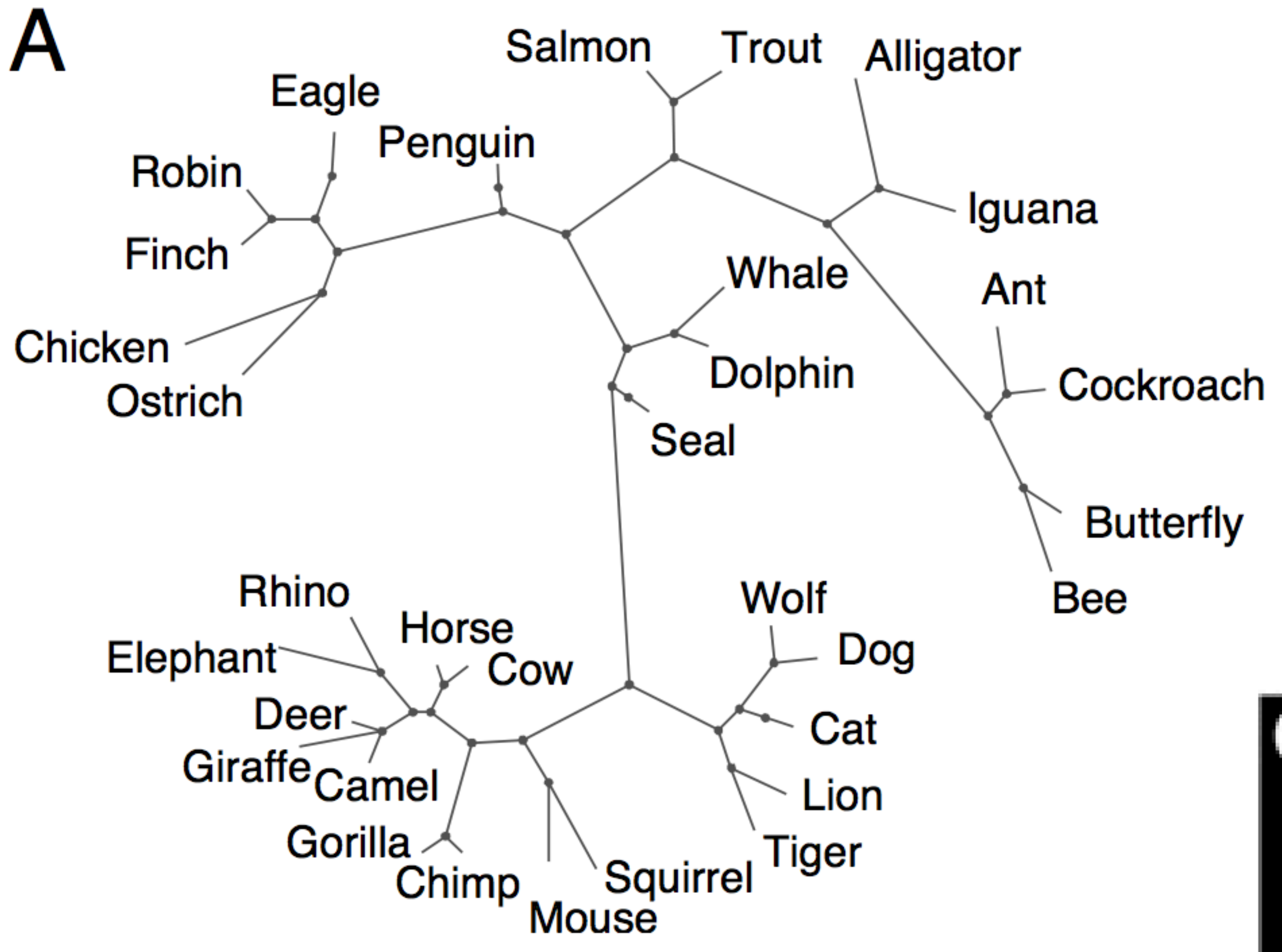
Relational Inductive Bias, Deep Learning, and Graph Networks

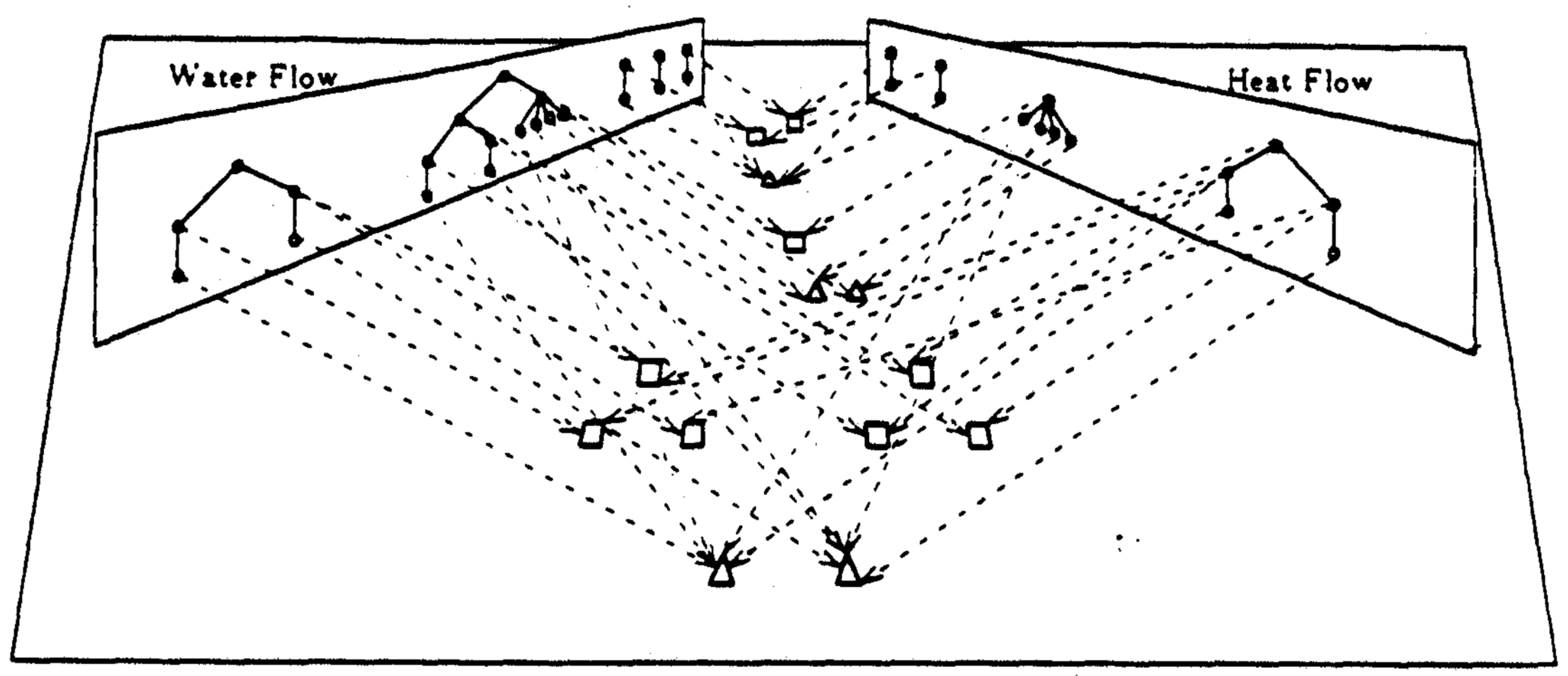
Jessica B. Hamrick
DeepMind

LAL Seminar
April 09, 2019

Human Thought is Highly Structured

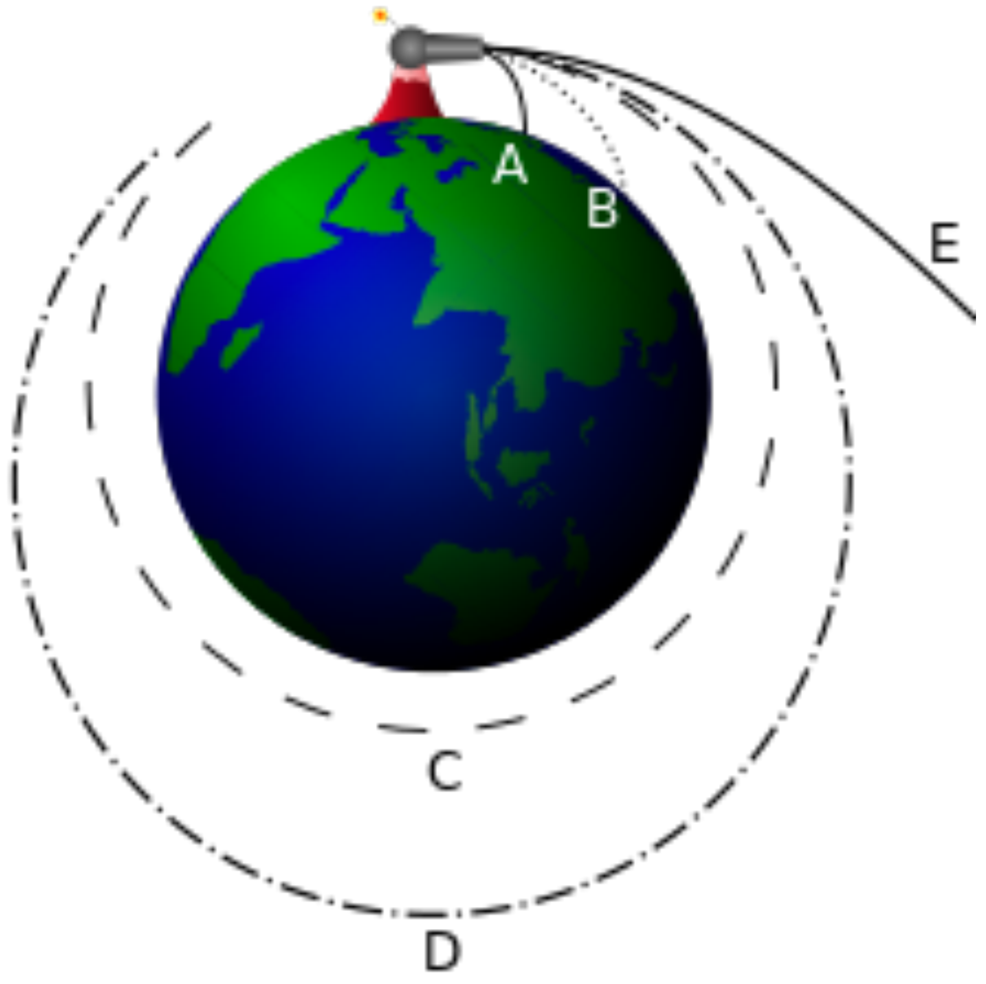


Kemp & Tenenbaum (2008)



Falkenhainer, Forbus & Gentner (1989)

Battaglia, Hamrick & Tenenbaum (2013)



Newton (1687)

How do we build AI that can…

- Understand hierarchical relationships, like family structure or evolutionary history?
- Understand how objects in the world interact with each other under different physical dynamics?
- Make analogies between disparate phenomena, despite a lack of surface similarity?
- Perform novel thought experiments—or even real experiments!—to gain insight about the world?

All of these are forms of reasoning that (in humans) require understanding the world in terms of ***entities,*** the ***relations*** between them, and the ***rules*** for composing them.

Structure: the product of composing a known set of ***entities*** and ***relations*** according to a particular set of ***rules***.

What should structure look like in modern AI systems?

Outline

1. Structure and inductive bias in deep learning
2. Graph networks for deep learning on graphs
3. Graph networks for physical inference
4. Graph networks for physical construction

Outline

1. **Structure and inductive bias in deep learning**
2. Graph networks for deep learning on graphs
3. Graph networks for physical inference
4. Graph networks for physical construction

Deep Learning

Supervised Learning

Unsupervised Learning

Reinforcement Learning

$$y = f(x; \theta)$$

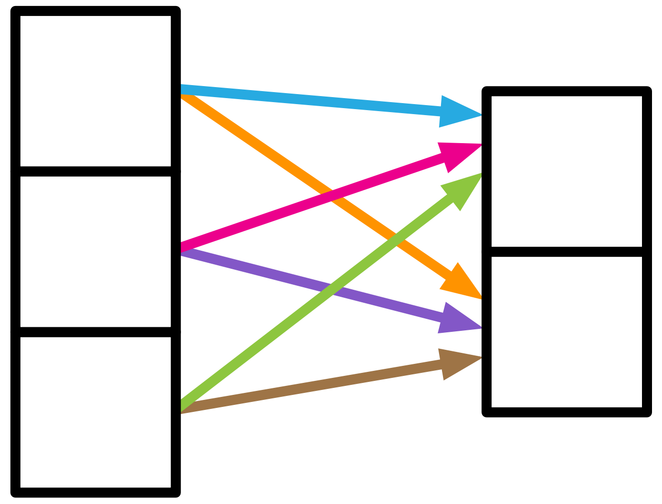
$$\min_{\theta} \mathcal{L}(y, f(x; \theta))$$

Gradient-based optimization ("backpropagation")
Usually stochastic gradient descent

"Layers" of differentiable,
nonlinear functions

The Multi-Layer Perceptron (MLP)

Rosenblatt (1961)

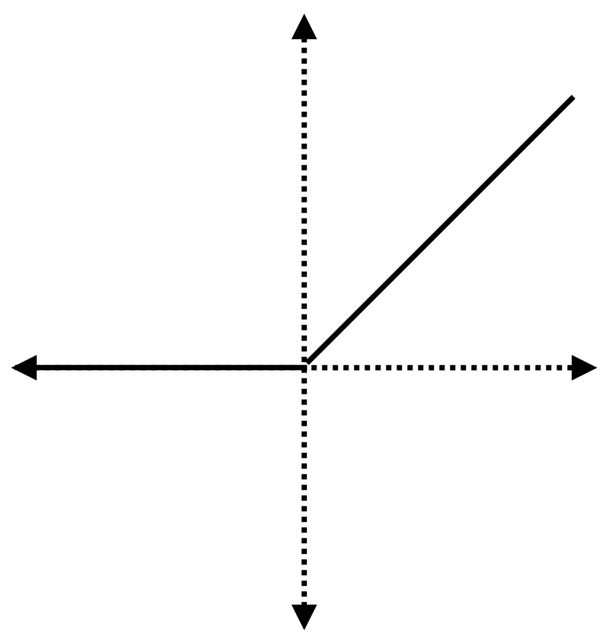


$$\text{Linear}(x) := Wx + b$$

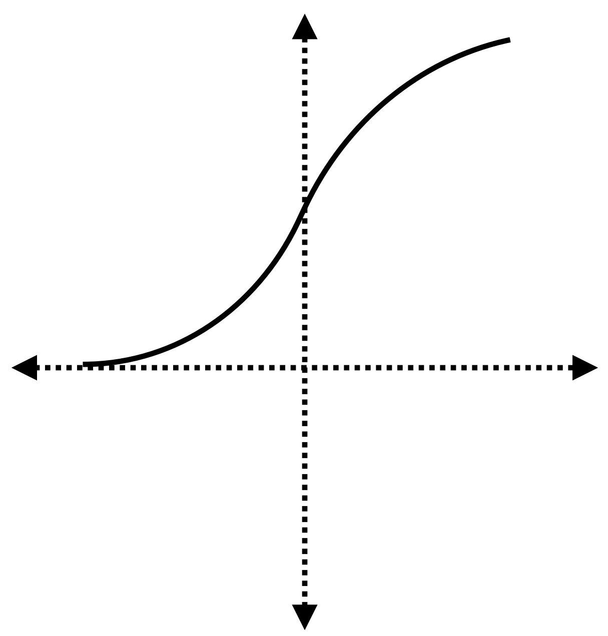
$$\text{FC}(x) := \text{ReLU}(\text{Linear}(x))$$

$$\text{MLP}(x) := \text{Linear}(\text{FC}(\cdots \text{FC}(x)))$$

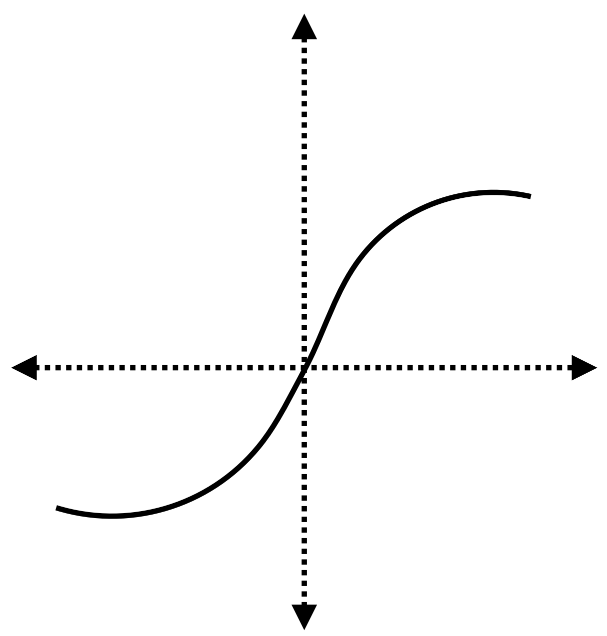
ReLU



Sigmoid

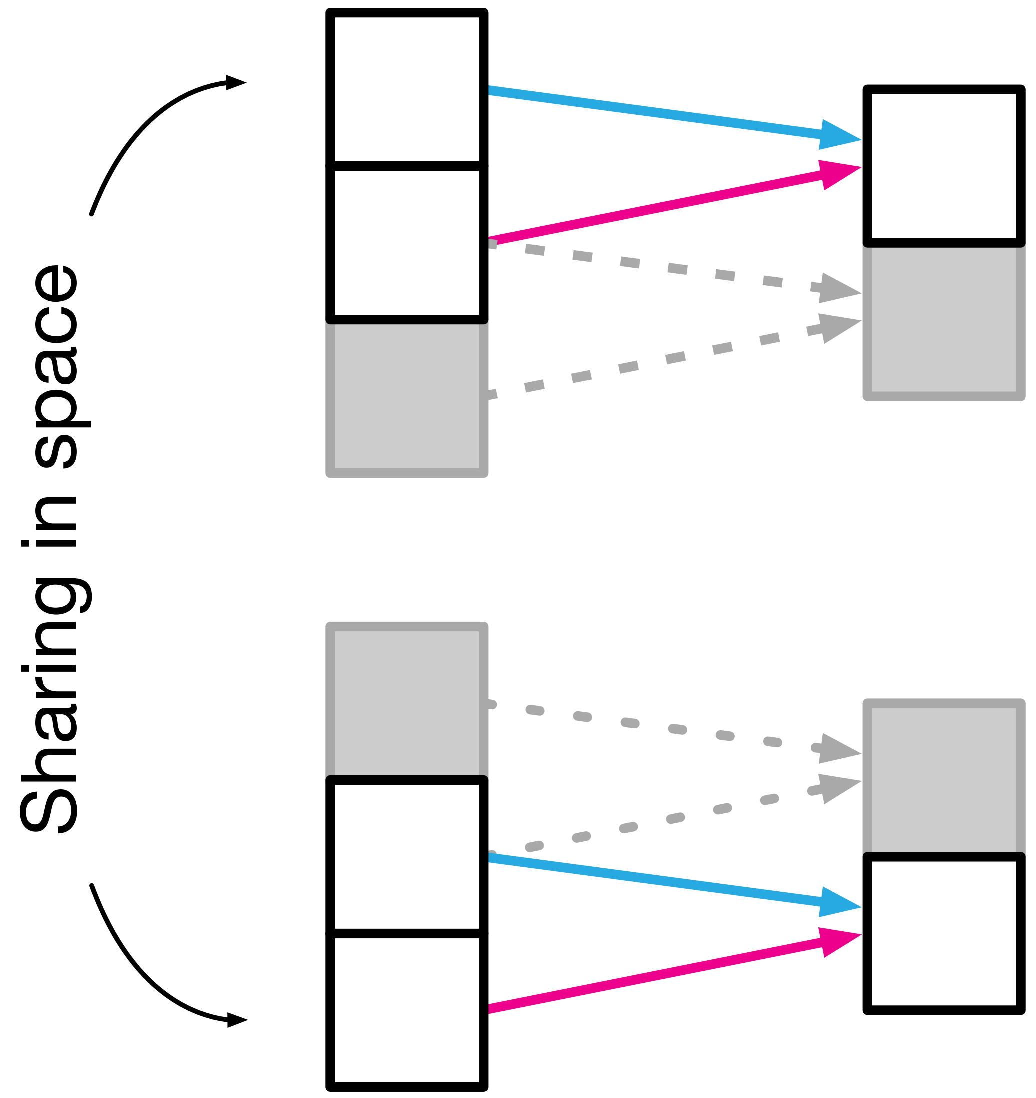


Tanh



The Convolutional Network (CNN)

Fukushima (1980), LeCun et al. (1989)



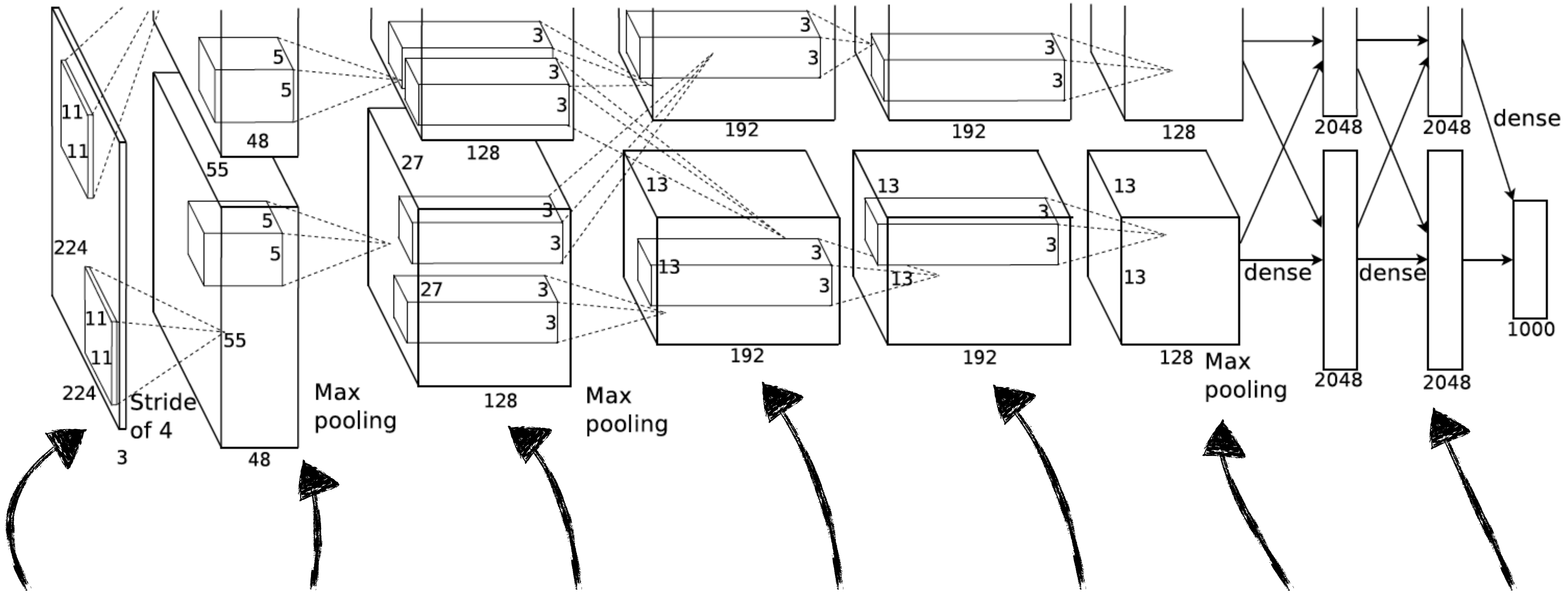
$$\mathrm{Conv}(x) := \mathrm{ReLU}(x * W)$$

Usually multiple convolutions with different weights are applied to the same input

Often followed by an additional nonlinearity to reduce the size, such as "max pooling"

AlexNet

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012)



Input

Convolution
ReLU
Max Pooling

Convolution
ReLU
Max Pooling

Convolution
ReLU

Convolution
ReLU

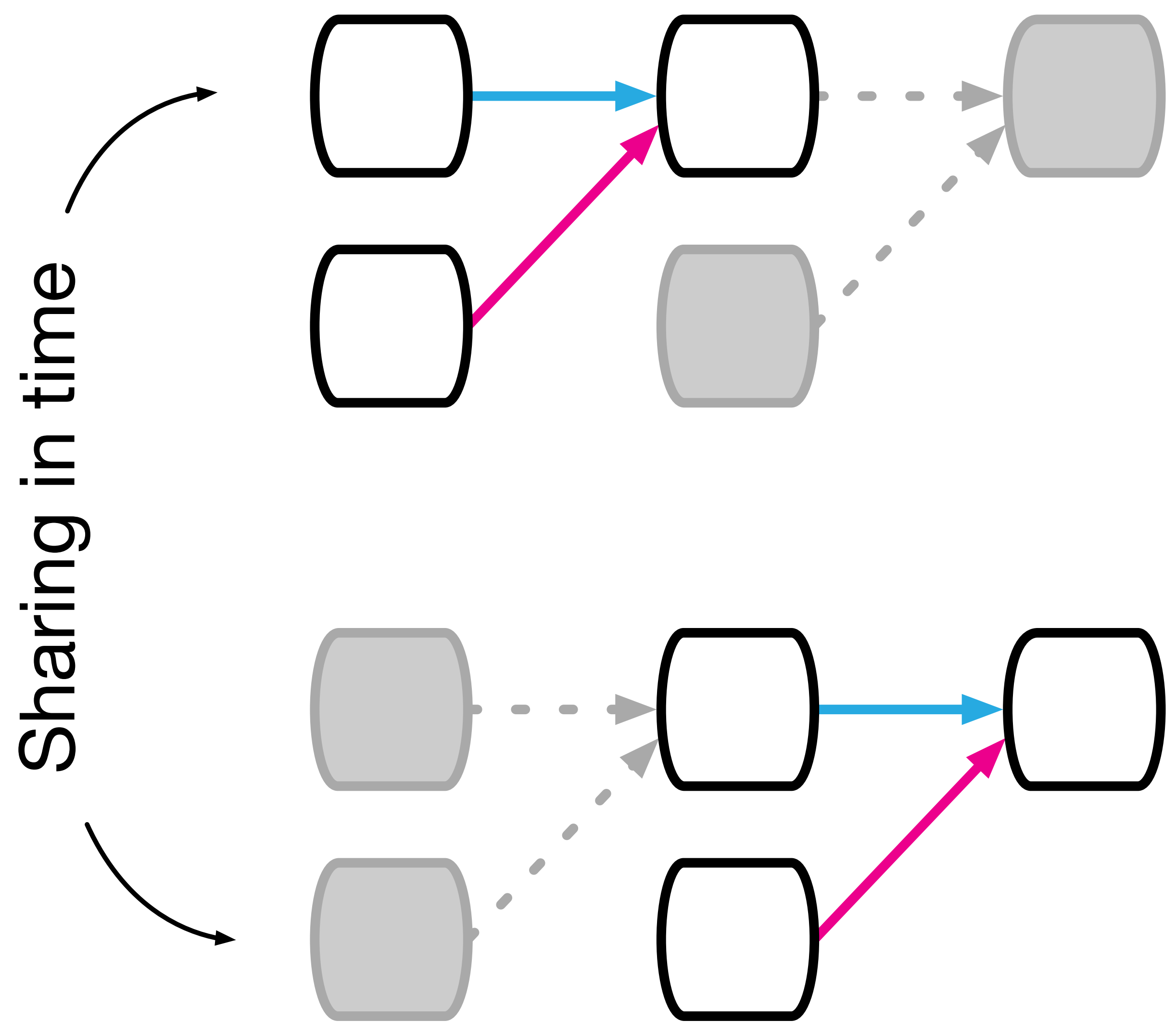
Convolution
ReLU
Max Pooling

MLP

“Cat”

The Recurrent Network (RNN)

Elman (1990)

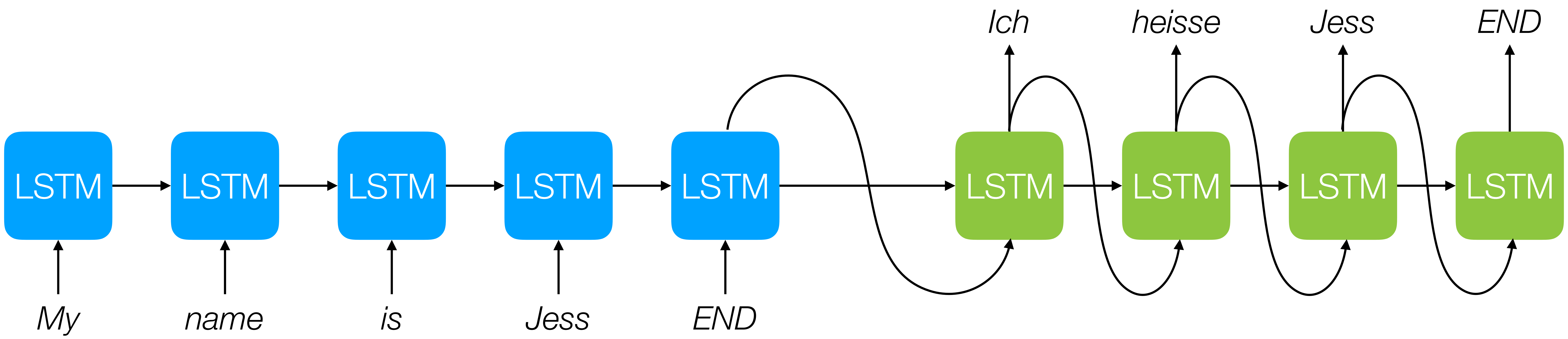


$$[y, h_{t+1}] = \mathrm{RNN}(x, h_t) := \mathrm{MLP}([x, h_t])$$

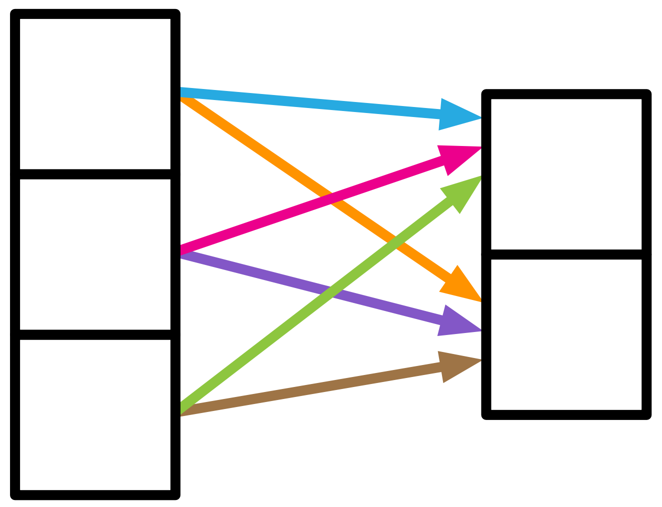
In practice, modern RNNs like LSTMs (Hochreiter & Schmidhuber, 1997) or GRUs (Cho et al. 2014) use a more complicated base function than an MLP

Sequence2Sequence

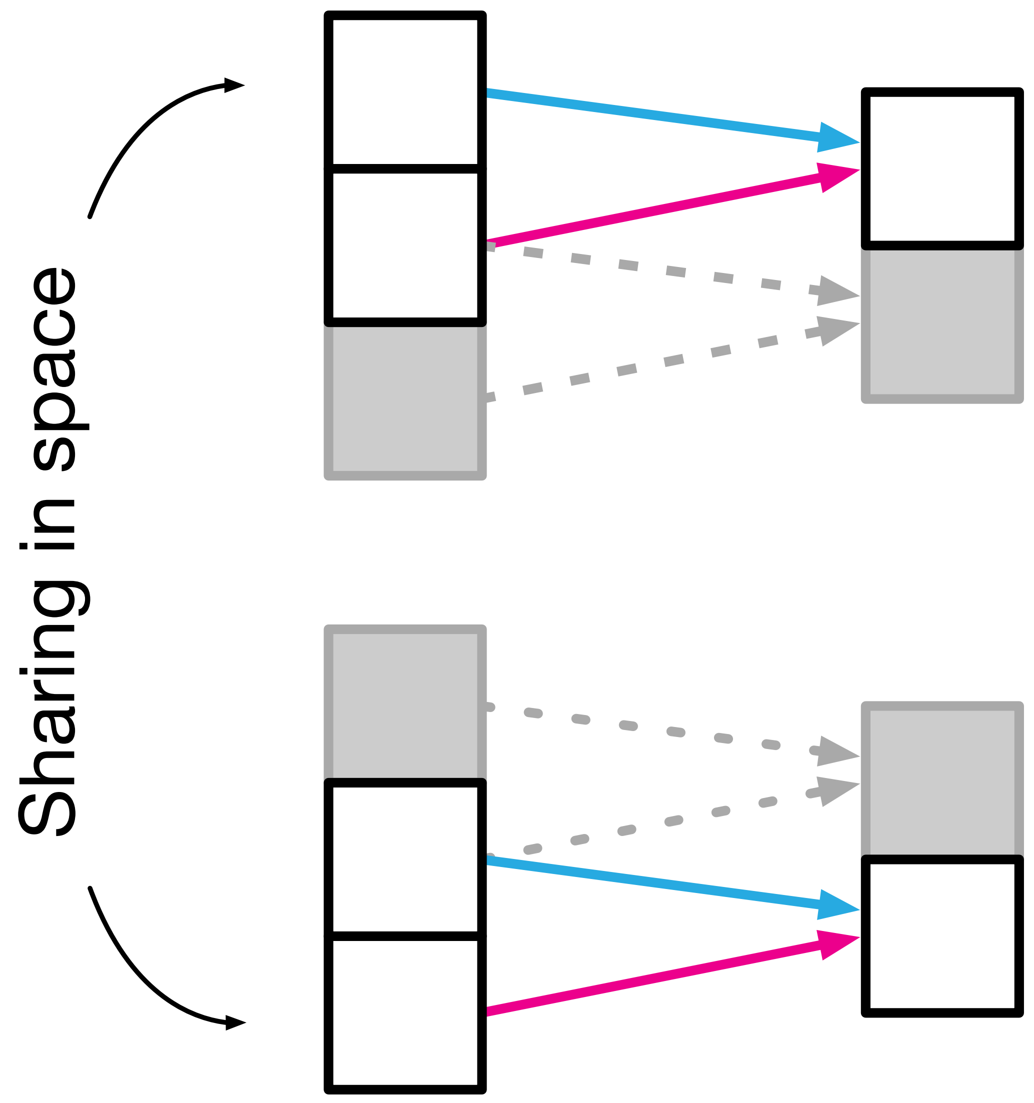
Sutskever, I., Vinyals, O., & Le, Q. V. (2014)



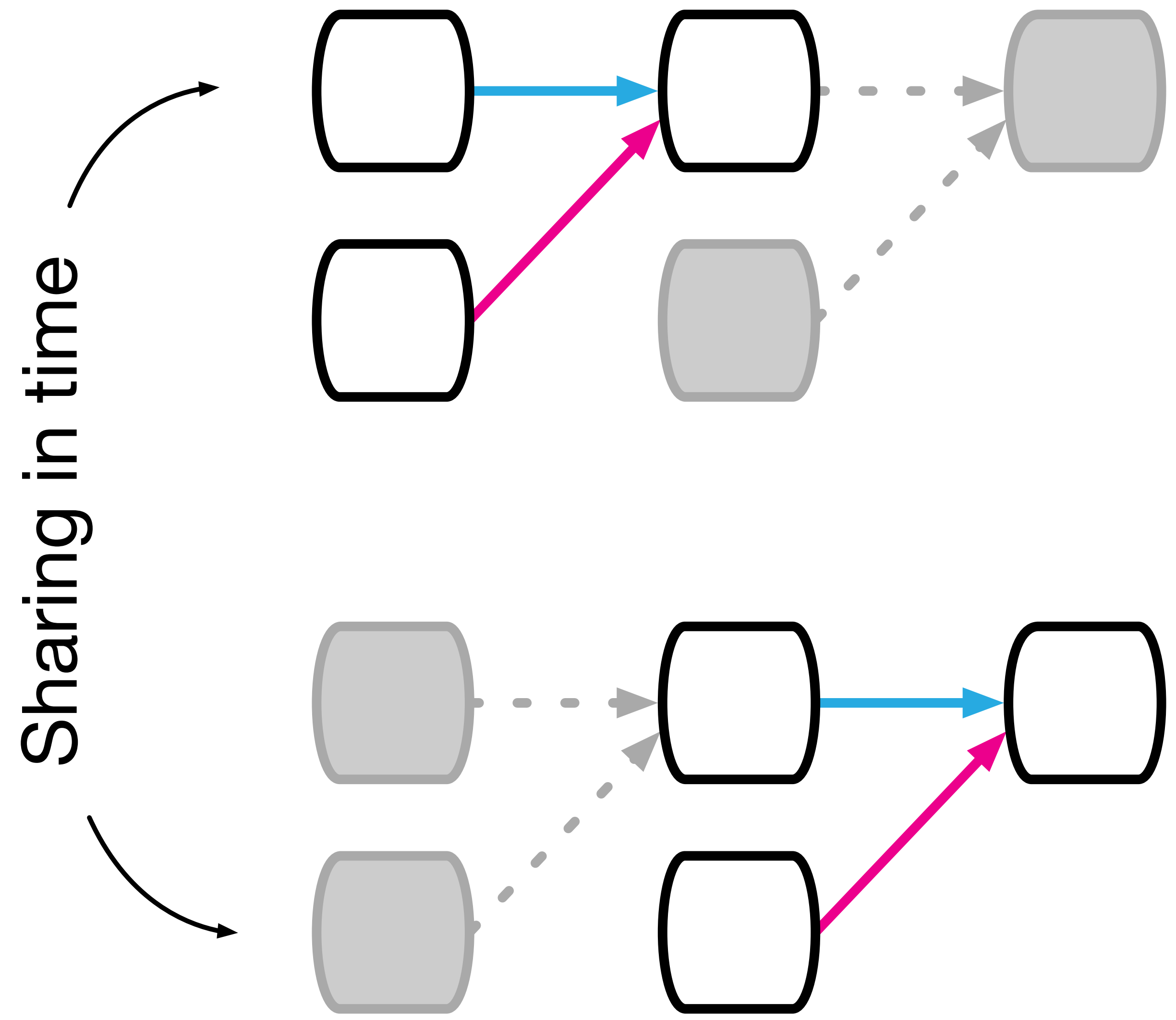
The Story So Far



Fully-Connected Layer
Fixed size tensors



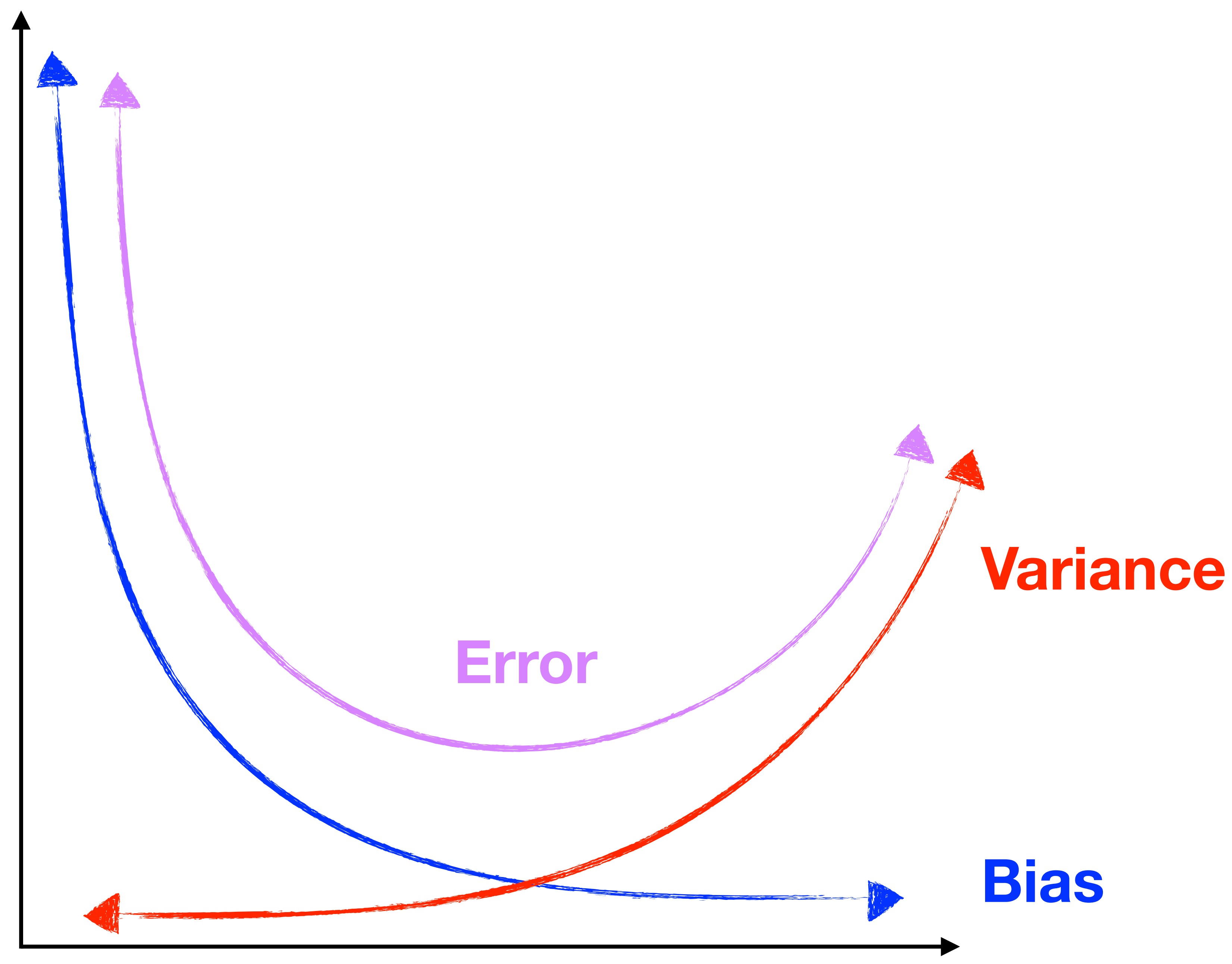
Convolutional Layer
Variable size tensors



Recurrent Layer
Variable-size ordered sequence of tensors

Inductive Bias

Inductive Bias: the part of a learning algorithm which allows it to prioritize one solution (or interpretation) over another, independent of the observed data (Mitchell, 1980)



CNNs prefer solutions which are spatially invariant.

RNNs prefer solutions in which order matters.

What other forms of (architectural) inductive bias might we be able to use?

Relational inductive bias: prioritizes solutions in which there are *rules* for combining *entities* and *relations*.

Outline

1. Structure and inductive bias in deep learning

2. Graph networks for deep learning on graphs

3. Graph networks for physical inference

4. Graph networks for physical construction

Beyond Tensor Data

History of Graph Neural Networks

Scarselli et al. (2009) "The Graph Neural Network Model”.
Summarizes the initial papers on the topic from ~2005-2009. Very general formalism.

Li et al. (2015) “Gated graph sequence neural networks”.
Used RNNs for sharing update steps across time.

Bronstein et al. (2016) “Geometric deep learning: going beyond Euclidean data”.
Survey of spectral and spatial approaches for deep learning on graphs.

Gilmer et al. (2017) “Neural Message Passing for Quantum Chemistry”.
Introduced “message-passing neural network” (MPNNs) formalism, unifying various approaches such as graph convolutional networks.

Battaglia et al. (2018). “Relational inductive biases, deep learning, and graph networks”.
Introduced “graph network” formalism, extends MPNNs, unifies non-local neural networks/self-attention/Transformer.

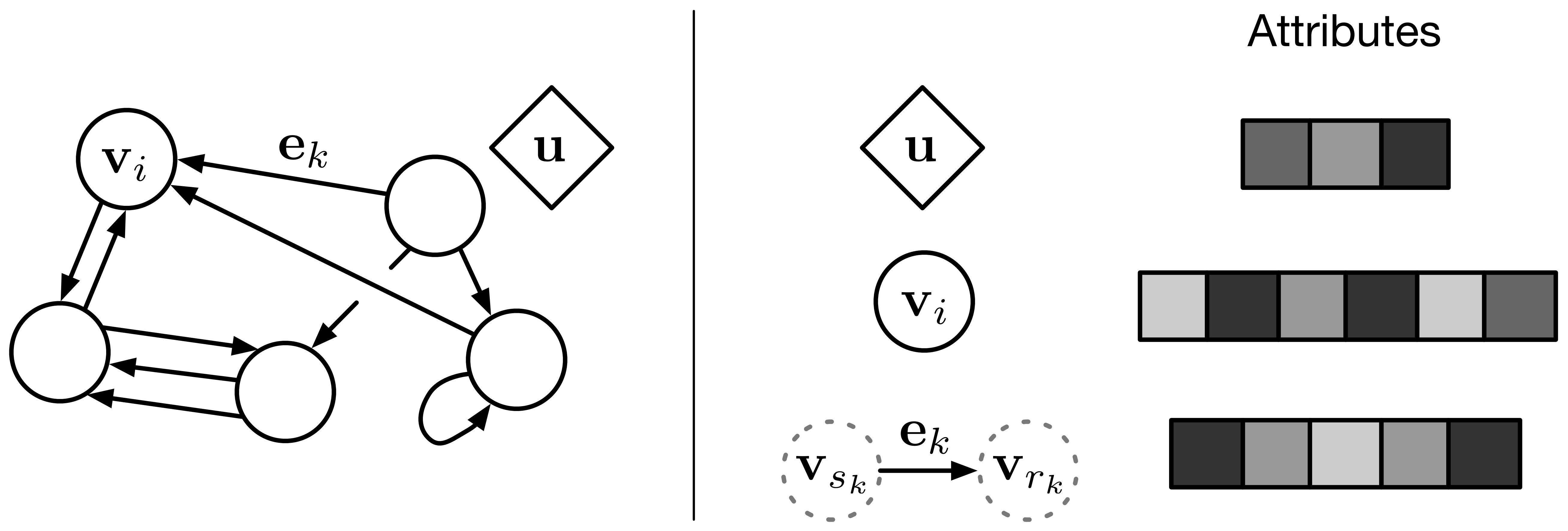
Graph Networks

Battaglia, Hamrick, Bapst, Sanchez-Gonzalez, Zambaldi, et al. (2018)

1. Takes **graphs** as input, return graphs as output

2. Invariant to the **permutation** of the nodes and edges

3. Scales to different **numbers** of nodes and edges

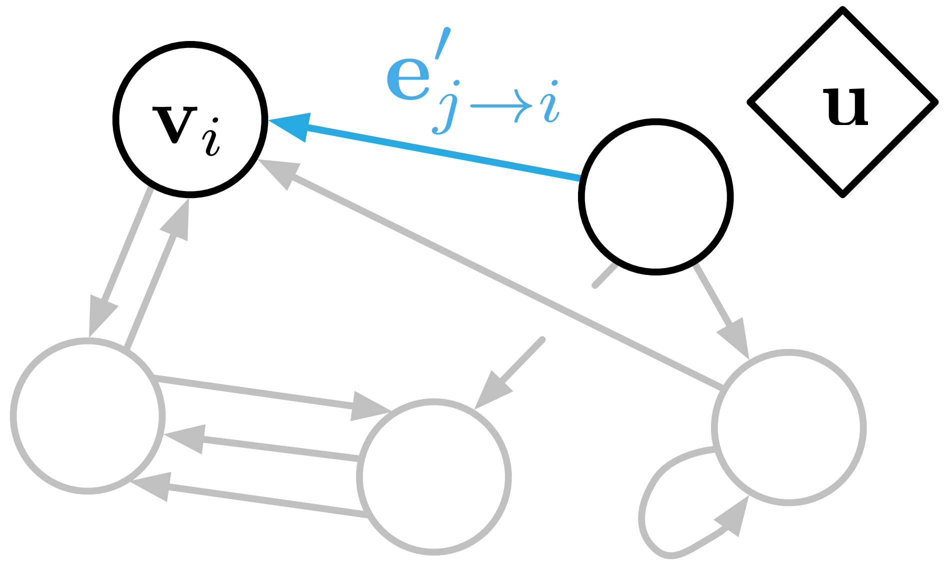


Graph Networks

Battaglia, Hamrick, Bapst, Sanchez-Gonzalez, Zambaldi, et al. (2018)

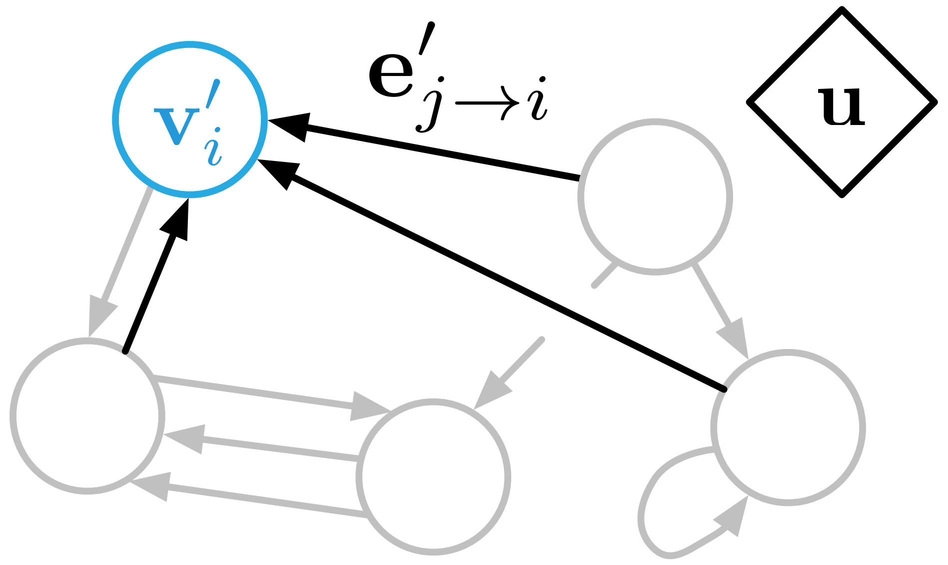
Edges	Nodes	Globals
E	V	$\mathbf{u}$

Edge update



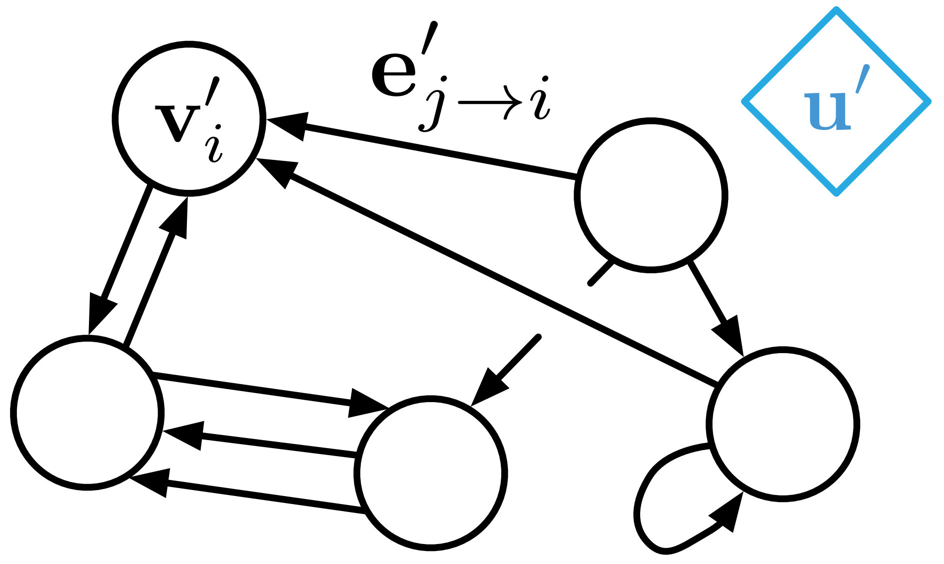
$$\mathbf{e}'_{i\to j} = \phi_e(\mathbf{v}_i, \mathbf{v}_j, \mathbf{e}_{i\to j}, \mathbf{u})$$

Node update



$$\mathbf{v}'_i = \phi_v(\mathbf{v}_i, \sum_j \mathbf{e}'_{j\to i}, \mathbf{u})$$

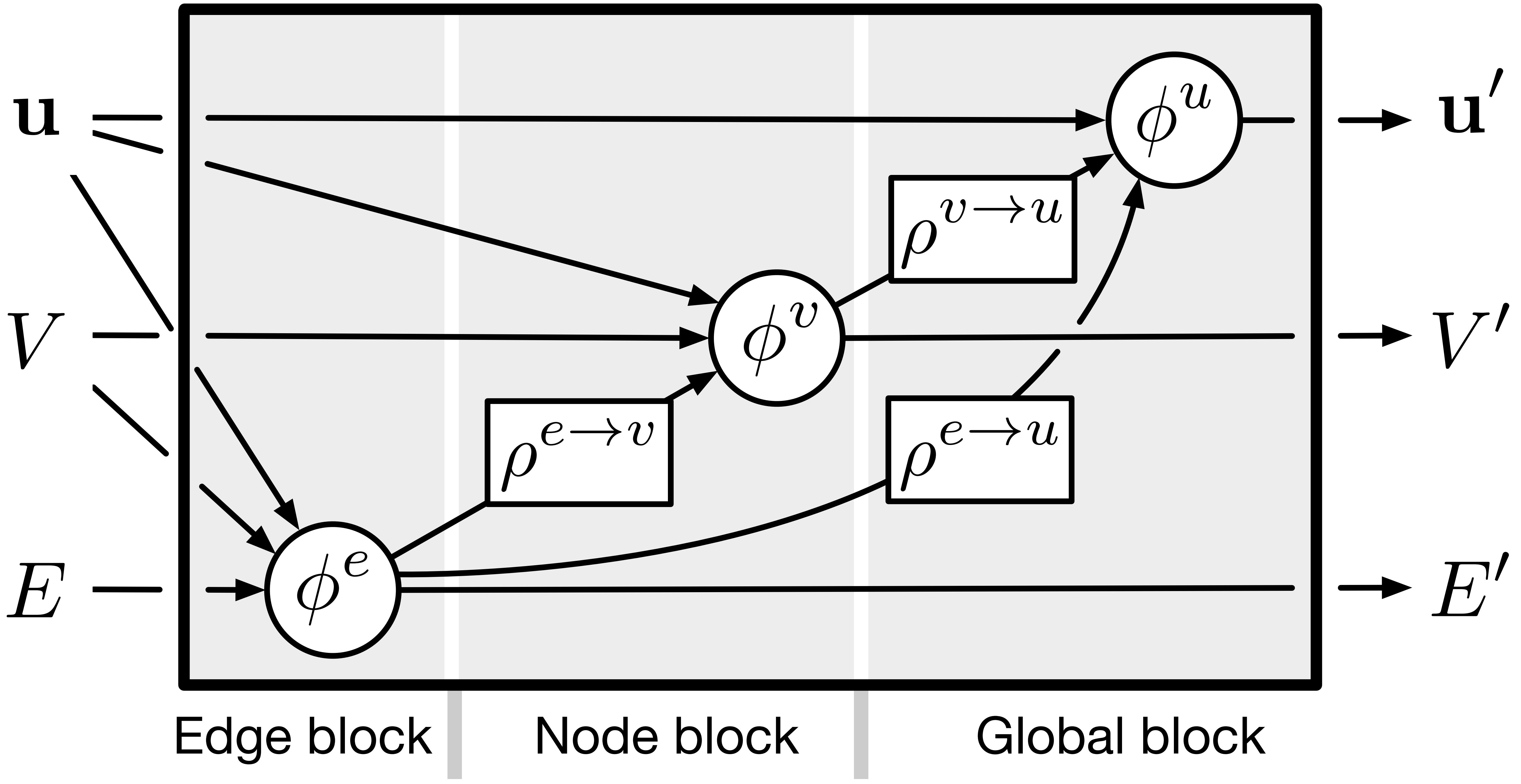
Globals update



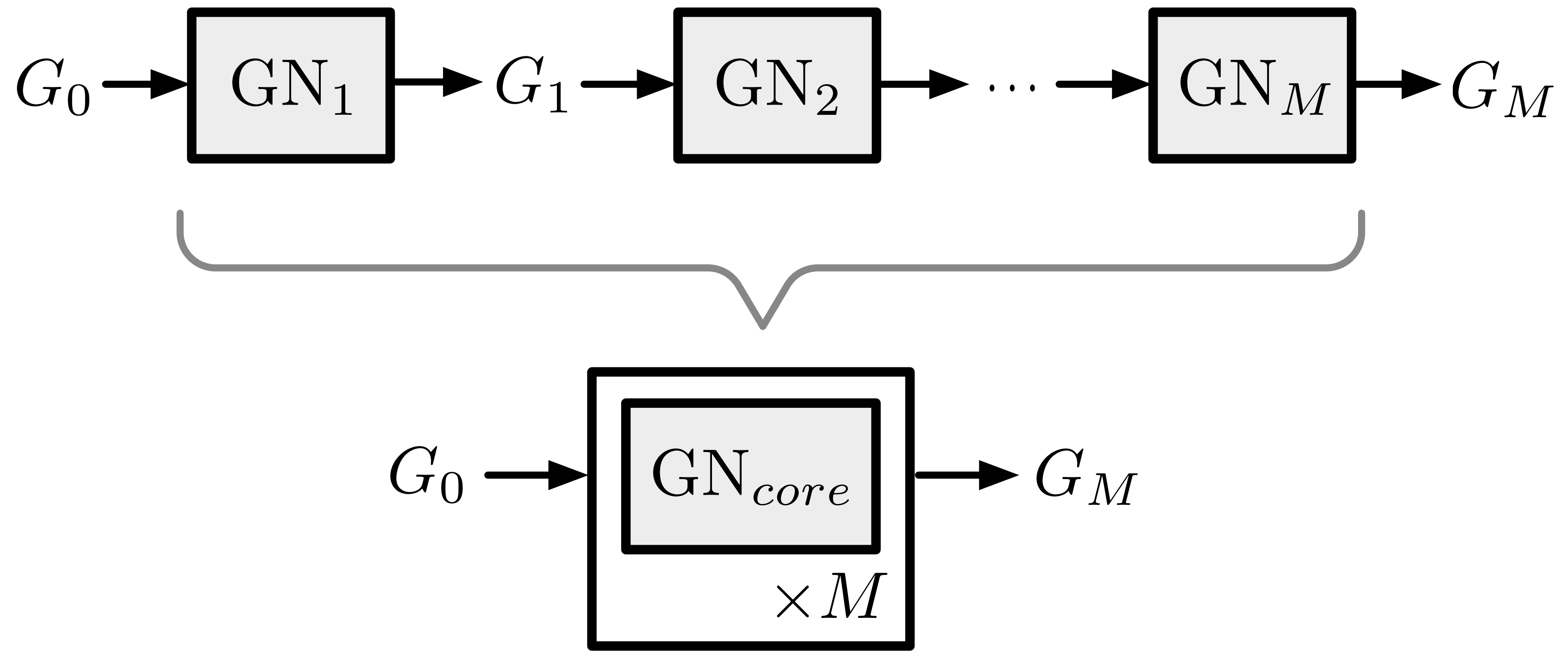
$$\mathbf{u}' = \phi_u(\sum_i \mathbf{v}'_i, \sum_{i,j} \mathbf{e}'_{i\to j}, \mathbf{u})$$

Graph Networks

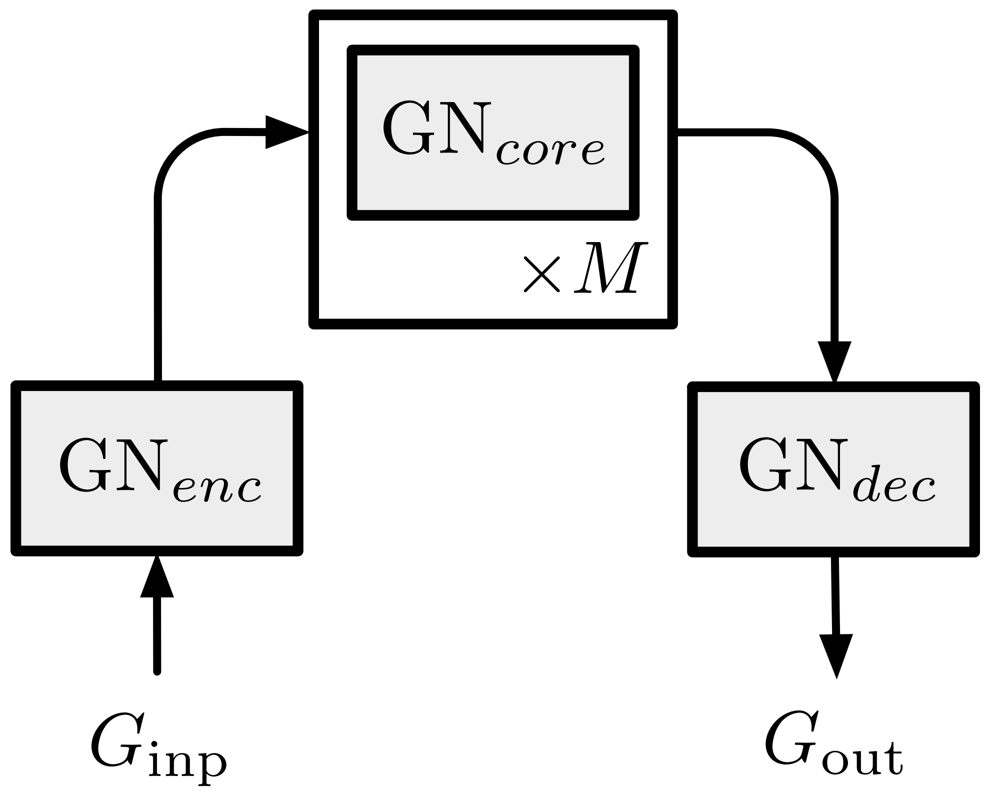
Battaglia, Hamrick, Bapst, Sanchez-Gonzalez, Zambaldi, et al. (2018)



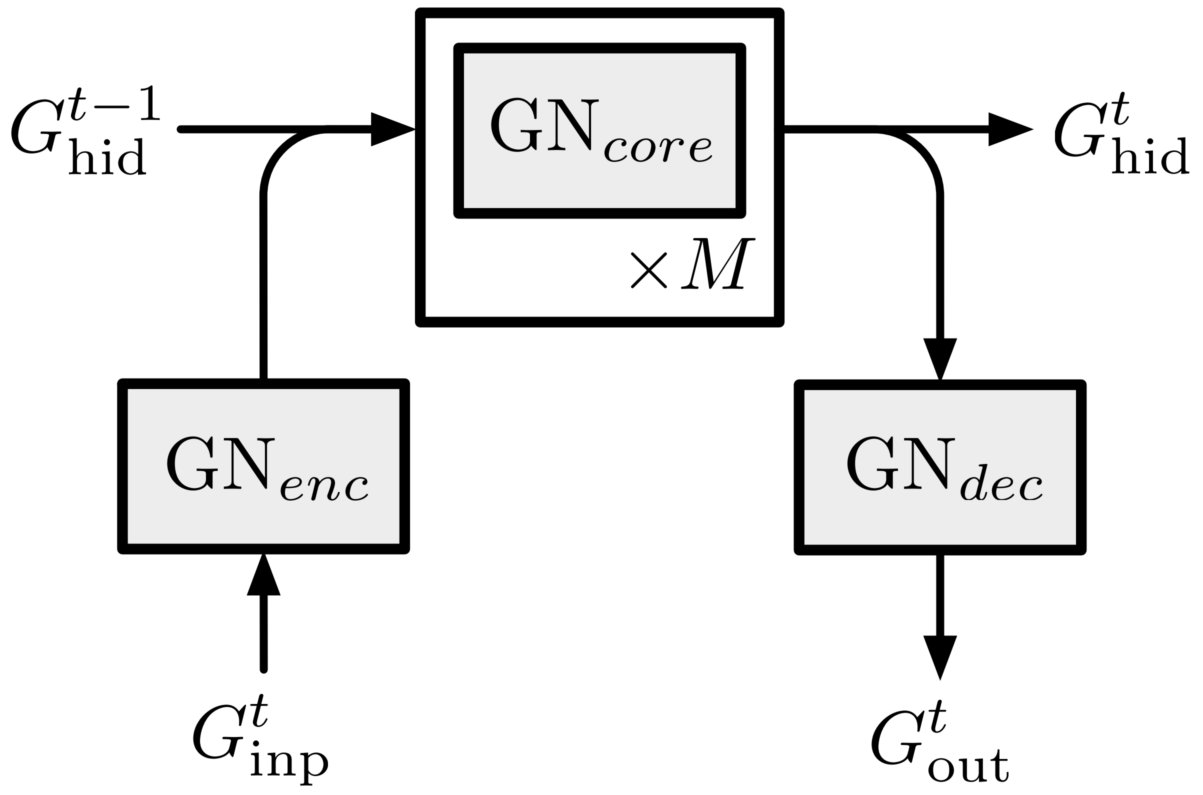
Composing Graph Networks



Shared GN Core



Encode-Process-Decode



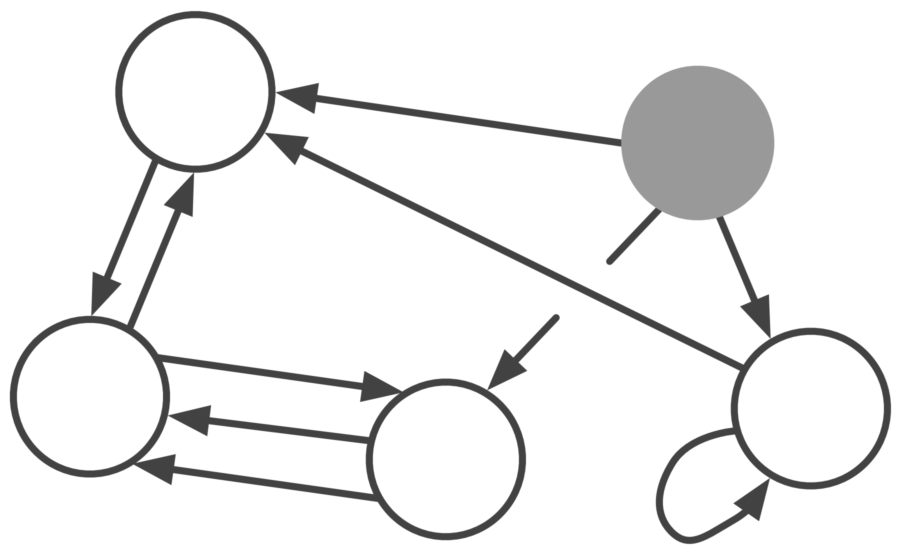
Recurrent GN Architecture

“Message Passing”

(Gilmer et al, 2017)

Message Passing

Gilmer et al. (2017)



$m = 0$

What are graph networks good for?

- Visual scene understanding
- Few-shot learning
- Predicting physical dynamics
- Predicting multi-agent systems
- Reasoning in knowledge graphs
- Predicting chemical properties of molecules
- Predicting road traffic
- Image segmentation
- 3D mesh classification
- Image region classification
- Semi-supervised text classification
- Machine translation
- Continuous control
- Object-oriented RL
- Symbolic planning
- Combinatorial optimization
- Boolean SAT problems
- Modeling cellular automata
- Inference in graphical models
- … and more!

Outline

1. Structure and inductive bias in deep learning

2. Graph networks for deep learning on graphs

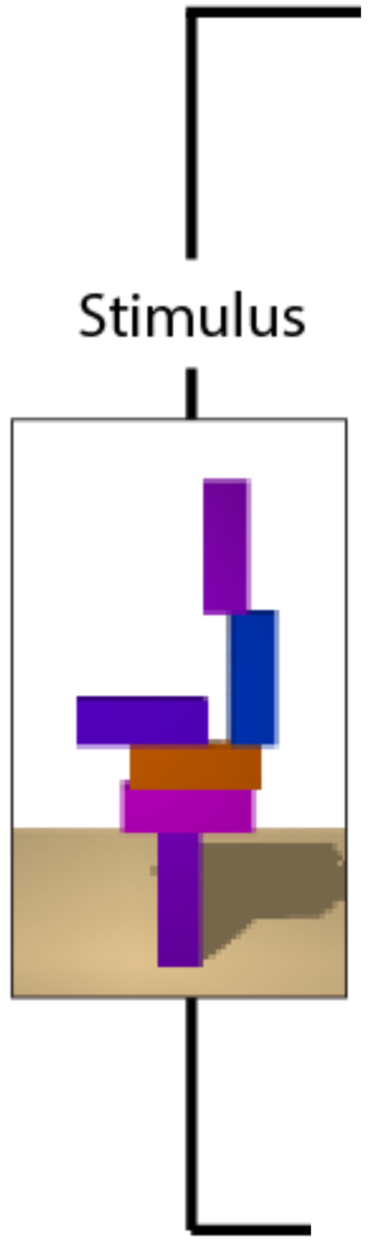
3. Graph networks for physical inference

4. Graph networks for physical construction

Humans are a “Construction Species”

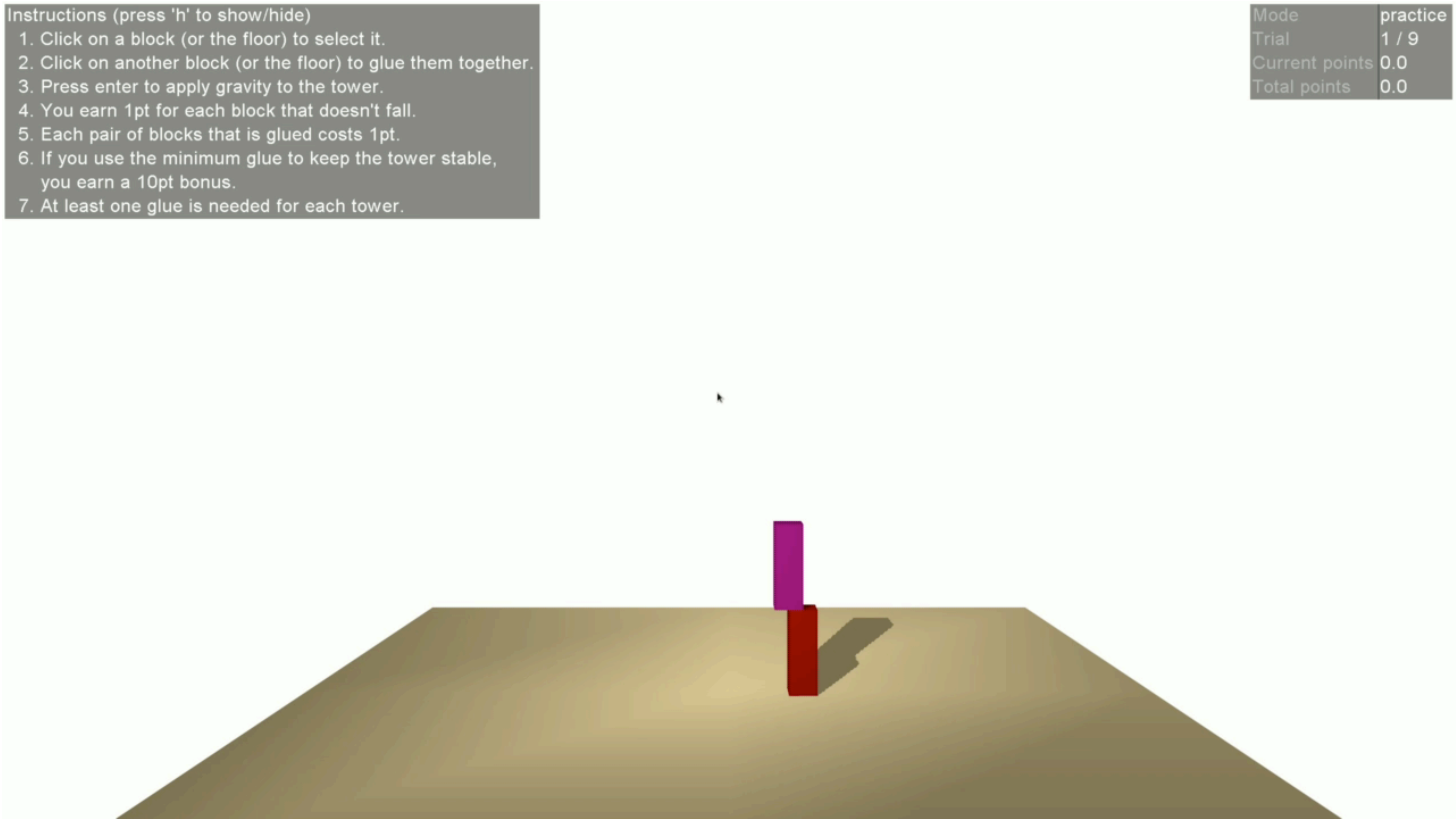
The Gluing Task

Hamrick*, Allen*, Bapst, Zhu, McKee, Tenenbaum & Battaglia (*CogSci* 2018)

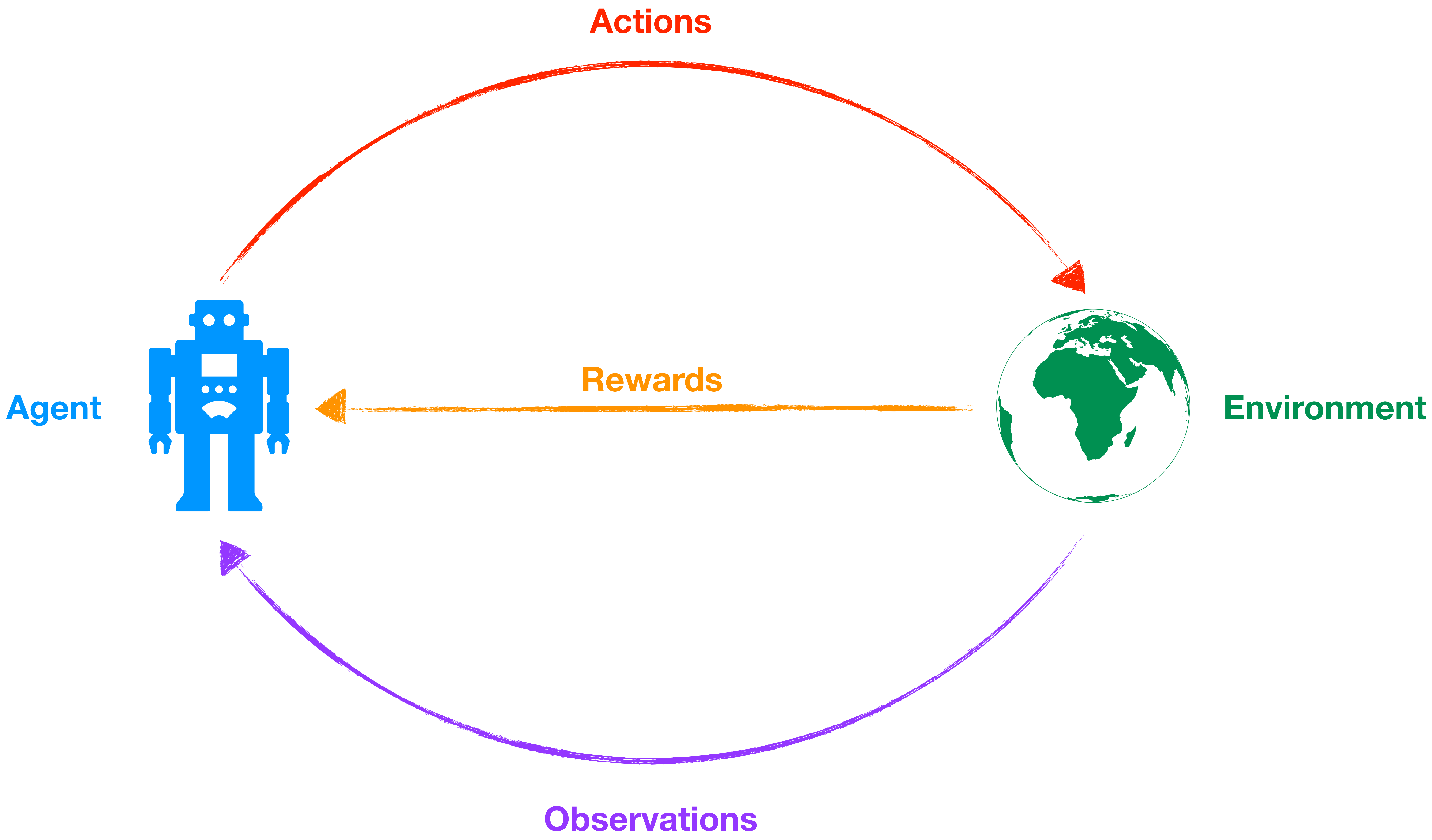


Joint work with
Kelsey Allen (MIT)

Goal: glue blocks together to make the tower stable, using the minimum amount of glue.


Instructions (press 'h' to show/hide)
1. Click on a block (or the floor) to select it.
2. Click on another block (or the floor) to glue them together.
3. Press enter to apply gravity to the tower.
4. You earn 1pt for each block that doesn't fall.
5. Each pair of blocks that is glued costs 1pt.
6. If you use the minimum glue to keep the tower stable, you earn a 10pt bonus.
7. At least one glue is needed for each tower.
Mode practice
Trial 1 / 9
Current points 0.0
Total points 0.0

Reinforcement Learning



Reinforcement Learning

Learn a "policy" which maximizes the sum of discounted expected future rewards

The Bellman Equation:

$$Q(s,a) = r + \gamma \cdot \max_{a'} Q(s',a')$$

Q-Learning:

$$\mathcal{L} := Q(s,a;\theta) - (r + \gamma \cdot \max_{a'} Q(s',a';\theta))$$

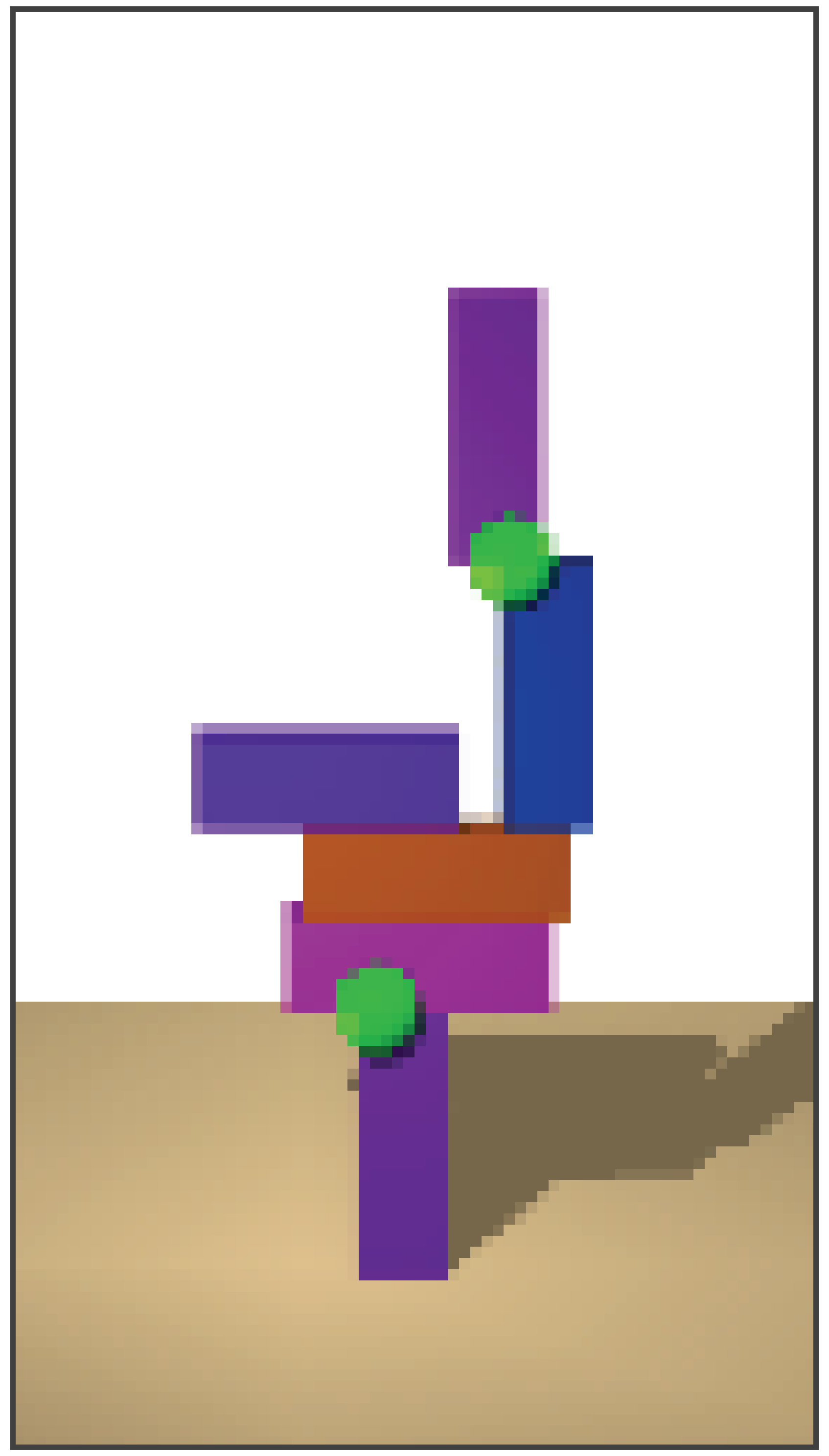
Policy:

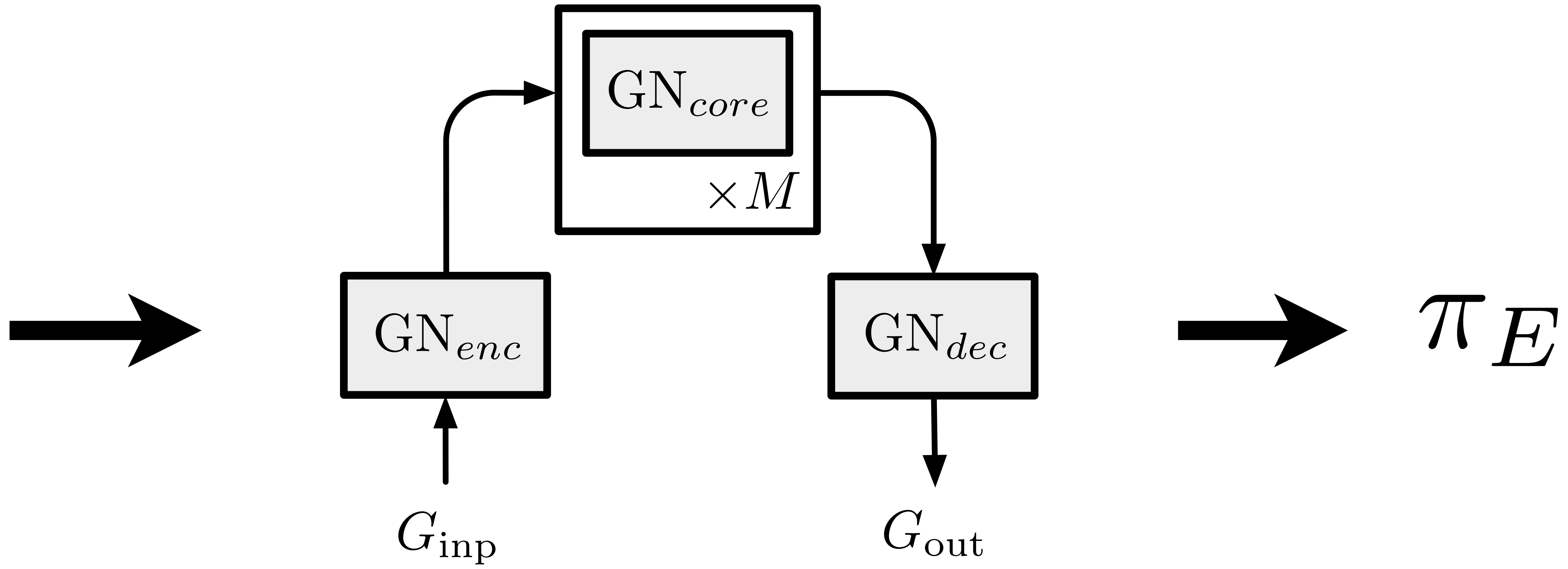
$$\pi(s) := \arg\max_{a} Q(s,a;\theta)$$

Deep Q-Learning (DQN) approximates Q with a neural network that takes a single state as input and returns Q-values for all possible actions.

Learning a Policy Over the Edges of a Graph

Hamrick*, Allen*, Bapst, Zhu, McKee, Tenenbaum & Battaglia (*CogSci* 2018)





Agent Variations

Hamrick*, Allen*, Bapst, Zhu, McKee, Tenenbaum & Battaglia (*CogSci* 2018)

(Trained & tested on towers of size 2-10 blocks)

Human: human baseline

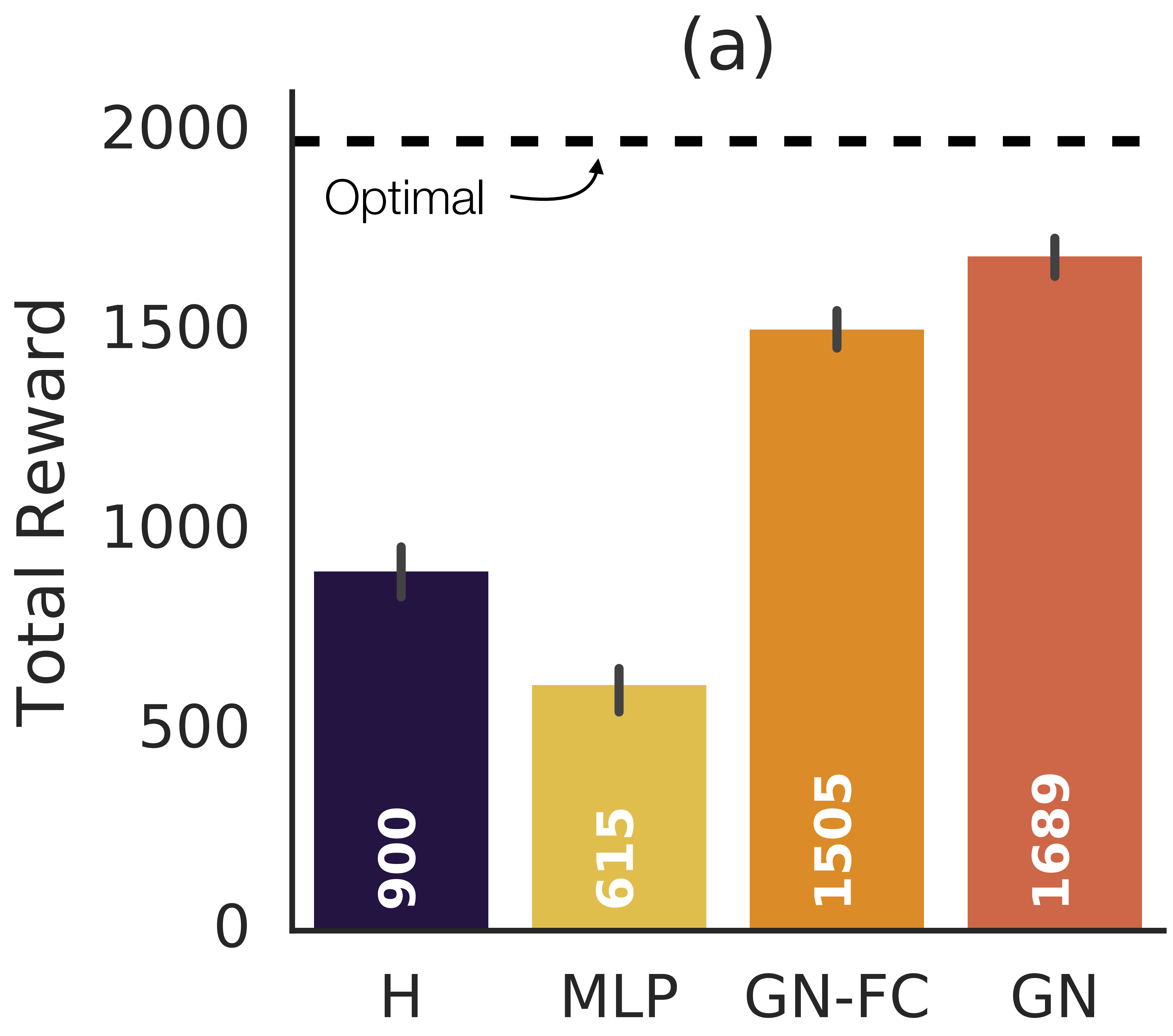
MLP: multilayer perceptron agent

GN-FC: fully connected graph network agent
(nodes=blocks, edges=all-to-all)

GN: sparse graph network agent
(nodes=blocks, edges=contacts)

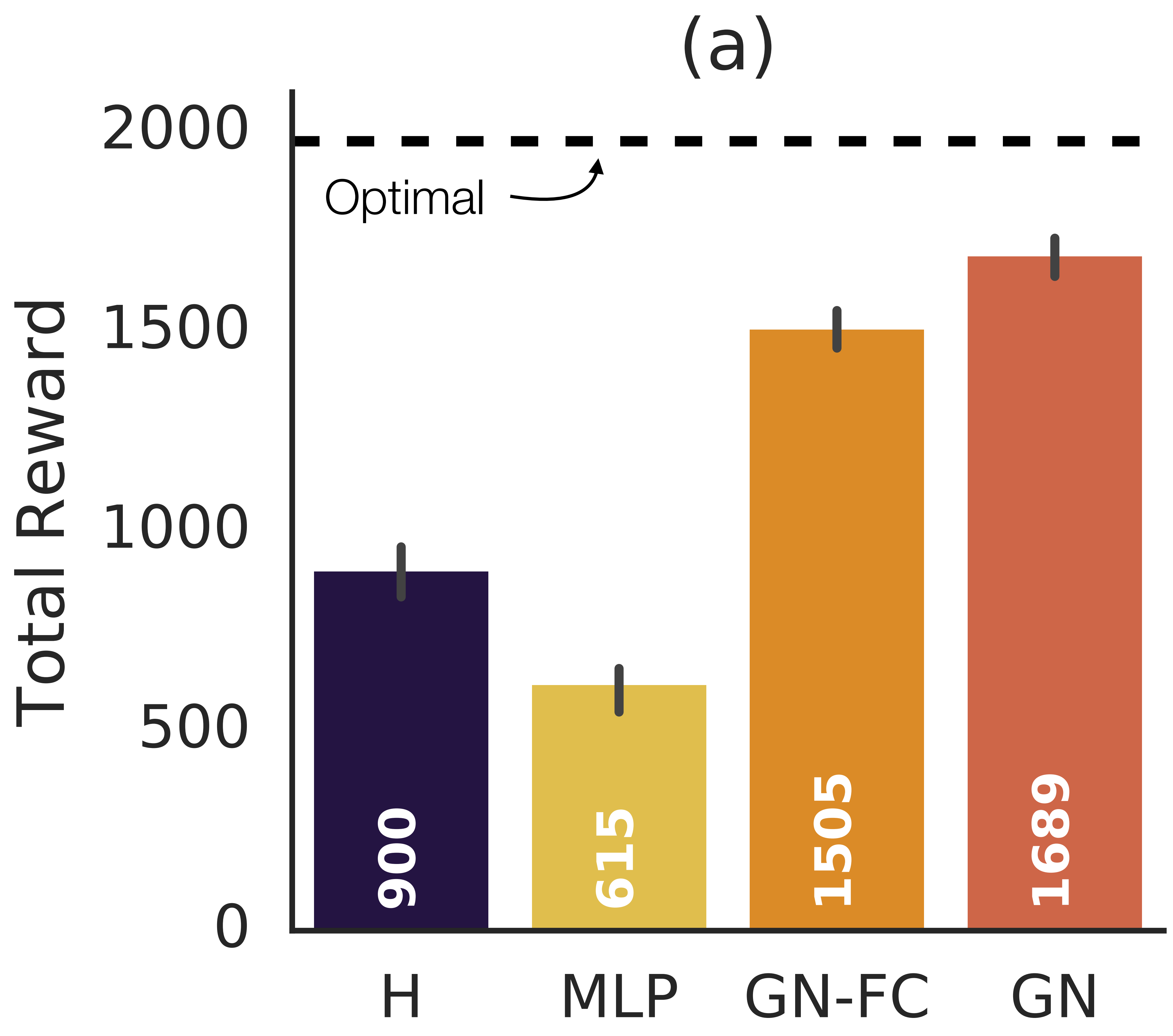
Results

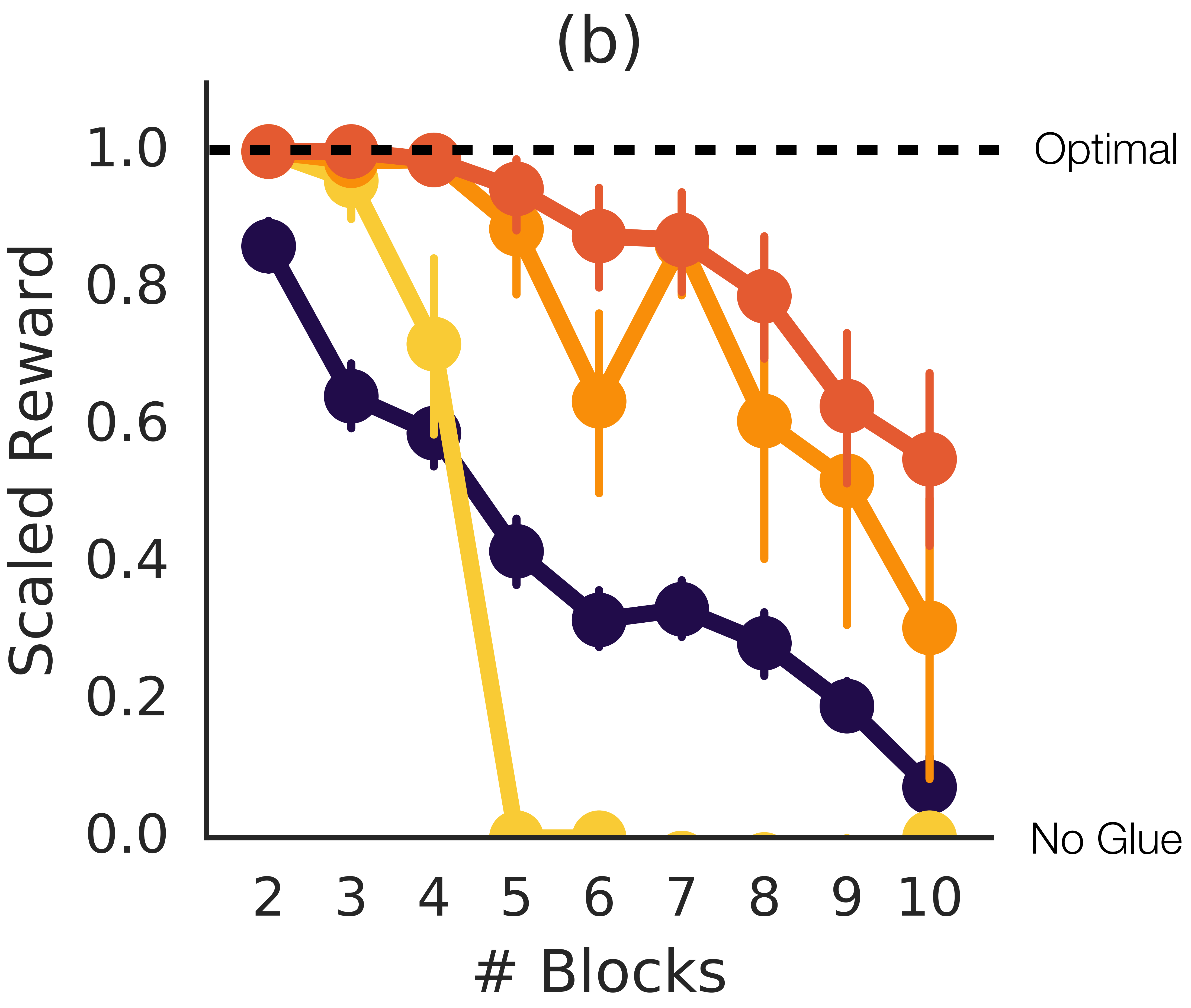
Hamrick*, Allen*, Bapst, Zhu, McKee, Tenenbaum & Battaglia (*CogSci* 2018)



Results

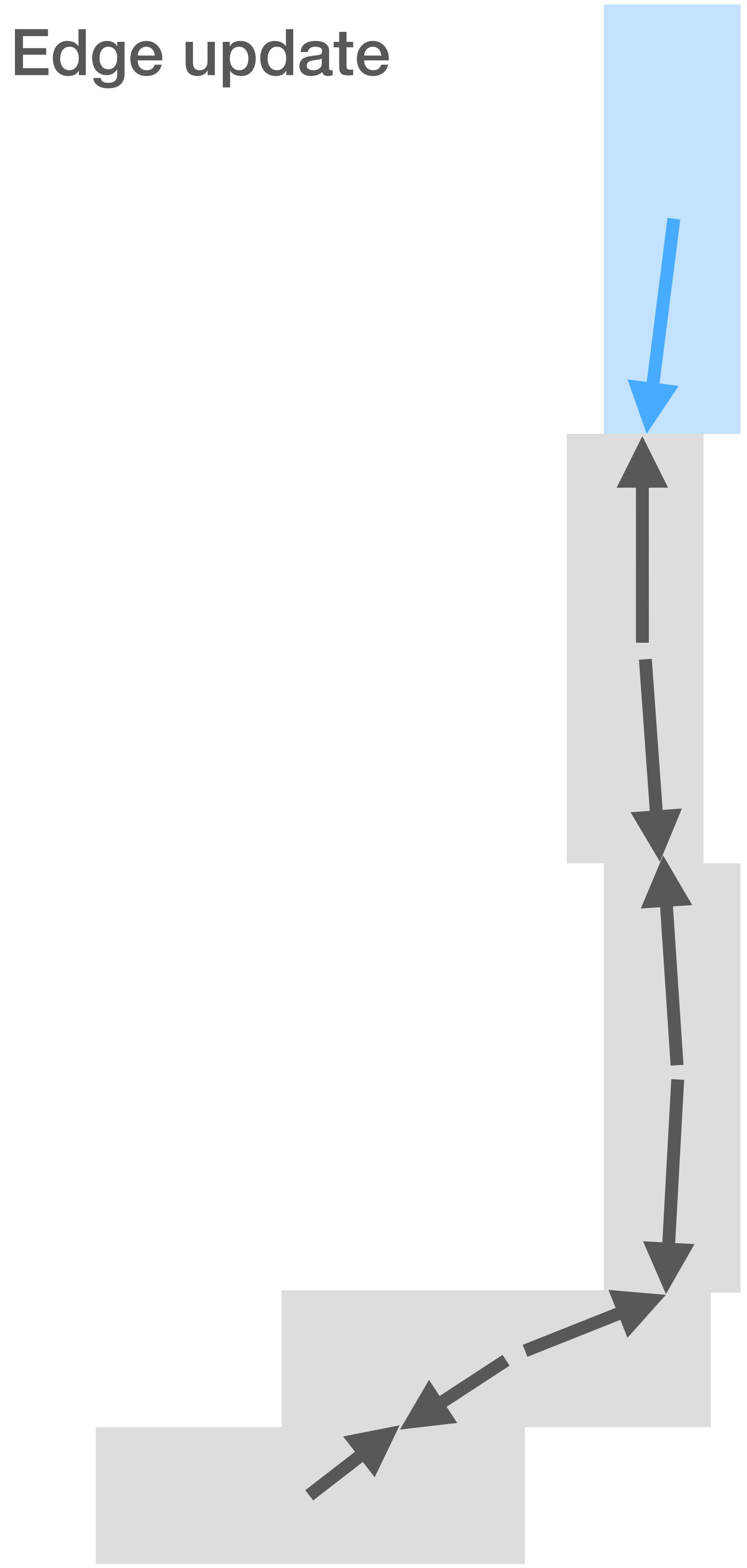
Hamrick*, Allen*, Bapst, Zhu, McKee, Tenenbaum & Battaglia (*CogSci* 2018)



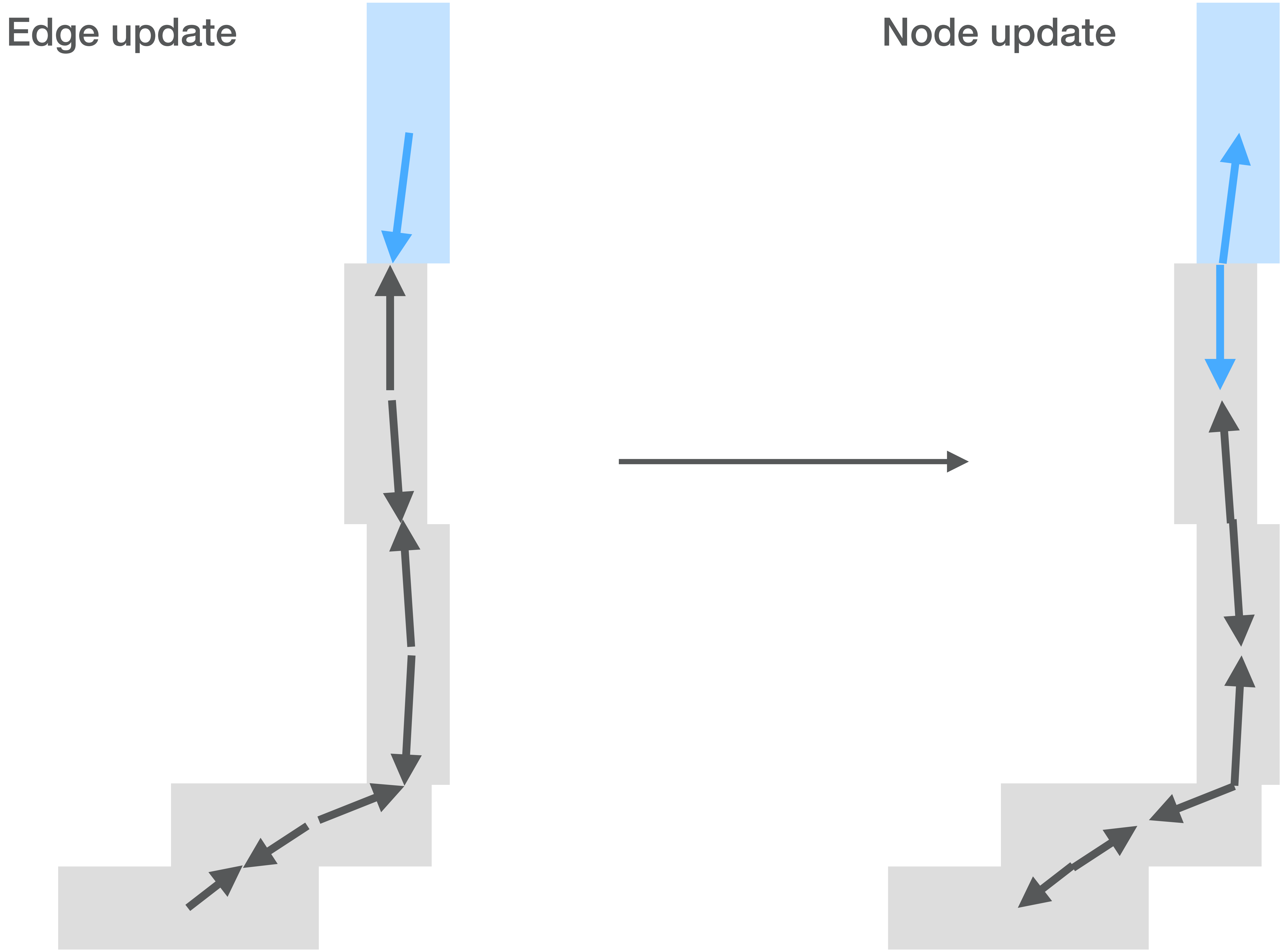


Force Propagation

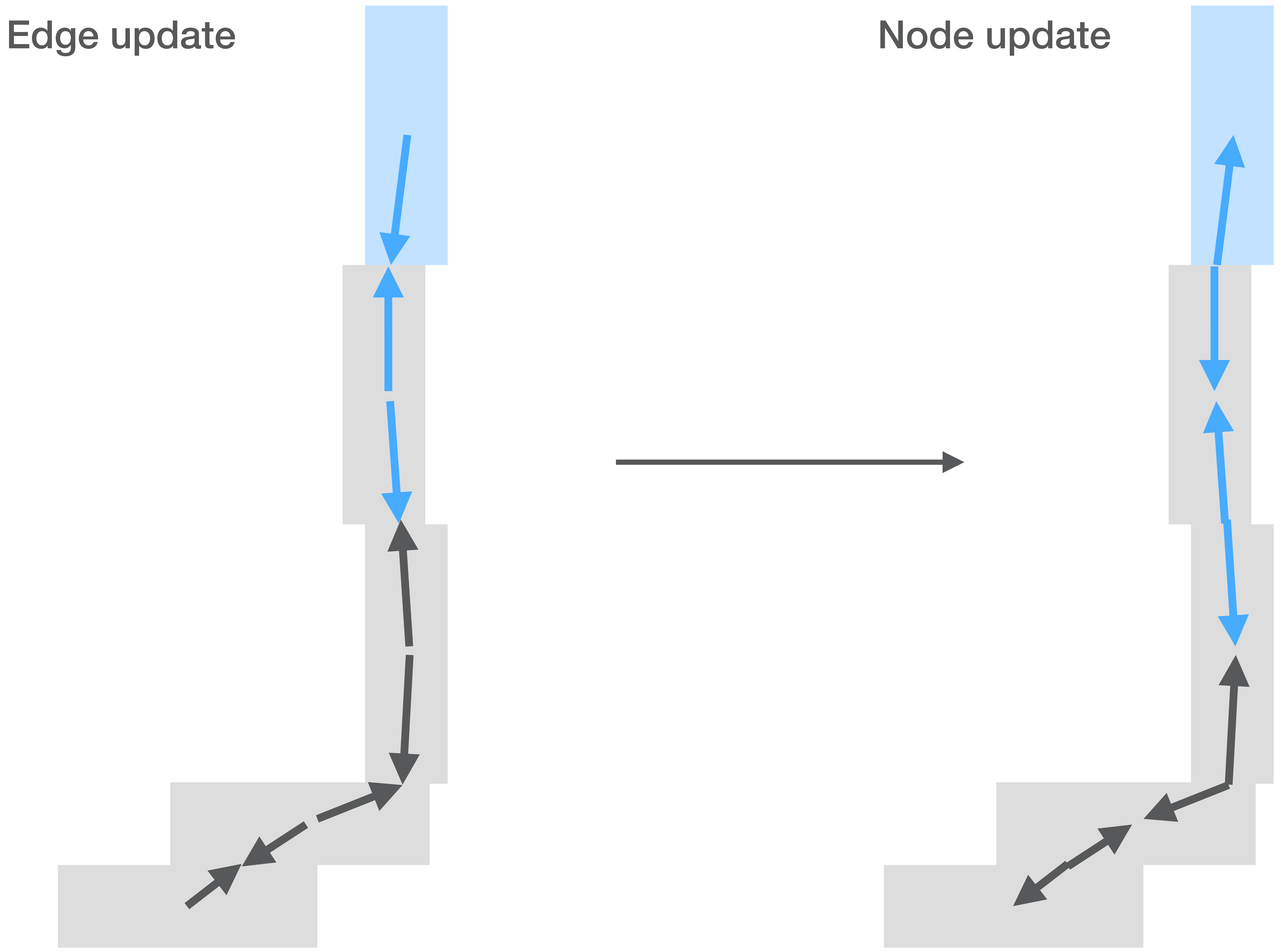
Force Propagation



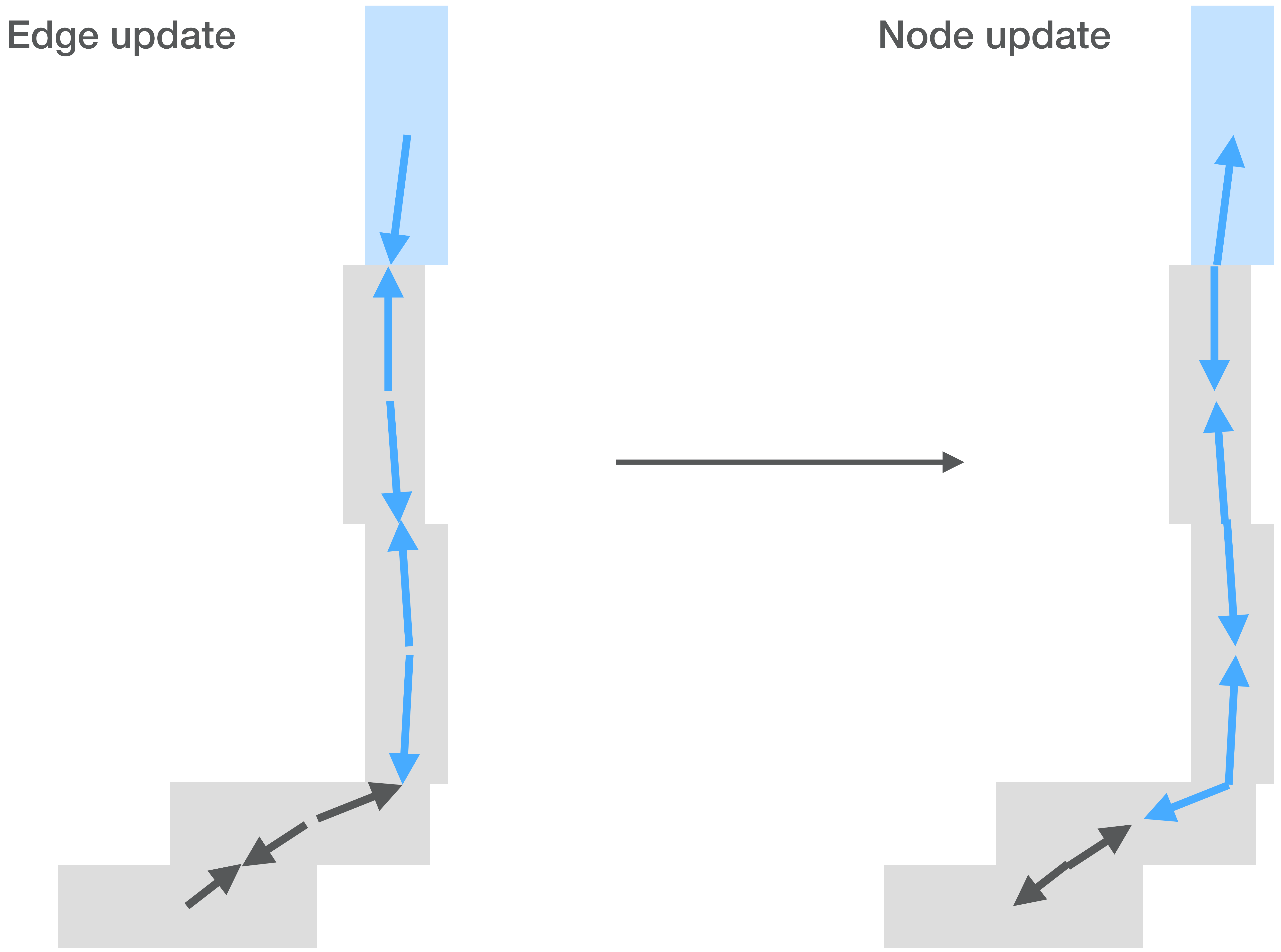
Force Propagation



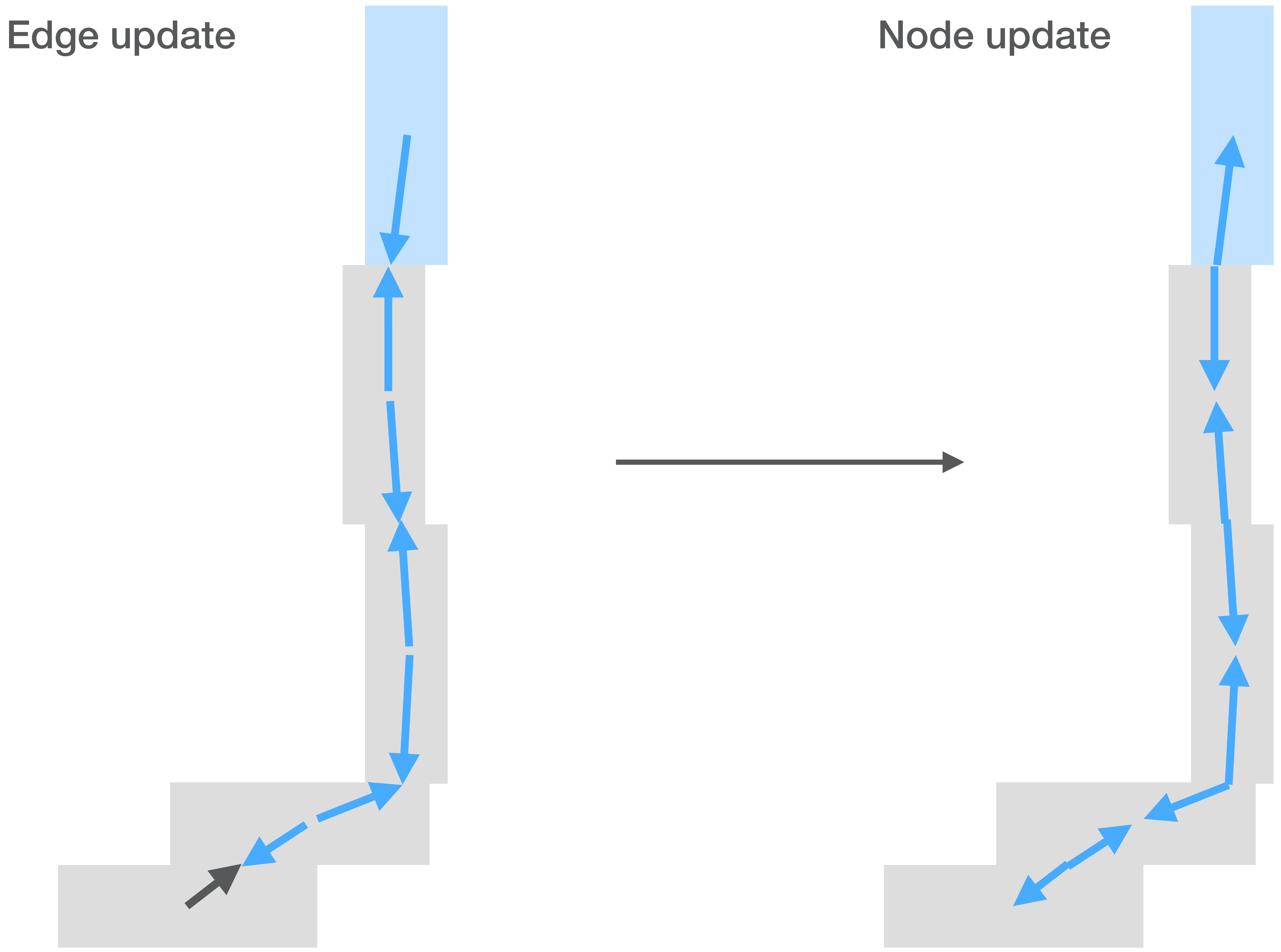
Force Propagation



Force Propagation



Force Propagation



Outline

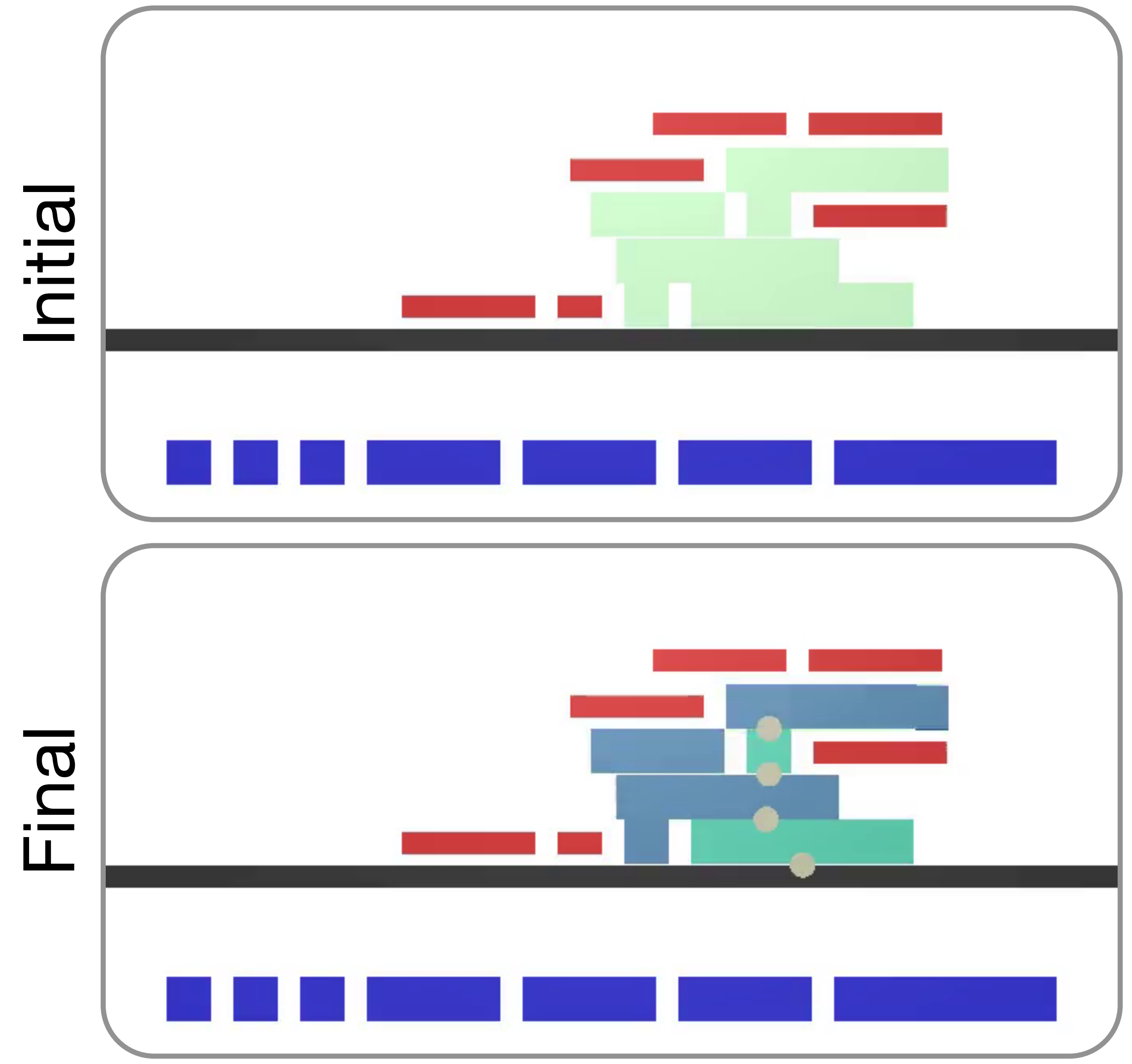
1. Structure and inductive bias in deep learning
2. Graph networks for deep learning on graphs
3. Graph networks for physical inference
4. **Graph networks for physical construction**

Humans are a “Construction Species”

Construction Tasks

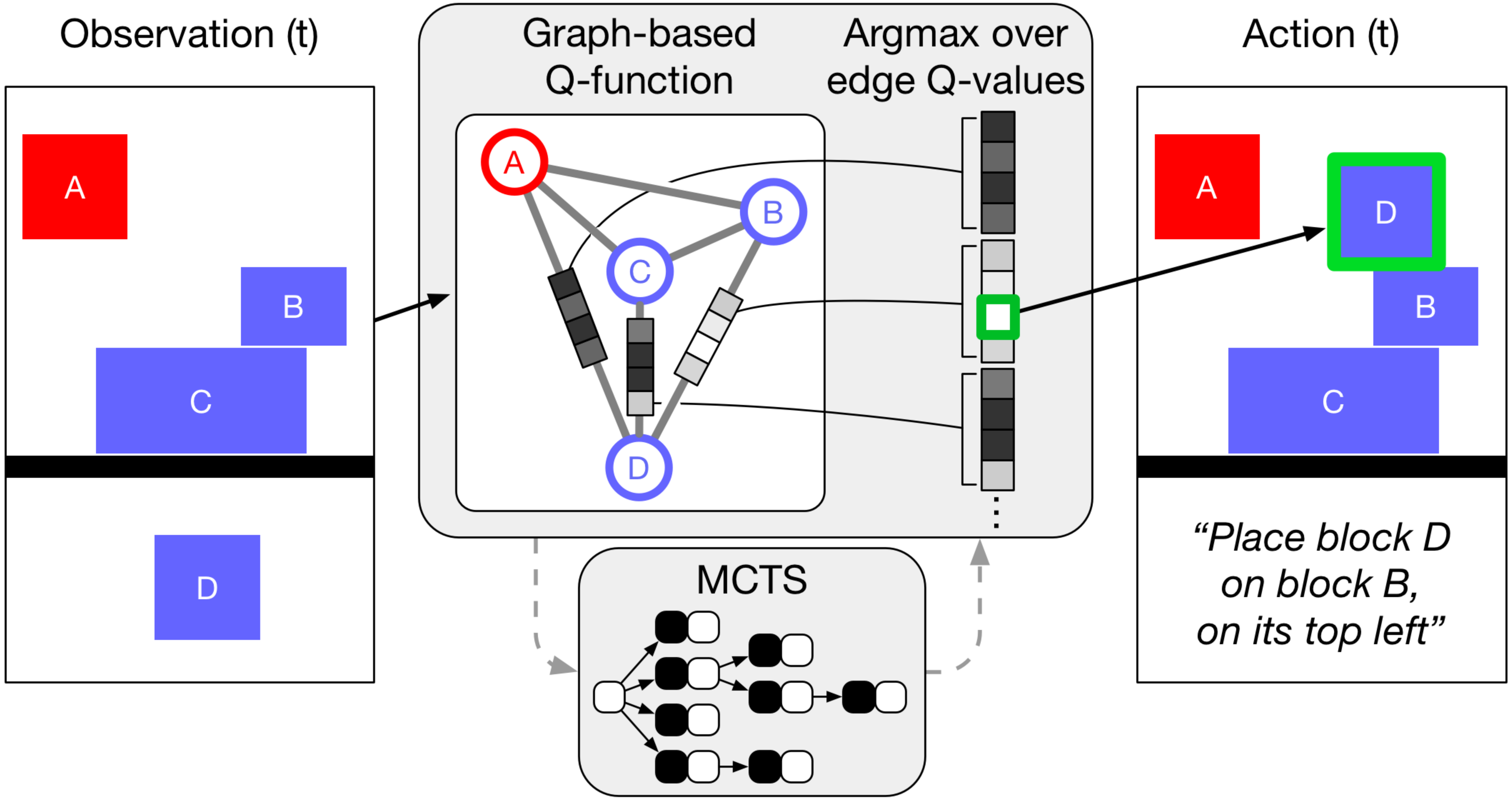
Bapst*, Sanchez-Gonzalez*, Doersch, Stachenfeld, Kohli, Battaglia & Hamrick (*arXiv*, 2019)

(a) Silhouette



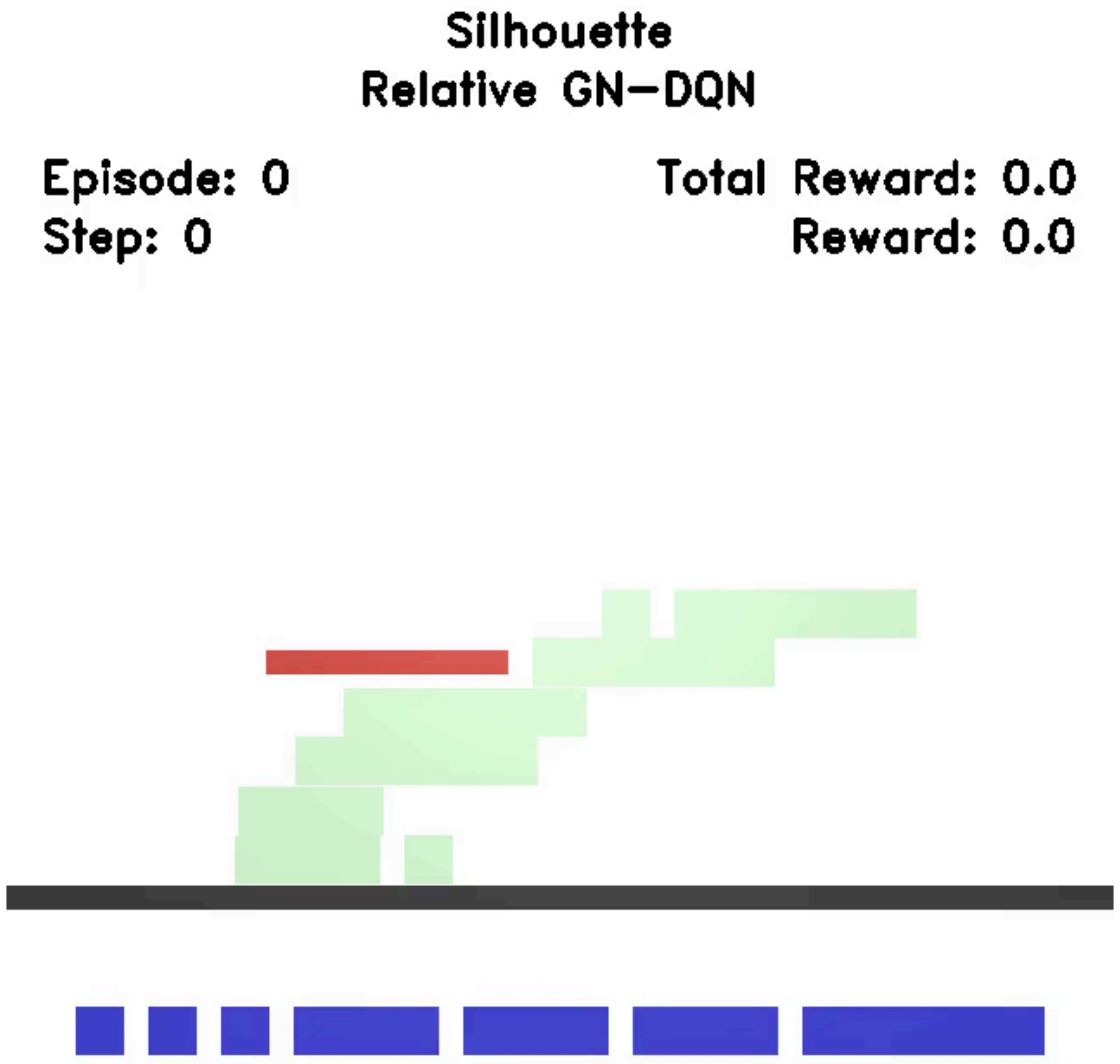
Graph Network Agent (GN-DQN)

Bapst*, Sanchez-Gonzalez*, Doersch, Stachenfeld, Kohli, Battaglia & Hamrick (*arXiv*, 2019)



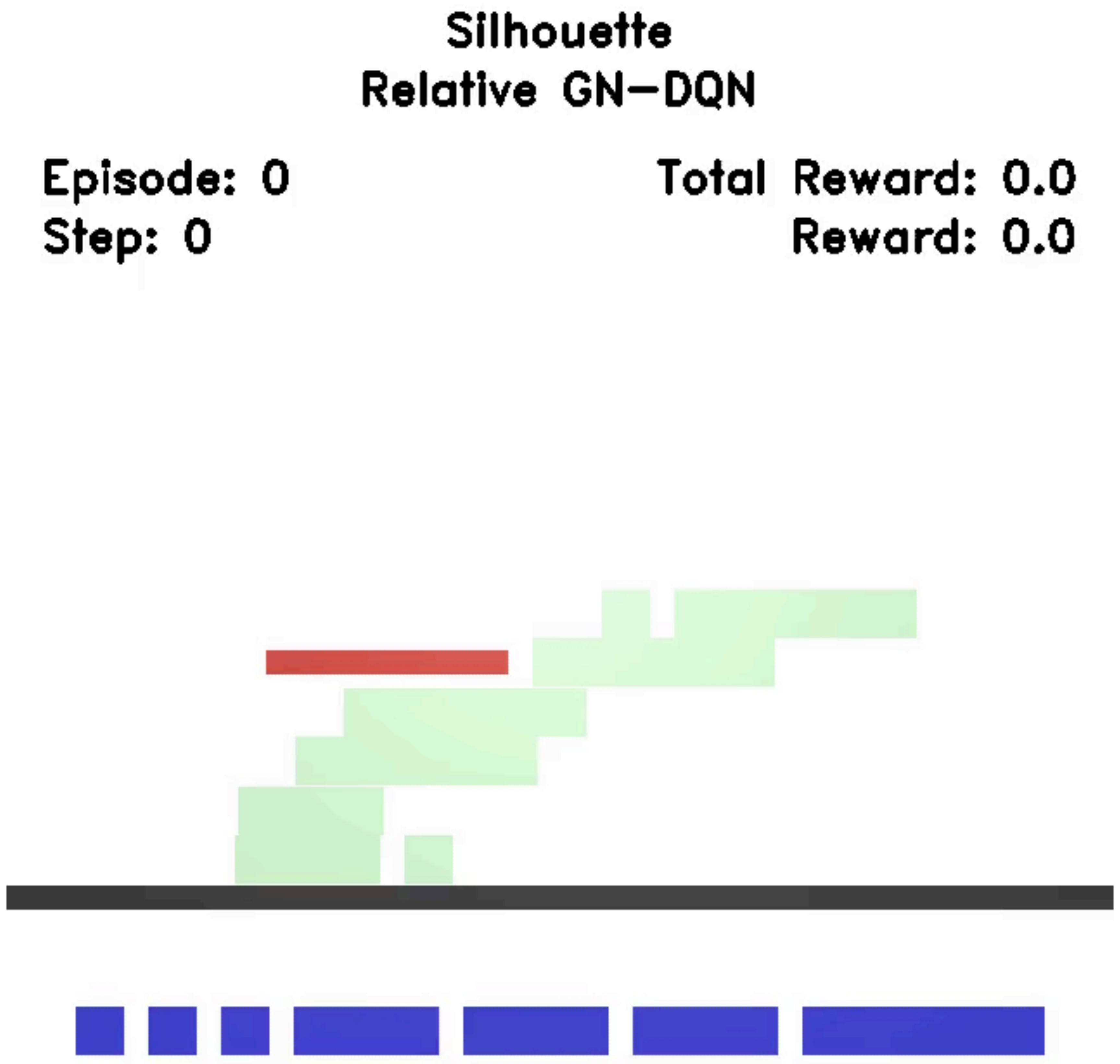
Silhouette

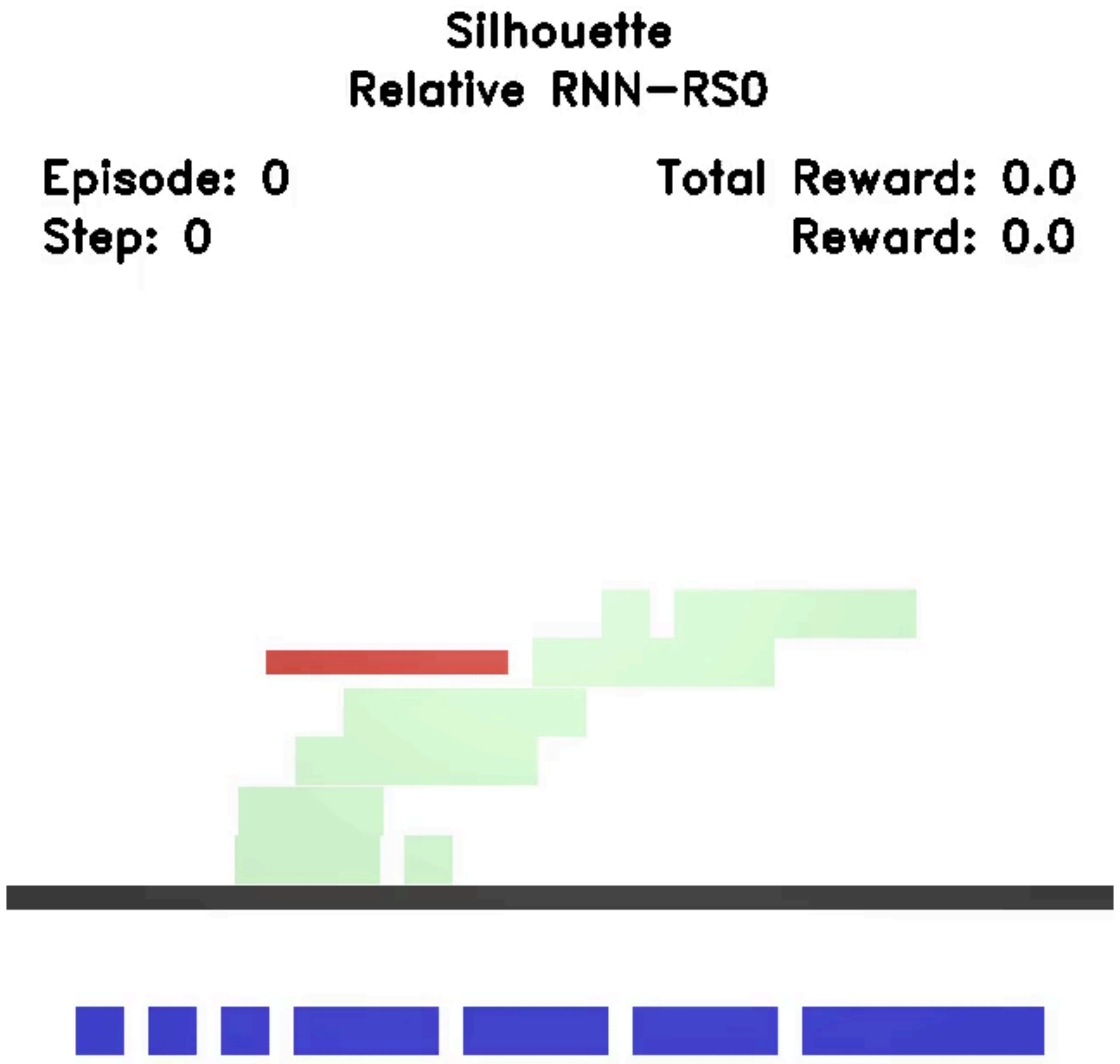
Bapst*, Sanchez-Gonzalez*, Doersch, Stachenfeld, Kohli, Battaglia & Hamrick (*arXiv*, 2019)



Silhouette

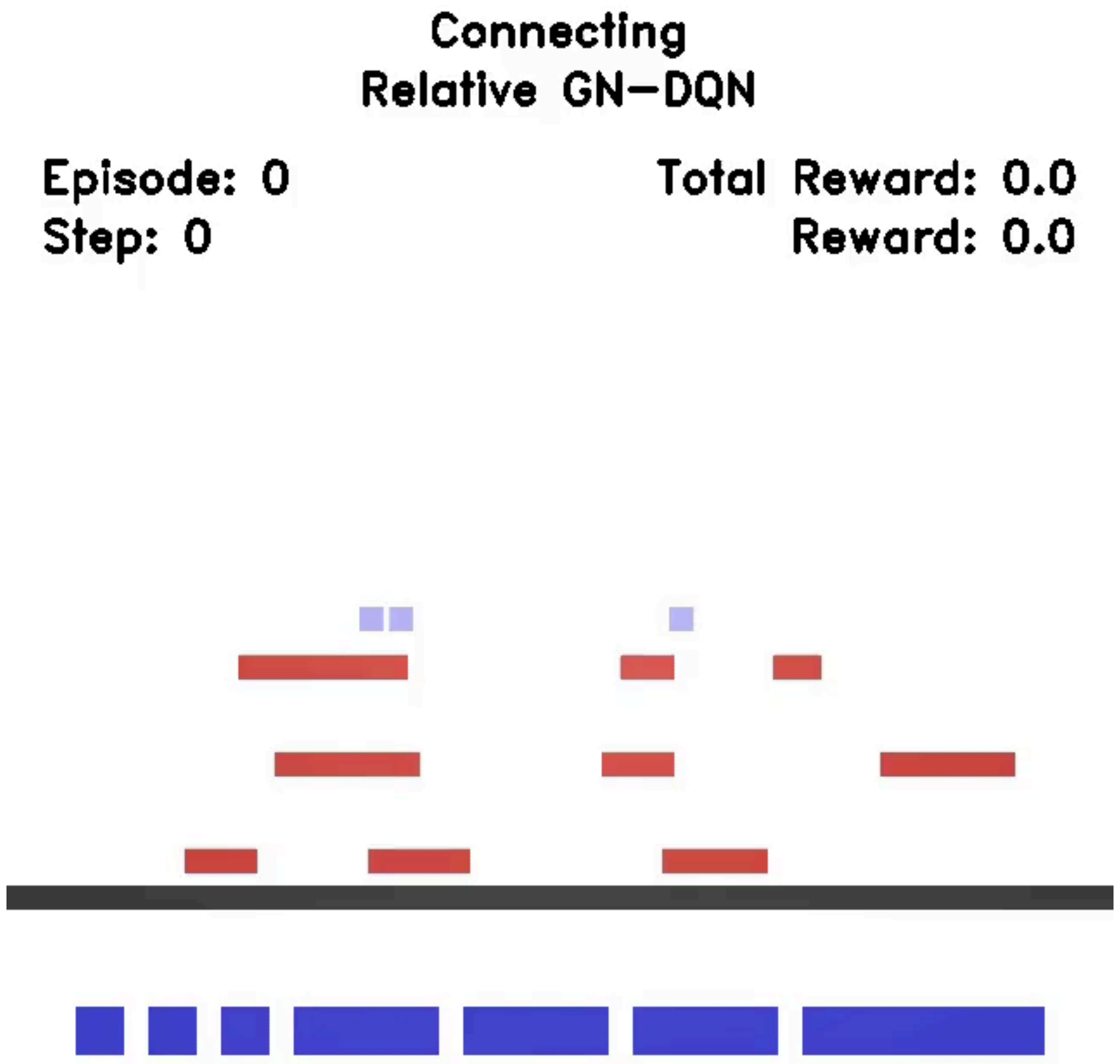
Bapst*, Sanchez-Gonzalez*, Doersch, Stachenfeld, Kohli, Battaglia & Hamrick (*arXiv*, 2019)





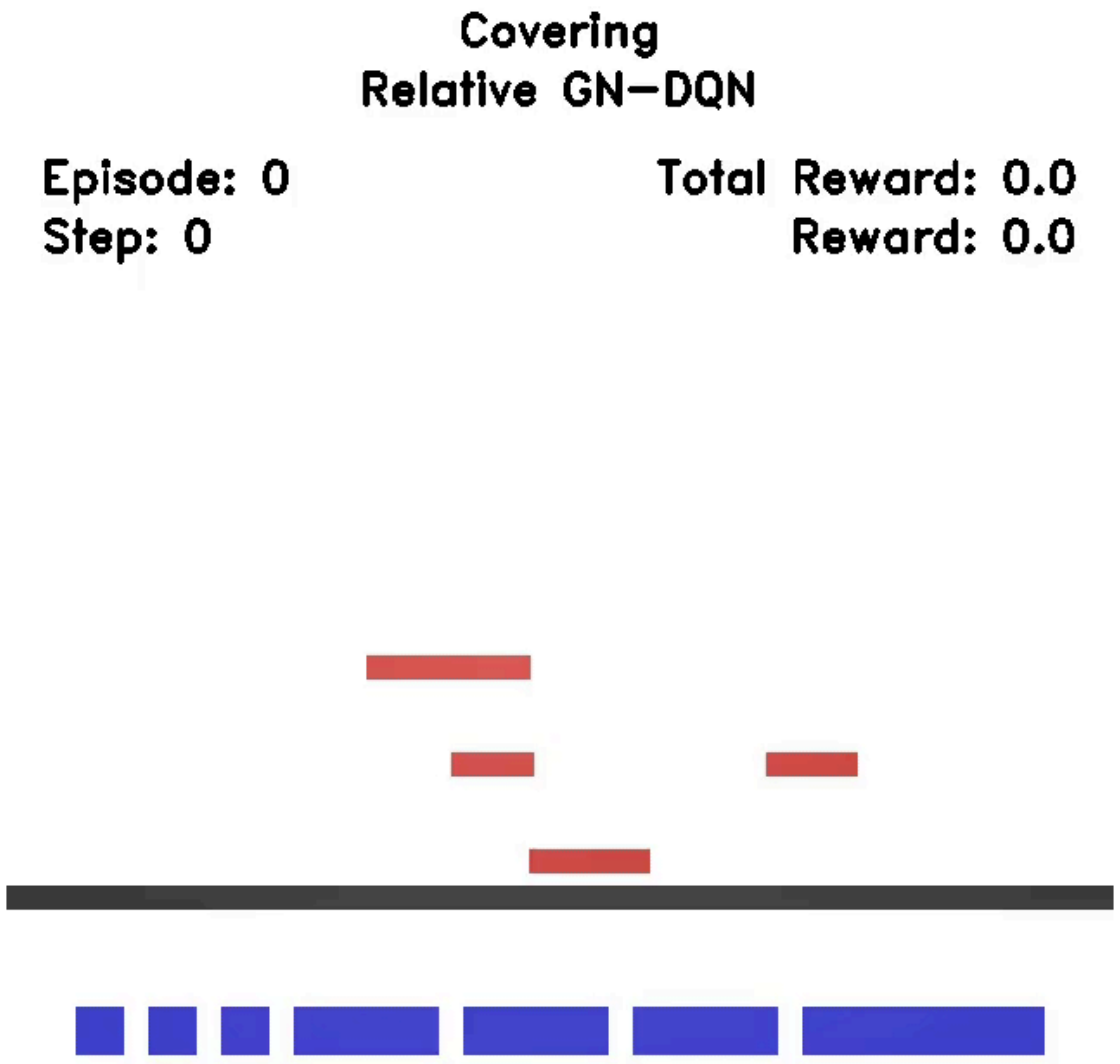
Connecting

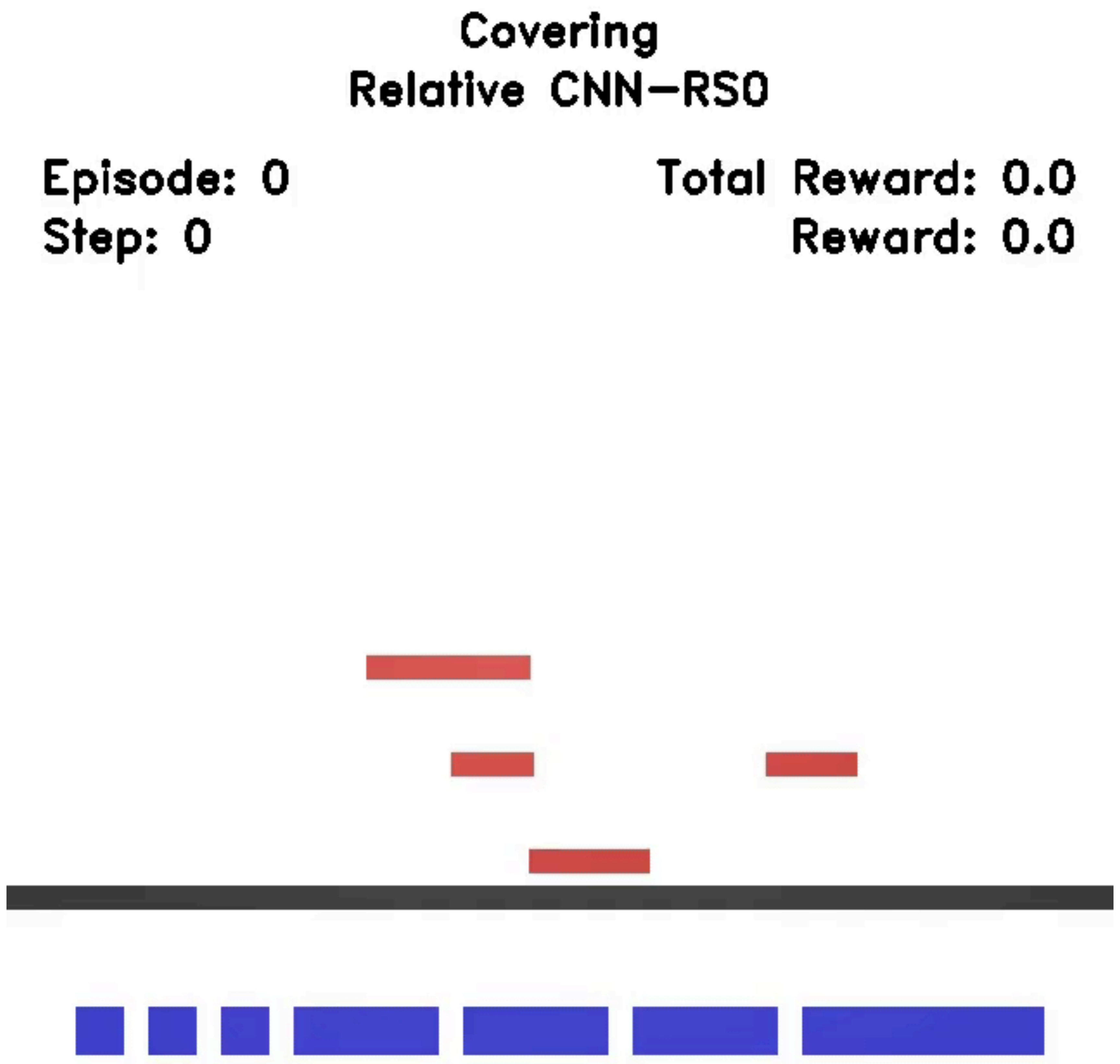
Bapst*, Sanchez-Gonzalez*, Doersch, Stachenfeld, Kohli, Battaglia & Hamrick (*arXiv*, 2019)



Covering

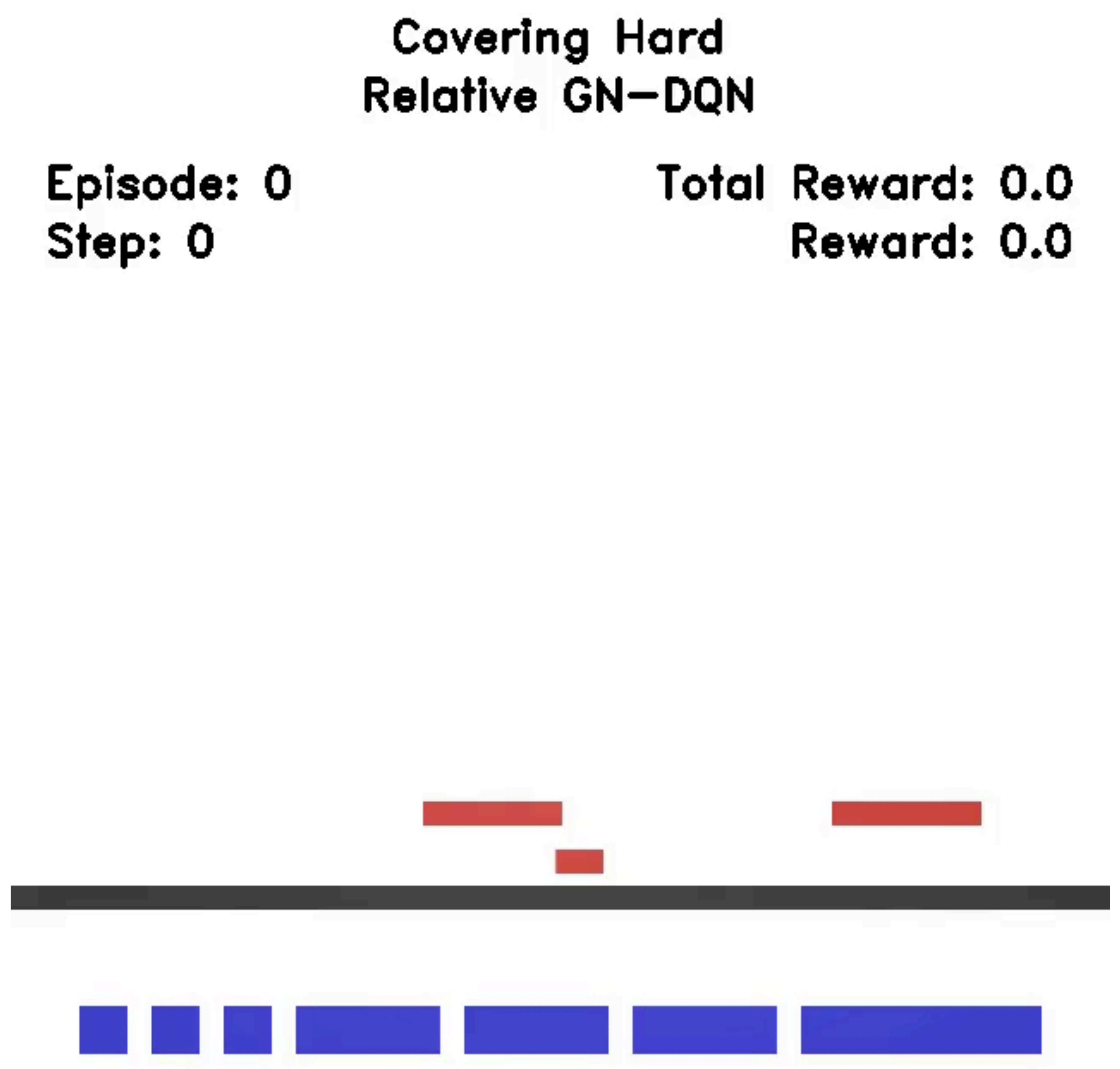
Bapst*, Sanchez-Gonzalez*, Doersch, Stachenfeld, Kohli, Battaglia & Hamrick (*arXiv*, 2019)

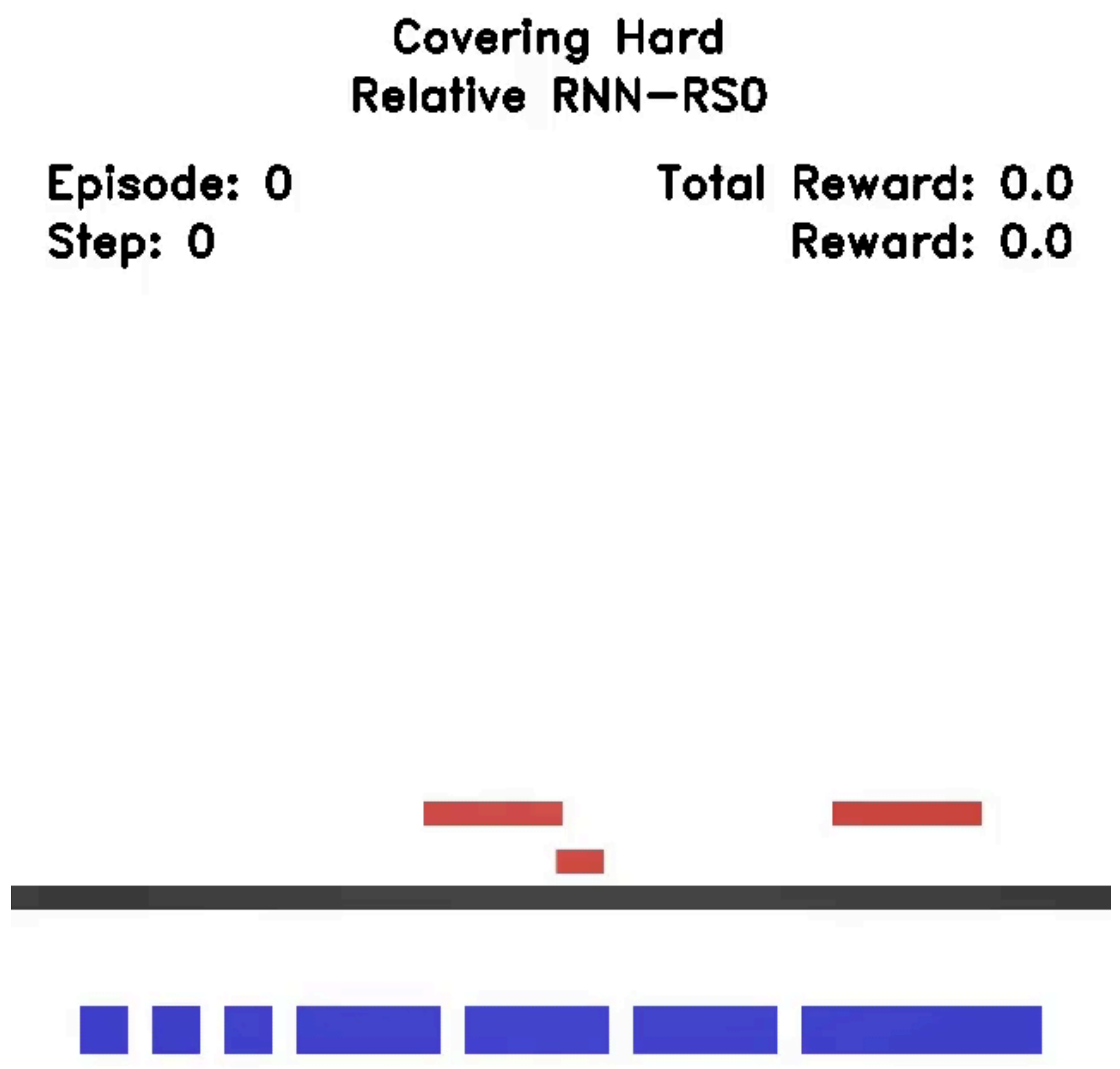




Covering Hard

Bapst*, Sanchez-Gonzalez*, Doersch, Stachenfeld, Kohli, Battaglia & Hamrick (*arXiv*, 2019)





Outline

1. Structure and inductive bias in deep learning
2. Graph networks for deep learning on graphs
3. Graph networks for physical inference
4. Graph networks for physical construction

Structure: the product of composing a known set of ***entities*** and ***relations*** according to a particular set of ***rules***.

What should structure look like in modern AI systems?

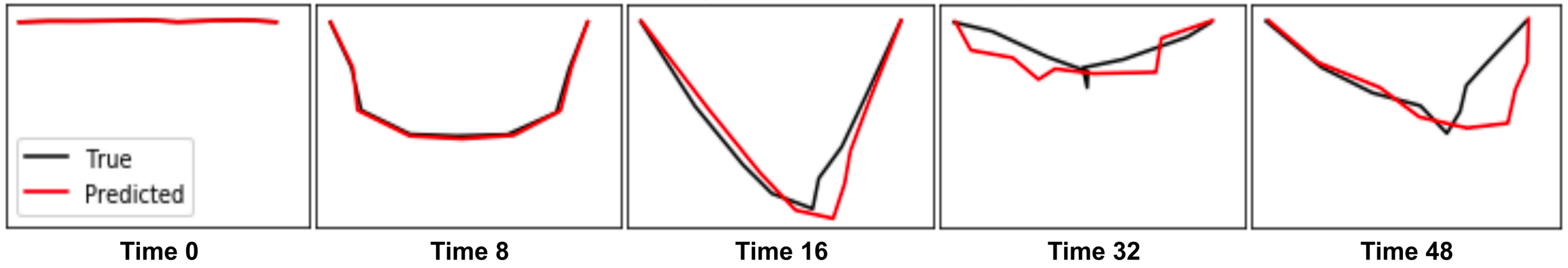
Graph Networks!

Graph Networks in TensorFlow

https://github.com/deepmind/graph_nets

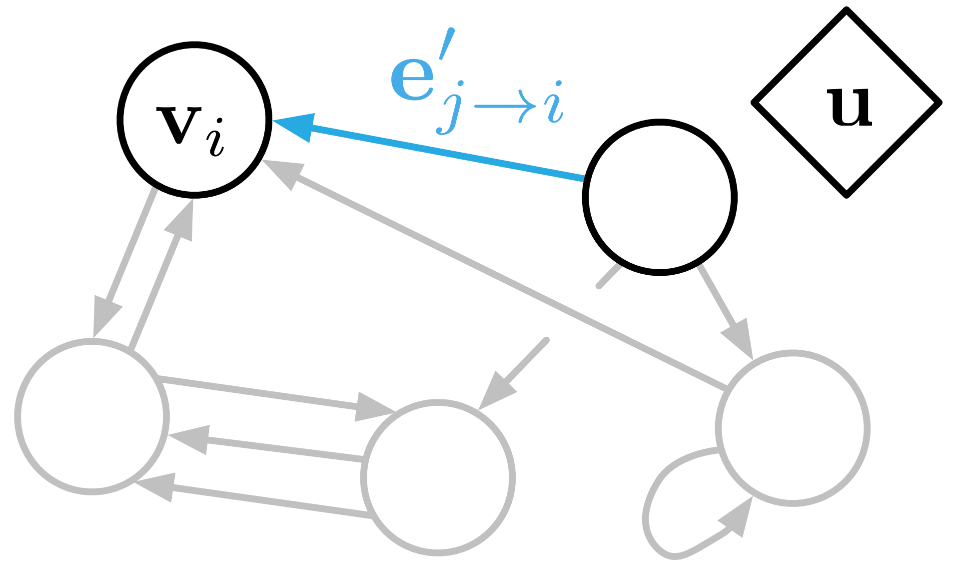
```
import graph_nets as gn
import sonnet as snt

# Provide your own functions to generate graph-structured data.
input_graphs = get_graphs()

# Create the graph network.
graph_net_module = gn.modules.GraphNetwork(
    edge_model_fn=lambda: snt.nets.MLP([32, 32]),
    node_model_fn=lambda: snt.nets.MLP([32, 32]),
    global_model_fn=lambda: snt.nets.MLP([32, 32]))

# Pass the input graphs to the graph network, and return the output graphs.
output_graphs = graph_net_module(input_graphs)
```

Physics: rollout of mass-spring system pinned at ends

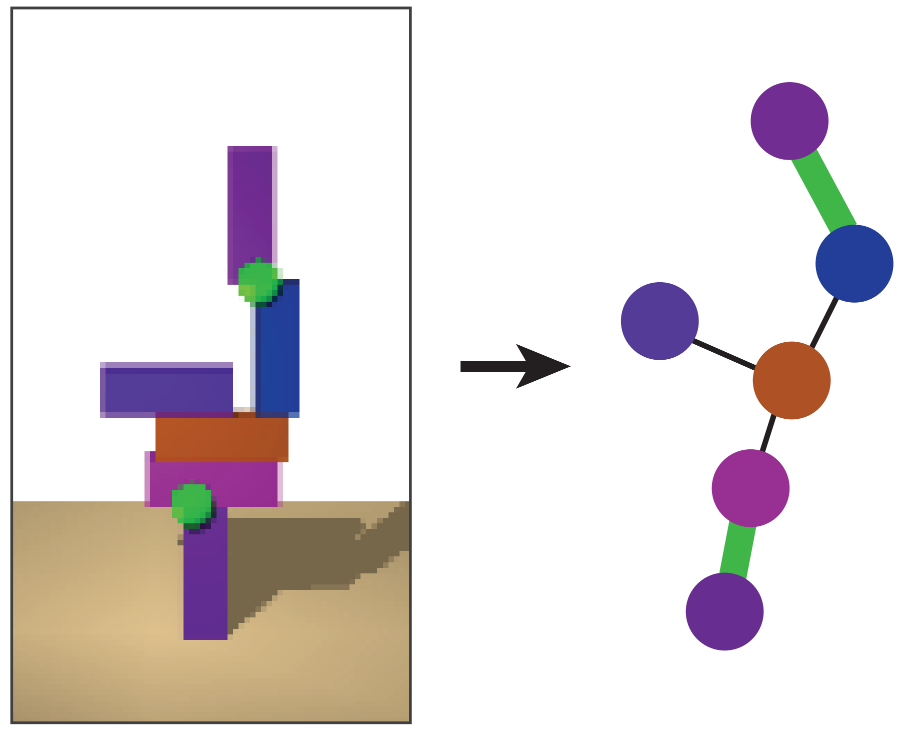


Thanks!

Peter Battaglia
Victor Bapst
Alvaro Sanchez-Gonzalez
Kelsey Allen
Tina Zhu
Kevin McKee
Josh Tenenbaum
Andy Ballard
Razvan Pascanu
Oriol Vinyals
Nicolas Heess
Yujia Li
Carl Doersch
Kim Stachenfeld
Pushmeet Kohli
Tobias Pfaff
DeepMind

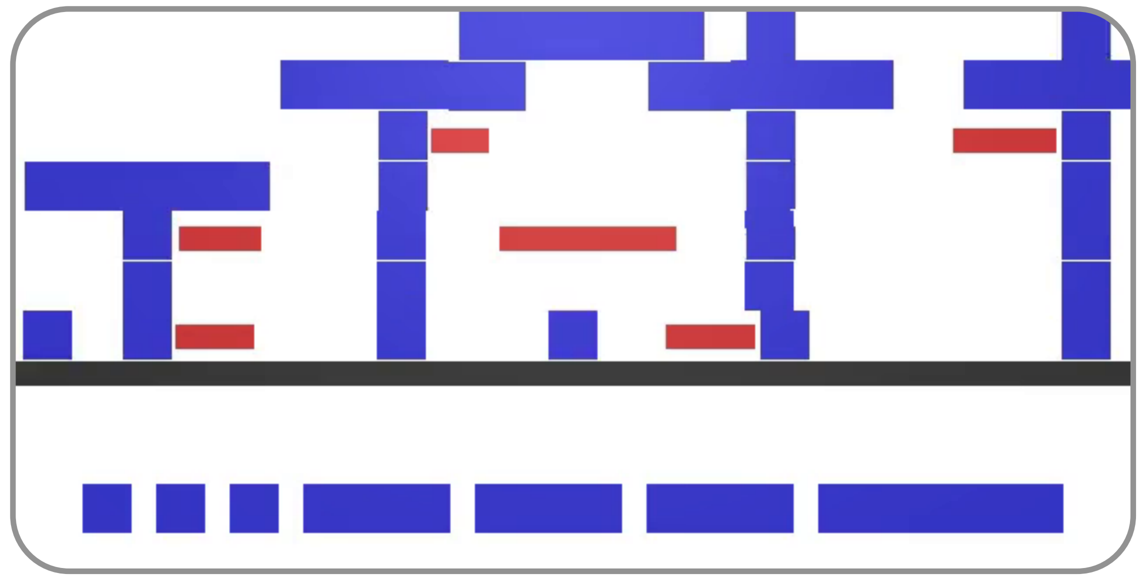


Battaglia, Hamrick, Bapst, Sanchez-Gonzalez, Zambaldi, et al. (2018)



Hamrick*, Allen*, Bapst, Zhu, McKee, Tenenbaum, & Battaglia (2018)

**equal contribution*



Bapst*, Sanchez-Gonzalez*, Doersch, Stachenfeld, Kohli, Battaglia & Hamrick (2019)

**equal contribution*