CS375 / Psych 249:
Large-Scale Neural Network Models for Neuroscience

Lecture 12: Models of Agents
2018.11.28

Daniel Yamins
Stanford Neurosciences Institute
Stanford Artificial Intelligence Laboratory
Departments of Psychology and Computer Science
Stanford University
1. \( A = \text{architecture class} \)
   e.g. \text{convRNNs}

2. \( L = \text{loss function} \)
   \( D = \text{dataset} \)
   e.g. \text{Object Categorization}

3. **Learning Rule**

   \[
   \arg\min_{a \in A} [L(p^*)] \\
   \text{where } p^* \text{ is result of } \text{backprop}
   \]

   \[
   \frac{dp_a}{dt} = -\lambda(t) \cdot \left\langle \nabla_{p_a} L(x) \right\rangle_{x \in D}
   \]
Goal-Driven Modeling

1. ✓ok
   \( A = \text{architecture class} \)
   e.g. convRNNs

2. ✓ok
   \( L = \text{loss function} \)
   \( D = \text{dataset} \)
   e.g. Object Categorization

3. Learning Rule
   \[
   \arg\min_{a \in A} [L(p^*)] \\
   \text{where } p^* \text{ is result of backprop}
   \]
   \[
   \frac{dp_a}{dt} = -\lambda(t) \cdot \langle \nabla_{p_a} L(x) \rangle_{x \in D}
   \]
   ✓ok = basically possible biologically, if not exactly right
1. ✓ ok
\[ A = \text{architecture class} \]
e.g. convRNNs

2. ❌ bad
\[ L = \text{loss function} \quad D = \text{dataset} \]
e.g. Object Categorization

3. Learning Rule
\[
\arg\min_{a \in A} [L(p^*)]
\]
geometric algorithms

where \( p^* \) is result of
\[
\frac{dp_a}{dt} = -\lambda(t) \cdot \langle \nabla_{p_a} L(x) \rangle_{x \in D}
\]

✓ ok = basically possible biologically, if not exactly right
The Problem

2. Imagenet Categorization \( \times \text{bad} \)

\[ L = \text{loss function} \quad \text{D} = \text{dataset} \]

There’s just no way that these creatures receive millions of high-level semantic labels during learning.

Effective proxy, but just obviously deeply wrong.
Self-supervised learning

Olshausen & Field (1996)

$x$

hidden layer $H(x)$

output layer $O(H(x))$

parameters
Self-supervised learning

\[ L(x) = |x - O(H(x))|^2 + \lambda \cdot |H(x)| \]

reconstruction loss
complexity penalty

Olshausen & Field (1996)
Self-supervised learning

Olshausen & Field (1996)

$$L(x) = |x - O(H(x))|^2 + \lambda \cdot |H(x)|$$

- reconstruction loss
- complexity penalty

hidden layer $H(x)$

output layer $O(H(x))$

parameters

(to some extent)
Self-supervised learning

Olshausen & Field (1996)

\[ L(x) = |x - O(H(x))|^2 + \lambda \cdot |H(x)| \]

Unfortunately, autoencoding quite weak in general

⇒

does not generate good features in deep networks
The Problem

2. Imagenet Categorization \( \times \text{bad} \)

\[ L = \text{loss function} \quad D = \text{dataset} \]

Must find some sort of semi-, self-, or unsupervised loss function / task that is “realistically costly” to the creature but is sufficiently powerful that it constructs useful representations.

There’s just no way that these creatures receive millions of high-level semantic labels during learning.

Effective proxy, but just obviously deeply wrong.
Self-supervised learning

Dynamics might give richer signal
Self-supervised learning

Dynamics might give richer signal

\[ L(x) = |x_{t+1} - \text{Decode}(\text{Encode}(x_t))|^2 + \lambda \cdot \text{Penalty}(\text{Encode}(x_t)) \]
Self-supervised learning

Dynamics might give richer signal... but most passive video sequences are quite boring

\[ L(x) = |x_{t+1} - \text{Decode}(\text{Encode}(x_t))|^2 + \lambda \cdot \text{Penalty}(\text{Encode}(x_t)) \]
Children learn through **play**. How does this work?
In contrast, robots — even very advanced robots — are brittle and inflexible.
Infants are curious and play with their environment!

There’s just **no way** that infants receive **millions of high-level semantic labels** during learning.
Self-supervised learning

Give agent some kind of volition to take actions

\[ L(x) = |x_{t+1}^{\text{action}} - \text{Decode(Encode}(x_t))|^2 + \lambda \cdot \text{Penalty(Encode}(x_t)) \]
Self-supervised learning

Give agent some kind of volition to take actions . . . but now the agent will be lazy

\[ L(x) = |x_{t+1}^{\text{action}} - \text{Decode(Encode}(x_t))|^2 + \lambda \cdot \text{Penalty}(\text{Encode}(x_t)) \]
Self-supervised learning

Give agent some kind of volition to take actions . . . but now the agent will be lazy

\[
L(x) = \left| x_{\text{action}}^{t+1} - \text{Decode}(\text{Encode}(x_t)) \right|^2 + \lambda \cdot \text{Penalty}(\text{Encode}(x_t)) + \text{Intrinsic Motivation}
\]
Environment
Environment  

Perception  

Agent
Environment

Perception

Agent
Perception: How does the child perceive its world?

Psych: How does the child perceive its world?
Automated object detection and agent prediction

Perception:

Environment

Perception

Action

Agent

Psych

How does the child perceive its world?

AI/Robotics

Automated object detection and agent prediction
How does the child decide what to — and how — to play with?

**Psych**
How does the child perceive its world?

**Action:**
How does the child decide what to — and how — to play with?

**Environment**

**Perception**

**Action**

**Agent**

**AI/Robotics**
Automated object detection and agent prediction
Optimal action choice policy for interactive goal learning.

Automated object detection and agent prediction.

**Perception:** How does the child perceive its world?

**Action:** How does the child decide what to — and how — to play with?

**Psych:** How does the child decide what to — and how — to play with?

**AI/Robotics:** Automated object detection and agent prediction.

Optimal action choice policy for interactive goal learning.
Environment \rightarrow Perception \rightarrow Action \rightarrow World Model \rightarrow Self Model \rightarrow Agent
Agent learns to predict the environment.
Environment

Perception

Agent

learns to predict
the environment

Action

World Model

learns to predict
own world-model

Self Model
**Curiosity principle:**

The **self-model** directs the agent toward **interesting** actions — the ones that the **world-model** doesn’t yet fully understand.
Curiosity principle:

The **self-model** directs the agent toward **interesting** actions — the ones that the **world-model** doesn’t yet fully understand.

![Diagram showing the interaction between the environment, perception, action, curiosity, self-model, world-model, and an agent.](image)
**Curiosity principle:**

The **self-model** directs the agent toward **interesting** actions — the ones that the **world-model** doesn’t yet fully understand.
**Curiosity principle:**

The *self-model* directs the agent toward *interesting* actions — the ones that the *world-model* doesn’t yet fully understand.
Learning to Play With Intrinsically-Motivated, Self-Aware Agents

Nick Haber¹,²,³,*, Damian Mrowca⁴,⁵; Stephanie Wang⁴; Li Fei-Fei⁴; and Daniel L. K. Yamins¹,²,⁶

Departments of Psychology¹, Pediatrics², Biomedical Data Science³, Computer Science³, and Wu Tsai Neurosciences Institute⁵, Stanford, CA 94305
{nhaber, mrowca}@stanford.edu

Abstract

Infants are experts at playing, with an amazing ability to generate novel structured behaviors in unstructured environments that lack clear extrinsic reward signals. We seek to mathematically formalize these abilities using a neural network that implements curiosity-driven intrinsic motivation. Using a simple but ecologically naturalistic simulated environment in which an agent can move and interact with objects it sees, we propose a “world-model” network that learns to predict the dynamic consequences of the agent’s actions. Simultaneously, we train a separate explicit “self-model” that allows the agent to track the error map of its world-model. It then uses the self-model to adversarially challenge the developing world-model. We demonstrate that this policy causes the agent to explore novel and informative interactions with its environment, leading to the generation of a spectrum of complex behaviors, including ego-motion prediction, object attention, and object gathering. Moreover, the world-model that the agent learns supports improved performance on object dynamics prediction, detection, localization and recognition tasks. Taken together, our results are initial steps toward creating flexible autonomous agents that self-supervise in realistic physical environments.
Agent ("baby") can (a) swivel its head
(b) move around the room
(c) apply forces to objects

... and receives back images of what happened, given action
Model has two pieces: (I) World-Model
Goal of **world-model** network is to predict consequences of actions.

Goal of **self-model** network is to predict errors of world-model ("self-aware").
Learning to Play - Model overview

Goal of **world-model** network is to predict consequences of actions

Goal of **self-model** network is to predict errors of world-model ("self-aware")

Action choice: self-model is **adversarial** to world-model ("curious intrinsic motivation")
Model has two pieces:  

(1) **World-Model**  

(2) **Self-Model**
Learning to Play - Model overview

Goal of **world-model** network is to predict consequences of actions

Goal of **self-model** network is to predict errors of world-model ("self-aware")
Goal of **world-model** network is to predict consequences of actions

Goal of **self-model** network is to predict errors of world-model ("self-aware")

Action choice: self-model is **adversarial** to world-model ("curious intrinsic motivation")
Goal of **world-model**: “Post-dict” the action taken given past and future states and actions
Learning to Play - World Model

Post-dict the action taken given past and future states and actions
Learning to Play - World Model

Post-dict the action taken given past and future states and actions

ID
Inverse Dynamics
Learning to Play - World Model

Predict (latent) future states given past states and actions

\[ o^{t+2} \]

\[ \alpha_{t+1}^{\text{ego}} \quad \alpha_{t+1}^{\text{obj}_1} \quad \alpha_{t+1}^{\text{obj}_2} \]

LF

Latent Future Prediction
Goal of **self-model**: Predict errors ("loss") of World-Model
Learning to Play - Self Model

Predict errors ("loss") of World-Model
Learning to Play - Self Model

Predict errors ("loss") of World-Model
Learning to Play - Self Model

Predict errors ("loss") of World-Model
Learning to Play - Self Model

Predict errors ("loss") of World-Model

Sample 1000x actions and choose the one that maximizes the World-Model loss

$$\pi(a) \sim \exp(\beta \sigma_\Lambda(a))$$

Policy mechanism
Learning to Play — Adversarial Policy

Action choice: **self-model** is *adversarial* to world-model (“curious intrinsic motivation”)

Goal of **world-model** network is to predict consequences of actions

Goal of **self-model** network is to predict errors of world-model (“self-aware”)
Self-supervised learning

Training Loss

Ego motion learning

Testing Loss

Ego-motion only (easy)

curious policy
random policy
Self-supervised learning

- **Training Loss**
  - Ego-motion learning
  - Emergence of object attention
  - Object interaction learning

- **Testing Loss**
  - Ego-motion only (easy)

- Curious policy vs. Random policy

- Training Steps from 0 to 420

- Loss values range from 0.5 to 0.1
Self-supervised learning

Emergence of:
(a) ego-motion understanding
(b) object attention
(c) improved world-model
Self-supervised learning

Simple navigation and planning behavior emerges …
Self-supervised learning

Simple navigation and planning behavior emerges …

Moreover, substantially improved transfer learning accuracy:

(a) object detection (present or not): \(~8\% \text{ vs } \sim 40\%\) accuracy

(b) object position: \(~6\text{px} \text{ vs } \sim 4\text{px}\) error

(c) object recognition (among 16 geometries): \(~12\% \text{ vs } \sim 30\%\) accuracy
Self-supervised learning

- curious policy
- random policy

Training Loss

One Object Frequency

Fraction of Frames
Self-supervised learning

- **Training Loss**
  - Curious policy (blue)
  - Random policy (green)

- **One Object Frequency**
- **Two Object Frequency**

- 2 object loss
- 1 object loss
Self-supervised learning

Object recognition in testing (one object per image): ~16% vs ~40% accuracy

... especially large gain compared to training in 1-obj case
Curiosity-driven Exploration by Self-supervised Prediction

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, Trevor Darrell

(Submitted on 15 May 2017)

In many real-world scenarios, rewards extrinsic to the agent are extremely sparse, or absent altogether. In such cases, curiosity can serve as an intrinsic reward signal to enable the agent to explore its environment and learn skills that might be useful later in its life. We formulate curiosity as the error in an agent's ability to predict the consequence of its own actions in a visual feature space learned by a self-supervised inverse dynamics model. Our formulation scales to high-dimensional continuous state spaces like images, bypasses the difficulties of directly predicting pixels, and, critically, ignores the aspects of the environment that cannot affect the agent. The proposed approach is evaluated in two environments: VizDoom and Super Mario Bros. Three broad settings are investigated: 1) sparse extrinsic reward, where curiosity allows for far fewer interactions with the environment to reach the goal; 2) exploration with no extrinsic reward, where curiosity pushes the agent to explore more efficiently; and 3) generalization to unseen scenarios (e.g. new levels of the same game) where the knowledge gained from earlier experience helps the agent explore new places much faster than starting from scratch. Demo video and code available at this URL

![Image](https://example.com/screenshot.jpg)

(a) learn to explore in Level-1  (b) explore faster in Level-2

Figure 1. Discovering how to play Super Mario Bros without rewards. (a) Using only curiosity-driven exploration, the agent makes significant progress in Level-1. (b) The gained knowledge helps the agent explore subsequent levels much faster than when starting from scratch. Watch the video at http://pathak22.github.io/noreward-rl/
Exploration by Random Network Distillation

Yuri Burda, Harrison Edwards, Amos Storkey, Oleg Klimov

(Submitted on 30 Oct 2018)

We introduce an exploration bonus for deep reinforcement learning methods that is easy to implement and adds minimal overhead to the computation performed. The bonus is the error of a neural network predicting features of the observations given by a fixed randomly initialized neural network. We also introduce a method to flexibly combine intrinsic and extrinsic rewards. We find that the random network distillation (RND) bonus combined with this increased flexibility enables significant progress on several hard exploration Atari games. In particular we establish state of the art performance on Montezuma's Revenge, a game famously difficult for deep reinforcement learning methods. To the best of our knowledge, this is the first method that achieves better than average human performance on this game without using demonstrations or having access to the underlying state of the game, and occasionally completes the first level.

Progress in Montezuma's Revenge

![Graph showing progress in Montezuma's Revenge](image-url)
Self-supervised learning

Next steps:

(a) improve realism environment and agent embodiment
Self-supervised learning

**ThreeDWorld**: Simulation environment visual, auditory, interaction learning
Self-supervised learning

Next steps:

(a) improve realism environment and agent embodiment
(b) seek more interesting behavior
Self-supervised learning

Next steps:

(a) improve realism environment and agent embodiment
(b) seek more interesting behavior

goal: “stacking”, peekaboo, building simple machines
Self-supervised learning

Next steps:

(a) improve realism environment and agent embodiment
(b) seek more interesting behavior

goal: “stacking”, peekaboo, building simple machines

(c) solve core technical issue with degeneracy
Self-supervised learning

Next steps:

(a) improve realism environment and agent embodiment
(b) seek more interesting behavior
   goal: “stacking”, peekaboo, building simple machines
(c) solve core technical issue with degeneracy
   better intrinsic motivation formulations, e.g.
   predicted gain-of-knowledge
Next steps:

(a) improve realism environment and agent embodiment
(b) seek more interesting behavior

goal: “stacking”, peekaboo, building simple machines

(c) solve core technical issue with **DEGENERACY**

better intrinsic motivation formulations, e.g
predicted gain-of-knowledge

(d) add *other* agents — incorporating theory of mind
Glossing over a key problem:

The above ideas rely on having the agent solve a dynamics prediction problem about the world.
Start with some data $H$...
Self-supervised learning: General Formulation

...Create input data $X$ from $H$...
Self-supervised learning: General Formulation

…Create output data Y from H…
...Predict $Y$ from $X$.
Self-supervised learning: General Formulation

Examples:

1) Forward future prediction

\[ \text{state}(t), \text{action}(t) \rightarrow \text{state}(t+1) \]
Self-supervised learning: General Formulation

Examples:

1) Forward future prediction

\[ \text{state}(t), \text{action}(t) \rightarrow \text{state}(t+1) \]

2) Inverse dynamics prediction

\[ \text{state}(t), \text{state}(t+1) \rightarrow \text{action}(t) \]
Self-supervised learning: General Formulation

Examples:
1) Forward future prediction
   \[ \text{state}(t), \text{action}(t) \implies \text{state}(t+1) \]

2) Inverse dynamics prediction
   \[ \text{state}(t), \text{state}(t+1) \implies \text{action}(t) \]

1) is hard, because … pixel prediction is hard!
Intuitive Physics as Underlying Goal

Obvious idea: just predict future pixels

Finn et. al (2016)

PredRNN(2017) ; Wang (2018) ; among many others
Intuitive Physics as Underlying Goal

Pixel prediction is hard.

Two blue objects in a room
Intuitive Physics as Underlying Goal

Pixel prediction is hard.

Two blue objects in a room

Objects acted on and camera moves
Intuitive Physics as Underlying Goal

Pixel prediction is hard.

Two blue objects in a room

Objects acted on and camera moves

Ground truth
Intuitive Physics as Underlying Goal

Pixel prediction is hard.

Two blue objects in a room

Objects acted on and camera moves

Wang et. al (2018)
Examples:

1) **Forward future prediction**
   
   \[
   \text{state}(t), \text{action}(t) \rightarrow \text{state}(t+1)
   \]

2) **Inverse dynamics prediction**
   
   \[
   \text{state}(t), \text{state}(t+1) \rightarrow \text{action}(t)
   \]

1) is hard, because … pixel prediction is hard!

2) is mostly what we did in the work described above because it’s easier …
Self-supervised learning: General Formulation

Examples:

1) Forward future prediction

\[ \text{state}(t), \text{action}(t) \rightarrow \text{state}(t+1) \]

2) Inverse dynamics prediction

\[ \text{state}(t), \text{state}(t+1) \rightarrow \text{action}(t) \]

1) Is hard, because … pixel prediction is hard!

2) Is mostly what we did in the work described above because it’s easier …

But degenerate!
Self-supervised learning: General Formulation

Examples:

1) Forward future prediction

\[ \text{state}(t), \text{action}(t) \rightarrow \text{state}(t+1) \]

2) Inverse dynamics prediction

\[ \text{state}(t), \text{state}(t+1) \rightarrow \text{action}(t) \]

1) is hard, because … pixel prediction is hard!

2) is mostly what we did in the work described above because it’s easier …

BUT DEGENERATE!
Self-supervised learning: General Formulation

Examples:

1) Forward future prediction

\[ \text{state}(t), \text{action}(t) \rightarrow \text{state}(t+1) \]

2) Inverse dynamics prediction

\[ \text{state}(t), \text{state}(t+1) \rightarrow \text{action}(t) \]

1) is hard, because … pixel prediction is hard!

2) is mostly what we did in the previous work because it’s easier …

BUT DEGENERATE!

Ex: pushing down on an object
Examples:
1) Forward future prediction

\[ \text{state}(t), \text{action}(t) \rightarrow \text{state}(t+1) \]

**Conclusion:**
we cannot escape having to do better future prediction — so let’s attack the problem directly.

1) is hard, because … pixel prediction is hard!

2) is mostly what we did in the previous work because it’s easier …

**BUT DEGENERATE!**

Ex: pushing down on an object
Flexible Neural Representation for Physics Prediction

Damian Mrowca\textsuperscript{1,*}, Chengxu Zhuang\textsuperscript{2,*}, Elias Wang\textsuperscript{3,*}, Nick Haber\textsuperscript{2,4,5}, Li Fei-Fei\textsuperscript{1}, Joshua B. Tenenbaum\textsuperscript{7,8}, and Daniel L. K. Yamins\textsuperscript{1,2,6}

\textsuperscript{1}Department of Computer Science, \textsuperscript{2}Psychology, \textsuperscript{3}Electrical Engineering, \textsuperscript{4}Pediatrics and \textsuperscript{5}Biomedical Data Science, and Wu Tsai Neurosciences Institute, Stanford, CA 94305
\textsuperscript{6}Department of Brain and Cognitive Sciences, and Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, MA 02139
\{mrowca, chengxuz, eliwang\}@stanford.edu

Abstract

Humans have a remarkable capacity to understand the physical dynamics of objects in their environment, flexibly capturing complex structures and interactions at multiple levels of detail. Inspired by this ability, we propose a hierarchical particle-based object representation that covers a wide variety of types of three-dimensional objects, including both arbitrary rigid geometrical shapes and deformable materials. We then describe the Hierarchical Relation Network (HRN), an end-to-end differentiable neural network based on hierarchical graph convolution, that learns to predict physical dynamics in this representation. Compared to other neural network baselines, the HRN accurately handles complex collisions and nonrigid deformations, generating plausible dynamics predictions at long time scales in novel settings, and scaling to large scene configurations. These results demonstrate an architecture with the potential to form the basis of next-generation physics predictors for use in computer vision, robotics, and quantitative cognitive science.
Choosing a good intermediate...

“Encoding”

“Physics”

“Rendering”
Intuitive Physics as Underlying Goal

Experimental results with infants: **object permanence** present very early, perhaps by 3 months.

Liz Spelke
Experimental results with infants: **object permanence** present very early, perhaps by 3 months.
In a classic series of tests of object permanence, Renée Baillargeon and her colleagues first habituated young infants to the sight of a screen rotating through 180 degrees. Then a box was placed in the path of the screen. In the possible event, the screen rotated up, occluding the box, and stopped when it reached the top of the box. In the impossible event, the screen rotated up, occluding the box, but then continued on through 180 degrees, appearing to pass through the space where the box was. Infants looked longer at the impossible event, showing they mentally represented the presence of the invisible box. (From Baillargeon, 1987)
Intuitive Physics as Underlying Goal

Experimental results with infants: **object permanence** present very early, perhaps by 3 months.

Conv2d structures, even with RNNs, have severe trouble with object permanence.
Conv3d structures are better for object permanence, but very inefficient: hard to achieve high resolution.

Experimental results with infants: **object permanence** present very early, perhaps by 3 months.

Liz Spelke
Spatial convolutions are not ideal for physics propagation

"Derendering"  "Physics"  "Rendering"

$I_{t-n} \ldots t$  $P_{3D-INV}$  $P_{2D-INV}$

$S_t$  $F_{2D}$  $S_{t+1}$

$F_{3D}$  $P_{3D}$

$S_t$  $S_{t+1}$

Object Permanency Problem

Back completion problem
Intuitive Physics as Underlying Goal

Experimental results with infants: **object permanence** present very early, perhaps by 3 months.

Alternative to spatially-uniform priors are **graph-based** priors

Liz Spelke

Relational Networks (Battaglia et. al., 2016)
Intuitive Physics as Underlying Goal

Experimental results with infants: **object permanence** present very early, perhaps by 3 months.

Alternative to spatially-uniform priors are **graph-based** priors

… still local and convolutional, just on the graph.

Liz Spelke

Relational Networks (Battaglia et. al., 2016)
Relational Networks
(Battaglia et. al., 2016)

Neural Physics Engine
(Chang et. al., 2016)
Intuitive Physics as Underlying Goal

Complex Scenes
Deformable Objects
Describe objects through complex graphs:

Cube
Cuboid
Pyramid
Flat Pyramid
Octahedron
Prism
Cylinder
Ellipsoid
Sphere
Mentos
Stick
Bowl
Cone
Pentagon
Domino
Torus
Duck
Bunny
Teddy
In fact, describe whole scenes.
In fact, describe whole scenes.

$$G = \langle N, E \rangle$$  
scene graph
In fact, describe whole scenes.

\[ G = \langle N, E \rangle \] scene graph

\( N = \) nodes corresponding to particles comprising objects

\( E = \) edges corresponding to relationships between particles
In fact, describe whole scenes.

\[ G = \langle N, E \rangle \]  

**scene graph**

\( N = \) nodes corresponding to particles comprising objects

\( E = \) edges corresponding to relationships between particles

can have different strengths
Intuitive Physics as Underlying Goal

Of course, humans don’t think about all the particles at once all the time.
Intuitive Physics as Underlying Goal

Of course, humans don’t think about all the particles at once all the time.

\[ G \longrightarrow G_H \]

\[ G_H = \text{dynamic “hierarchicization” of underlying scene graph} \]

(right now computed via k-means)
Intuitive Physics as Underlying Goal

Of course, humans don’t think about all the particles at once all the time.

$G \longrightarrow G_H$

$G_H$ = dynamic “hierarchization” of underlying scene graph

(right now computed via $k$-means)

graph convolution $\rightarrow$ hierarchical graph convolution
Intuitive Physics as Underlying Goal

\[ \phi^{L2A} \]  graph conv. leaves to ancestors

\[ \phi^{WS} \]  graph conv. with siblings

\[ \phi^{A2D} \]  graph conv. ancestors to descendants
Intuitive Physics as Underlying Goal

\[ \phi^{L2A} \quad \text{graph conv. leaves to ancestors} \]
\[ \phi^{WS} \quad \text{graph conv. with siblings} \]
\[ \phi^{A2D} \quad \text{graph conv. ancestors to descendants} \]

\[ \eta \quad \text{module composing these three operations from one up-down cycle, adding physical effects} \]
Hierarchical graph convolution propagates interactions efficiently
Hierarchical Relational Network (HRN) generates momentum updates ($P$) from hierarchical graph state ($G$):
Deformable cone bouncing off a flat floor
Deformable cone bouncing off a flat floor

Stanford bunny
Deformable box bouncing off an incline
Deformable box bouncing off an incline

Multiple rigid objects colliding
rigid sphere rolling out of rigid bowl
rigid sphere rolling out of rigid bowl

floppy teddybear bouncing off floor and recovering
knocking over an unstable block tower

in GT the tower does fall, but prediction falls too fast . . .
knocking over an unstable block tower

in GT the tower does fall, but prediction falls too fast . . .

slinky

shape is not preserved super well . . .
Results - Folding Cloth

<table>
<thead>
<tr>
<th>Ground Truth</th>
<th>Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="t+1" alt="Image" /></td>
<td><img src="t+1" alt="Image" /></td>
</tr>
<tr>
<td><img src="t+3" alt="Image" /></td>
<td><img src="t+3" alt="Image" /></td>
</tr>
<tr>
<td><img src="t+5" alt="Image" /></td>
<td><img src="t+5" alt="Image" /></td>
</tr>
<tr>
<td><img src="t+7" alt="Image" /></td>
<td><img src="t+7" alt="Image" /></td>
</tr>
<tr>
<td><img src="t+9" alt="Image" /></td>
<td><img src="t+9" alt="Image" /></td>
</tr>
</tbody>
</table>
Results - Videos

Rigid Body Collision
Work in progress: End-to-end prediction of dynamical scenes

Scene at time $T = 0, 1, \ldots, t$

ConvRNN

Physical graph of scene

Hierarchical Relation Network*

Scene at time $T = t+1, t+2, \ldots$

Predictions of graph in future

Work in progress: End-to-end prediction of dynamical scenes

Scene at time $T = 0, 1, \ldots, t$

ConvRNN

Unsupervised Learning

Hierarchical Relation Network*

Scene at time $T = t+1, t+2, \ldots$

Physical graph of scene

Predictions of graph in future

Work in progress: End-to-end prediction of dynamical scenes

Scene at time $T = 0, 1, \ldots, t$

ConvRNN

Physical, 3-D Latent Representation

Object 1
Object 2
Object N

Hierarchical Relation Network*

Scene at time $T = t+1, t+2, \ldots$

Predictions of graph in future

Human-centered feedback loop

1. **A** = architecture class  
   e.g. CNNs

2. **L** = loss function  
   **D** = dataset  
   e.g. Object Categorization

3. **Learning Rule**
   \[
   \text{argmin}_{a \in A} \left[ L(p^*_a) \right]
   \]
   where \( p^* \) is result of
   \[
   \frac{dp_a}{dt} = -\lambda(t) \cdot \langle \nabla_{p_a} L(x) \rangle \forall x \in D
   \]

---

**Artificial Intelligence Algorithms**  
**Experimental Testing**

- **AI Improvement**
- **Comparision**
- **Refinement**

**Model Testing**

- **Computational Predictions**
- **Developmental Studies**
- **Neuroscientific Validation**
animal fixates; sees fixation cue; sees stimulus; takes action (saccade)
animal fixates; sees fixation cue; sees stimulus; takes action (saccade)

goal depends on cue
animal fixates; sees fixation cue; sees stimulus; takes action (saccade)

goal depends on cue

there are axes of coherence for both motion and color
animal fixates; sees fixation cue; sees stimulus; takes action (saccade)

goal depends on cue

there are axes of coherence for both motion and color

animal can do task
Executive Control and Task Switching

effect of motion direction/strength input

gray = motion input effect
blue = color input effect

effect of color “direction” input
Executive Control and Task Switching

effect of motion direction/strength input

1. evidence accumulation = gradual motion of the population response along choice axis

grey = motion input effect
blue = color input effect

effect of color “direction” input
Executive Control and Task Switching

effect of motion direction/strength input

grey = motion input effect
blue = color input effect

2. population response to sensory input different from choice directions ... e.g. arcs away from choice axis

effect of color “direction” input
Executive Control and Task Switching

Effect of motion direction/strength input

**Gray = motion input effect**

**Blue = color input effect**

3. Context doesn’t have effect on the axes; and only weak effect on the strength of the signals (e.g. compare a-c to d-f)

Effect of color “direction” input
Executive Control and Task Switching

effect of motion direction/strength input

grey = motion input effect
blue = color input effect

4. but location in state space are *not* context independent

effect of color “direction” input
Context-dependent computation by recurrent dynamics in prefrontal cortex

Valerio Mante1++, David Sussillo2*, Krishna V. Shenoy2,3 & William T. Newsome1

actual responses (schematic)
Executive Control and Task Switching

Context-dependent computation by recurrent dynamics in prefrontal cortex

Valerio Mante1, David Sussillo2, Krishna V. Shenoy2,3 & William T. Newsome1

actual responses
(schematic)

Wrong models:
Executive Control and Task Switching

Context-dependent computation by recurrent dynamics in prefrontal cortex

Valerio Mante1,*, David Sussillo2, Krishna V. Shenoy2,3 & William T. Newsome1

actual responses (schematic)  gating of input by context signal
Context-dependent computation by recurrent dynamics in prefrontal cortex

Valerio Mante1,*, David Sussillo2,*, Krishna V. Shenoy2,3 & William T. Newsome1

Executive Control and Task Switching

actual responses (schematic)  gating of input by context signal  modifying of angle between choice and motion axes ...
Executive Control and Task Switching

Context-dependent computation by recurrent dynamics in prefrontal cortex

Valerio Mante\(^1\), David Sussillo\(^2\), Krishna V. Shenoy\(^2,3\) & William T. Newsome\(^1\)

actual responses (schematic)

gating of input by context signal

modifying of angle between choice and motion axes ...

changing direction of motion axis
Context-dependent computation by recurrent dynamics in prefrontal cortex

Valerio Mante1†*, David Sussillo2*, Krishna V. Shenoy2,3 & William T. Newsome1

actual responses (schematic)
gating of input by context signal

modifying of angle between choice and motion axes . . .
changing direction of motion axis
changing direction of choice axis
Context-dependent computation by recurrent dynamics in prefrontal cortex

Valerio Mante1‡*, David Sussillo2*, Krishna V. Shenoy2,3 & William T. Newsome1
Executive Control and Task Switching

Context-dependent computation by recurrent dynamics in prefrontal cortex

Valerio Mante1*, David Sussillo2*, Krishna V. Shenoy2,3 & William T. Newsome1

a. gradual motion along choice axis
Context-dependent computation by recurrent dynamics in prefrontal cortex

Valerio Mante1*, David Sussillo2*, Krishna V. Shenoy2,3 & William T. Newsome1

Executive Control and Task Switching

a. gradual motion along choice axis

b. parametric ordering of trajectories along the motion and color axes
Context-dependent computation by recurrent dynamics in prefrontal cortex

Valerio Mante¹**, David Sussillo²*, Krishna V. Shenoy²,³ & William T. Newsome¹

a. gradual motion along choice axis

b. parametric ordering of trajectories along the motion and color axes

c. direction of axes of choice, motion and color are invariant to context
a. gradual motion along choice axis

b. parametric ordering of trajectories along the motion and color axes

c. direction of axes of choice, motion and color are invariant to context

d. motion and color trajectories are separated along context axis
How does same input pulse lead to evidence integration in one context, but ignored in other?
How does same input pulse lead to evidence integration in on context, but ignored in other?

Relaxation toward line attractor is orthogonal to context-dependent selection vector, reversing effect of irrelevant pulse.
Learning to Reason: End-to-End Module Networks for Visual Question Answering

Ronghang Hu\textsuperscript{1} Jacob Andreas\textsuperscript{1} Marcus Rohrbach\textsuperscript{1,2} Trevor Darrell\textsuperscript{1} Kate Saenko\textsuperscript{3}  
\textsuperscript{1}University of California, Berkeley \textsuperscript{2}Facebook AI Research \textsuperscript{3}Boston University  
{ronghang, jda, trevor, rohrbach}@eecs.berkeley.edu, saenko@bu.edu
Learning to Reason: End-to-End Module Networks for Visual Question Answering

Ronghang Hu1  Jacob Andreas1  Marcus Rohrbach1,2  Trevor Darrell1  Kate Saenko3
1University of California, Berkeley  2Facebook AI Research  3Boston University
{ronghang,jda,trevor,rohrbach}@eecs.berkeley.edu, saenko@bu.edu

<table>
<thead>
<tr>
<th>Module name</th>
<th>Att-inputs</th>
<th>Features</th>
<th>Output</th>
<th>Implementation details</th>
</tr>
</thead>
<tbody>
<tr>
<td>find</td>
<td>(none)</td>
<td>x_{vis}, x_{txt}</td>
<td>att</td>
<td>a_{out} = conv2(conv1(x_{vis}) \odot W_{x_{txt}})</td>
</tr>
<tr>
<td>relocate</td>
<td>a</td>
<td>x_{vis}, x_{txt}</td>
<td>att</td>
<td>a_{out} = conv2(conv1(x_{vis}) \odot W_{1}\text{sum}(a \odot x_{vis}) \odot W_{2}x_{txt})</td>
</tr>
<tr>
<td>and</td>
<td>a, a_1, a_2</td>
<td>(none)</td>
<td>att</td>
<td>a_{out} = \text{minimum}(a_1, a_2)</td>
</tr>
<tr>
<td>or</td>
<td>a, a_1, a_2</td>
<td>(none)</td>
<td>att</td>
<td>a_{out} = \text{maximum}(a_1, a_2)</td>
</tr>
<tr>
<td>filter</td>
<td>a</td>
<td>x_{vis}, x_{txt}</td>
<td>att</td>
<td>a_{out} = a \odot (a, \text{find}(x_{vis}, x_{txt})(s)), i.e. reusing find and and</td>
</tr>
<tr>
<td>[exist, count]</td>
<td>a</td>
<td>(none)</td>
<td>ans</td>
<td>y = W^T \text{vec}(a)</td>
</tr>
<tr>
<td>describe</td>
<td>a</td>
<td>x_{vis}, x_{txt}</td>
<td>ans</td>
<td>y = W^T \text{vec}(a_1) + W^T_{2} \text{vec}(a_2)</td>
</tr>
<tr>
<td>[eq.count, more, less]</td>
<td>a, a_1, a_2</td>
<td>(none)</td>
<td>ans</td>
<td>y = W^T_{1} \text{sum}(a_1 \odot x_{vis}) \odot W_{2}\text{sum}(a_2 \odot x_{vis}) \odot W_{4}x_{txt})</td>
</tr>
</tbody>
</table>
A Dataset and Architecture for Visual Reasoning with a Working Memory

Guangyu Robert Yang, Igor Ganichev, Xiao-Jing Wang, Jonathon Shlens, David Sussillo; The European Conference on Computer Vision (ECCV), 2018, pp. 714-731
A Dataset and Architecture for Visual Reasoning with a Working Memory

Guangyu Robert Yang, Igor Ganchev, Xiao-Jing Wang, Jonathon Shlens, David Sussillo; The European Conference on Computer Vision (ECCV), 2018, pp. 714-731
Gluck and Meyers (1993)

- Hypothesized that many of the behaviors the hippocampus supports result from two representational biases in the system:
  - Predictive differentiation: enhance the discriminability of ambiguous cues that predict future outcomes
  - Redundancy compression: combining or clustering non-predictive stimulus features
Unsupervised Predictive Memory in a Goal-Directed Agent

Greg Wayne*,1, Chia-Chun Hung*,1, David Amos*,1, Mehdi Mirza1,

- $I_t$: image at time $t$
- $r_t$: reward
- $v_t$: velocity
- $T_t$: text input
- $h_t$: LSTM-memory
- $n_t$: action distribution
- $a_t$: action sample

![Diagram of RL-LSTM](image)
Unsupervised Predictive Memory in a Goal-Directed Agent

Greg Wayne*,1, Chia-Chun Hung*,1, David Amos*,1, Mehdi Mirza1,

\[
\begin{align*}
I_t & \quad \text{image at time } t \\
r_t & \quad \text{reward} \\
v_t & \quad \text{velocity} \\
T_t & \quad \text{text input} \\
h_t & \quad \text{LSTM-memory} \\
n_t & \quad \text{action distribution} \\
a_t & \quad \text{action sample} \\
M_t & \quad \text{neural turing machine-like external memory}
\end{align*}
\]
Unsupervised Predictive Memory in a Goal-Directed Agent

Greg Wayne*,¹, Chia-Chun Hung*,¹, David Amos*,¹, Mehdí Mirza¹,

$I_t$ image at time $t$  
$r_t$ reward  
$v_t$ velocity  
$T_t$ text input  

$M_t$ NTM memory  
$n_t$ action distribution  
$a_t$ action sample

$p, q$ prior, posterior on world state + reconstruction loss
Memory Game

(a) Memory Game

(b) 1 2 3 4 5 6

Episode Score

Number of Environment Steps
Tolman-Like Navigation

https://www.youtube.com/watch?v=YFx-D4eEs5A
Arbitrary visuomotor mapping

https://youtu.be/liR_NOomcpk
Rapid Reward Evaluation

https://www.youtube.com/watch?v=dQMKJtLScmk
Episodic Water Mazes

https://youtu.be/xrYDlTXyC6Q
Executing Instructions

https://www.youtube.com/watch?v=04H28-qA3f8
T-Maze

https://youtu.be/3iA19h0Vvq0
T-Maze

![Graph showing the episode score over the number of environment steps for different models: MERLIN, RL-LSTM, NEG. VLB, and RL-MEM. The y-axis represents the episode score ranging from 0 to 18, and the x-axis represents the number of environment steps ranging from 0 to $10^9$. The graph compares the performance of these models with negative VLB values.]