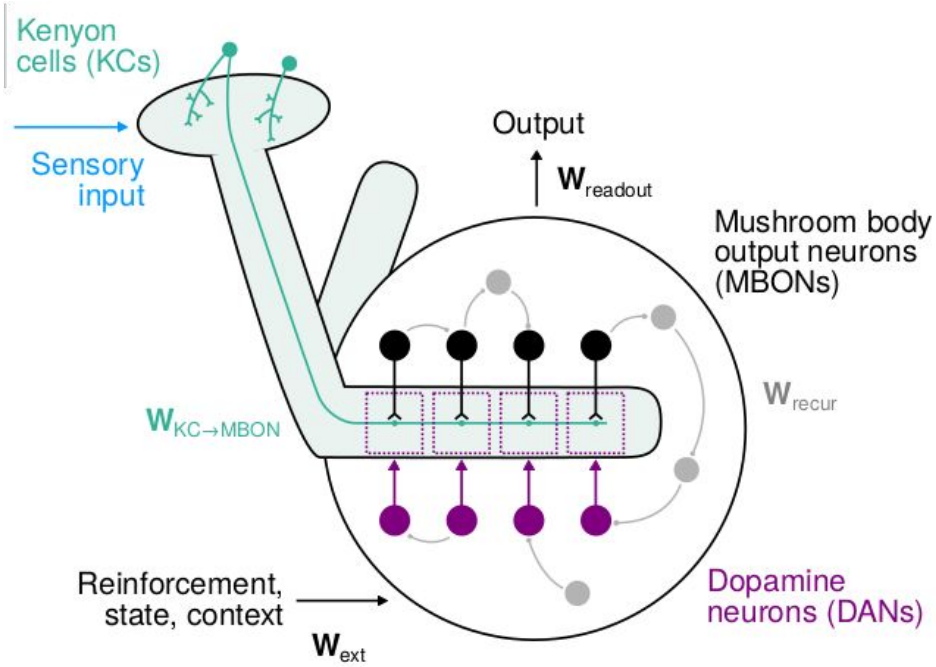


# L. Jiang (2020) Model

## Six neuron populations

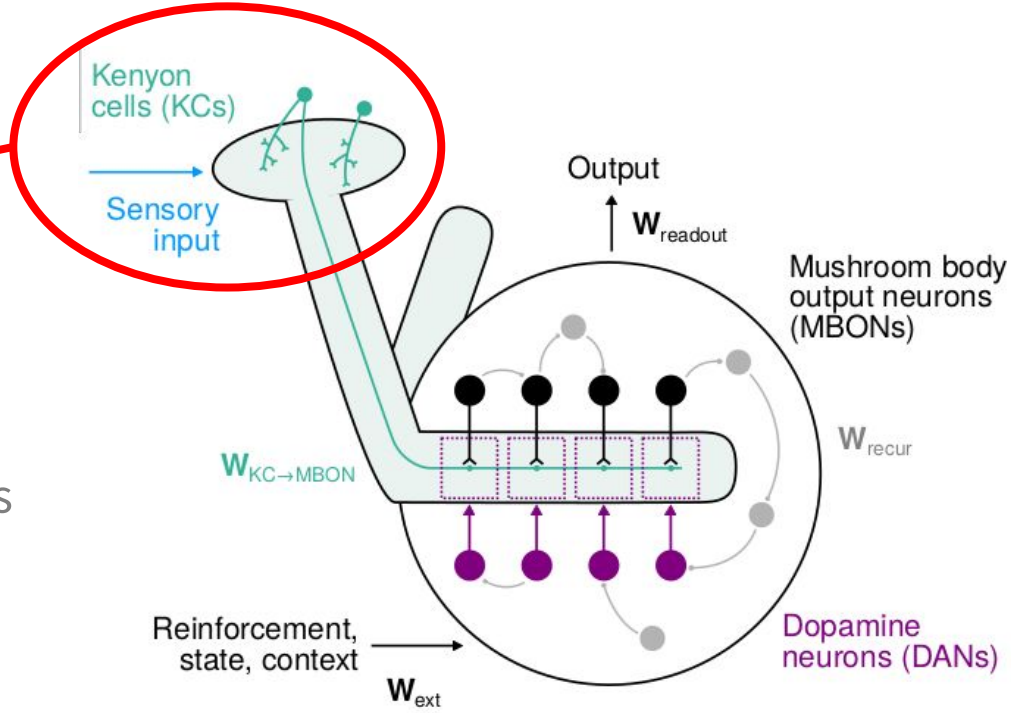
1. Kenyon Cells (KCs)
2. Output (readout)
3. Context (ext)
4. Mushroom-Body Output Neurons (MBONs)
5. Dopaminurgic Neurons (DANs)
6. Feedback Neurons (FBNs)



# L. Jiang (2020) Model

Six neuron populations

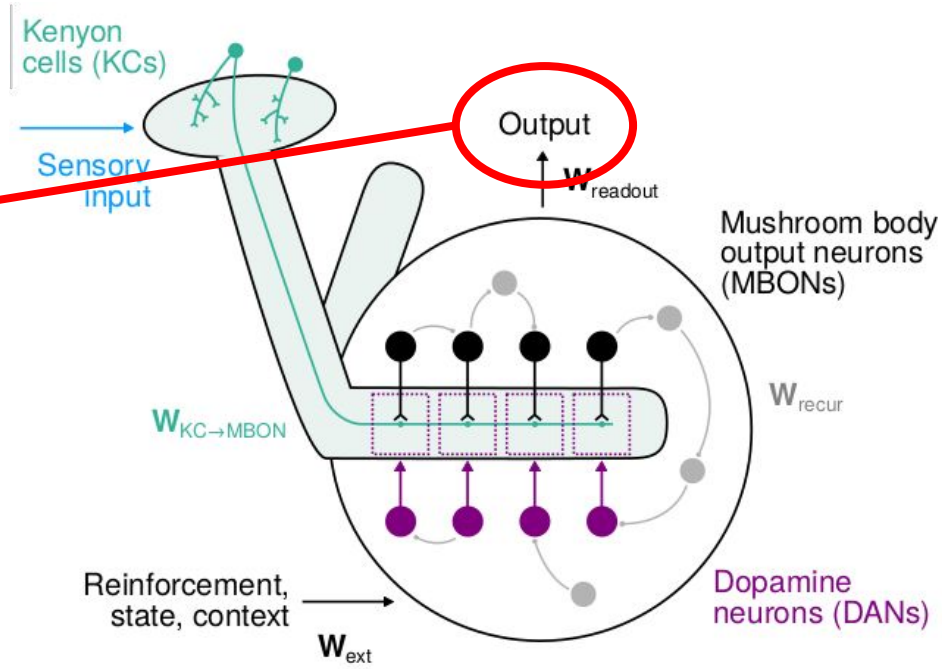
1. Kenyon Cells (KCs)
2. Output (readout)
3. Context (ext)
4. Mushroom-Body Output Neurons (MBONs)
5. Dopaminurgic Neurons (DANs)
6. Feedback Neurons (FBNs)



# L. Jiang (2020) Model

## Six neuron populations

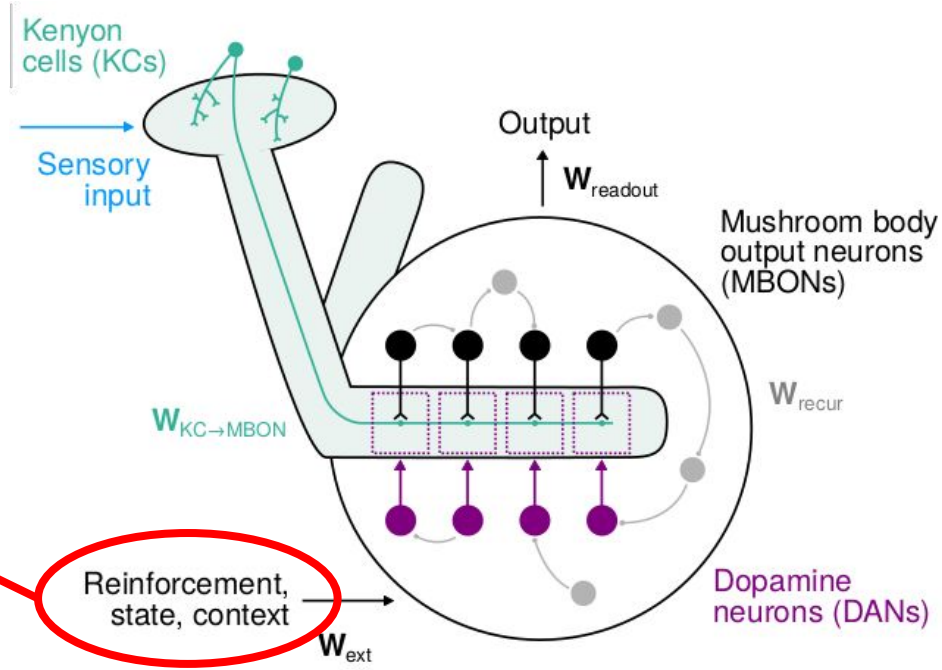
1. Kenyon Cells (KCs)
2. Output (readout)
3. Context (ext)
4. Mushroom-Body Output Neurons (MBONs)
5. Dopaminurgic Neurons (DANs)
6. Feedback Neurons (FBNs)



# L. Jiang (2020) Model

## Six neuron populations

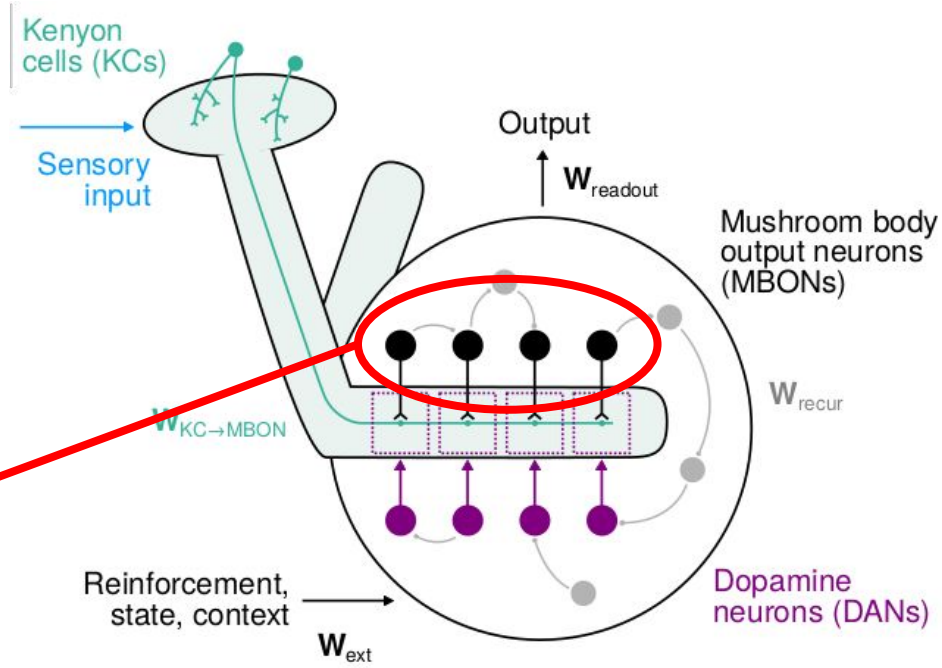
1. Kenyon Cells (KCs)
2. Output (readout)
3. Context (ext)
4. Mushroom-Body Output Neurons (MBONs)
5. Dopaminurgic Neurons (DANs)
6. Feedback Neurons (FBNs)



# L. Jiang (2020) Model

## Six neuron populations

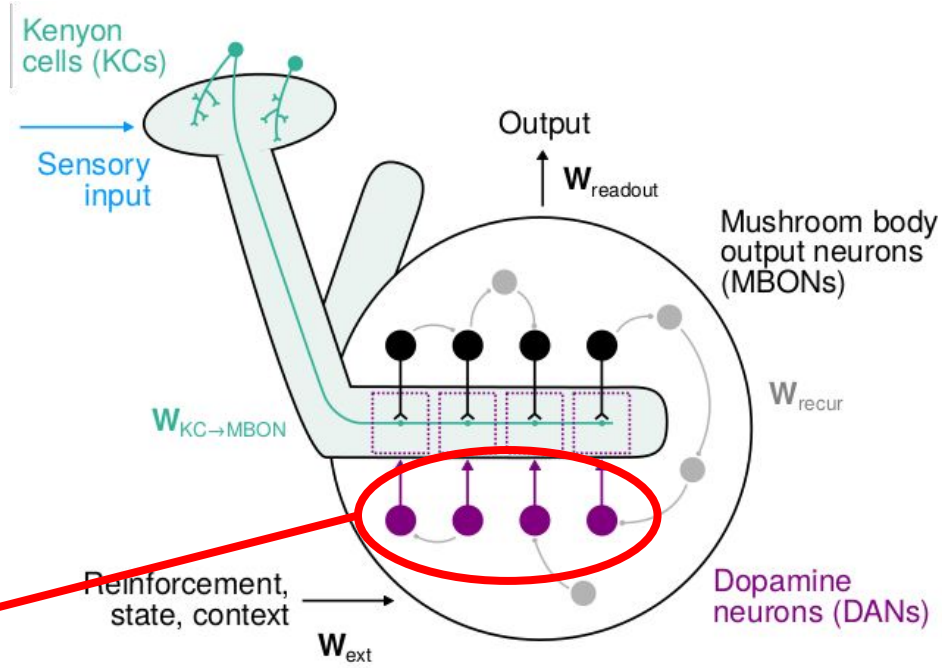
1. Kenyon Cells (KCs)
2. Output (readout)
3. Context (ext)
4. Mushroom-Body Output Neurons (MBONs)
5. Dopaminurgic Neurons (DANs)
6. Feedback Neurons (FBNs)



# L. Jiang (2020) Model

## Six neuron populations

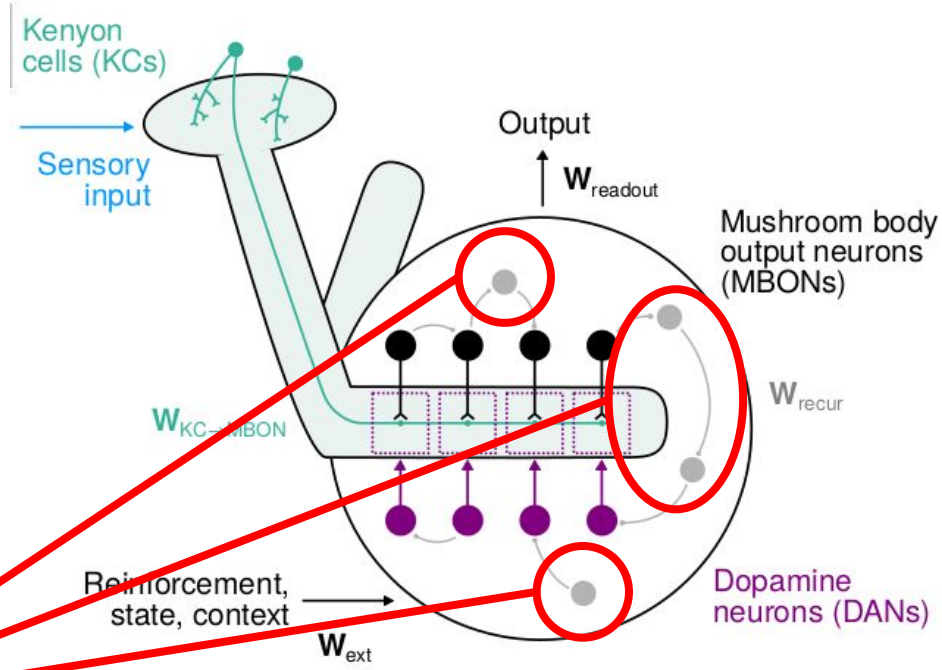
1. Kenyon Cells (KCs)
2. Output (readout)
3. Context (ext)
4. Mushroom-Body Output Neurons (MBONs)
5. Dopaminurgic Neurons (DANs)
6. Feedback Neurons (FBNs)



# L. Jiang (2020) Model

## Six neuron populations

1. Kenyon Cells (KCs)
2. Output (readout)
3. Context (ext)
4. Mushroom-Body Output Neurons (MBONs)
5. Dopaminurgic Neurons (DANs)
6. Feedback Neurons (FBNs)

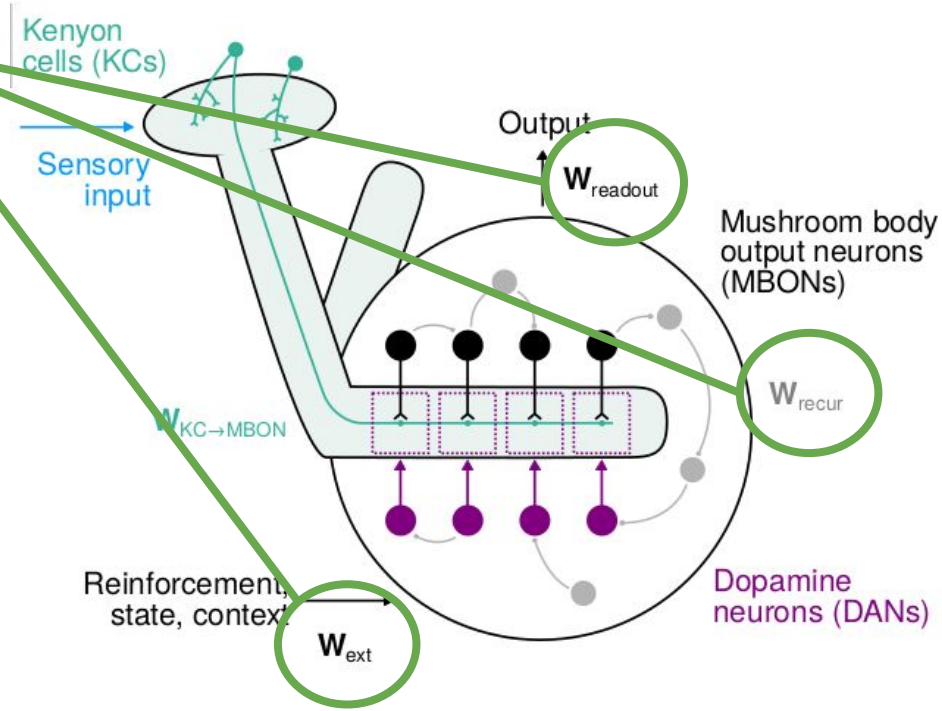


# L. Jiang (2020) Model

Backprop trained weights

- Context to feedback neurons ($W^{ext}$)
- Recurrent weights among output circuitry ($W^{recur}$)
- MBONs to the output ($W^{readout}$)

Represents evolution and lifetime learning



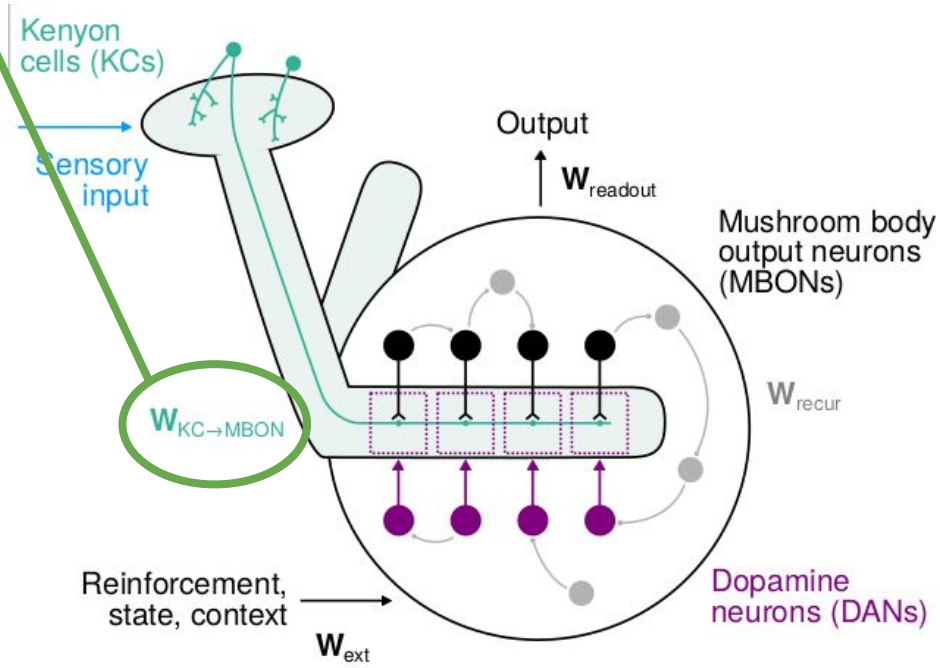
# L. Jiang (2020) Model

## Dynamically updated weights

- KCs→MBONs ($W^{KC→MBON}$)
- LTD/LTP mechanism ($\tau$ = 5s)
- Model of synaptic plasticity

Represents short-term learning

$$\tau_W \dot{\mathbf{W}}_{ij}^{KC→MBON} = -\mathbf{W}_{ij}^{KC→MBON} + \mathbf{w}_{ij}$$



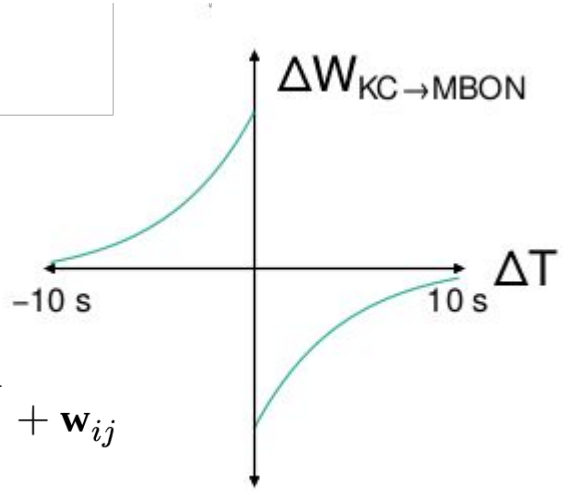
# L. Jiang (2020) Model

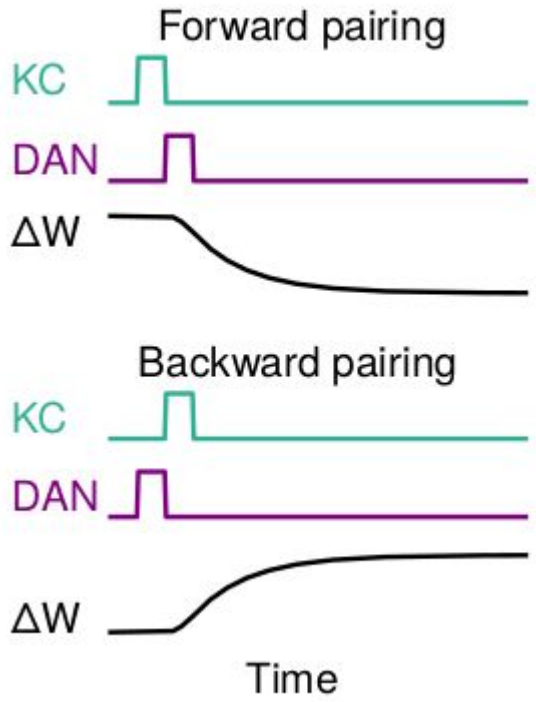
## Dynamically updated weights

- KCs→MBONs ($W^{KC→MBON}$)
- LTD/LTP mechanism ($\tau$ = 5s)
- Model of synaptic plasticity

Represents short-term learning

$$\tau_W \dot{\mathbf{W}}_{ij}^{KC→MBON} = -\mathbf{W}_{ij}^{KC→MBON} + \mathbf{w}_{ij}$$



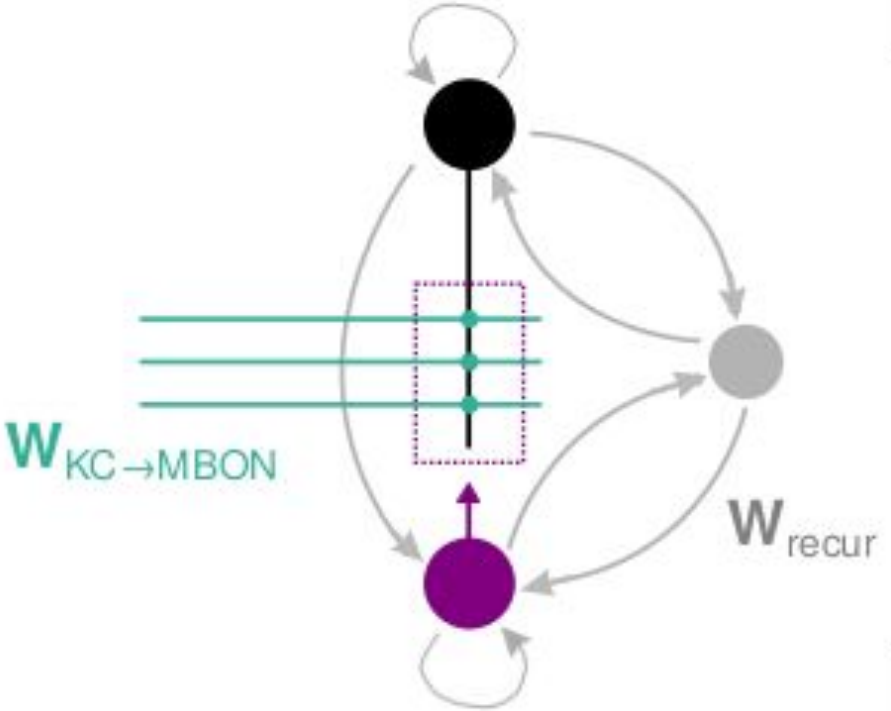
# L. Jiang (2020) Model

## Output circuitry

- Output circuitry =

  MBONs + DANs + FBNs

- DAN→MBON weights = 0
  - So that DANs only affect MBONs through plasticity



# L. Jiang (2020) Model

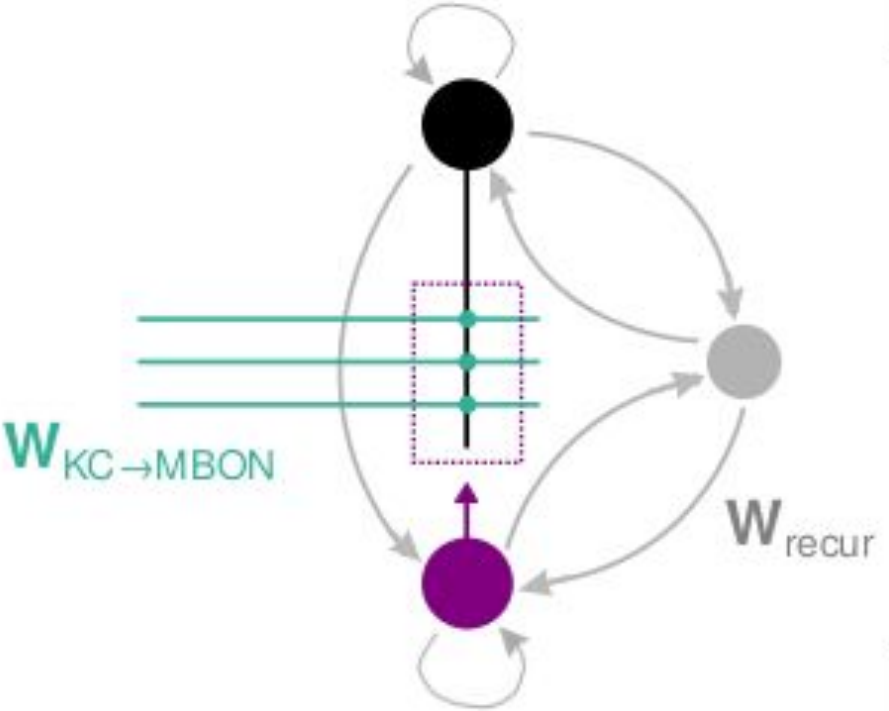
## Output circuitry

- DANs are paired 1-to-1 with MBONs through weight update equation
  - $N_{MBON} = N_{DAN}$

$$\frac{dw_{ij}}{dt} = \bar{r}_{DAN,i}(t) r_{KC,j}(t) - \bar{r}_{KC,j}(t) r_{DAN,i}(t)$$

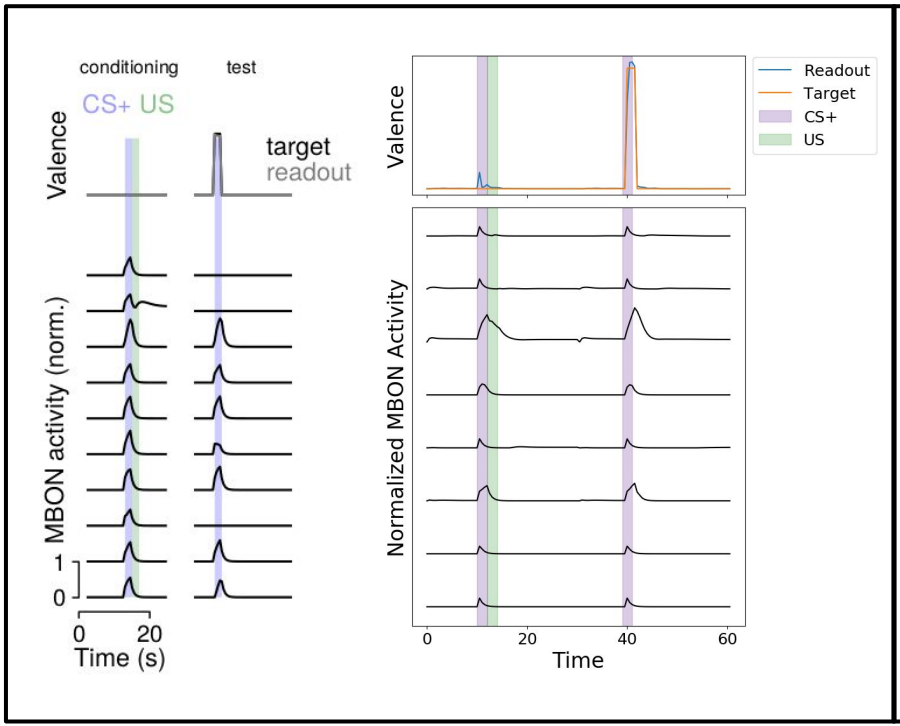
Where $i \in [0, 1, \ldots, N_{MBON}]$

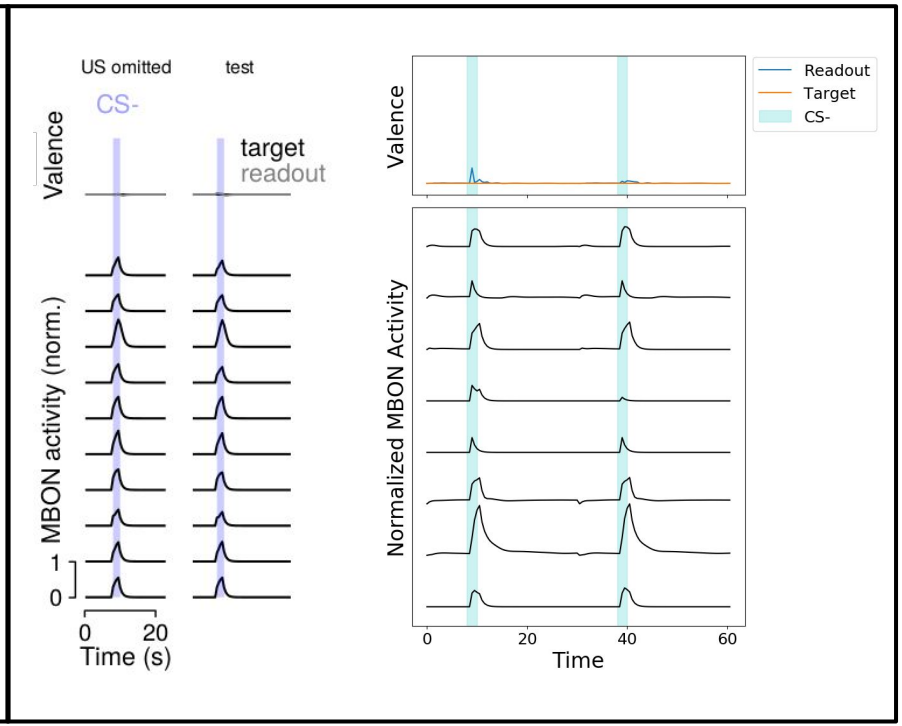
$j \in [0, 1, \ldots, N_{KC}]$



# Comparison of Results

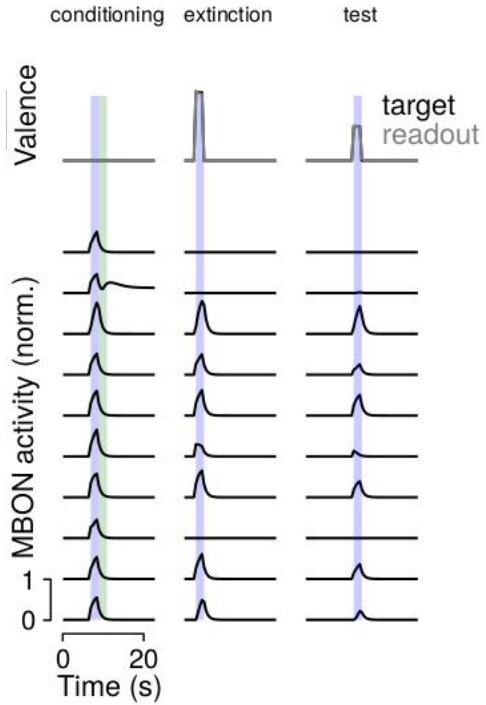
## 1st-order Conditioning

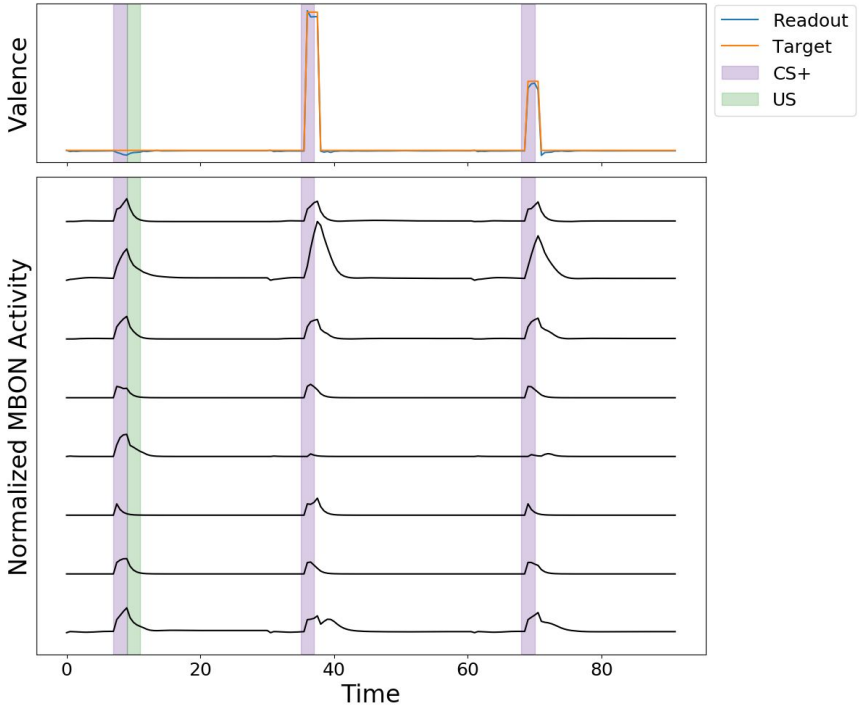


# Comparison of Results

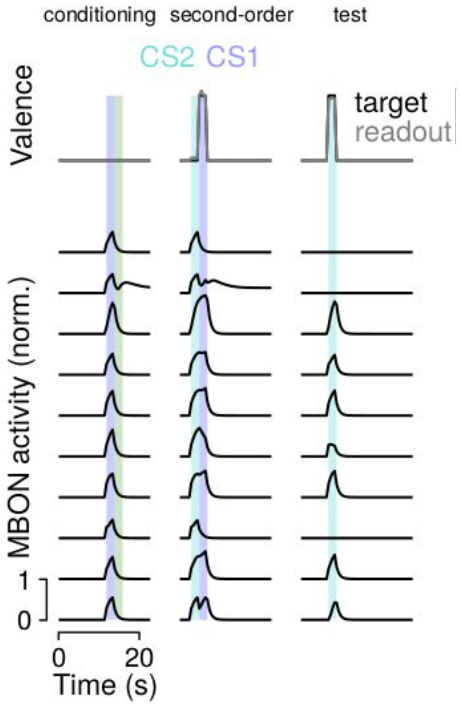
## Extinction Conditioning

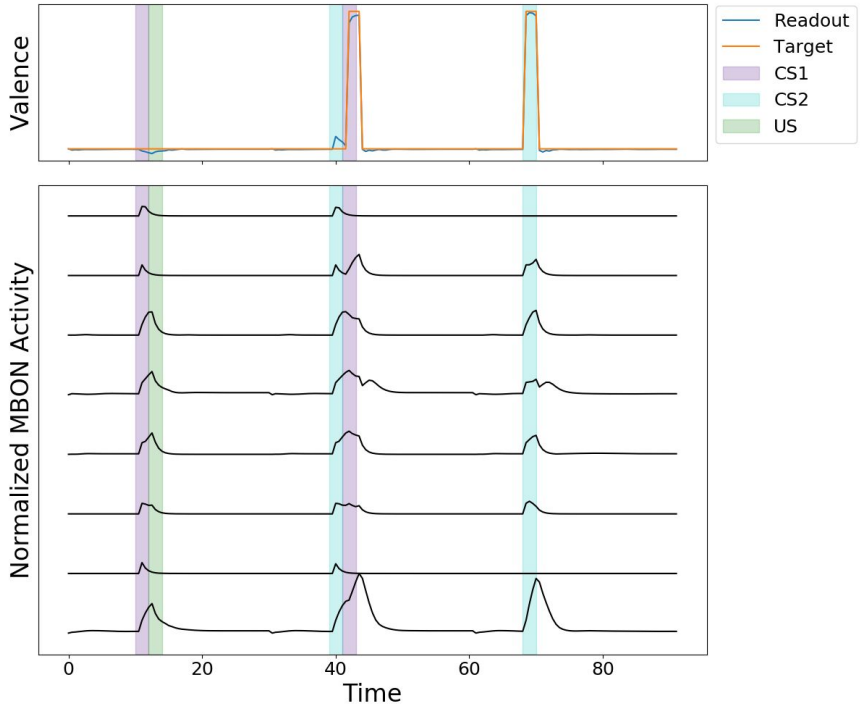


# Comparison of Results

## 2nd-order Conditioning



# Model Simplification

## Modifications to Network

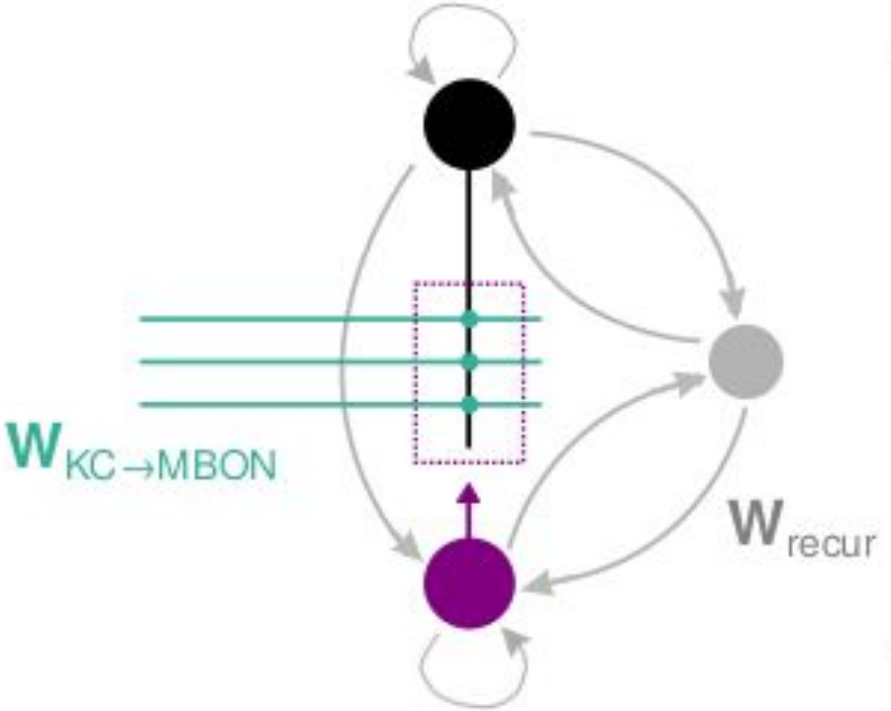
Remove Feedback Connections

- No feedback (2-hop MBON→DAN)
- No feedback (1-hop MBON→DAN)
- No feedback (no MBON→DAN)

Train only on second-order conditioning

Eliminate LTP weight updates (LTD only)

Use a minimum number of MBONs



# Model Simplification

## Modifications to Network

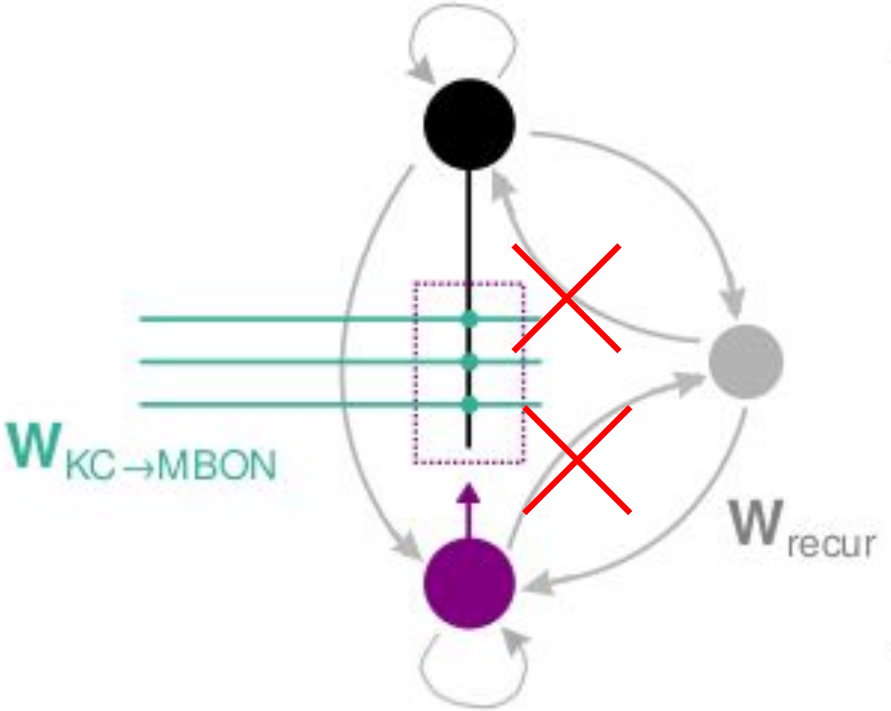
Remove Feedback Connections

- No feedback (2-hop MBON→DAN)
- No feedback (1-hop MBON→DAN)
- No feedback (no MBON→DAN)

Train only on second-order conditioning

Eliminate LTP weight updates (LTD only)

Use a minimum number of MBONs



# Model Simplification

## Modifications to Network

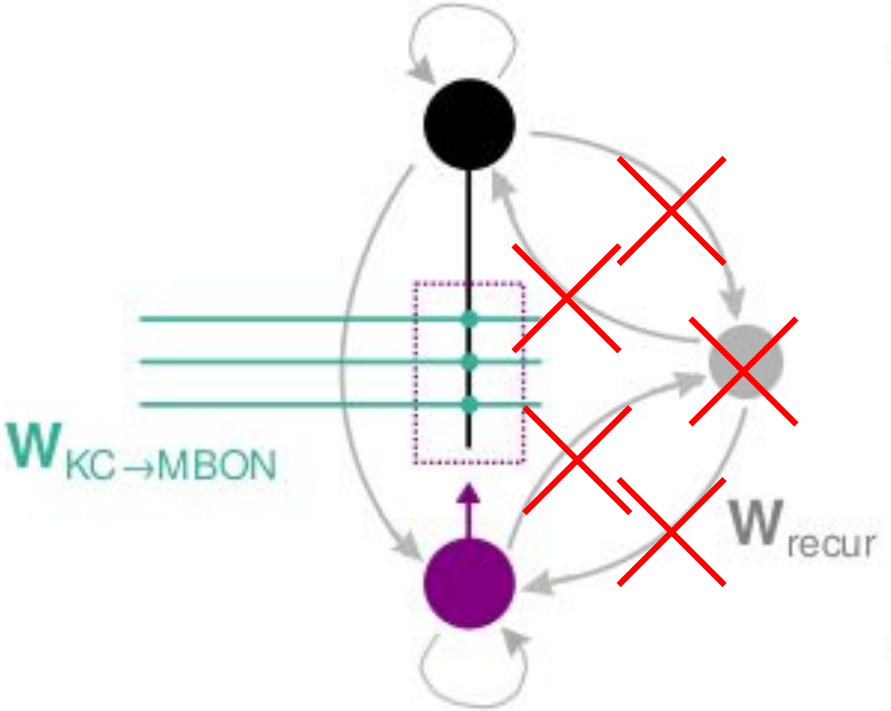
Remove Feedback Connections

- No feedback (2-hop MBON→DAN)
- No feedback (1-hop MBON→DAN)
- No feedback (no MBON→DAN)

Train only on second-order conditioning

Eliminate LTP weight updates (LTD only)

Use a minimum number of MBONs



# Model Simplification

## Modifications to Network

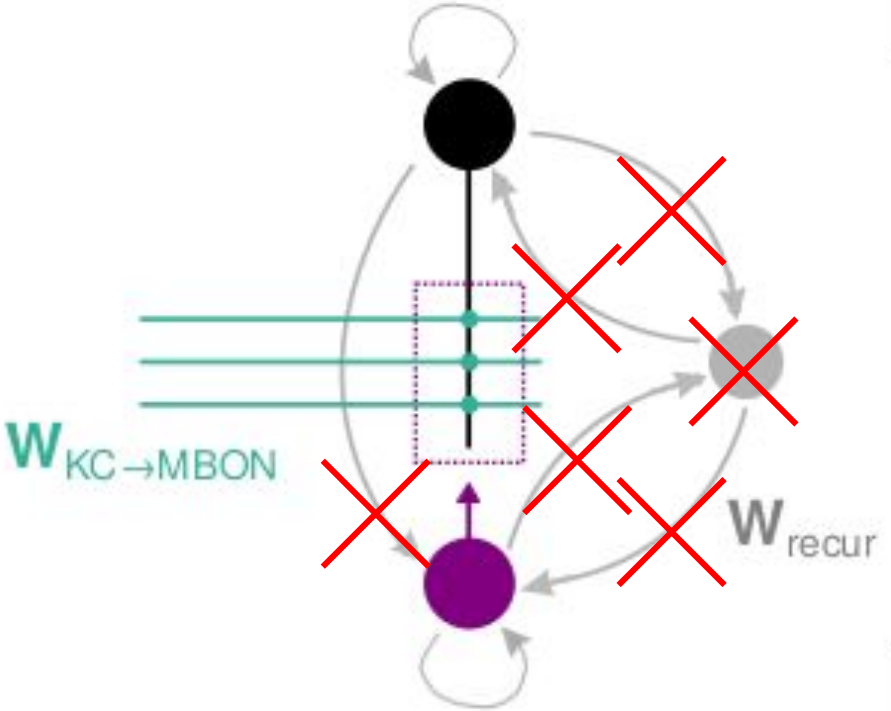
Remove Feedback Connections

- No feedback (2-hop MBON→DAN)
- No feedback (1-hop MBON→DAN)
- No feedback (no MBON→DAN)

Train only on second-order conditioning

Eliminate LTP weight updates (LTD only)

Use a minimum number of MBONs



# Model Simplification

## Modifications to Network

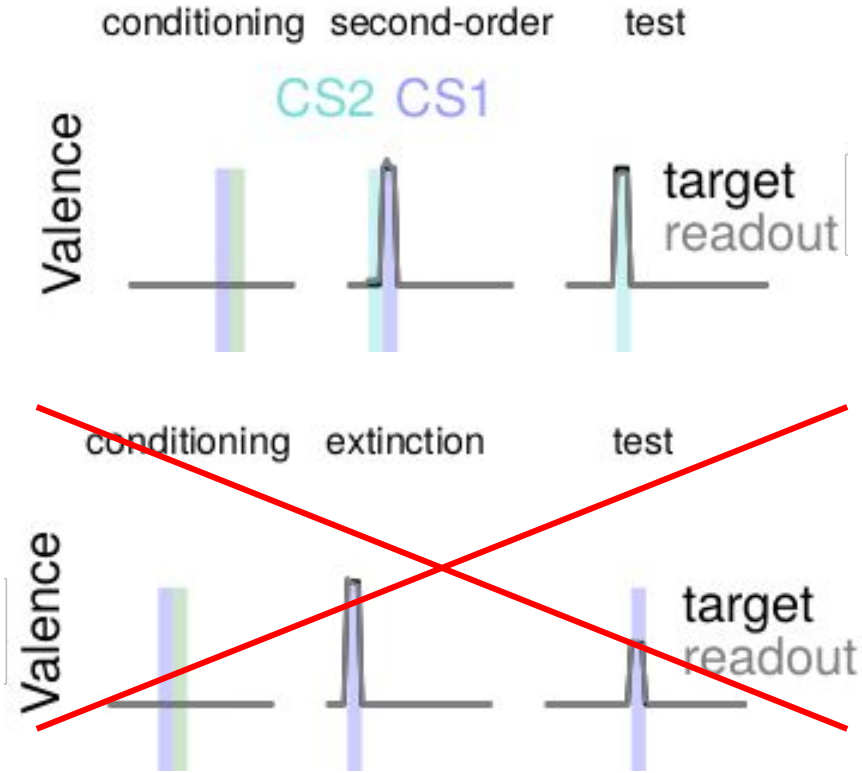
Remove Feedback Connections

- No feedback (2-hop MBON→DAN)
- No feedback (1-hop MBON→DAN)
- No feedback (no MBON→DAN)

Train only on second-order conditioning

Eliminate LTP weight updates (LTD only)

Use a minimum number of MBONs



# Model Simplification

## Modifications to Network

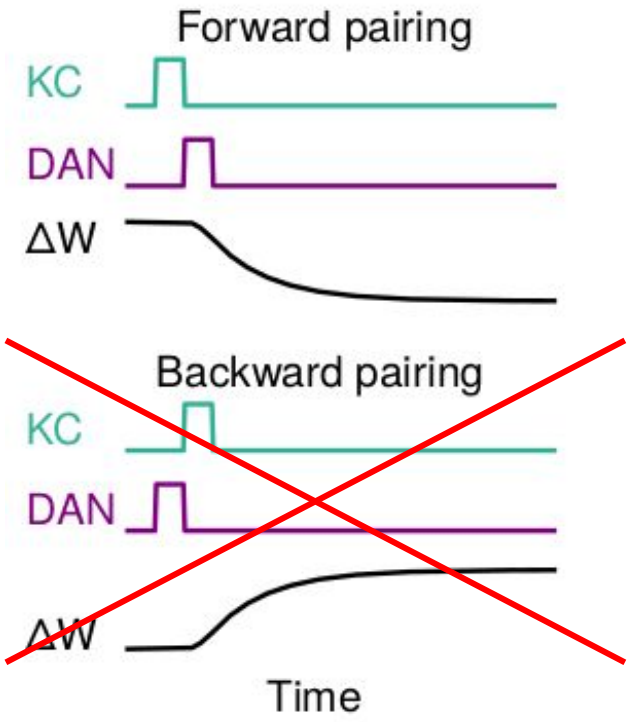
Remove Feedback Connections

- No feedback (2-hop MBON→DAN)
- No feedback (1-hop MBON→DAN)
- No feedback (no MBON→DAN)

Train only on second-order conditioning

Eliminate LTP weight updates (LTD only)

Use a minimum number of MBONs



# Model Simplification

## Modifications to Network

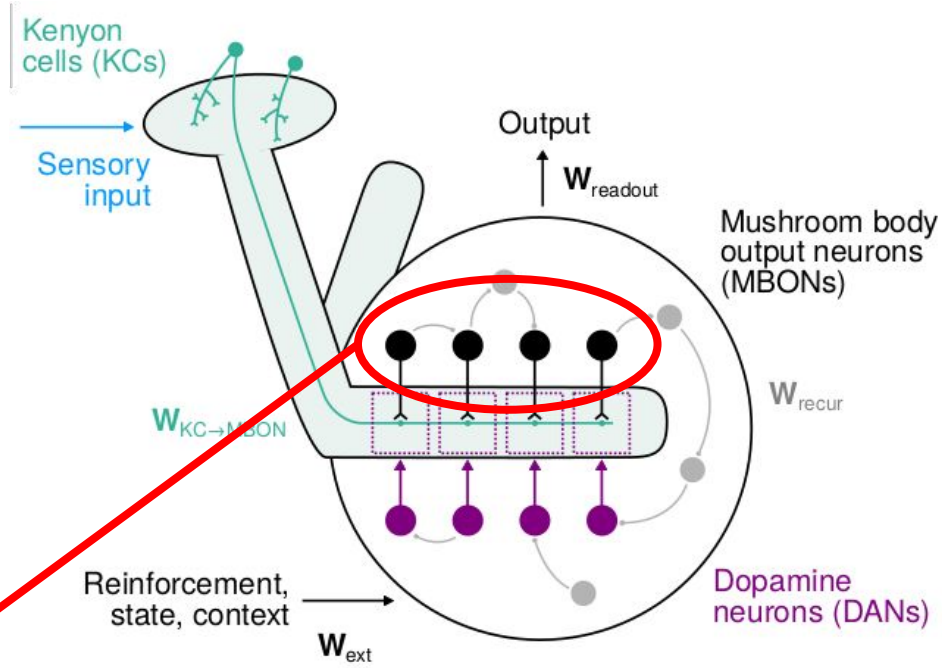
Remove Feedback Connections

- No feedback (2-hop MBON→DAN)
- No feedback (1-hop MBON→DAN)
- No feedback (no MBON→DAN)

Train only on second-order conditioning

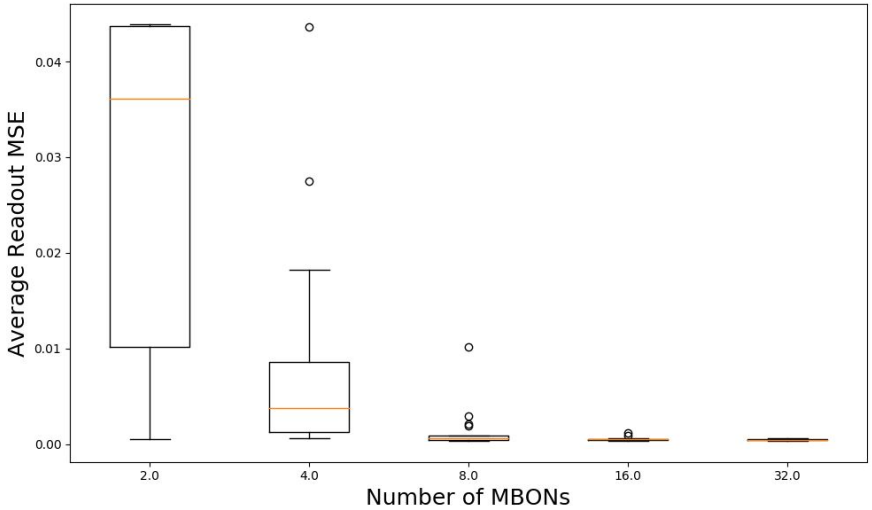
Eliminate LTP weight updates (LTD only)

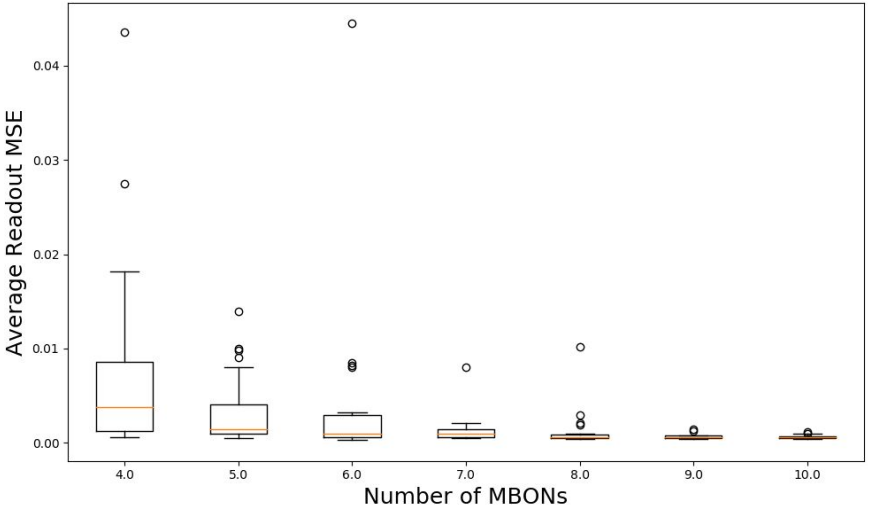
Use a minimum number of MBONs



# Model Simplification

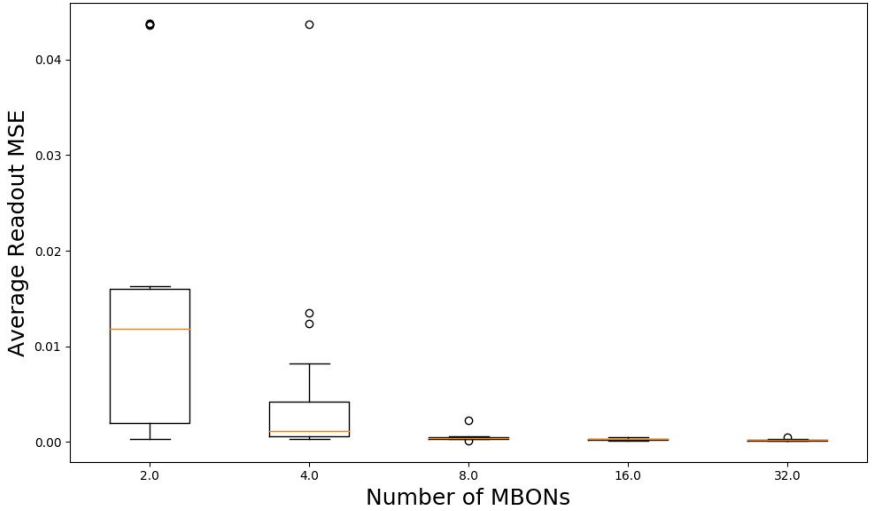
## MBON Sensitivity - All Classical Conditioning

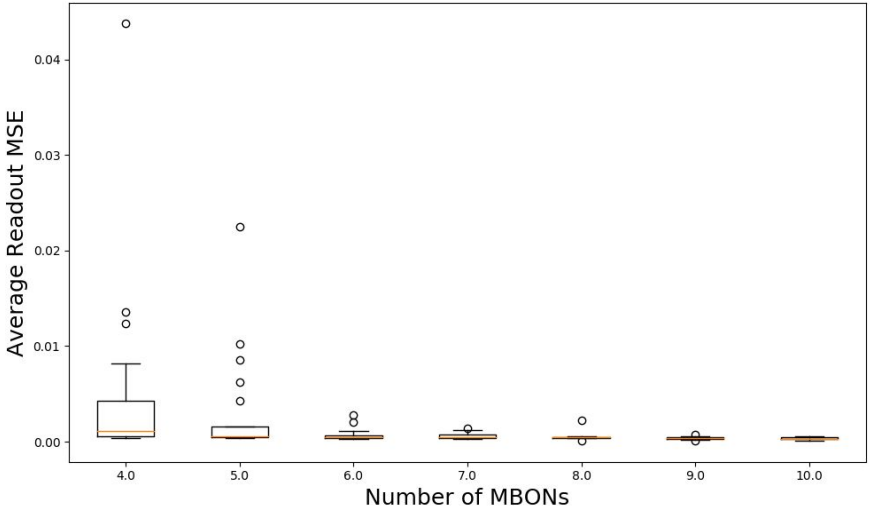


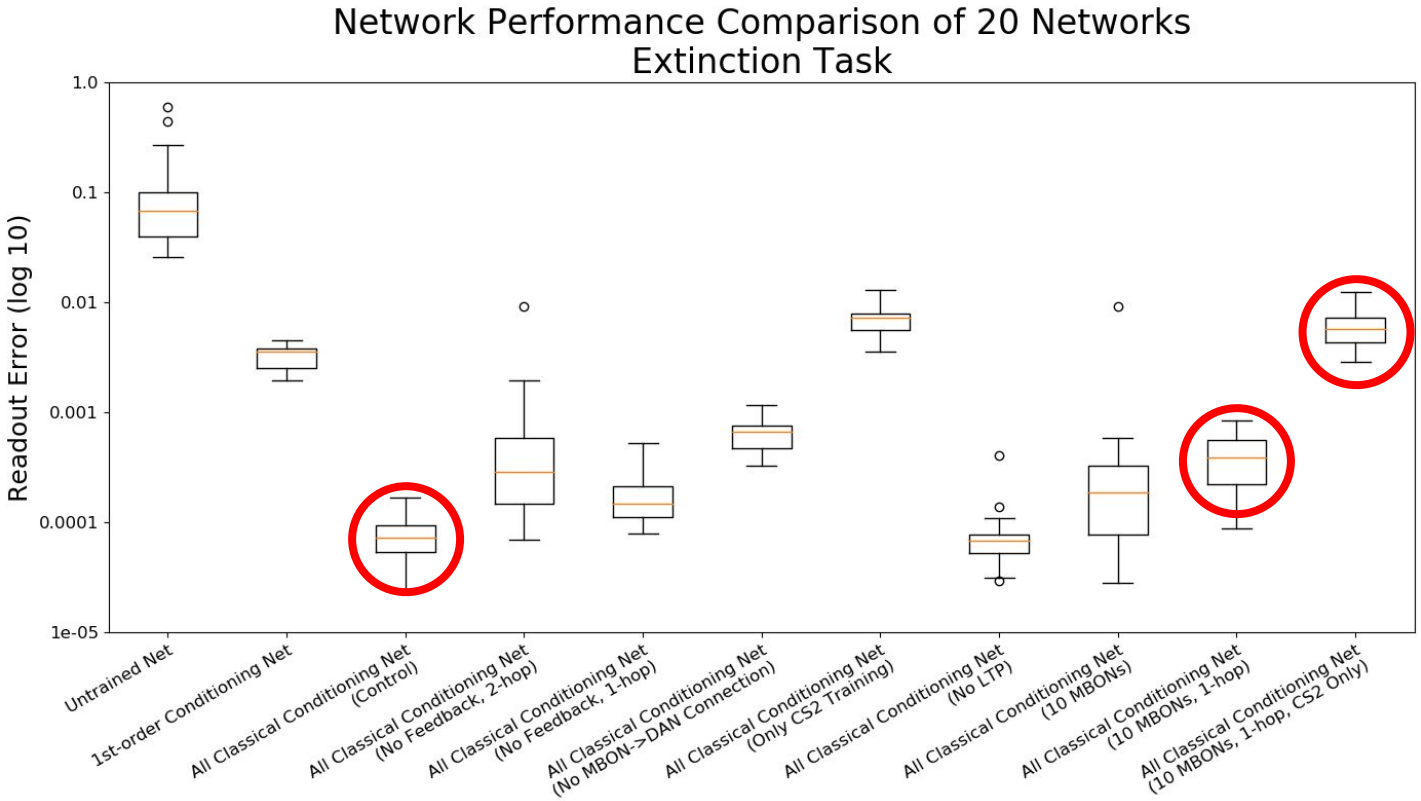
# Model Simplification

## MBON Sensitivity - Only 2nd-order Conditioning



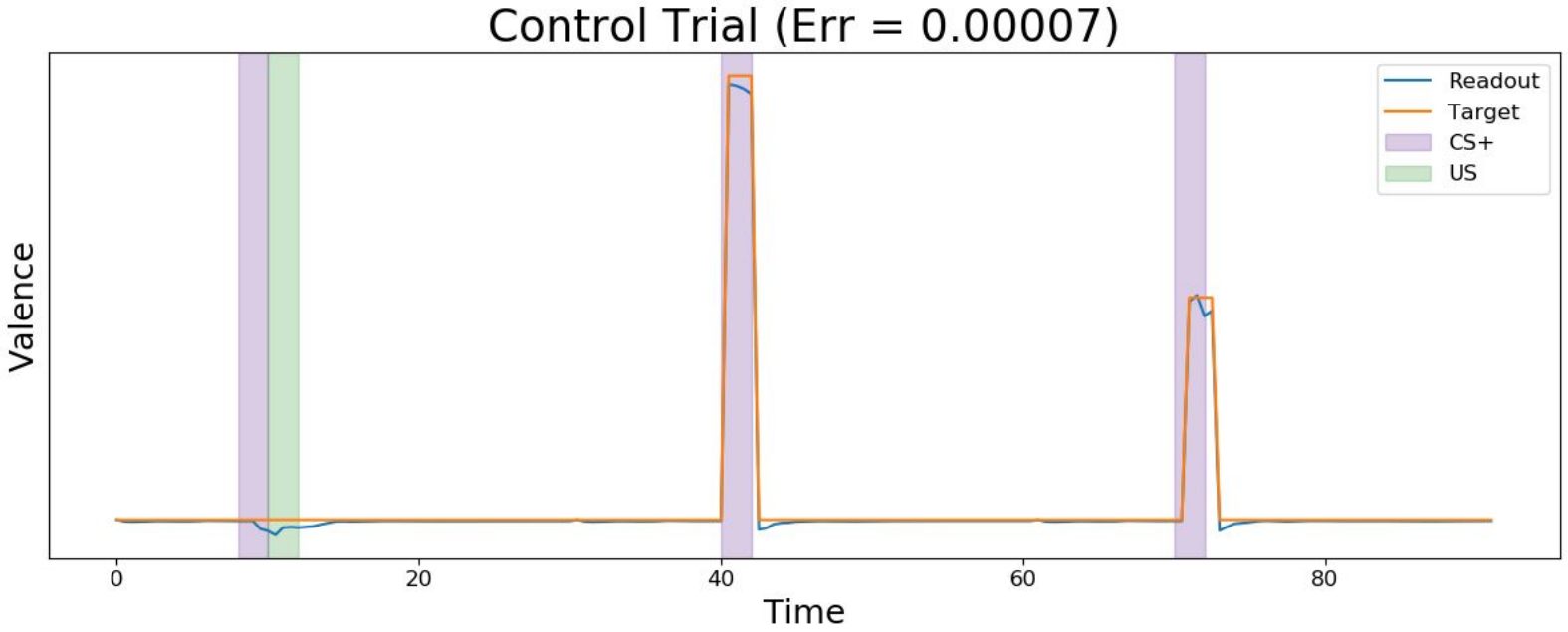



# Model Simplification Results



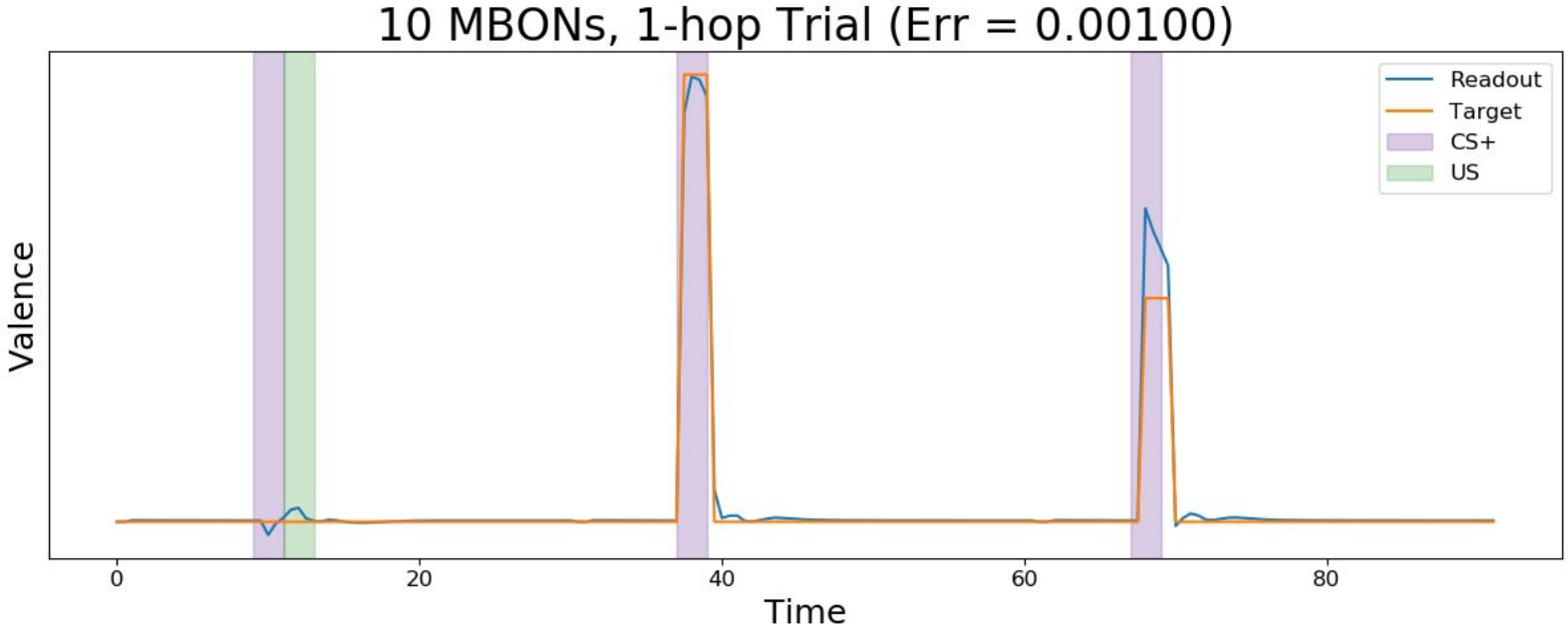
# Model Simplification Results

## Qualitative Assessment - Error ~$10^{-4}$



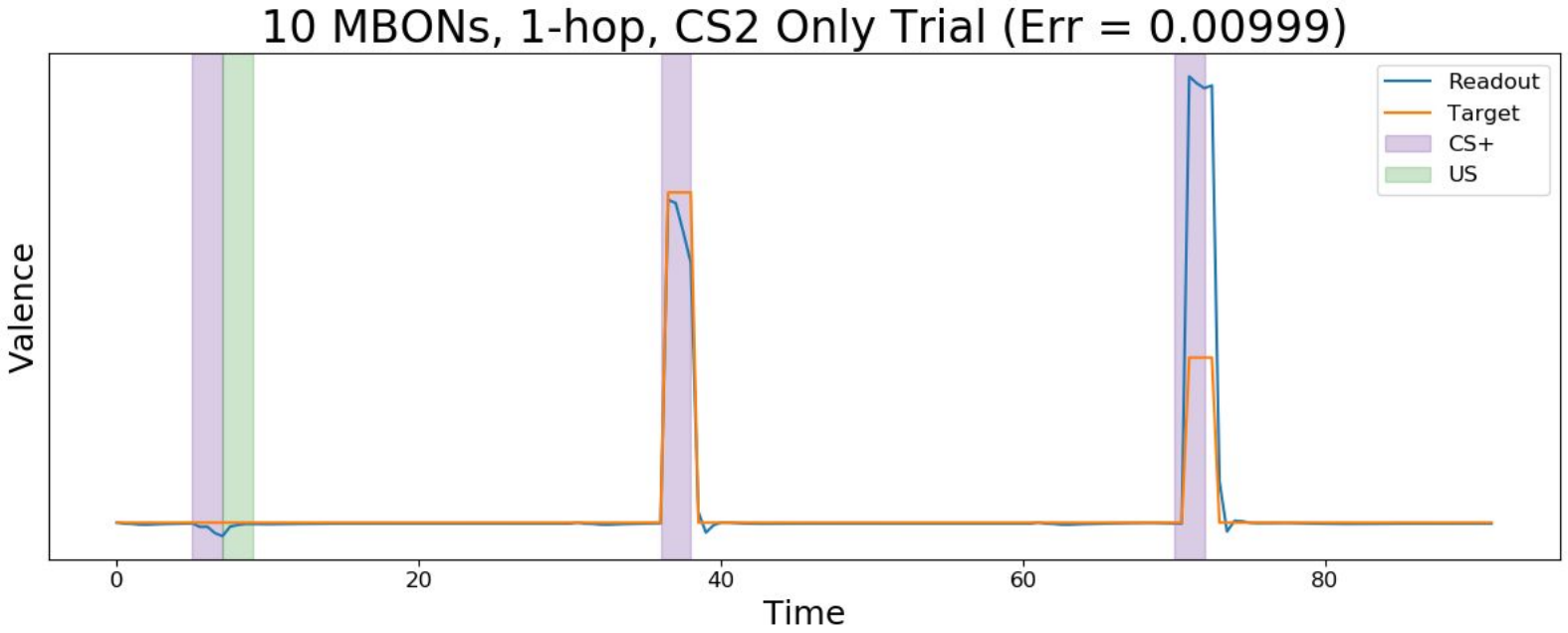
# Model Simplification Results

## Qualitative Assessment - Error ~$10^{-3}$

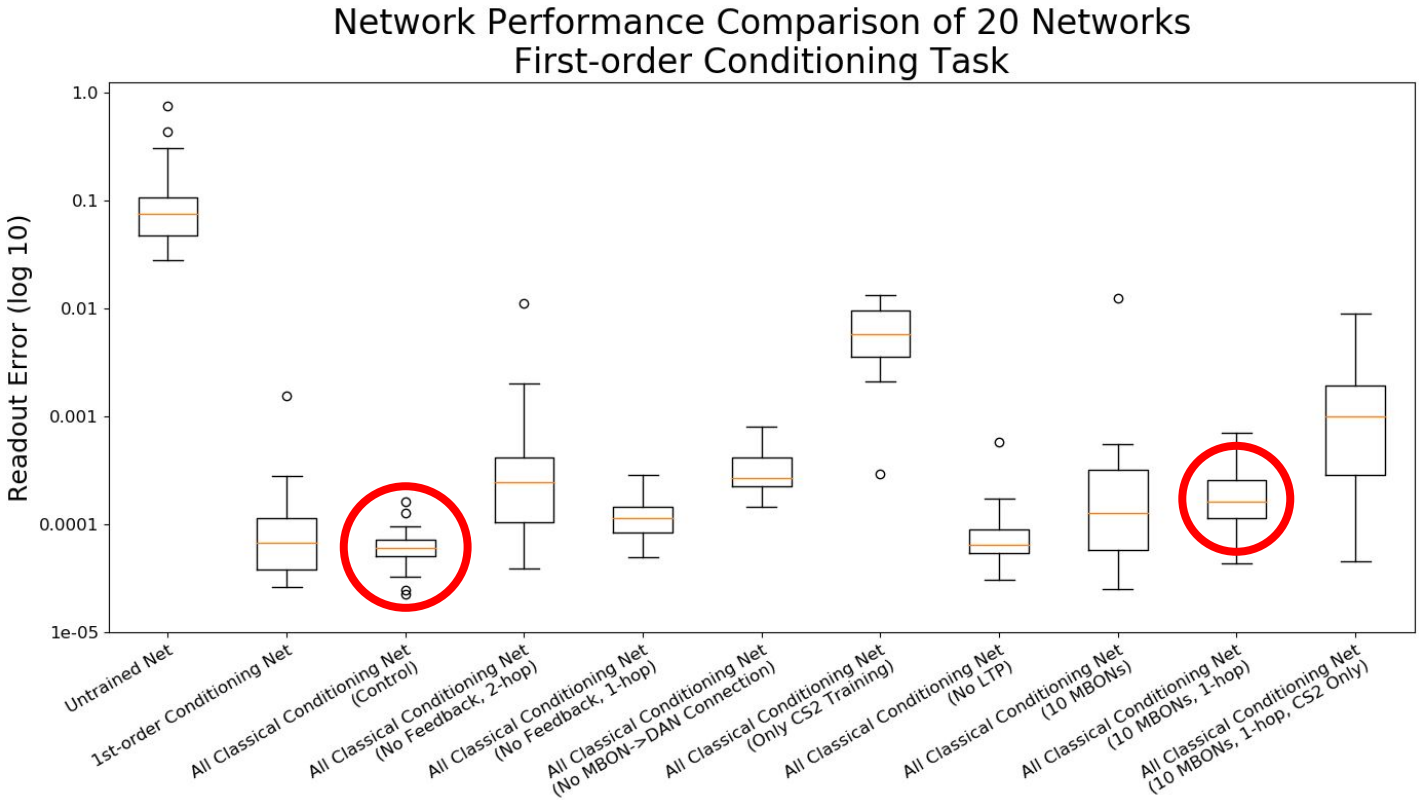


# Model Simplification Results

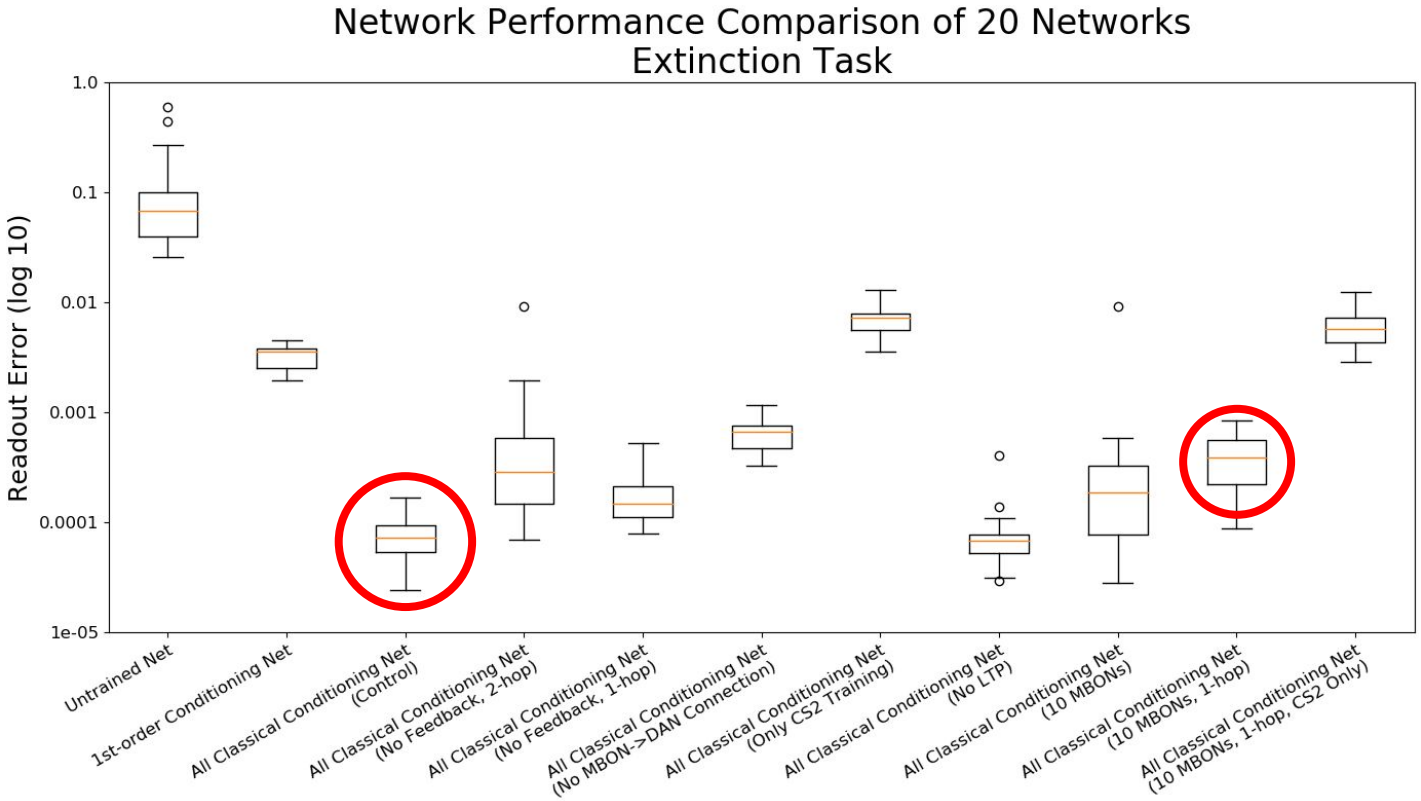
## Qualitative Assessment - Error ~$10^{-2}$



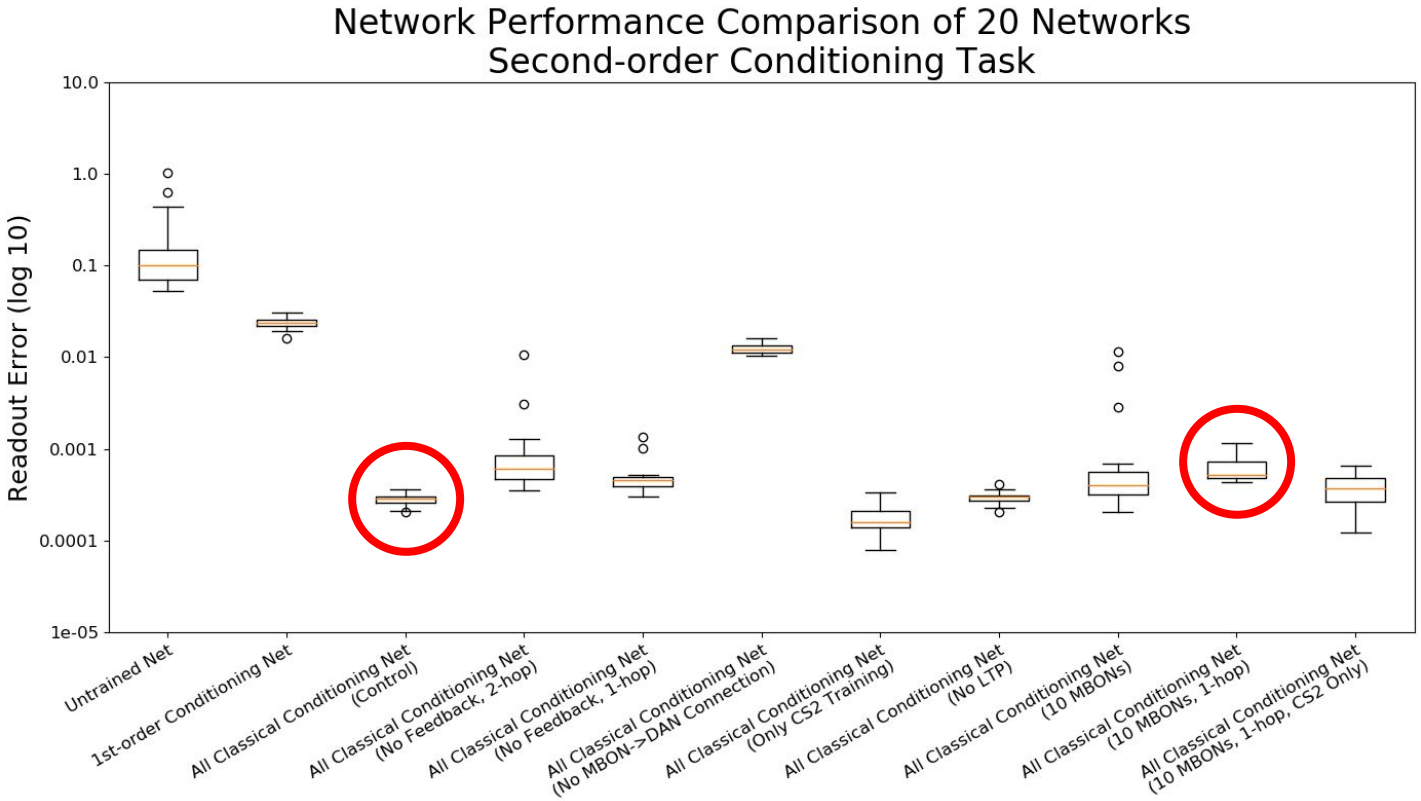
# Model Simplification Results



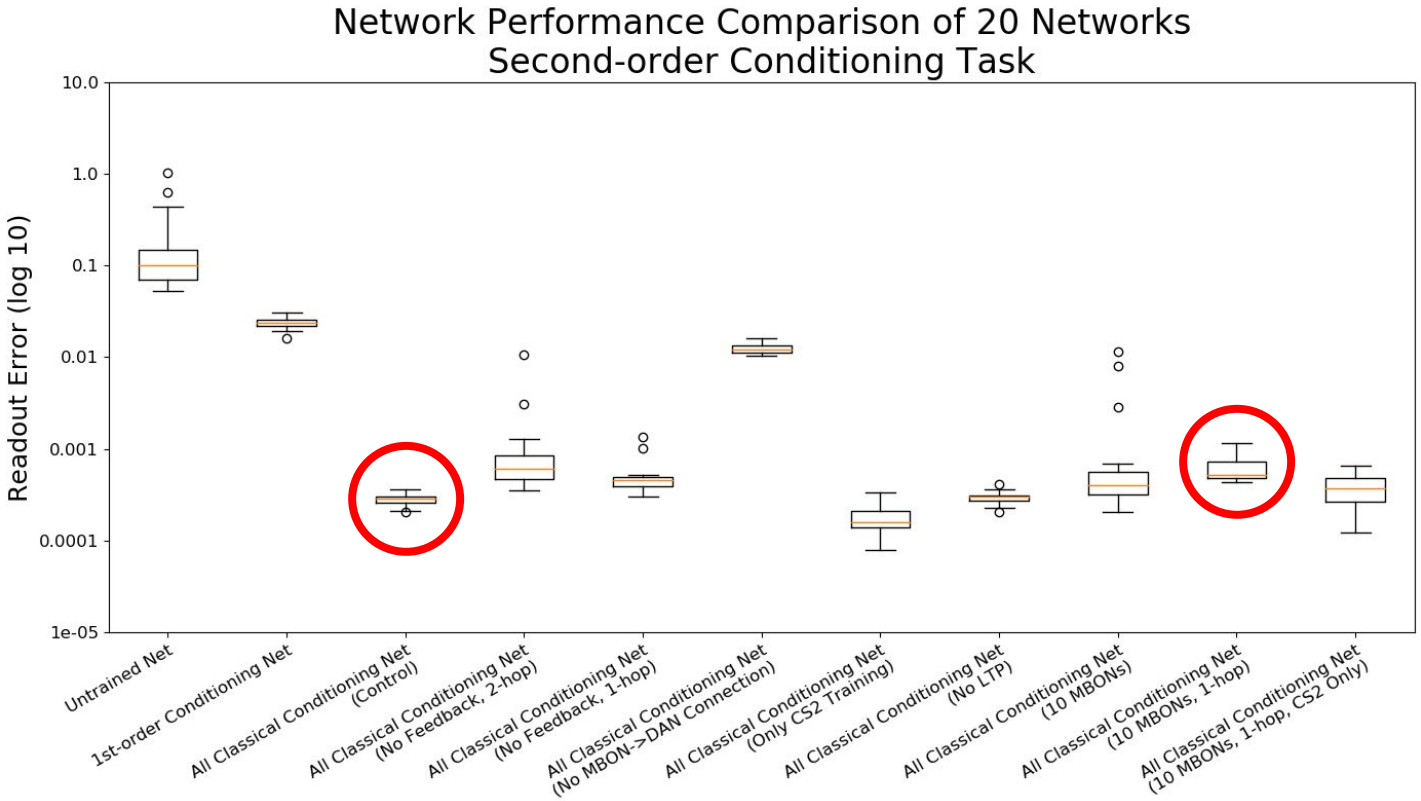
# Model Simplification Results



# Model Simplification Results



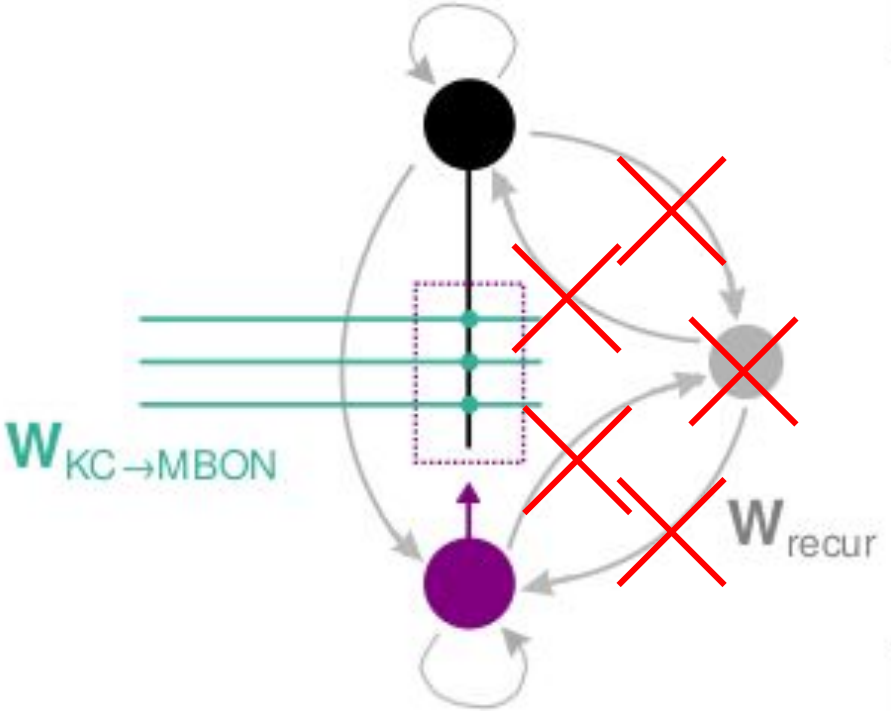
# Model Simplification Results



# Model Simplification - Summary

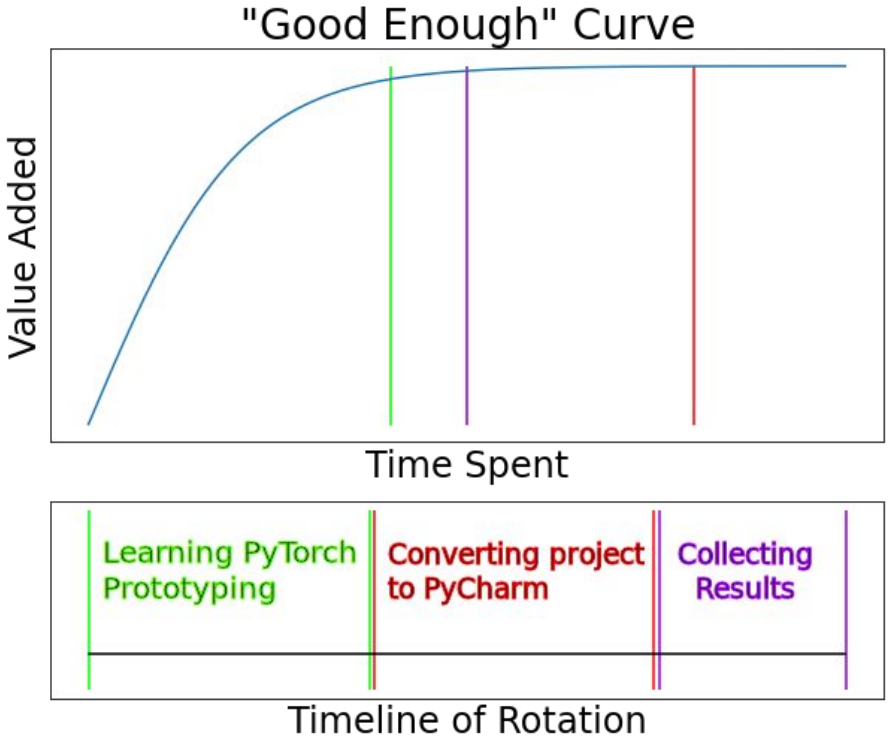
## Minimal Model

- Halved $N_{MBON}$ (20 -> 10)
- Removed FBN population (1-hop)
- Eliminated LTP from weight updates (optional)



# Skills Learned

- PyTorch
- Engineering-prototyping balance
- Packages and modules vs. notebooks
- Periodic reassessment of completed work



# Lab Rotation - Summary

- Reproduced model of memory and learning
  - Similar performance to Jiang 2020
- Reduced their model
  - Simplest form while maintaining performance
  - Halved size of MBON and DAN populations
  - Removed feedback neuron population
- Began to analyze the network dynamics (not shown here)