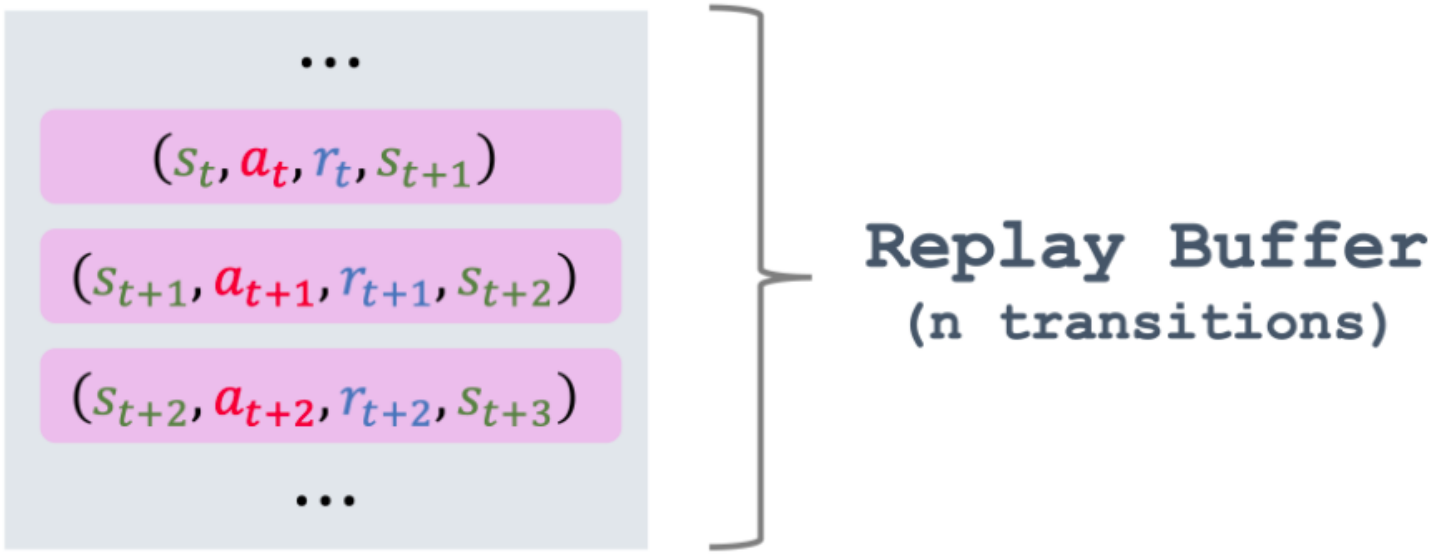
Experience Replay

- **A transition:** (s_t, a_t, r_t, s_{t+1}) .
- Store recent n transitions in a **replay buffer**.

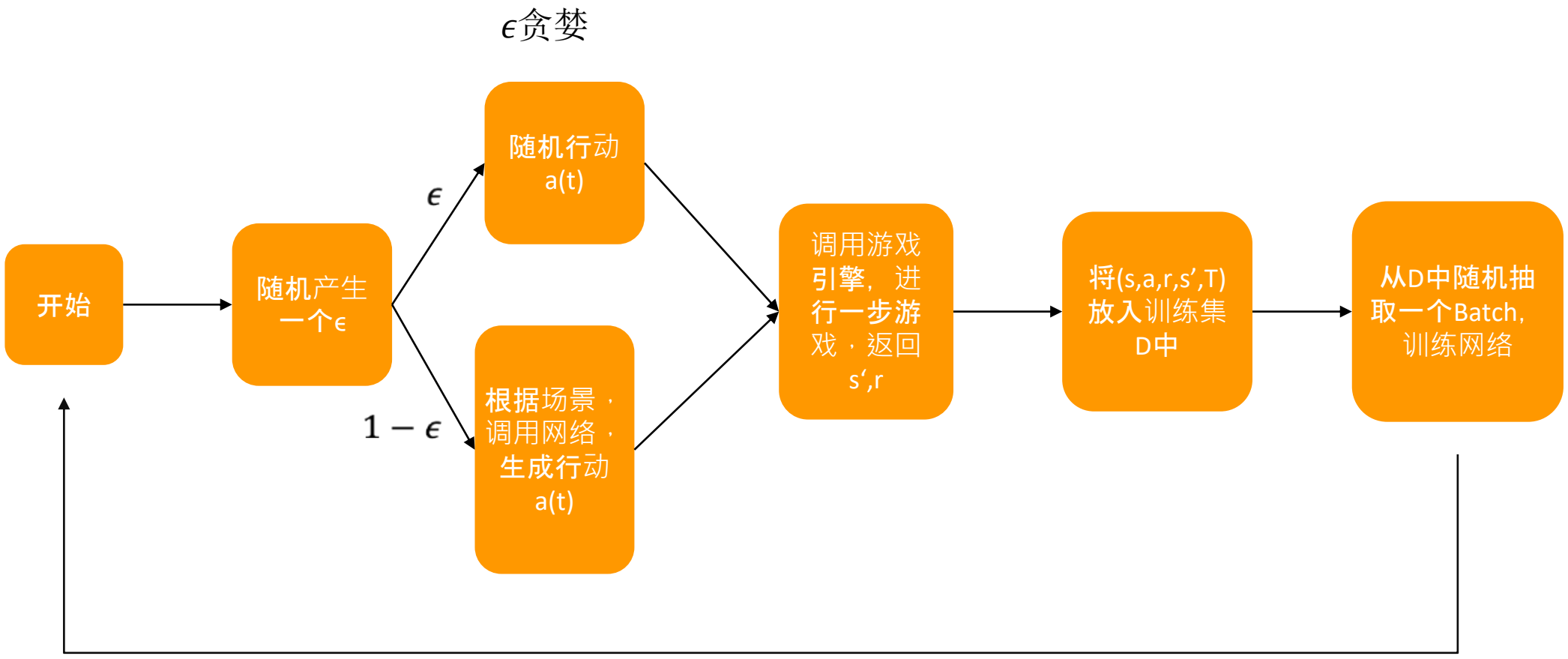
Benefits of Experience Replay

1. Make the updates uncorrelated.
2. Reuse collected experience many times.



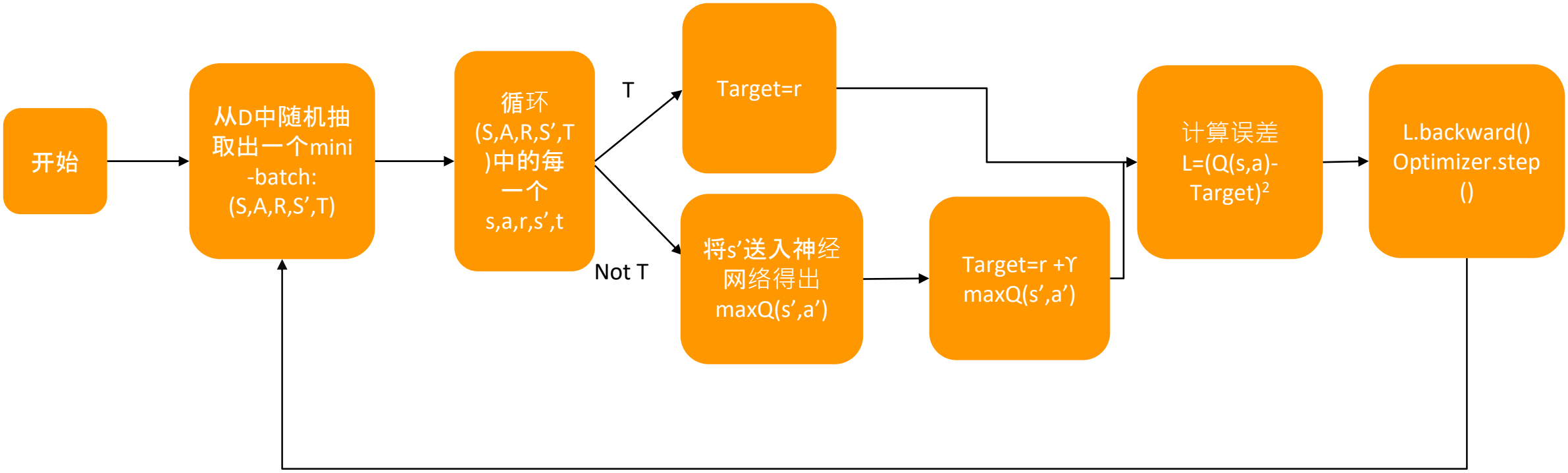
Human-level control through deep reinforcement learning
https://www.nature.com/articles/nature14236

One Step Action



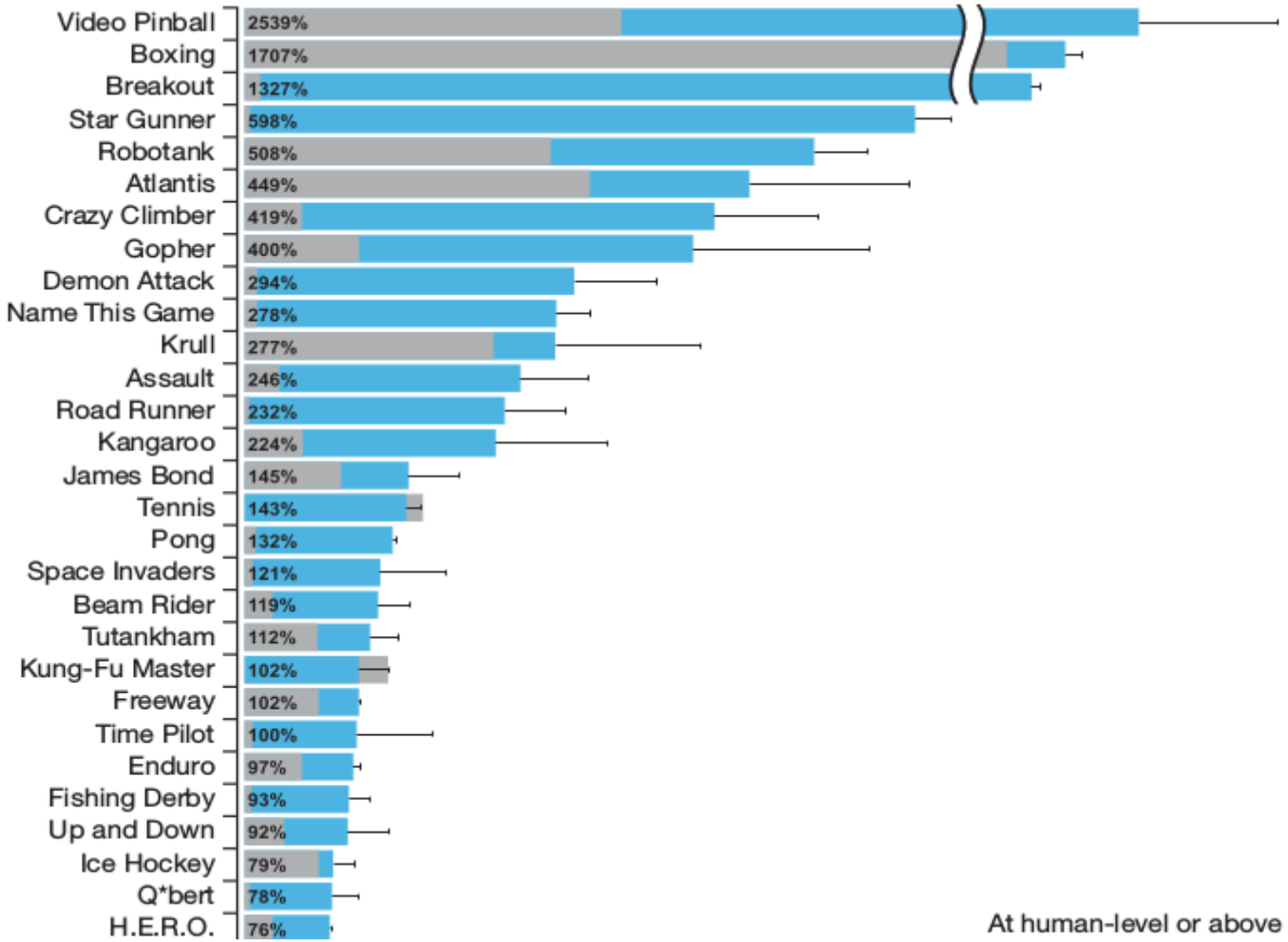
Human-level control through deep reinforcement learning
https://www.nature.com/articles/nature14236

One Step Learning



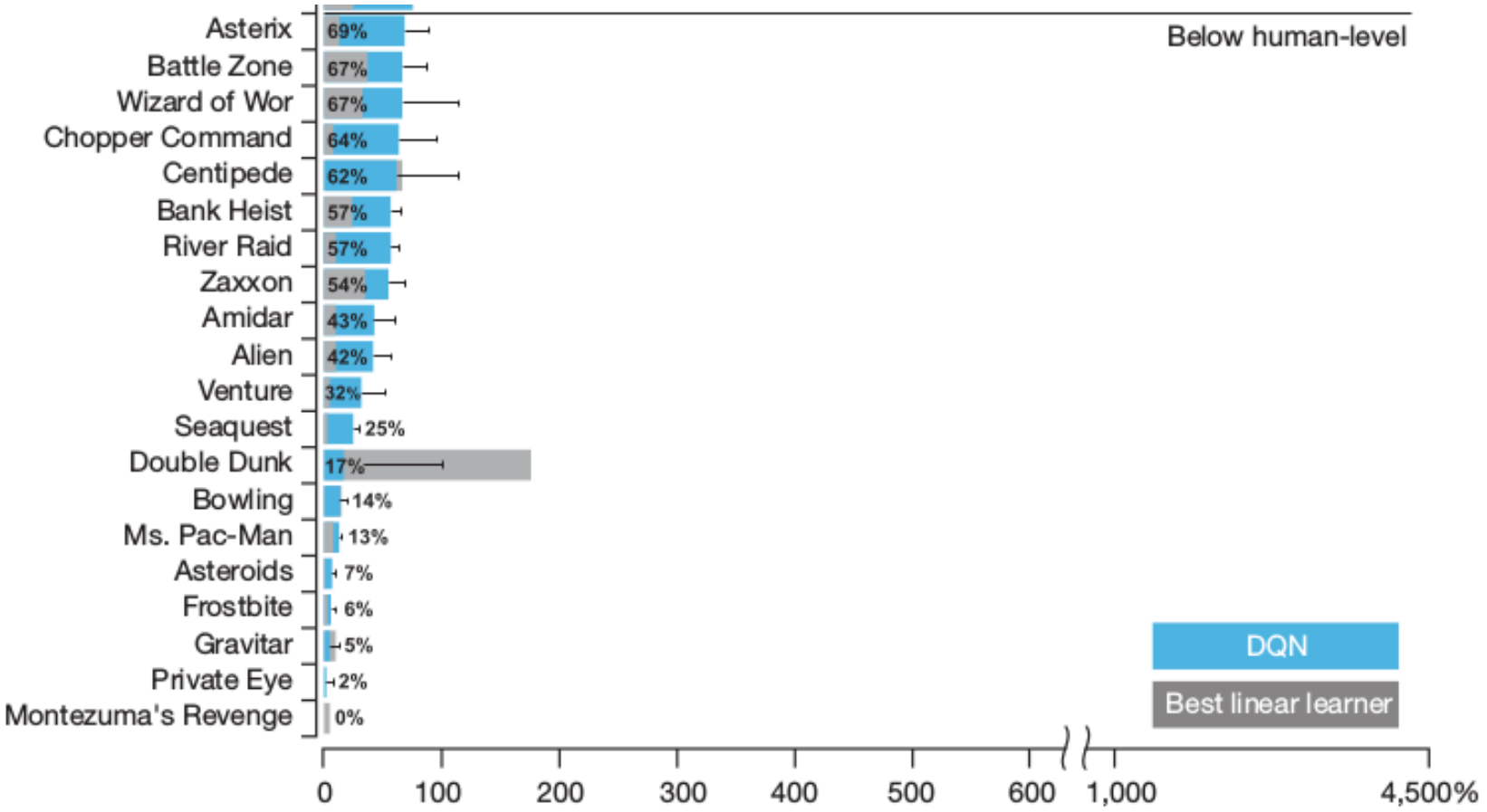
Human-level control through deep reinforcement learning
https://www.nature.com/articles/nature14236

Results



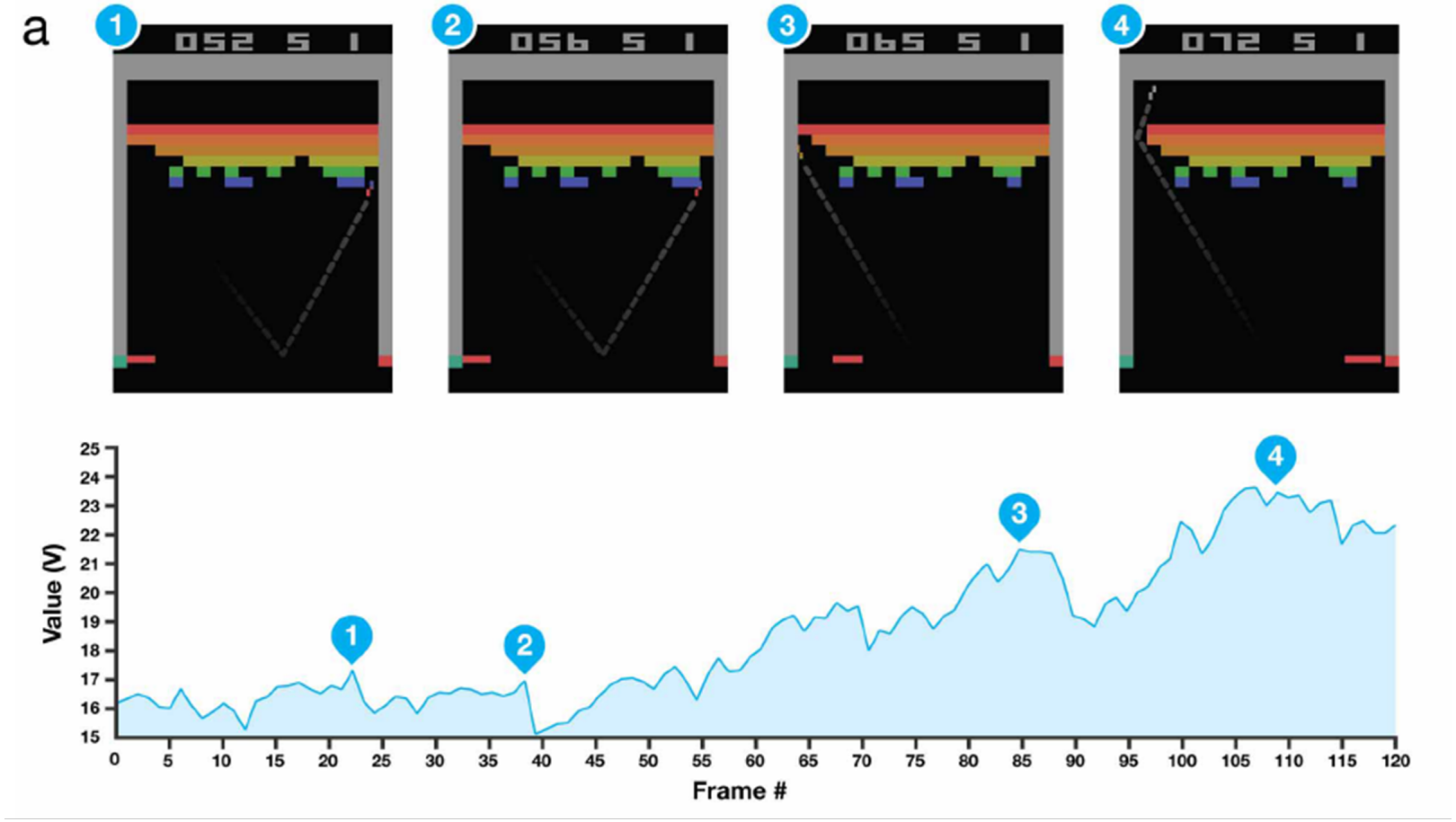
Human-level control through deep reinforcement learning
https://www.nature.com/articles/nature14236

Results



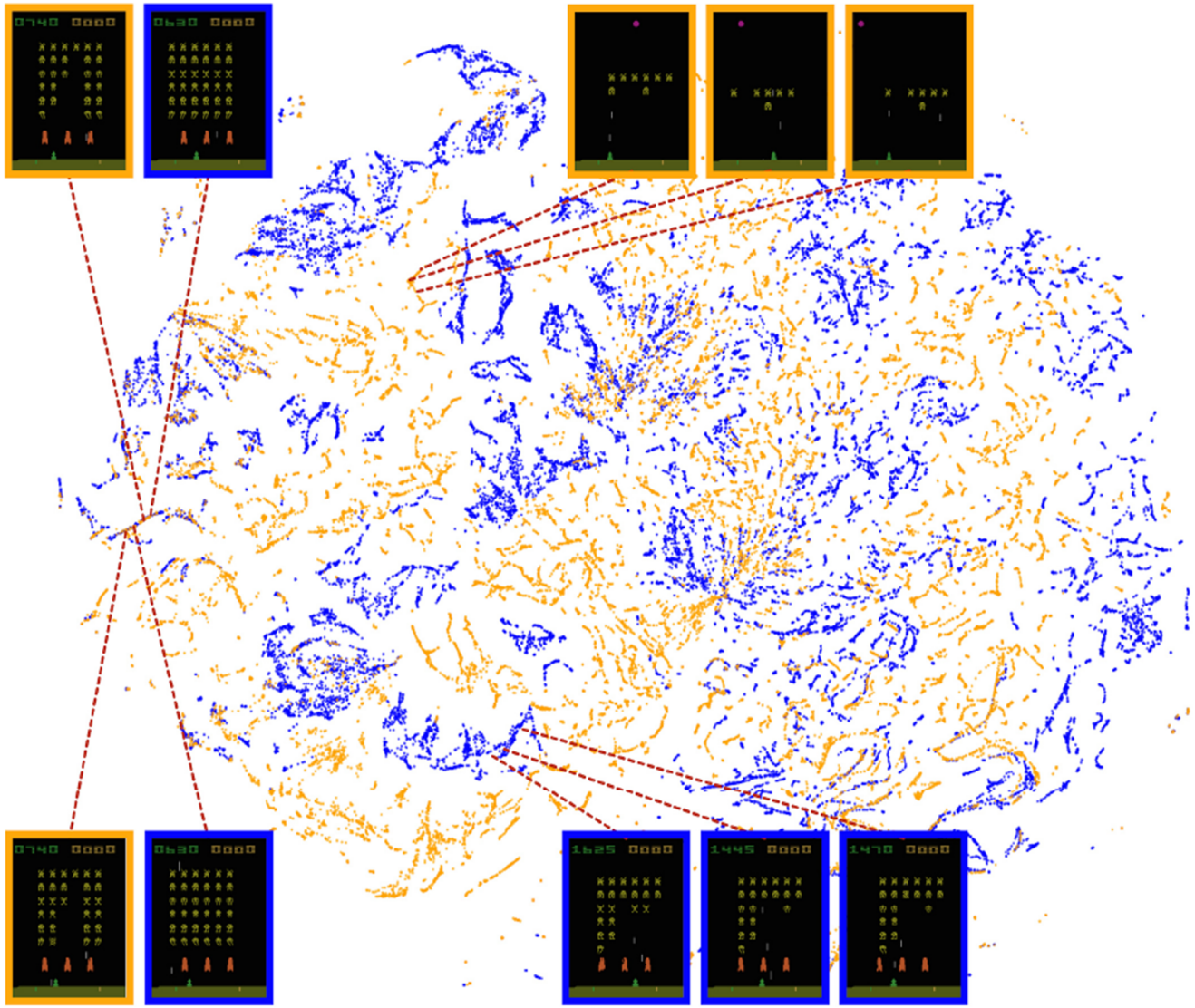
Human-level control through deep reinforcement learning
https://www.nature.com/articles/nature14236

Emergent Abilities



Human-level control through deep reinforcement learning
https://www.nature.com/articles/nature14236

Visualization



- 蓝色：人类
- 橙色：机器

Human-level control through deep reinforcement learning
https://www.nature.com/articles/nature14236

DQN - AGI

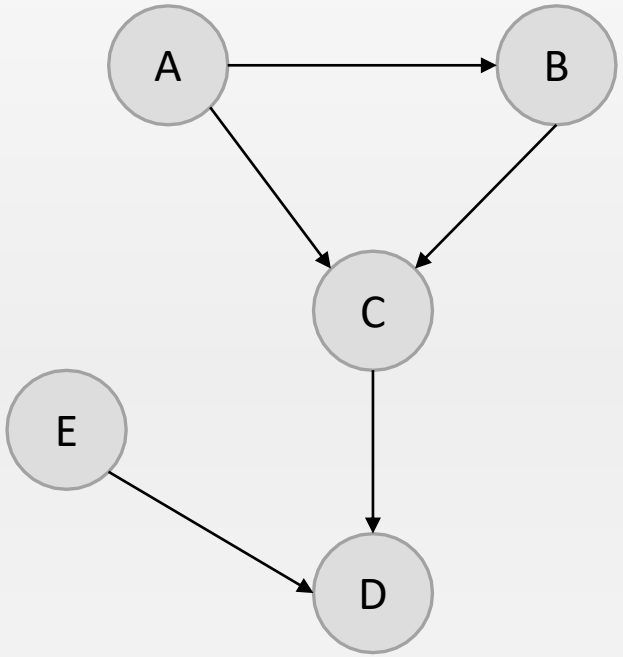
- **Artificial general intelligence** (**AGI**) is the intelligence of a machine that could successfully perform any intellectual task that a human being can.
- DQN可以看作游戏世界中的AGI

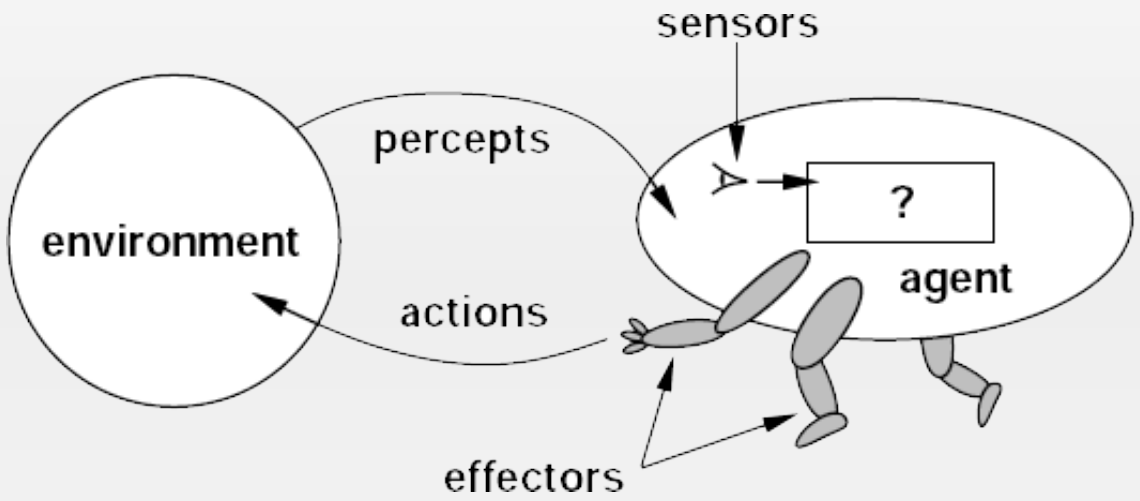
Human-level control through deep reinforcement learning
https://www.nature.com/articles/nature14236

Outline

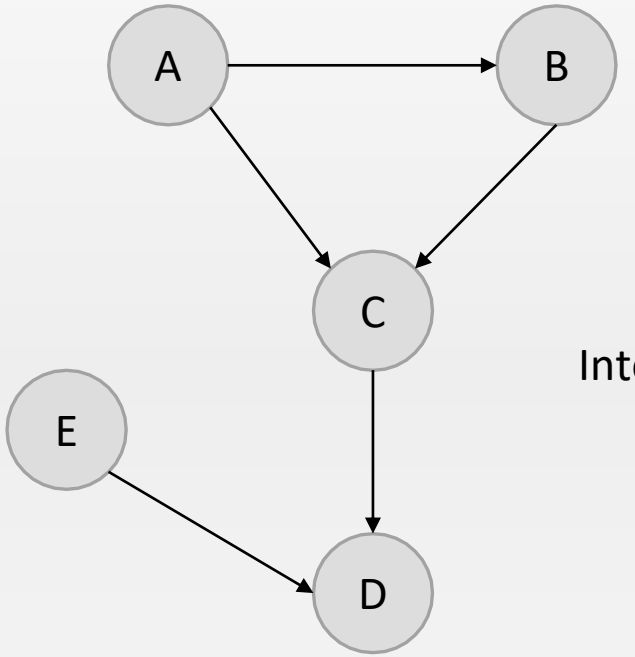
- Basic Notions
- AlphaGo
- Deep Q-Learning
- Causal Reinforcement Learning
- World Models

Causality and Intervention

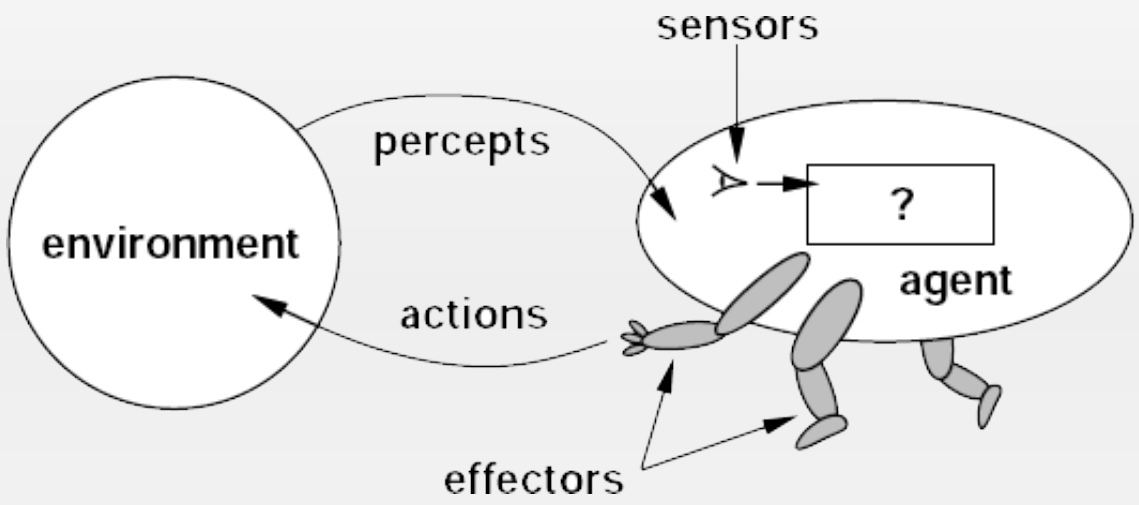




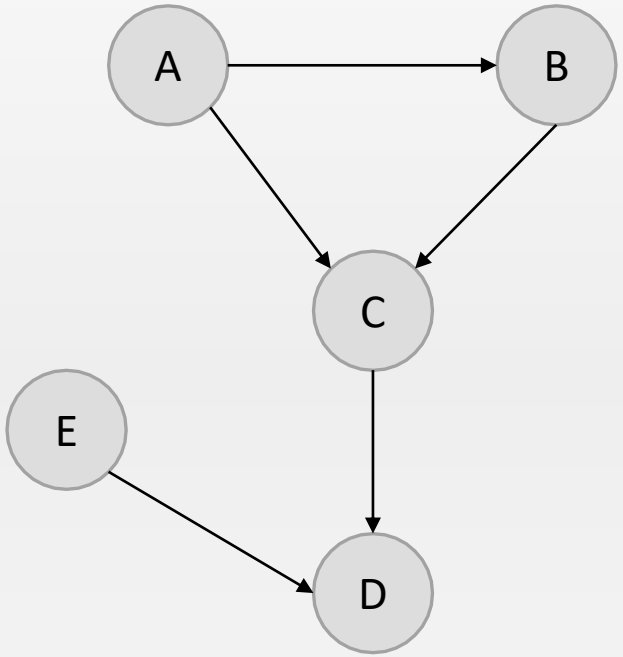
Causality and Intervention



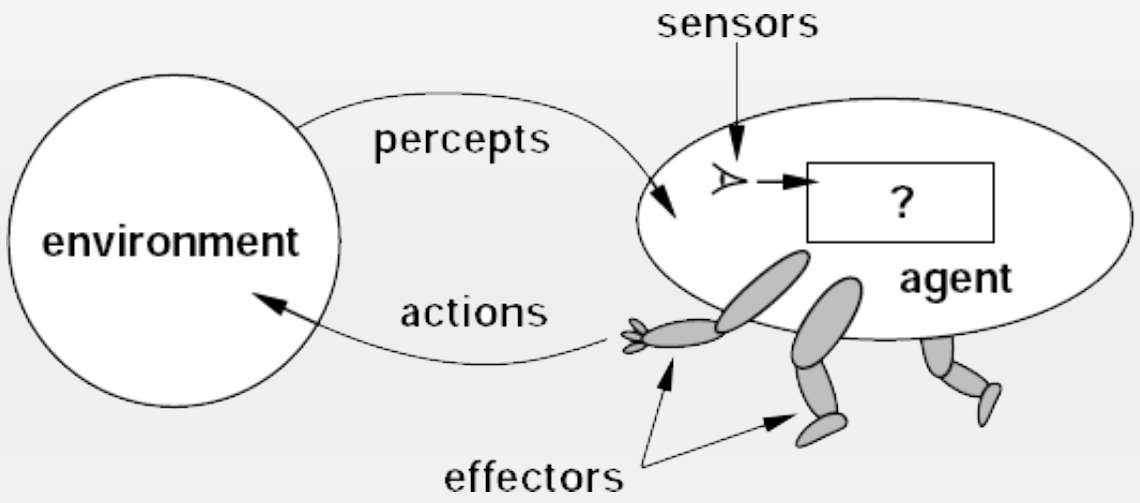
Interaction, Intervention



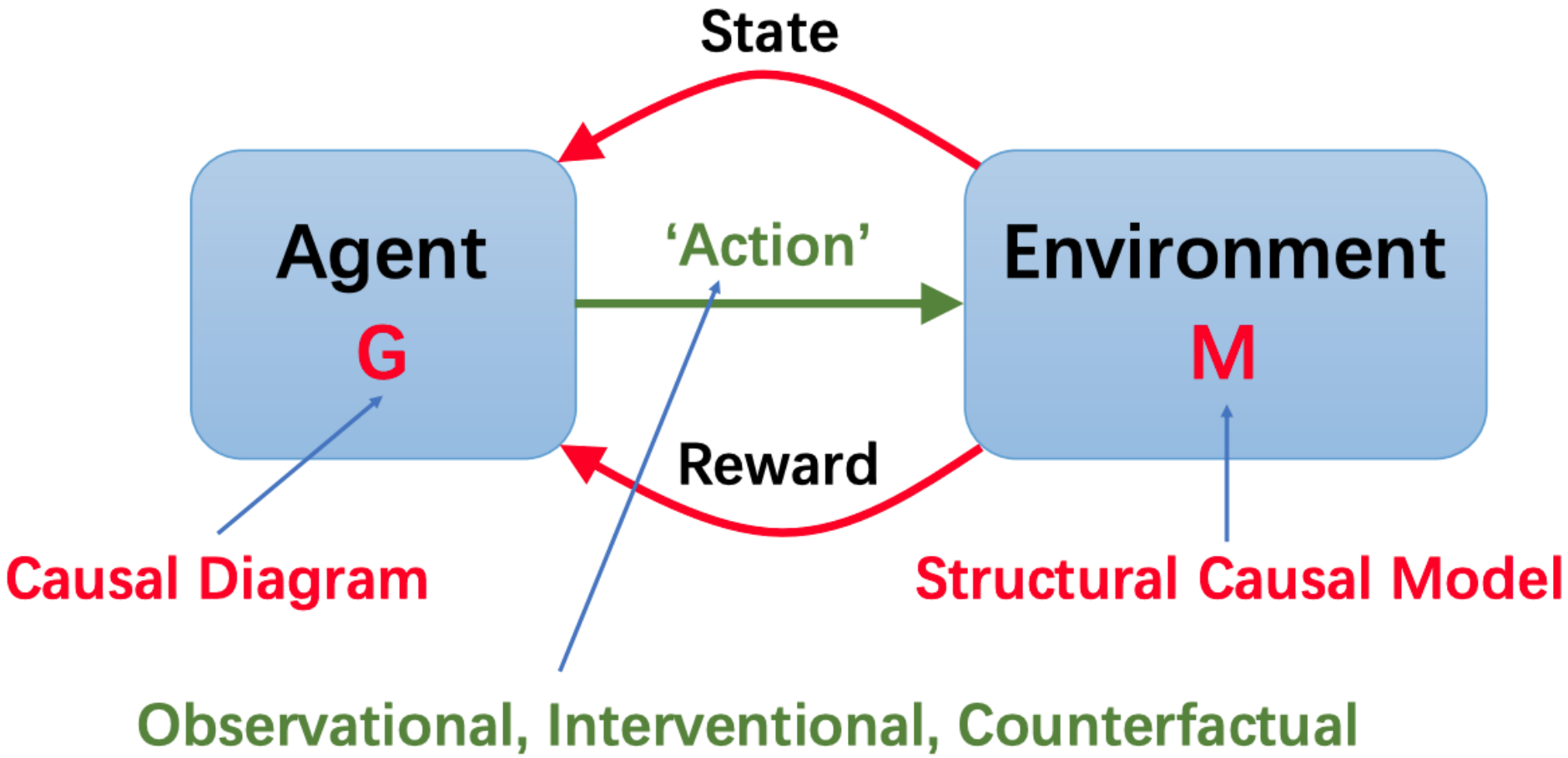
Causality and Intervention



How to do action



Causal Reinforcement Learning



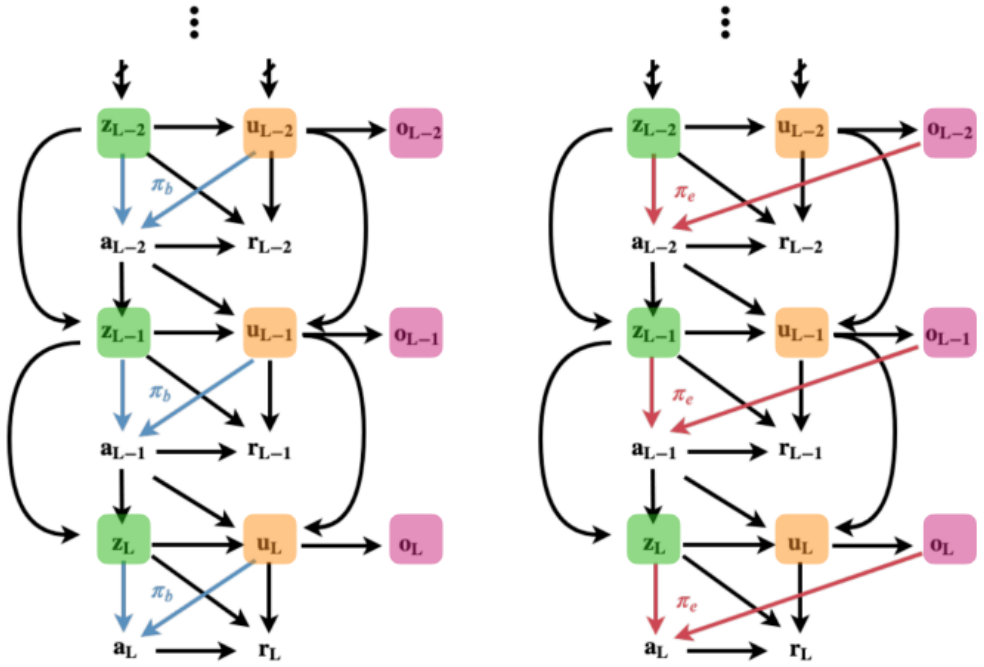
Observational: off-policy evaluation

z_L

medical state

u_L

doctor's current mood
&
how good the patient looks to



(b) Decoupled POMDP

Figure 1: A causal diagram of a POMDP (a) and a Decoupled POMDP (b). In Decoupled POMDPs, observed and unobserved states are separated into two distinct processes, with a coupling between them at each time step. Diagrams depicts the causal dependence of a behavior policy and evaluation policies. While evaluation policies are depicted to depend on the current observation alone, they can depend on any observable history h_t^o.

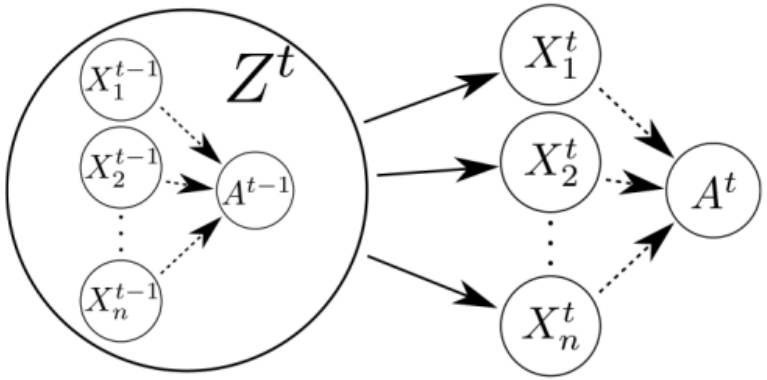
Tennenholtz et al. Off-Policy Evaluation in Partially Observable Environments. 2019

Observational: Imitation Learning

$$\mathcal{D} = \{(s_1, a_1), (s_2, a_2), (s_3, a_3), \ldots\}$$

de Hann et al. Causal Confusion in Imitation Learning, 2018

Interventional: self-driving



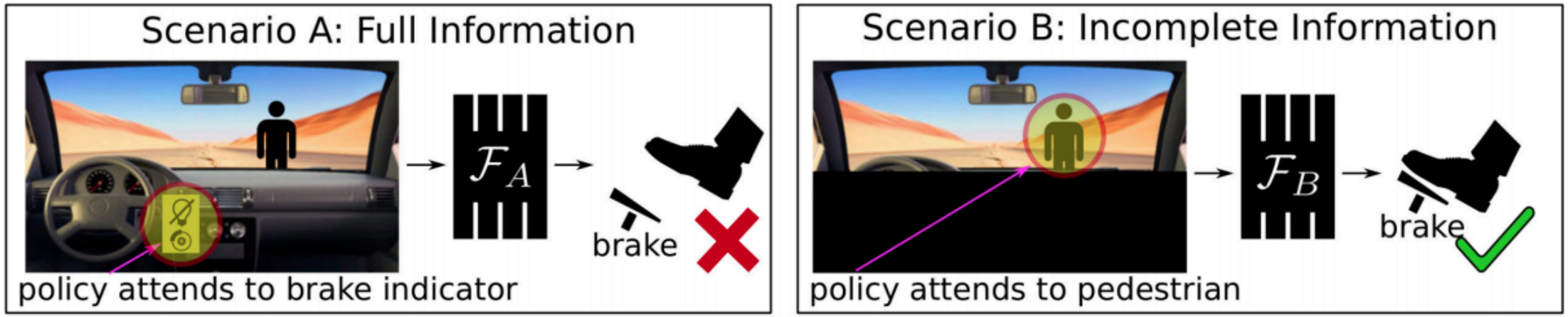
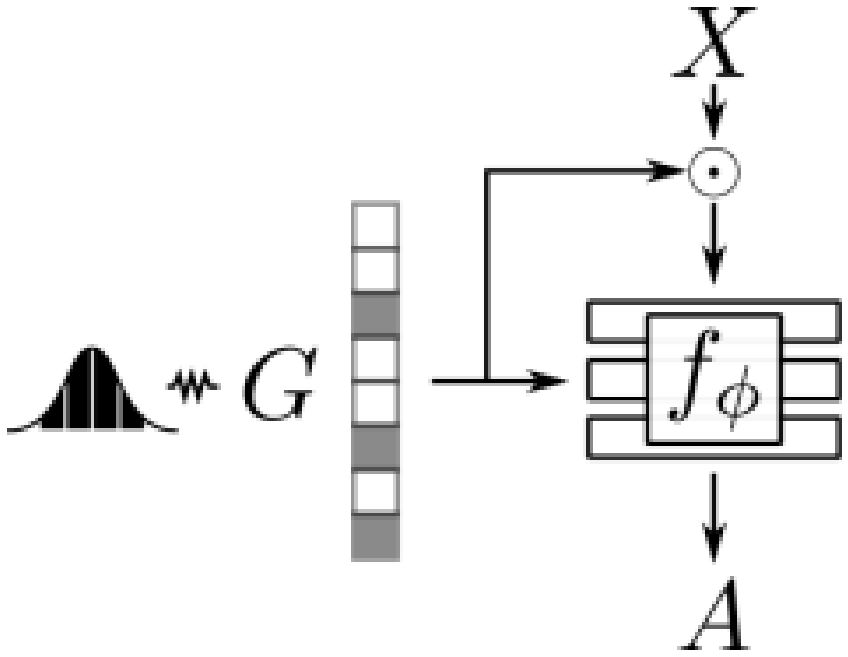


Figure 1: Causal misidentification: *more* information yields worse imitation learning performance. Model A relies on the braking indicator to decide whether to brake. Model B instead correctly attends to the pedestrian.

de Hann et al. Causal Confusion in Imitation Learning, 2018

Interventional: self-driving



Algorithm 1 Expert query intervention

Input: policy network f_ϕ s.t. $\pi_G(X) = f_\phi([X \odot G, G])$
Initialize $w = 0, \mathcal{D} = \emptyset$.
Collect states $\mathcal{S}$ by executing π_{mix}, the mixture of policies π_G for uniform samples G.
For each X in S, compute disagreement score:
$D(X) = \mathbb{E}_G[D_{KL}(\pi_G(X), \pi_{mix}(X))]$
Select $\mathcal{S}' \subset \mathcal{S}$ with maximal $D(X)$.
Collect state-action pairs $\mathcal{T}$ by querying expert on $\mathcal{S}'$.
for $i = 1 \ldots N$ **do**
  Sample $G \sim p(G) \propto \exp\langle w, G\rangle$.
  $\mathcal{L} \leftarrow \mathbb{E}_{s,a\sim\mathcal{T}}[\ell(\pi_G(s), a)]$
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(G, \mathcal{L})\}$
  Fit w on $\mathcal{D}$ with linear regression.
end for
Return: $\arg\max_G p(G)$

de Hann et al. Causal Confusion in Imitation Learning, 2018

Counterfactual: policy search

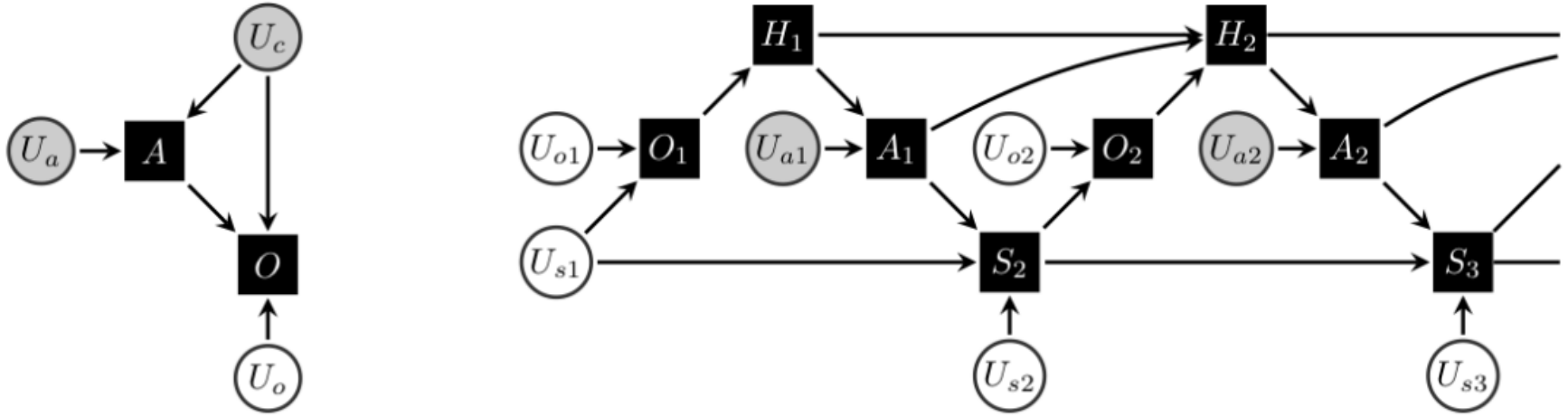
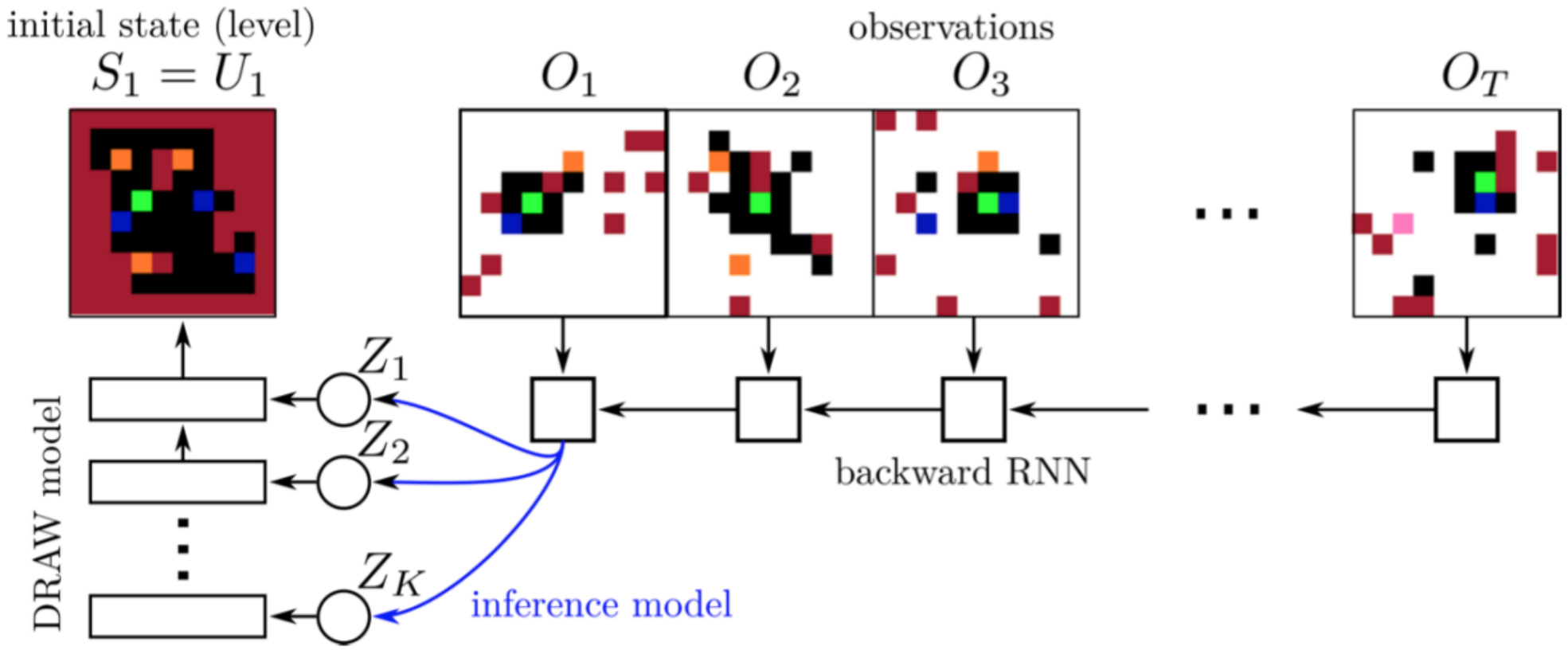


Figure 1: Structural causal models (SCMs) model environments using random variables U (circles, 'scenarios'), that summarize immutable aspects, some of which are observed (grey), some not (white). These are fed into deterministic functions f_i (black squares) that approximate causal mechanisms. **Left:** SCM for a contextual bandit with context U_c, action A, feedback O and scenario U_o. **Right:** SCM for a POMDP, with initial state $U_{s1} = S_1$, states S_t and histories H_t. The mechanism that generates the actions A_t is the policy π.

Buesing et al. Woulda, Coulda, Shoulda: Counterfactually-Guided Policy Search, 2019

Counterfactual: policy search



Buesing et al. Woulda, Coulda, Shoulda: Counterfactually-Guided Policy Search, 2019

Causal Discovery

A major class of such causal discovery methods are score-based, which assign a score $\mathcal{S}(\mathcal{G})$, typically computed with the observed data, to each directed graph $\mathcal{G}$ and then search over the space of all Directed Acyclic Graphs (DAGs) for the best scoring:

$$\min_{\mathcal{G}} \mathcal{S}(\mathcal{G}), \text{ subject to } \mathcal{G} \in \text{DAGs}.$$

$$J(\psi \mid \mathbf{s}) = \mathbb{E}_{A \sim \pi(\cdot|\mathbf{s})} \{ - [\mathcal{S}(\mathcal{G}) + \lambda_1 \mathbf{I}(\mathcal{G} \notin \text{DAGs}) + \lambda_2 h(A)] \} .$$

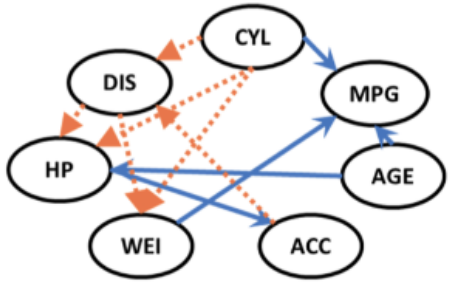
$$h(A) \coloneqq \operatorname{trace}\left(e^A\right) - d = 0,$$

CAUSAL DISCOVERY WITH REINFORCEMENT LEARNING

Shengyu Zhu[†] **Ignavier Ng**[§*] **Zhitang Chen**[†]
[†]Huawei Noah's Ark Lab [§]University of Toronto
[†]{zhushengyu,chenzhitang2}@huawei.com [§]ignavierng@cs.toronto.edu

ABSTRACT

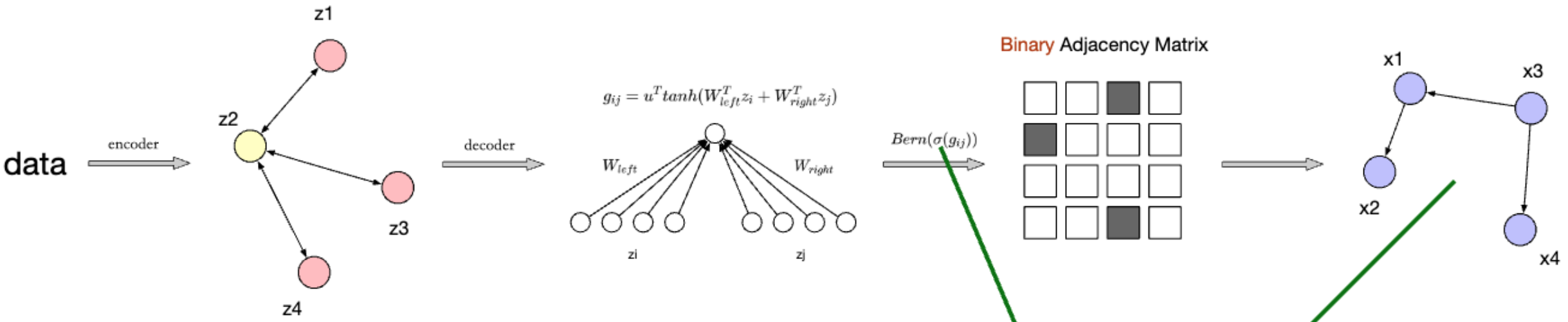
Discovering causal structure among a set of variables is a fundamental problem in many empirical sciences. Traditional score-based casual discovery methods rely on various local heuristics to search for a Directed Acyclic Graph (DAG) according to a predefined score function. While these methods, e.g., greedy equivalence search, may have attractive results with infinite samples and certain model assumptions, they are less satisfactory in practice due to finite data and possible violation of assumptions. Motivated by recent advances in neural combinatorial optimization, we propose to use Reinforcement Learning (RL) to search for the DAG with the best scoring. Our encoder-decoder model takes observable data as input and generates graph adjacency matrices that are used to compute rewards. The reward incorporates both the predefined score function and two penalty terms for enforcing acyclicity. In contrast with typical RL applications where the goal is to learn a policy, we use RL as a search strategy and our final output would be the graph, among all graphs generated during training, that achieves the best reward. We conduct experiments on both synthetic and real datasets, and show that the proposed approach not only has an improved search ability but also allows a flexible score function under the acyclicity constraint.

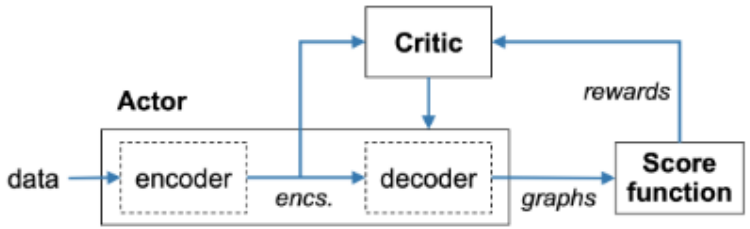


- Zhu S, Ng I, Chen Z. **Causal discovery with reinforcement learning**[J]. In ICLR 2020.

Zhu S, Ng I, Chen Z. Causal discovery with reinforcement learning, ICLR 2020

Causal Discovery





$$Cross\_Entropy\left(\begin{bmatrix}0.3 & 0.8 & 0.2 & 0.4\\ 0.5 & 0.2 & 0.6 & 0.1\\ 0.7 & 0.9 & 0.3 & 0.1\\ 0.2 & 0.4 & 0.8 & 0.4\end{bmatrix}, \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 1\\ 0 & 0 & 0 & 0\end{bmatrix}\right)$$

1. the predefined score function；
2. two penalty terms to enforce acyclicity

Zhu S, Ng I, Chen Z. Causal discovery with reinforcement learning, ICLR 2020

Problems Overview

Problem	Output	Benefits over non-causal RL
Causal Bandits	$\hat{\pi} = \arg\min_{\pi \in \Pi} L_n(\pi)$	Optimal simple regret guarantees
Model-Based RL	$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \ell\left(\boldsymbol{\theta}, \left(R_{t+1}, S_{t+1}\right)\right)$	Deconfounding
Multi-Environment RL	$\hat{\pi} = \arg\max_{\pi \in \Pi} \mathbb{E}_{c \sim p(c)}\left[\mathcal{R}\left(\pi, \mathcal{M}^c\right)\right]$	Interpretable task embeddings, systematic generalization
Off-Policy Policy Evaluation	$\hat{v}_\pi(s) = \mathbb{E}_{\boldsymbol{x} \sim d_0}\left[\sum_{t=0}^{T-1} \gamma^t r_t \mid \boldsymbol{x}_0 = \boldsymbol{x}\right]$	Deconfounding
Imitation Learning	$\hat{\pi} = \arg\min_{\pi \in \Pi} \mathbb{E}_{\boldsymbol{x} \sim d_{\pi^*}}\left[\ell\left(\boldsymbol{x}, \pi, \pi^*\left(\boldsymbol{x}\right)\right)\right]$	Deconfounding
Credit Assignment	$\mathcal{M}_{a_t \to r_{t+k}}$ or $\mathcal{M}_{a_t \to s_{t+1}}$ or $\mathcal{M}_{a_t^i \to a_t^j}$	Intrinsic reward, Data-efficiency
Counterfactual Data Augmentation	$\tilde{\tau} = \{\tilde{\boldsymbol{x}}_t, \tilde{a}_t, \tilde{\boldsymbol{x}}_{t+1}\}_{t=1}^T$	Data-efficiency

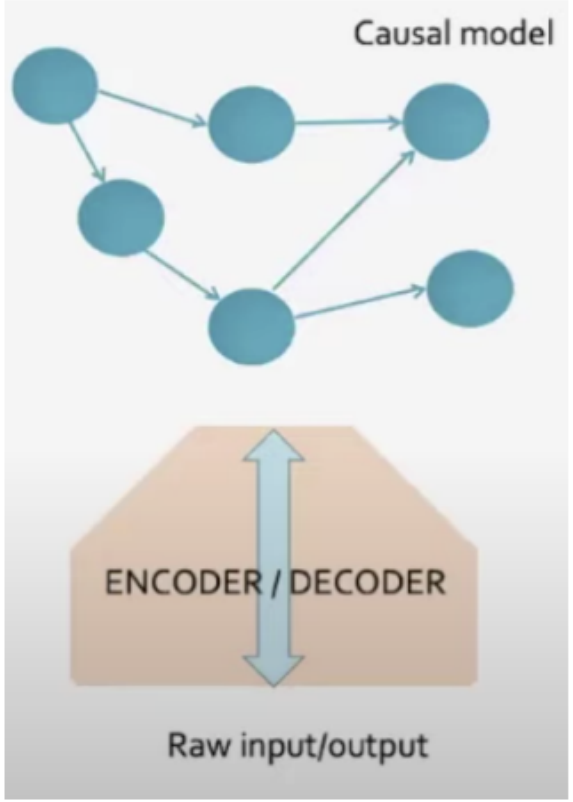
Kaddour et al. 2022: https://arxiv.org/abs/2206.15475

Outline

- Basic Notions
- AlphaGo
- Deep Q-Learning
- Causal Reinforcement Learning
- World Models

Learning Causal Representations

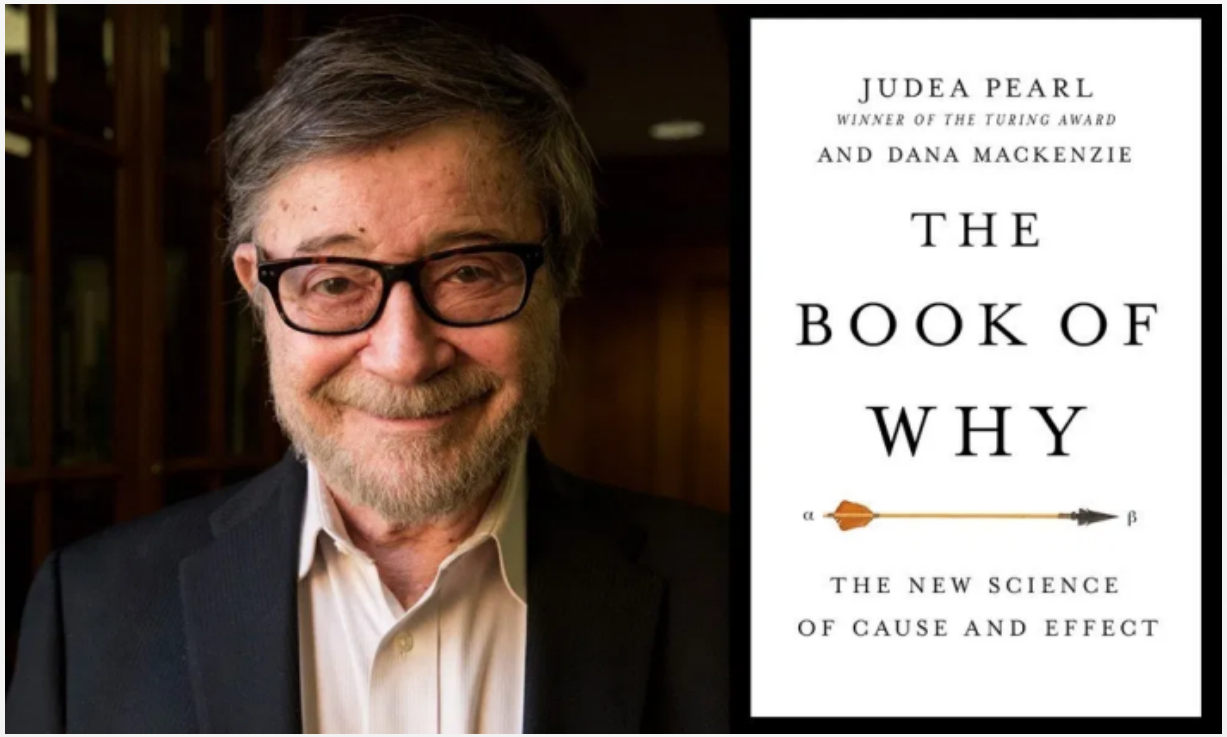
- Given possibly high-dim $X = (X_1, X_2, \dots, X_d)$, we should construct causal variables $S = (S_1, S_2, \dots, S_n)$, where $n \ll d$ as well as causal mechanisms :
  - $S_i = f_i(PA_i, U_i)$ (i = 1,..., n) .
- **Causal Feature Learning**: an injective mapping q: $\mathbb{R}^n \to \mathbb{R}^d$ $s.t. X = q(S)$
- **Causal Graph Discovery**: a causal graph $\mathcal{G}_S$ among the causal variables S
- **Causal Mechanism Learning**: the causal mechanisms f_i or $p(S_i|PA_i)$

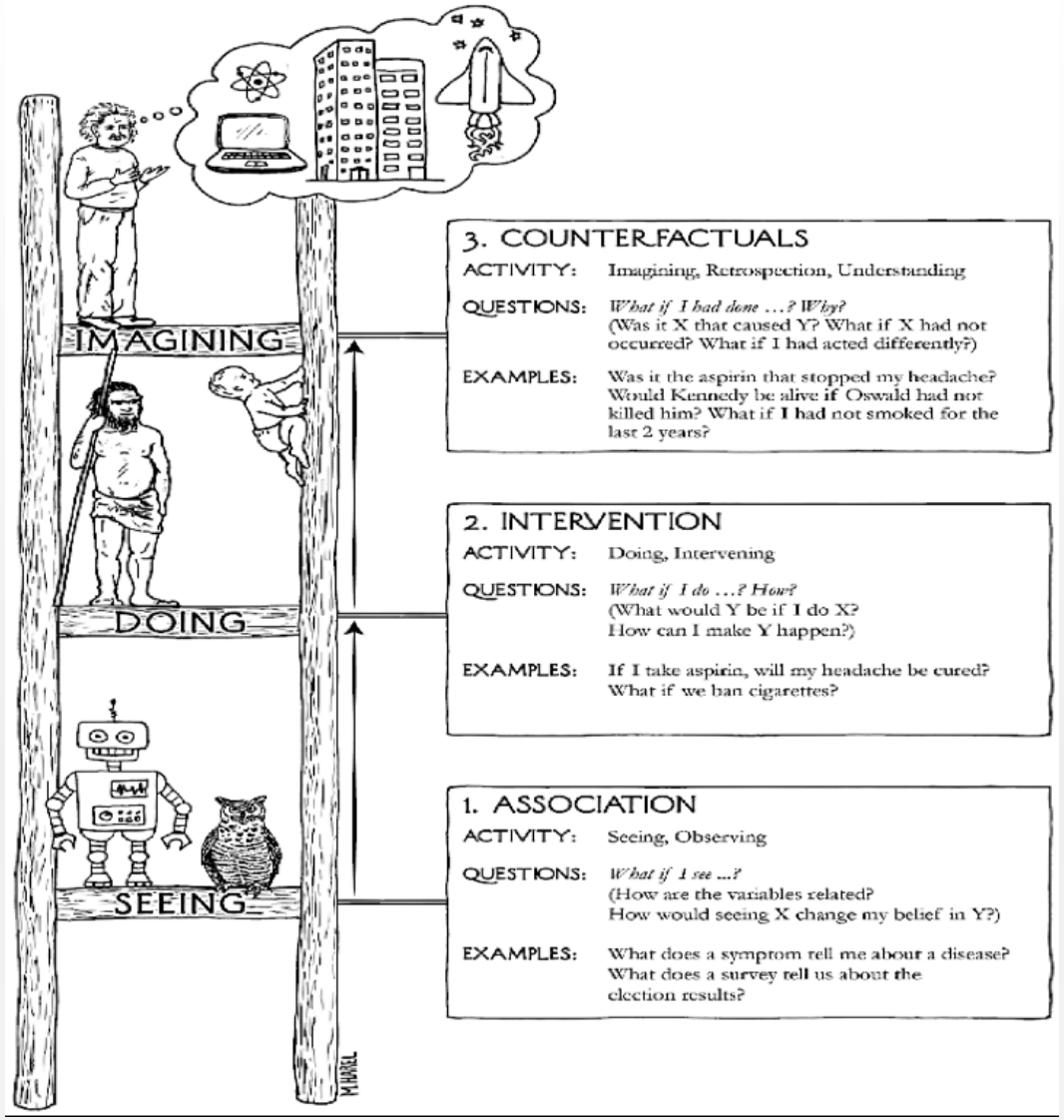


[Credit to Bengio]

https://arxiv.org/pdf/2102.11107.pdf

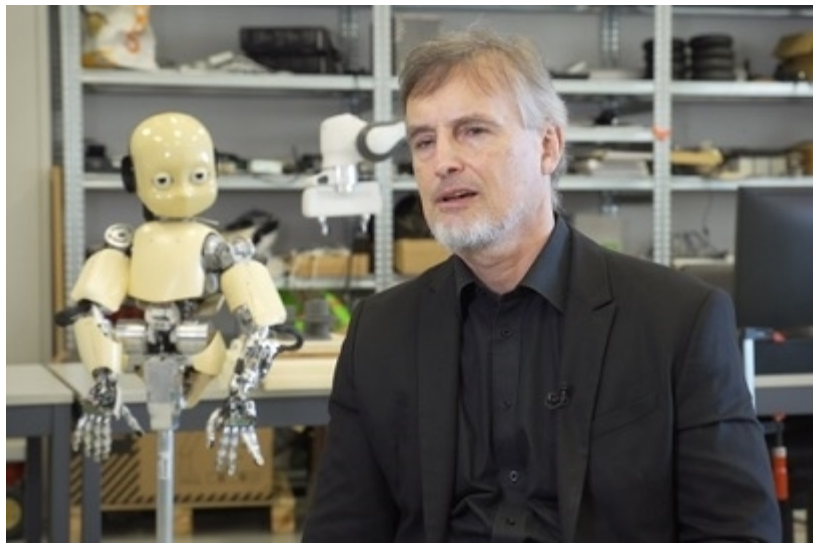
Hierarchy of Causality





Judea Pearl: The Book of Why

World Models



World Models

David Ha [1] Jürgen Schmidhuber [2 3]

Abstract

We explore building generative neural network models of popular reinforcement learning environments. Our *world model* can be trained quickly in an unsupervised manner to learn a compressed spatial and temporal representation of the environment. By using features extracted from the world model as inputs to an agent, we can train a very compact and simple policy that can solve the required task. We can even train our agent entirely inside of its own hallucinated dream generated by its world model, and transfer this policy back into the actual environment.

An interactive version of this paper is available at https://worldmodels.github.io

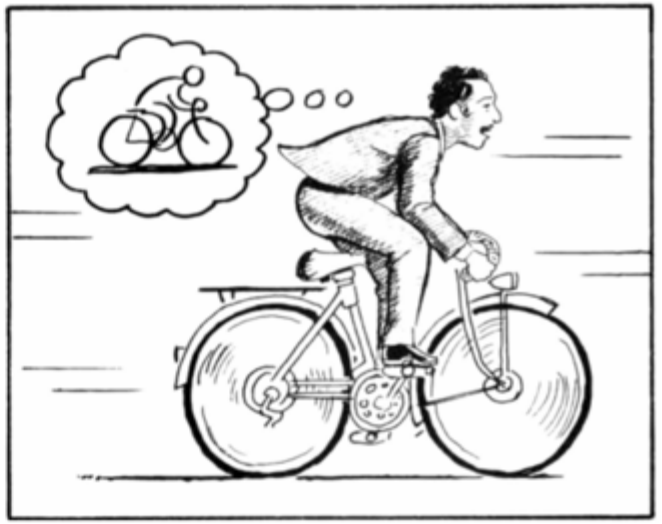


Figure 1. A World Model, from Scott McCloud's *Understanding Comics.* (McCloud, 1993; E, 2012)

https://arxiv.org/pdf/1803.10122.pdf

World Models

At each time step, our agent receives an **observation** from the environment.

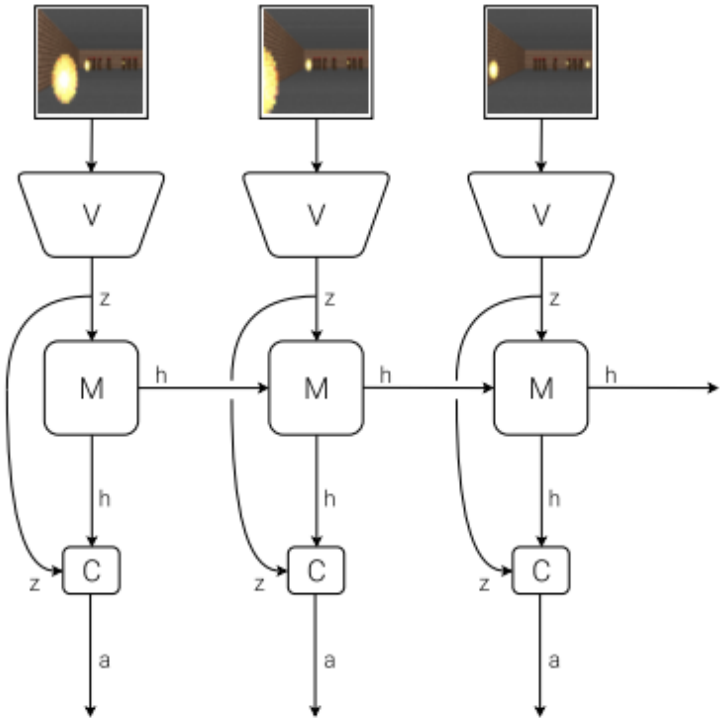
World Model

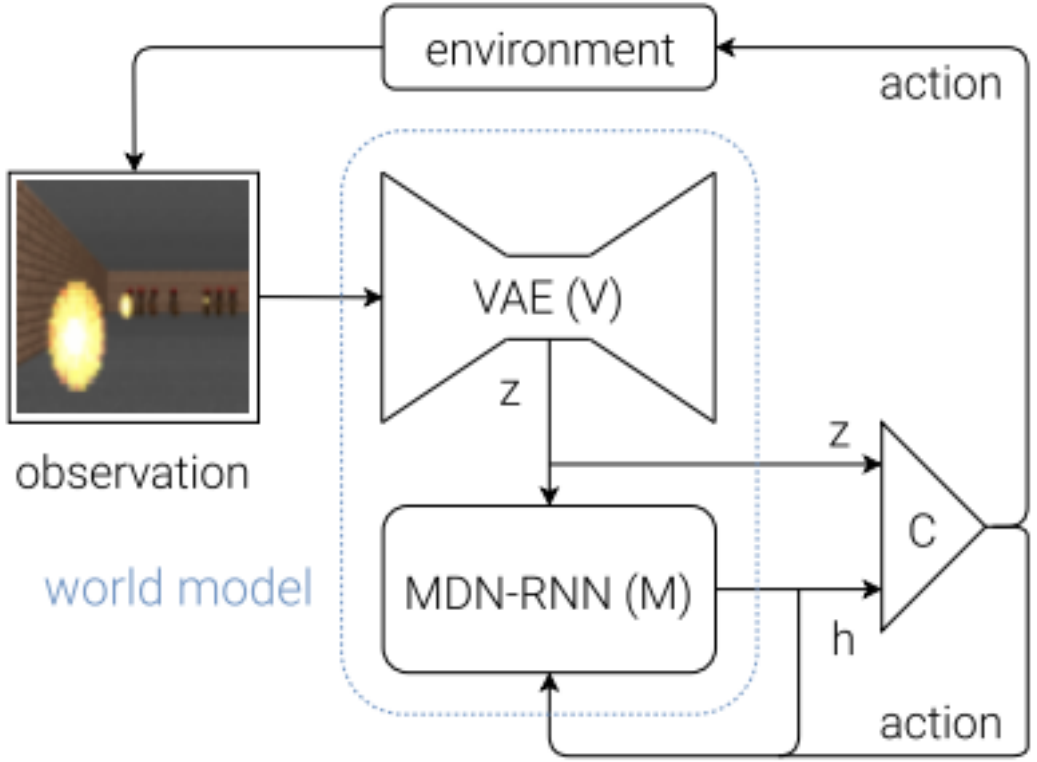
The **Vision Model (V)** encodes the high-dimensional observation into a low-dimensional latent vector.

The **Memory RNN (M)** integrates the historical codes to create a representation that can predict future states.

A small **Controller (C)** uses the representations from both V and M to select good actions.

The agent performs **actions** that go back and affect the environment.





https://arxiv.org/pdf/1803.10122.pdf

World Model – VAE encoder

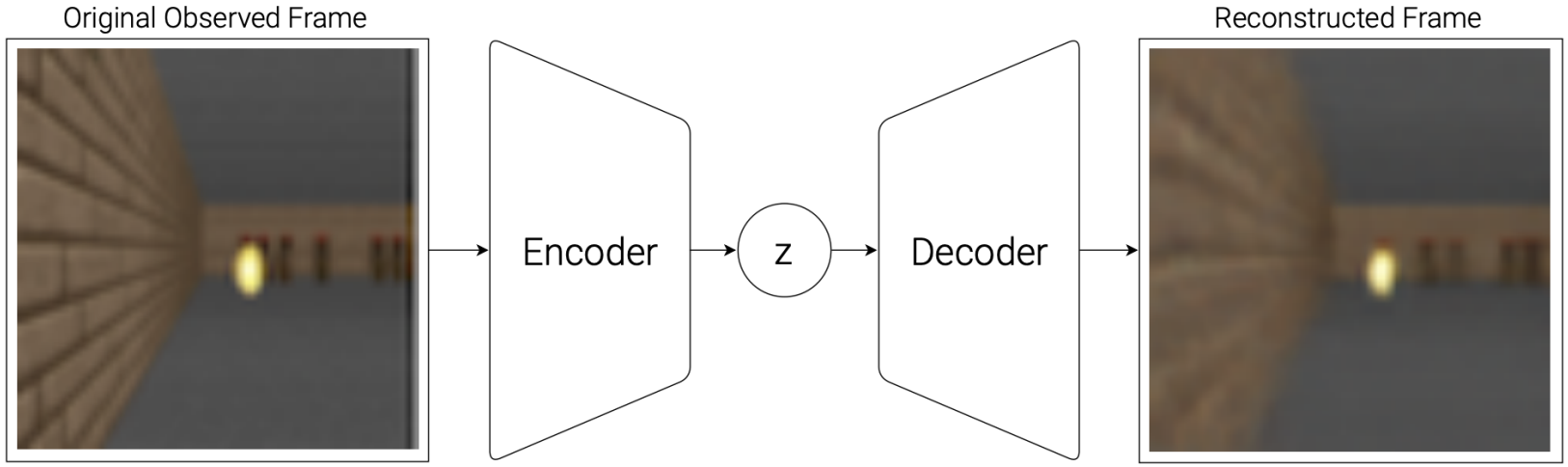
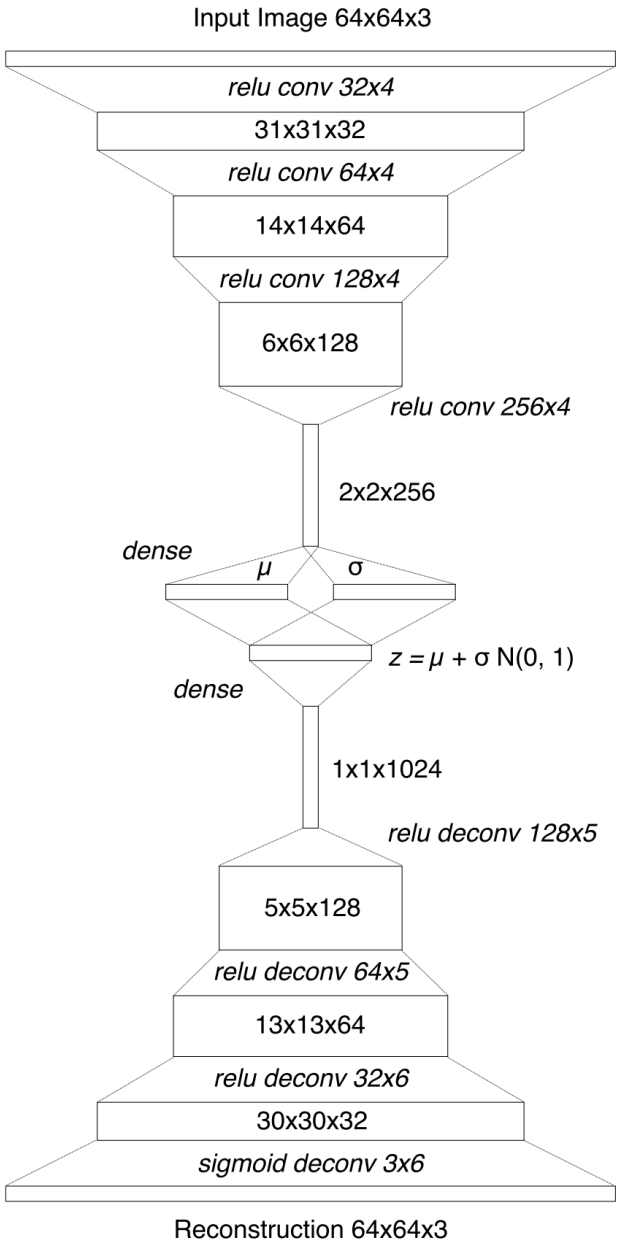
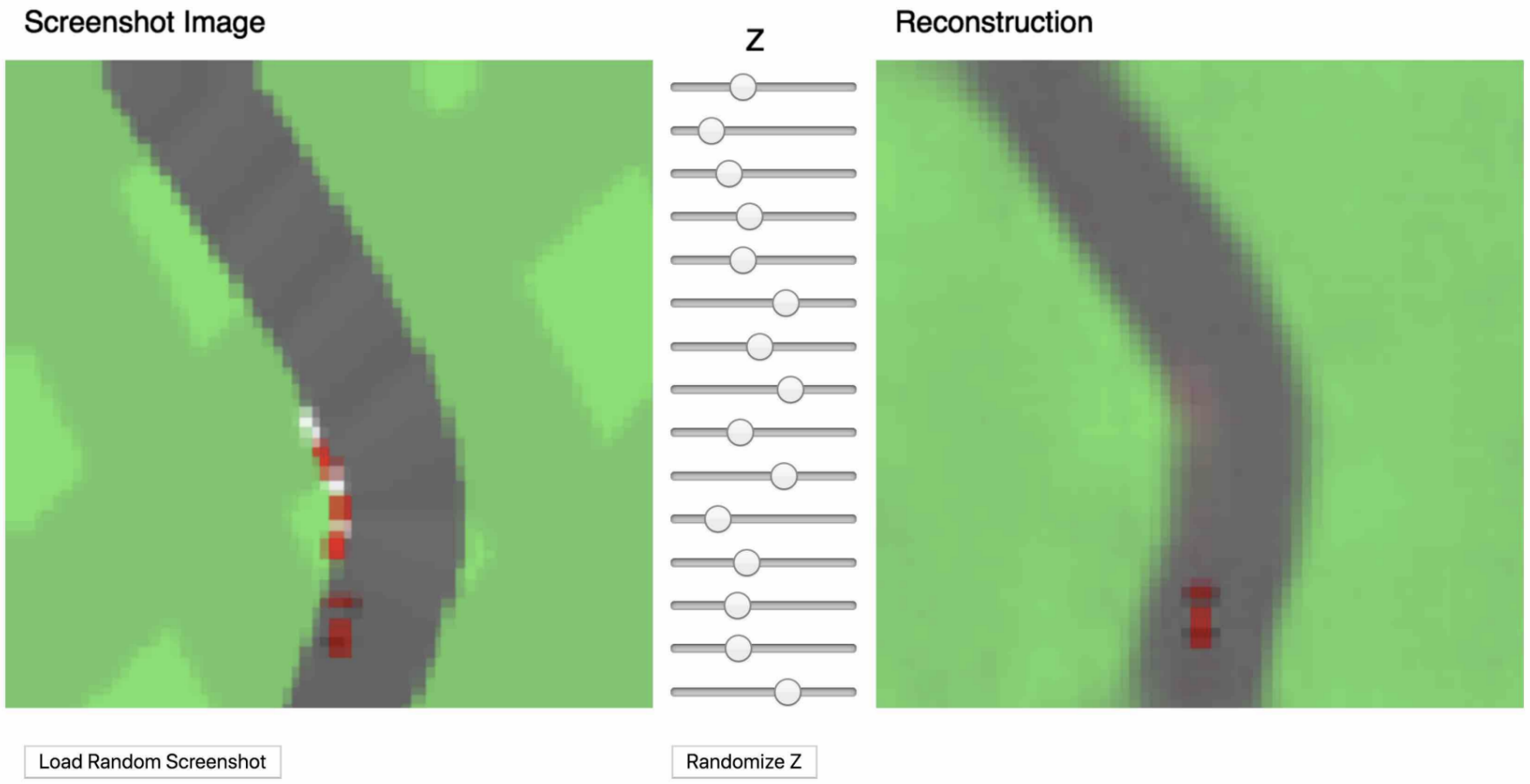


Figure 5. Flow diagram of a Variational Autoencoder (VAE).



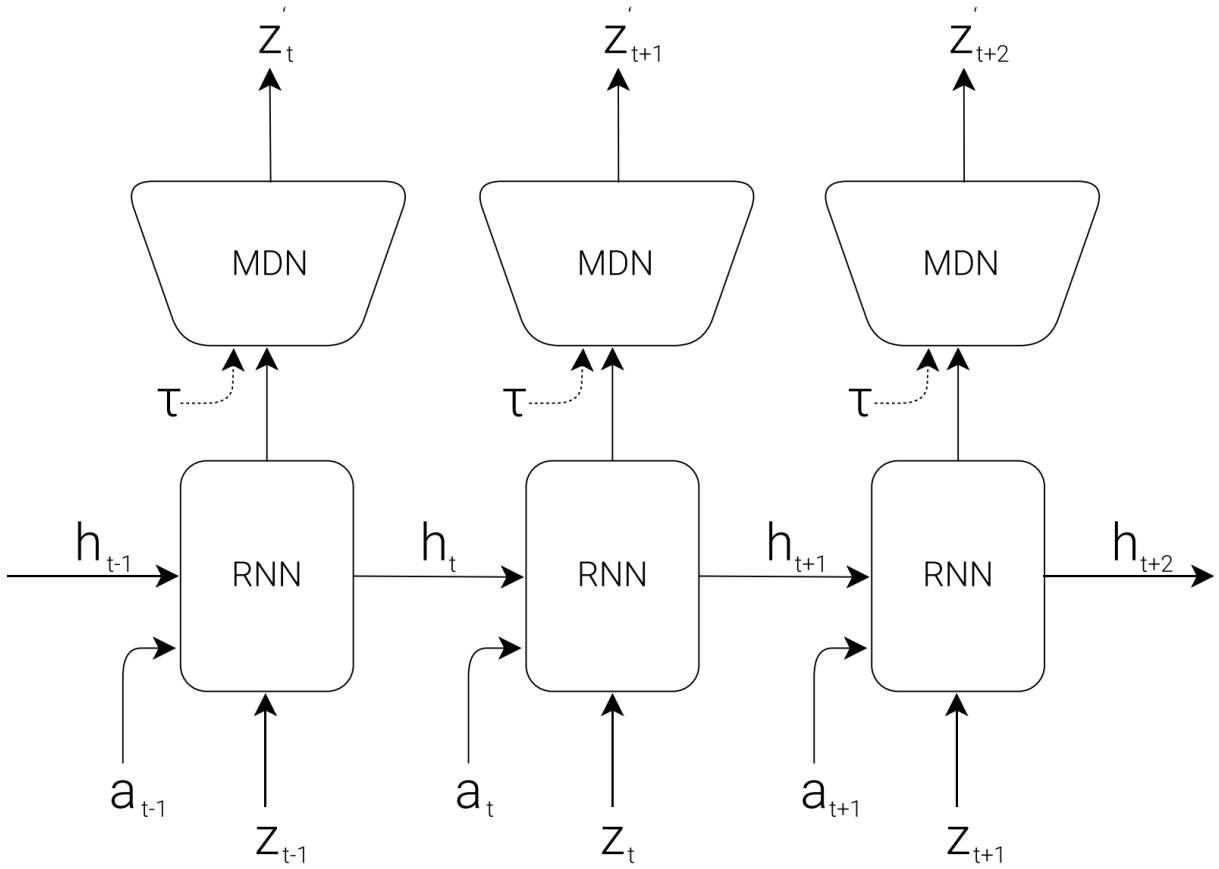
https://arxiv.org/pdf/1803.10122.pdf

World Model – VAE encoder

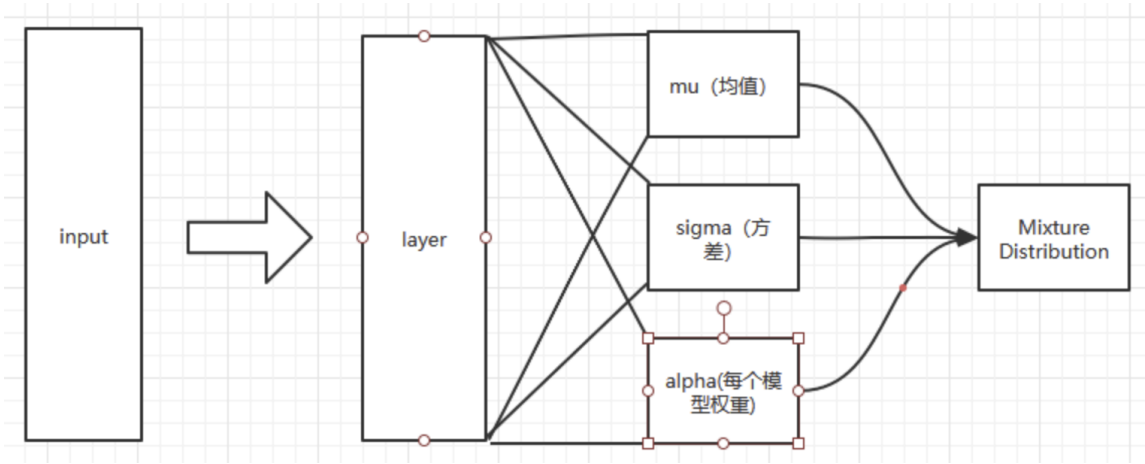


https://arxiv.org/pdf/1803.10122.pdf

World Model – MDP RNN



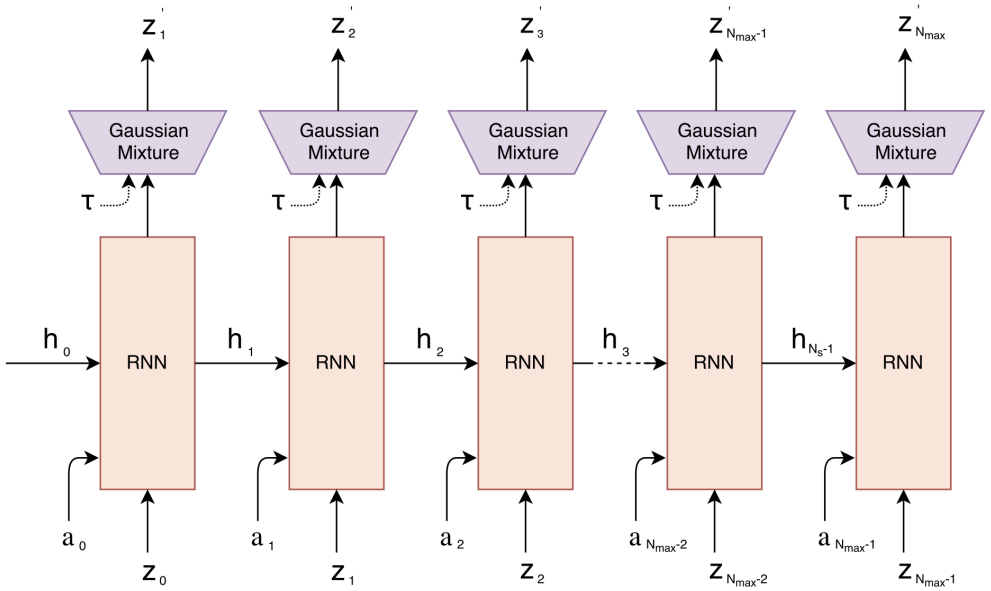
$$p(y|\mathbf{x}) = \sum_{c=1}^{C} \alpha_c(\mathbf{x}) \mathcal{D}\left(y|\lambda_{1,c}(\mathbf{x}), \lambda_{2,c}(\mathbf{x}), ...\right)$$



https://blog.csdn.net/KuXiaoQuShiHuai/article/details/109692063

https://arxiv.org/pdf/1803.10122.pdf

World Model – MDP RNN



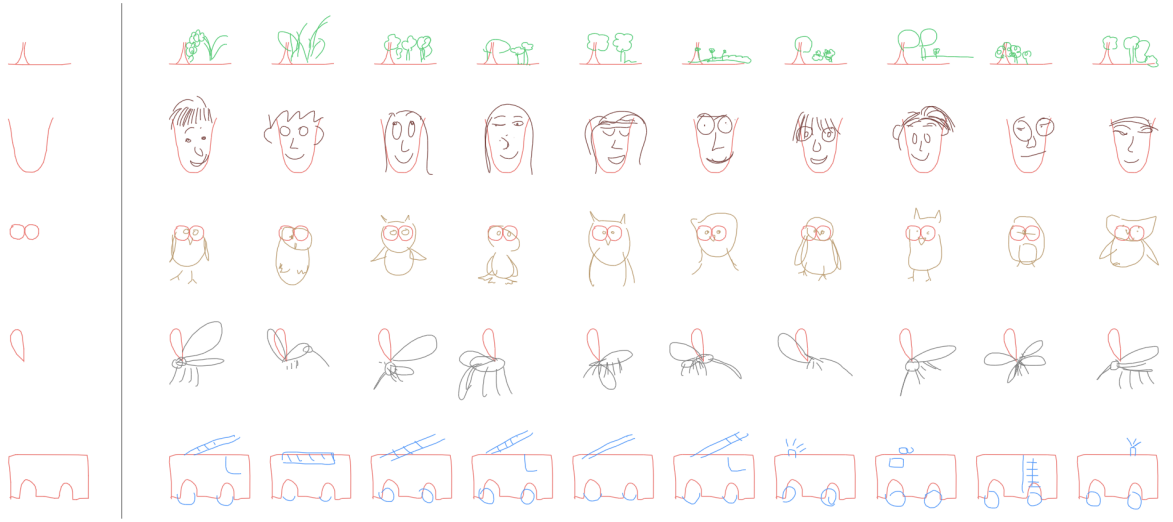
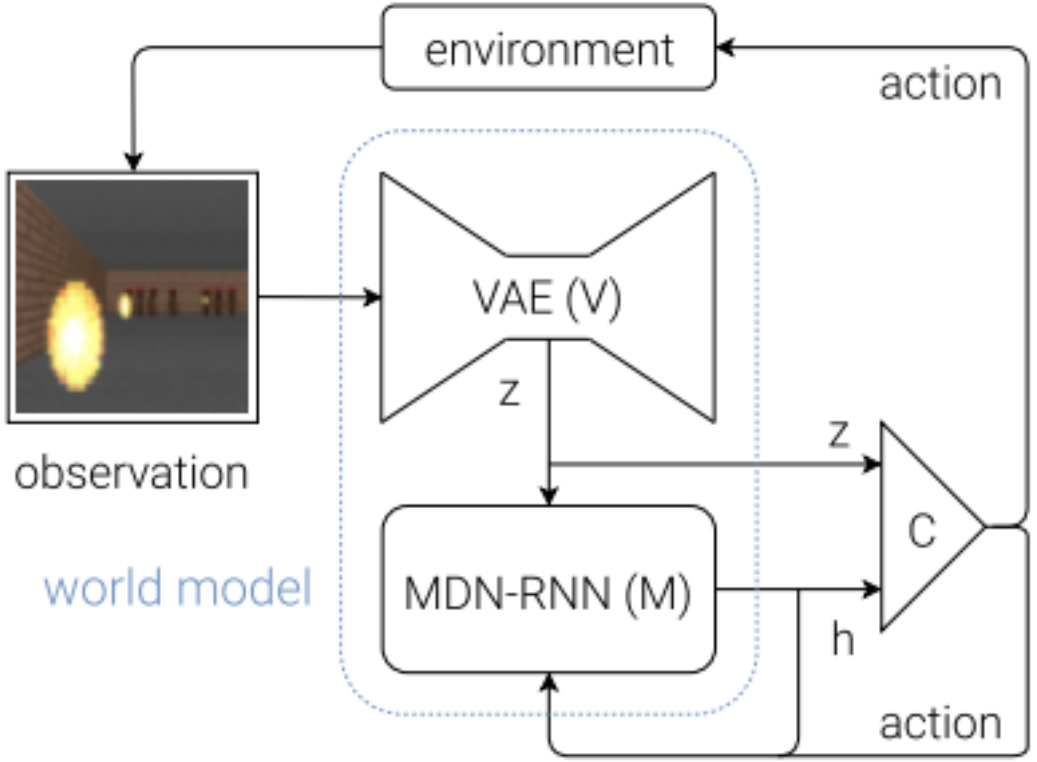


Figure 7. SketchRNN (Ha & Eck, 2017) is an example of a MDN-RNN used to predict the next pen strokes of a sketch drawing. We use a similar model to predict the next latent vector z_t.

https://arxiv.org/pdf/1803.10122.pdf

Controller



$$a_t = W_c \left[z_t \; h_t \right] + b_c$$

We used Covariance-Matrix Adaptation Evolution Strategy (CMA-ES) (Hansen, 2016) to evolve the weights for our C Model. Following the approach described in Evolving Stable Strategies (Ha, 2017b), we used a population size of 64, and had each agent perform the task 16 times with different initial random seeds. The fitness value for the agent is the average cumulative reward of the 16 random rollouts. The diagram below charts the best performer, worst performer, and mean fitness of the population of 64 agents at each generation

https://arxiv.org/pdf/1803.10122.pdf

Car racing experiments

1. Collect 10,000 rollouts from a random policy.
2. Train VAE (V) to encode frames into $z \in \mathcal{R}^{32}$.
3. Train MDN-RNN (M) to model $P(z_{t+1} \mid a_t, z_t, h_t)$.
4. Define Controller (C) as $a_t = W_c \left[z_t \; h_t\right] + b_c$.
5. Use CMA-ES to solve for a W_c and b_c that maximizes the expected cumulative reward.

Model	Parameter Count
VAE	4,348,547
MDN-RNN	422,368
Controller	867

https://arxiv.org/pdf/1803.10122.pdf

Dreaming

METHOD	AVG. SCORE
DQN (PRIEUR, 2017)	343 ± 18
A3C (CONTINUOUS) (JANG ET AL., 2017)	591 ± 45
A3C (DISCRETE) (KHAN & ELIBOL, 2016)	652 ± 10
CEOBILLIONAIRE (GYM LEADERBOARD)	838 ± 11
V MODEL	632 ± 251
V MODEL WITH HIDDEN LAYER	788 ± 141
FULL WORLD MODEL	$\mathbf{906 \pm 21}$

Table 1. `CarRacing-v0` scores achieved using various methods.

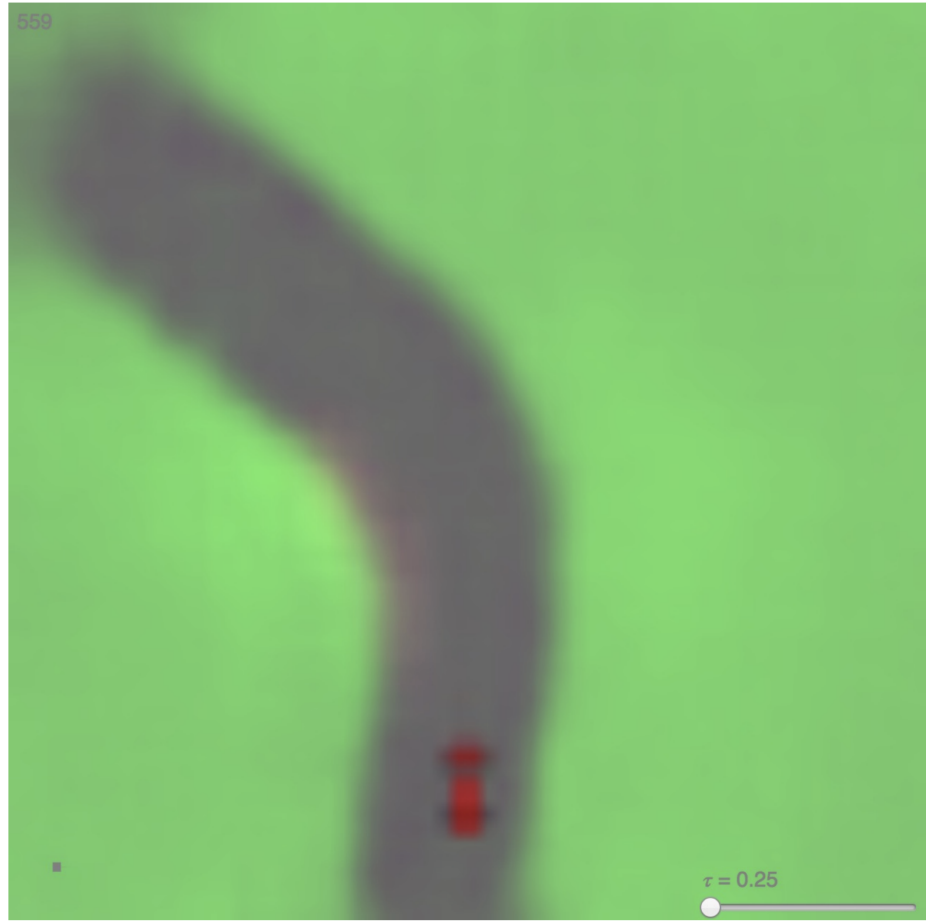


Figure 13. Our agent driving inside of its own dream world. Here, we deploy our trained policy into a fake environment generated by the MDN-RNN, and rendered using the VAE's decoder. In the demo, one can override the agent's actions as well as adjust τ to control the uncertainty of the environment generated by M.

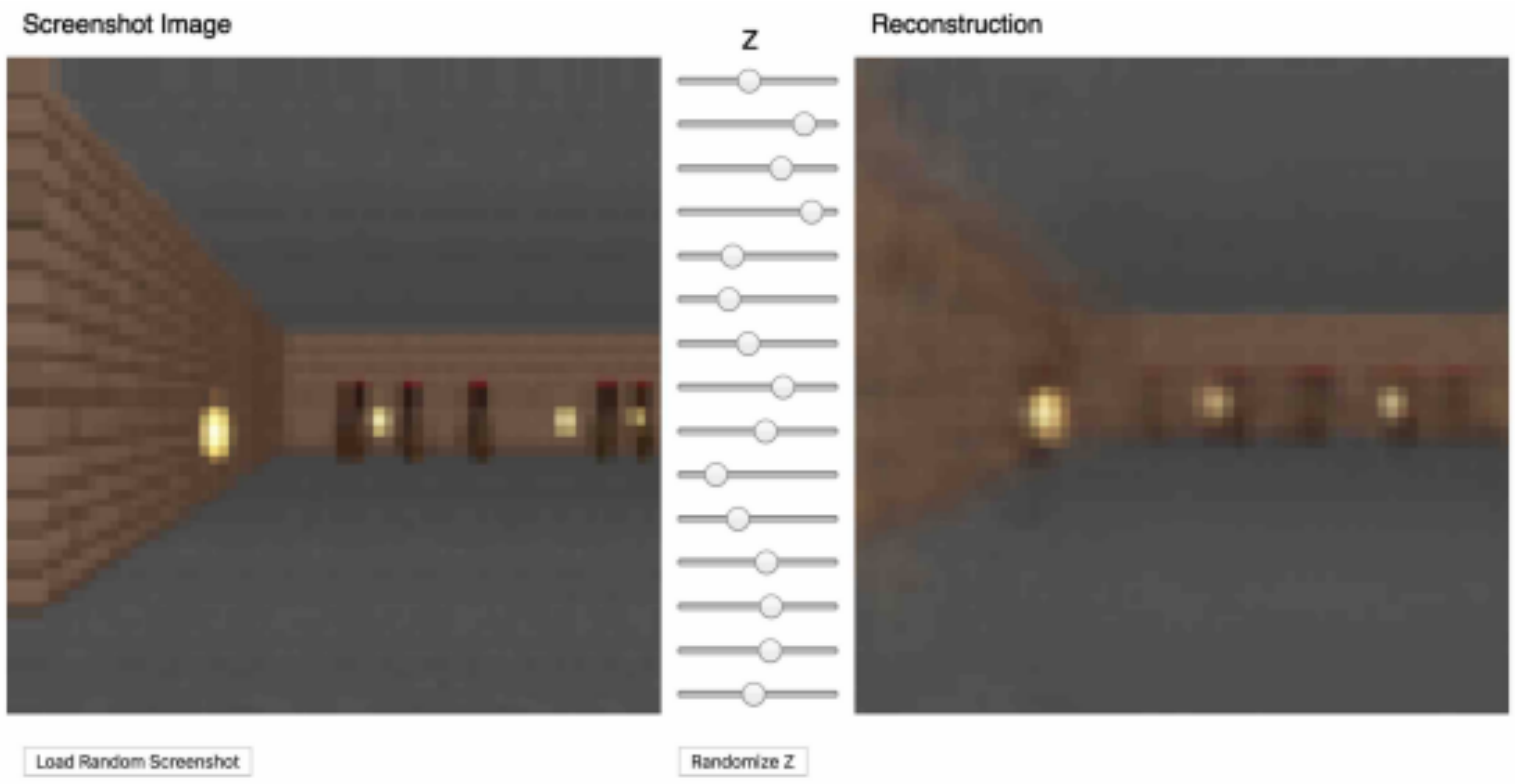
https://arxiv.org/pdf/1803.10122.pdf

Dreaming

1. Collect 10,000 rollouts from a random policy.
2. Train VAE (V) to encode each frame into a latent vector $z \in \mathcal{R}^{64}$, and use V to convert the images collected from (1) into the latent space representation.
3. Train MDN-RNN (M) to model $P(z_{t+1}, d_{t+1} \mid a_t, z_t, h_t)$.
4. Define Controller (C) as $a_t = W_c \, [z_t \; h_t]$.
5. Use CMA-ES to solve for a W_c that maximizes the expected survival time inside the virtual environment.
6. Use learned policy from (5) on actual environment.

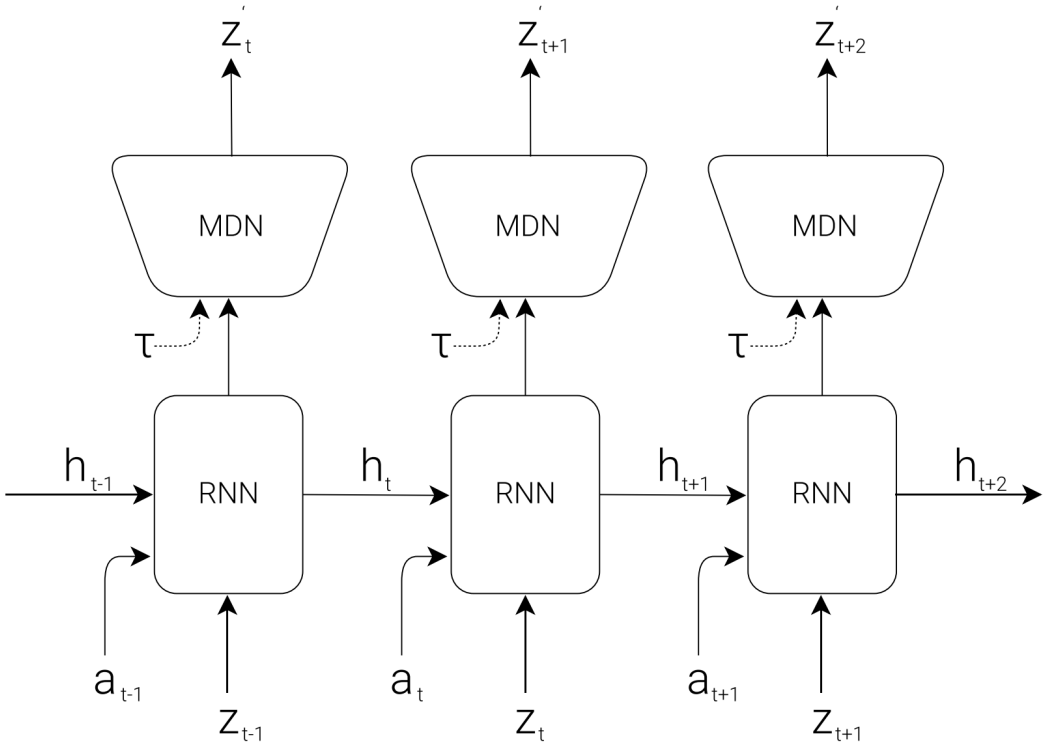
dead

Model	Parameter Count
VAE	4,446,915
MDN-RNN	1,678,785
Controller	1,088



https://arxiv.org/pdf/1803.10122.pdf

About temperature

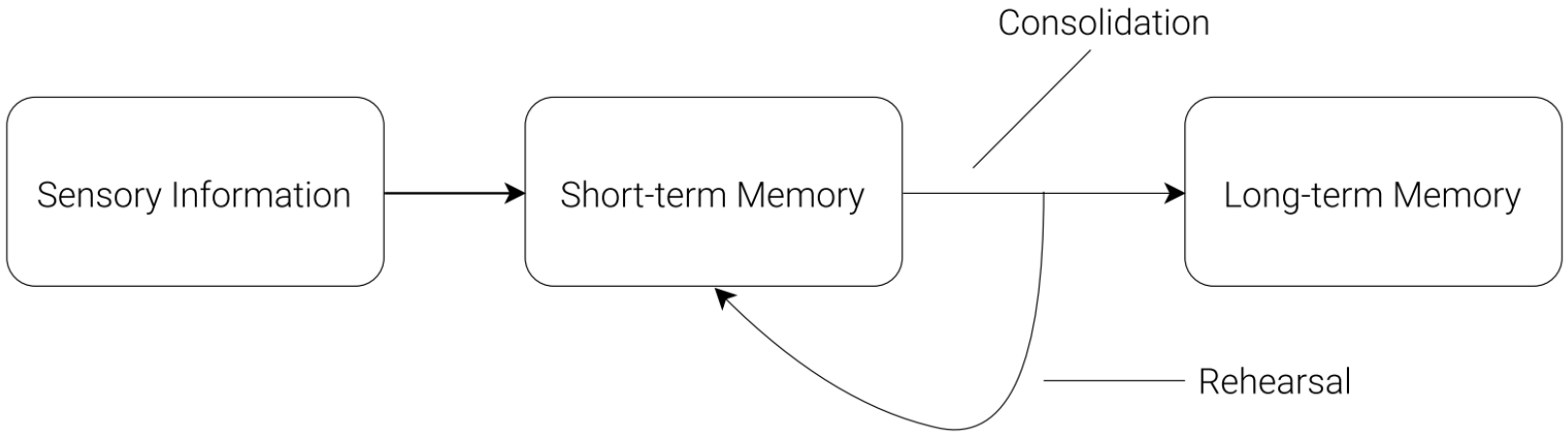


Temperature τ	Virtual Score	Actual Score
0.10	2086 ± 140	193 ± 58
0.50	2060 ± 277	196 ± 50
1.00	1145 ± 690	868 ± 511
1.15	918 ± 546	1092 ± 556
1.30	732 ± 269	753 ± 139
Random Policy	N/A	210 ± 108
Gym Leader	N/A	820 ± 58

Table 2. Take Cover scores at various temperature settings.

https://arxiv.org/pdf/1803.10122.pdf

Iterative Training Procedure



1. Initialize M, C with random model parameters.
2. Rollout to actual environment N times. Save all actions a_t and observations x_t during rollouts to storage.
3. Train M to model $P(x_{t+1}, r_{t+1}, a_{t+1}, d_{t+1}|x_t, a_t, h_t)$ and train C to optimize expected rewards inside of M.
4. Go back to (2) if task has not been completed.

https://arxiv.org/pdf/1803.10122.pdf

Dreamers

Published as a conference paper at ICLR 2020

DREAM TO CONTROL: LEARNING BEHAVIORS BY LATENT IMAGINATION

Danijar Hafner * University of Toronto Google Brain
Timothy Lillicrap DeepMind
Jimmy Ba University of Toronto
Mohammad Norouzi Google Brain

Abstract

Learned world models summarize an agent's experience to facilitate learning complex behaviors. While learning world models from high-dimensional sensory inputs is becoming feasible through deep learning, there are many potential ways for deriving behaviors from them. We present Dreamer, a reinforcement learning agent that solves long-horizon tasks from images purely by latent imagination. We efficiently learn behaviors by propagating analytic gradients of learned state values back through trajectories imagined in the compact state space of a learned world model. On 20 challenging visual control tasks, Dreamer exceeds existing approaches in data-efficiency, computation time, and final performance.

https://arxiv.org/abs/1912.01603

Mastering Diverse Domains through World Models

Danijar Hafner[1,2] Jurgis Pasukonis[1] Jimmy Ba[2] Timothy Lillicrap[1]

[1]DeepMind [2]University of Toronto

Abstract

General intelligence requires solving tasks across many domains. Current reinforcement learning algorithms carry this potential but are held back by the resources and knowledge required to tune them for new tasks. We present DreamerV3, a general and scalable algorithm based on world models that outperforms previous approaches across a wide range of domains with fixed hyperparameters. These domains include continuous and discrete actions, visual and low-dimensional inputs, 2D and 3D worlds, different data budgets, reward frequencies, and reward scales. We observe favorable scaling properties of DreamerV3, with larger models directly translating to higher data-efficiency and final performance. Applied out of the box, DreamerV3 is the first algorithm to collect diamonds in Minecraft from scratch without human data or curricula, a long-standing challenge in artificial intelligence. Our general algorithm makes reinforcement learning broadly applicable and allows scaling to hard decision making problems.

https://arxiv.org/abs/2301.04104

Dreamers

Published as a conference paper at ICLR 2020

DREAM TO CONTROL: LEARNING BEHAVIORS BY LATENT IMAGINATION

Danijar Hafner *
University of Toronto
Google Brain

Timothy Lillicrap
DeepMind

Jimmy Ba
University of Toronto

Mohammad Norouzi
Google Brain

Abstract

Learned world models summarize an agent's experience to facilitate learning complex behaviors. While learning world models from high-dimensional sensory inputs is becoming feasible through deep learning, there are many potential ways for deriving behaviors from them. We present Dreamer, a reinforcement learning agent that solves long-horizon tasks from images purely by latent imagination. We efficiently learn behaviors by propagating analytic gradients of learned state values back through trajectories imagined in the compact state space of a learned world model. On 20 challenging visual control tasks, Dreamer exceeds existing approaches in data-efficiency, computation time, and final performance.

Mastering Diverse Domains through World Models

Danijar Hafner,[12] Jurgis Pasukonis,[1] Jimmy Ba,[2] Timothy Lillicrap[1]

[1]DeepMind [2]University of Toronto

Abstract

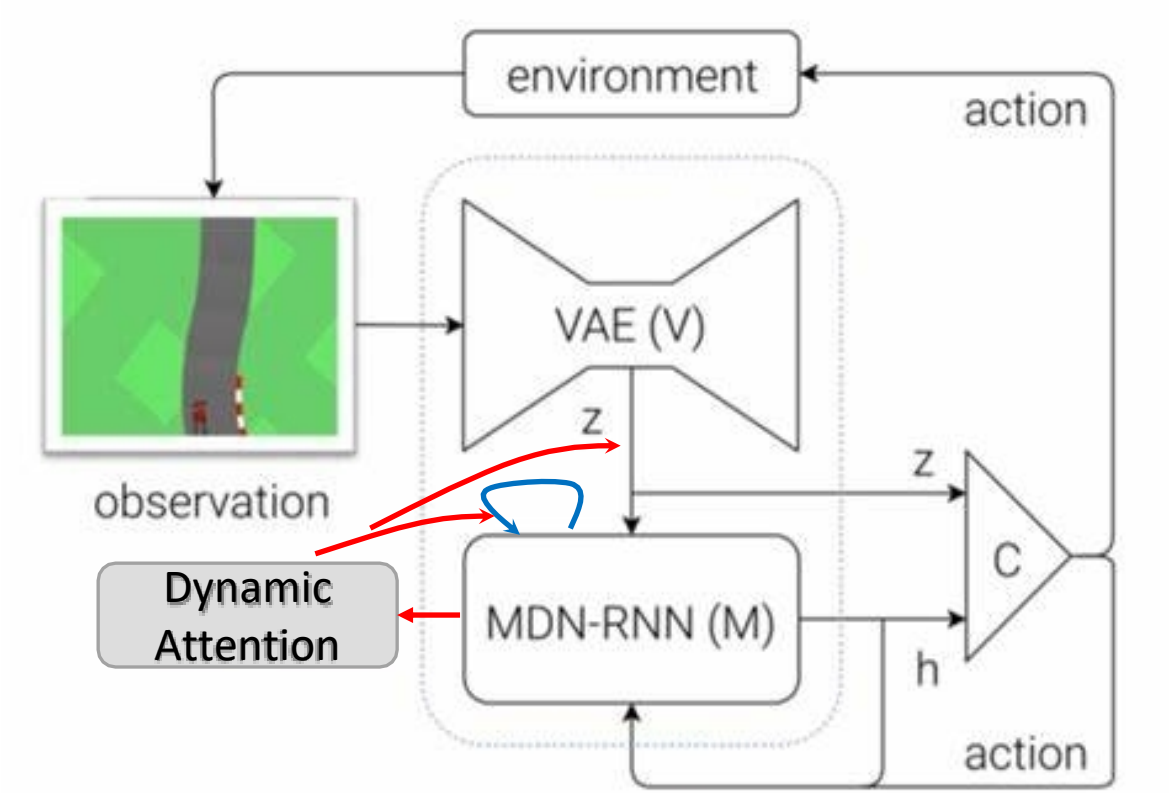
General intelligence requires solving tasks across many domains. Current reinforcement learning algorithms carry this potential but are held back by the resources and knowledge required to tune them for new tasks. We present DreamerV3, a general and scalable algorithm based on world models that outperforms previous approaches across a wide range of domains with fixed hyperparameters. These domains include continuous and discrete actions, visual and low-dimensional inputs, 2D and 3D worlds, different data budgets, reward frequencies, and reward scales. We observe favorable scaling properties of DreamerV3, with larger models directly translating to higher data-efficiency and final performance. Applied out of the box, DreamerV3 is the first algorithm to collect diamonds in Minecraft from scratch without human data or curricula, a long-standing challenge in artificial intelligence. Our general algorithm makes reinforcement learning broadly applicable and allows scaling to hard decision making problems.

Problems:

- World without self
- Dreaming is not autonomous

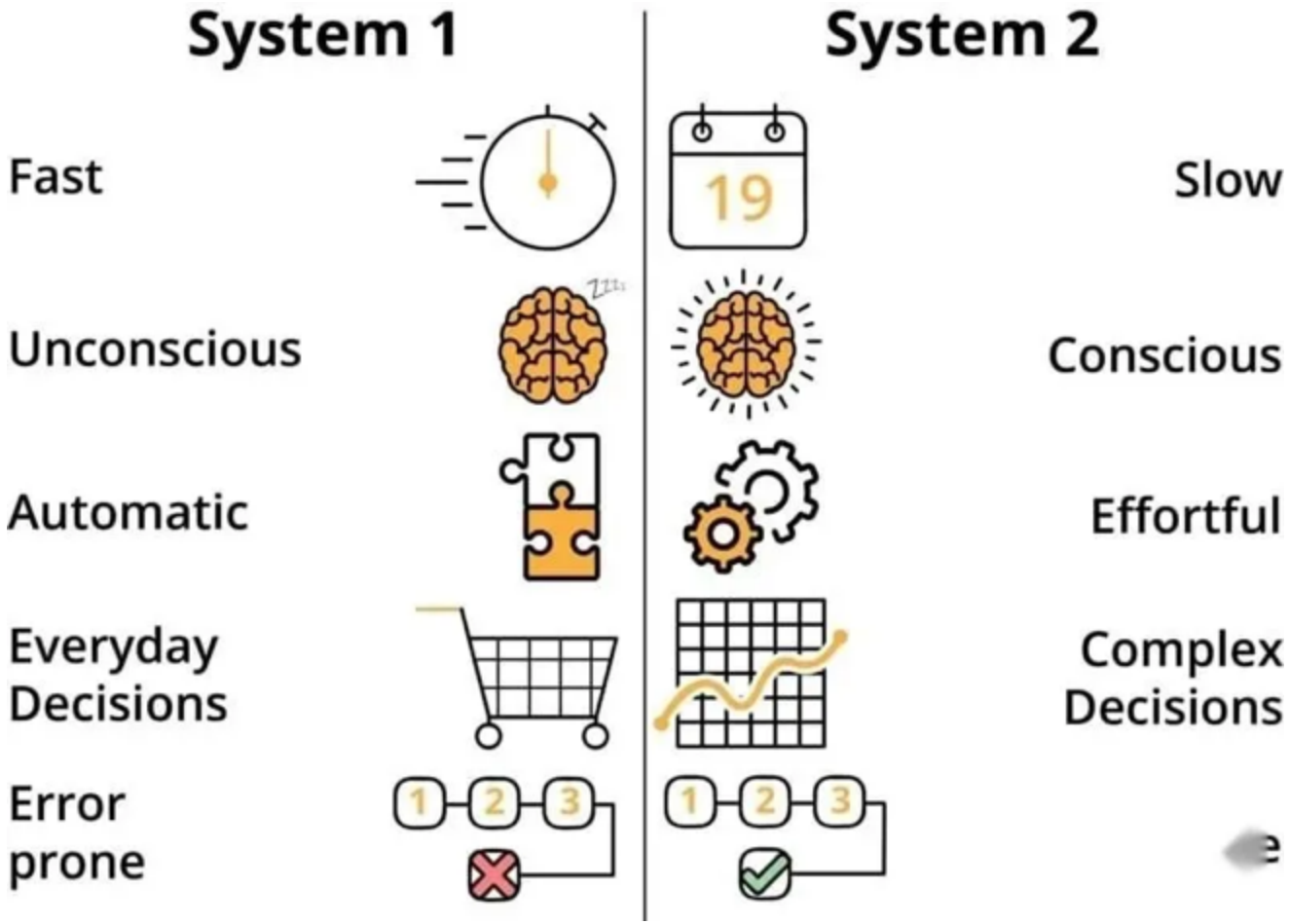
https://arxiv.org/abs/2301.04104

Self awareness



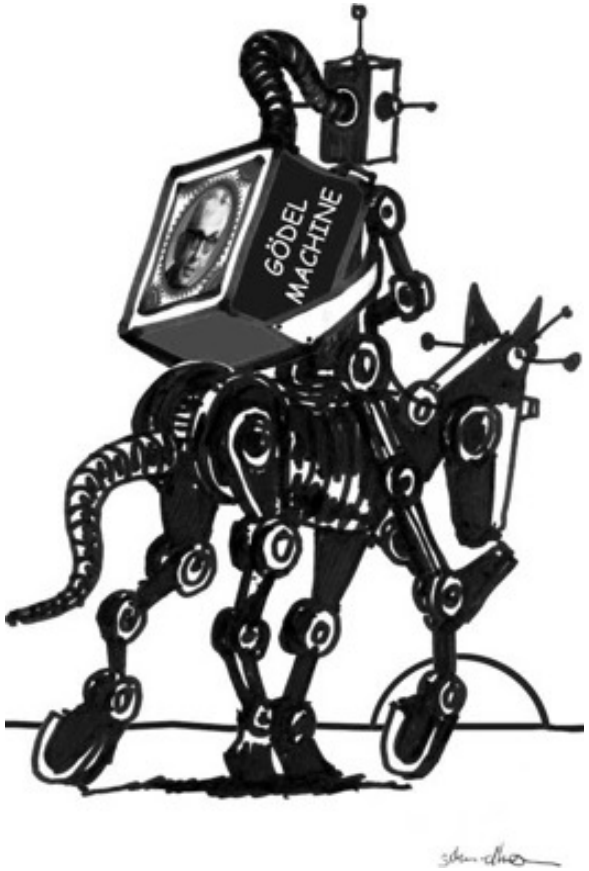
https://arxiv.org/pdf/2204.07184.pdf

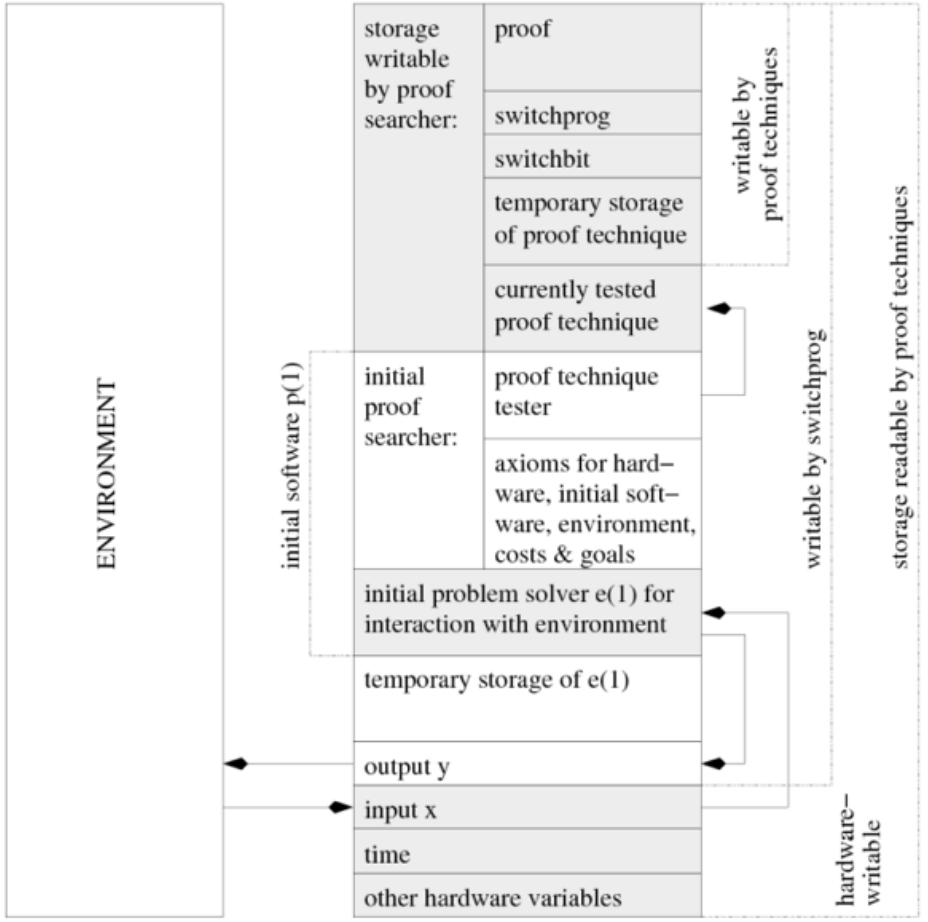
From System 1 to System 2



卡尼曼：思考，快与慢，中信出版社

Consciousness Machine?





https://arxiv.org/pdf/cs/0309048.pdf

Outline

- AI history and Introduction
- Automated Differential & PyTorch
- Machine Learning Basics
- Neural Networks I
- Neural Networks II
- Neural Ordinary Differential Equations
- Representation Learning and Transfer Learning
- Generative Models I
- Generative Models II
- Transformers and ChatGPT
- Graph Neural Networks
- Automated Modelling of Complex Systems
- Causal Inference
- Causal Machine Learning
- Reinforcement Learning