北京大学人工智能研究院
INSTITUTE FOR ARTIFICIAL INTELLIGENCE, PEKING UNIVERSITY

PKU-IAI Technical Report: PKU-IAI-2023-0003

OpenLock Grid: Learning Causal Structure with Deep RL Models

Yutong Yin
Yuanpei College
Peking University
ytyin@pku.edu.cn

Xingbo Wang
Yuanpei College
Peking University
jacksimbol@stu.pku.edu.cn

Ziyin Xiong
Yuanpei College
Peking University
xiongziyin@stu.pku.edu.cn

Abstract

Causal reasoning is considered by many to be the cornerstone of human intelligence. Humans have the ability to develop causal schemas to solve novel problems and establish explanations of latent constraints while interacting with the environment. In this paper, we explore and examine agents' ability to discover causal schemas and generalize to more complex environments based on the OpenLock task, in which participants are required to escape a room by moving levers, and the sequential movements of the correct levers form a causal sequence beginning with either a common-cause (CC) or a common-effect (CE) structure. We develop a deep Q-learning model to solve a simplified OpenLock problem by learning the transition probability between solutions. We argue that the simplification is reasonable since the original task is a complex combination of several diverse sub-tasks and it's difficult to solve all sub-tasks in one single model. The simplified task is equivalent to the original one on the ability to examine agents' causal transferring capability, as argued in Sec. 1. We focus on the agents' exploring the latent causal structure of levers, which is challenging and meaningful enough to show the potential of RL in causal reasoning. Experiments show that our method learns the underlying causal mechanisms perfectly and pertains to a great generalization ability that can transfer the knowledge trained in previous rooms to new rooms, and in simple conditions to more complex tasks.

1 Introduction

Causality is the abstract notion of cause and effect derived from our perceived environment and thus can be used as a prior foundation to construct notions of time and space (Robb [4]). It is the foundation of the other four cognitive elements (functionality, physics, intent, and utility). To a certain degree, much of human understanding depends on the ability to comprehend causality. Without understanding what causes an action, it is very difficult to consider what may happen next and respond effectively. (Zhu et al. [7])

An interesting research question in the field of causal learning is how various intelligent systems (ranging from rats, and humans to artificial intelligent agents) can acquire knowledge about cause-effect relations in novel situations. Years ago, a number of researchers suggested that causal knowledge can be acquired by a basic learning mechanism, associative learning, that non-human animals commonly employ in classical conditioning paradigms to learn the relationship between stimuli and responses.

Some models of human causal learning assume the hypothesis space of causal variables and structures is given and that inference focuses on selecting a causal structure to explain the observed contingency information relating causal cues to effects. In situations where outcomes depend on the learner's actions rather than simply observations, reinforcement learning (RL) is a widely-used modeling tool. RL focuses on learning what to do by mapping situations to actions to maximize a reward signal. RL has historically been closely linked with associative learning theory and conceives learning as a process of trial and error. Then it's essential to answer the question: with the significant developments in RL, is it possible for modern learning models to acquire human-like causal knowledge?

Our contribution is as follows. In this work, we focus on the exploration of RL in an equivalent simplified setting of the OpenLock task in [1]. It's worth mentioning that our simplification of the task is reasonable. Our reasons are as follows. From the perspective of cognitive science, the OpenLock task itself is very complex. Humans need to divide the task into several modules (for example, understanding that manipulating these levers might change the state of the lock, understanding that we need to press an additional button to finally unlock the door) to solve one by one, and these sub-tasks are diverse. Hence, trying to use one model to solve the compound task of multiple modules at the same time is difficult to succeed. Our simplification essentially ignores several sub-tasks that are not necessary compared to the most important causal structure inference. We assume that the agent already knows that manipulating the lever can change the state of the lock. We focus on exploring the correct levers and inferring the causal structure of unlocking, which make the simplified task still challenging and meaningful in digging into the potential of RL in causal reasoning.

We use a grid to represent the simplified levers. The specific settings will be detailed in the fourth section. Each row or column of the grid represents a fixed-numbered lever. Every time the scene is changed, randomly select levers to form a CC or CE structure, and the value is assigned to the grid accordingly. The shape of the rows and columns is fixed, corresponding to the constant visual observation in the original task; the causal relationship of the movable levers is changed, corresponding to the changing causal environment. Therefore, we believe that task simplification preserves OpenLock's ability to detect agent causal transfer capabilities. Besides, we have also found the excellent generalization ability of RL through this task, which has not been mentioned in the original work [1].

When considering the level of cognitive reasoning of agents reflected in this task, we focus on the ability of agents, trained in the 'three movable levers' task, to adapt its door-opening policies to tasks with more movable levers. We will first introduce the original task design of OpenLock and the results of human experiments in Sec. 3. We will then perform a homogeneous replacement and simplification of the task's environment and illustrate the equivalence of the original task and our designed replacement task in Sec. 4. Next, we will introduce the learning details of the models and algorithms we used in Sec. 5, and in Sec. 6 we'll show our experiments, results, and analysis.

2 Related Work

2.1 Causal Transfer

Zhu et al. [7] argued that learning causal concepts is of great significance to agents that are expected to operate in observationally varying domains with common latent dynamics. Much of our world is designed by other humans and largely adheres to common causal concepts. Even though objects in different settings appear different, their causal effect is constant because they all fit and cohere to a consistent causal design. Thus, for agents expected to work in varying but human-designed environments, the ability to learn generalizable and transferable causal understanding is crucial.

2.2 Causal Structures in OpenLock

Edmonds et al. [1] designed a novel task called OpenLock to examine the learning of action sequences governed by different causal structures, allowing them to determine in what situations humans can transfer their learned causal knowledge. Their design involves two types of basic causal structures (common cause (CC) and common effect (CE); see Fig. 1). When multiple causal chains are consolidated into a single structure, they can form either CC or CE schemas. As shown in Fig. 1, Common Cause means the agent is required to push lever L0 first. Only after lever L0 is pushed could levers L1 and L2 become useful levers. Thus, L0 is the common factor that causes L1 and L2 to be effective at the door opening. Common Effect means the agent is required to push levers L1 and L2 first. Lever L0 is effective for door opening only after L1 and L2 are pushed. Thus, L1 and L2 are common causes of L0 and L0 is their common effect. Edmonds et al. [1] developed a virtual "escape room", and its underlying theory is referred to as a causal schema, a conceptual organization of events identified as cause and effect. A detailed description of this task is presented in Sec. 3.

2.3 Reinforcement Learning

As discussed in [6], RL algorithms are most commonly classified into two categories: model-free RL (MFRL), which directly learns a value function or a policy by interacting with the environment, and model-based RL (MBRL), which uses interactions with the environment to learn a model of it. While model-free algorithms have achieved success in areas including robotics, video games, and motion animation, their high sample complexity limits largely their application to simulated domains. By learning a model of the environment, model-based methods learn with significantly lower sample complexity.

Model-free RL Q-learning, a representative model-free RL technique, seeks to learn an action-value function using expected discounted rewards. The optimal Q function is defined as follows:

$$Q^{\star}(s,a) = max_{\pi}\mathbf{E}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots | s_t = s, a_t = a, \pi],$$

where s_t is the state at time t, a_t the action, $\pi = P(a|s)$ agent's policy, $\gamma \in [0,1]$ a discount factor, and r_t the reward.

DQN uses experience replay to mitigate networks from over-fitting to recent correlations in the observation sequence. DQN also introduced a target network that is only updated every τ steps to further mitigate over-fitting. Double DQN(DDQN) expands on DQN by reducing over-estimations of the Q function. While DQN uses a single value estimator to both select and evaluate a particular action, DDQN decouples selection and evaluation by learning two value estimators, one for selection and another for evaluation. In [1], they used the state-of-art model-free algorithm DDQN as the computational model and proved that DDQN performed badly on the OpenLock task.

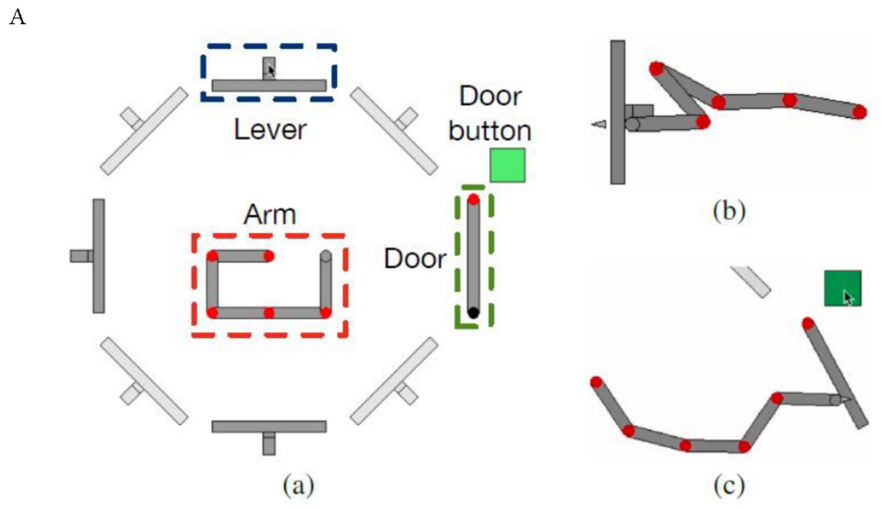
3 OpenLock Task

3.1 Task Setting

The OpenLock game is a virtual "escape room" environment proposed in [1]. In the OpenLock task, participants are asked to escape a virtual room by moving the levers that serve as "locks" to open the door (Fig.1). Each situation consists of seven levers and a door that begins in a locked state. The sequence of lever movements that open the door forms a branching causal sequence with either a common effect (CE) or a common cause (CC) structure (Fig.1). And the task is to figure out which lever mechanisms can open the door. Participants are given 30 trials to discover all the solutions in a situation. In each trial, the participants are allowed three operations to (1) interact with the levers (two operations) or (2) push open the door (one operation). The levers pertinent to the locking mechanism are highlighted in advance. If the participant pushes the lever in an incorrect order, the lever will remain stationary and the operation will be expended.

3.2 Human Results

Researchers from UCLA recruited 240 undergraduates to participate in the experiment. In the baseline conditions participants only completed the CC4 and CE4 trials. A clear learning effect was detected as fewer attempts were needed for later trials (Fig.2a). In the transfer group, participants were trained



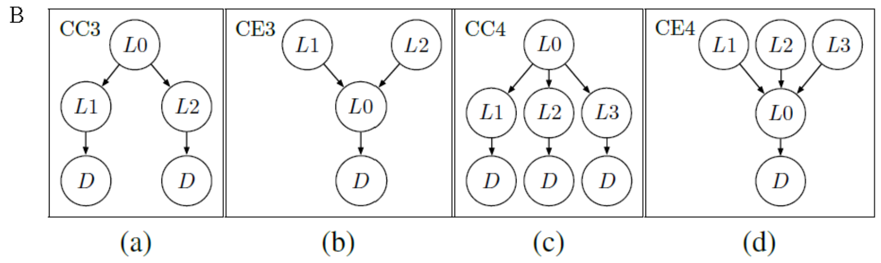
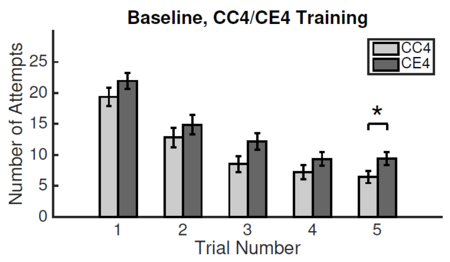
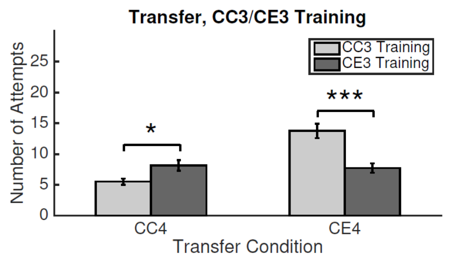


Figure 1: Reproduced from Ref.[1]. **A**. Examples of OpenLock task setting. (a) An example of the 3-lever trial. Active levers are marked in light gray. The black circle located opposite the door's red hinge represents the door lock indicator: present if locked, absent if unlocked. (b) Push to open a lever by the robot arm. (c) Push the door by clicking the green button. **B**. Underlying causal mechanisms. Common cause (CC) and common effect (CE) structures. (a) CC3 condition, 3 active levers. (b) CE3 condition, 3 active levers. (c) CC4 condition, 4 active levers. (d) CE4 condition, 4 active levers.



(a) Average number of attempts needed to find all unique solutions in the 4-lever common cause (CC4) and common effect (CE4) baseline conditions.



(b) Average number of attempts needed to find all unique solutions in 4-lever (CC4 and CE4) conditions with training in 3-lever conditions.

Figure 2: Results from the experiments in [1].

under either CE3 or CC3 structures before they performed CE4 and CC4 trials. The resulting plot shows that participants trained under a CC3 structure performed better in the CC4 condition than those trained under a CE3 causal structure. Similarly, participants trained under a CE3 structure performed better in the CE4 test trials than those trained under a CC3 structure (Fig.2b).

4 Problem Analysis and Simplification

In this section, we seek to simplify the problem into a more concise setting. We first introduce the OpenLock Grid Game, the simplified version of OpenLock. Then we explain that this simplification is necessary for people's understanding about Reinforcement Learning's ability in deducing the causal structure. Also, we indicate that the simplified version is still a challenging task and solving it perfectly means a lot.

4.1 OpenLock Grid Game

We simplify the original OpenLock task into the OpenLock Grid Game in order to model and learn the causal mechanisms in a more concise form. Generally, the solution space of an OpenLock task with n active levers can be mapped to an $n \times n$ matrix, wherein (i, j) represents the sequence of pushing the lever i, pushing the lever j and then pushing the door. The sequences that open the door are marked 1 in the matrix, and the wrong sequences are marked -1. And the model can move strategically in the grid to find all the solutions. For example, a 3-lever task can be simplified to a 3×3 Grid (Fig.3), and the positions $(1, 2)$ and $(1, 3)$ marked 1 indicate that the correct sequences are $L1 - L2$ and $L1 - L3$. We make this simplification because the original setting of OpenLock tasks is too complex, involving observation of the environment and interactions with the levers and locks. For a human participant, completing the original task may engage different brain regions, and for a model the complex settings will make it confused. What we want to do is focus on the ability to learn causal mechanism, thus it is necessary to simplify the problem setting. Note that the simplified task is the same challenging as the original task, as the agent does not know whether a lever can be pushed twice, nor does it know anything about the underlying CC or CE structures. Also the simplified task shows the generalization ability of our method more intuitively.

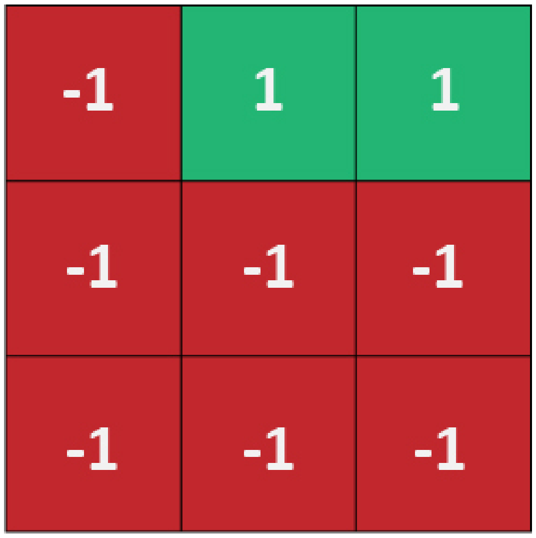


Figure 3: An example of OpenLock Grid Game. In the CC3 condition, correct solutions $(1, 2)$ and $(1, 3)$ are marked 1. The other positions are marked -1. Note that 3×3 grids stand for 3-lever tasks, not 9-lever tasks.

5 Solving the OpenLock Grid Game by Reinforcement Learning

5.1 The Markov Decision Process

For an OpenLock Grid environment with n active levers, the state space is denoted as $\mathcal{S} = \{n \times n$ matrix whose elements are one of $\{-1, 0, 1\}\}$ and the action space is denoted as $\mathcal{A} = \{e_{ij} | i, j \in \{1, 2, \cdots, n\}\}$ where e_{ij} represents a one-hot matrix whose entry (i, j) is 1 and all other entries are 0. $\forall a \in \mathcal{A}$, $a = e_{ij}$ means that the agent chooses the sequence of pushing the lever i, pushing the lever j, and then pushing the door, and the action space both consist of $n \times n$ dimensions. $\forall s \in \mathcal{S}$; $s_{ij} = 0$ means that the agent has not taken the action e_{ij}, $s_{ij} = 1$ means that the agent has taken the action e_{ij} and found that it is one of the solutions to the lock; $s_{ij} = -1$ means that the agent has taken the action e_{ij} and found that it is not the solution to the lock. At the beginning of the game, elements in s are all zeros. As time goes by, certain a may make certain element of s become one of $\{-1, 1\}$. The dynamic of this MDP is shown in Figure 4.

Notice that it is quite possible that the agent keeps choosing the same a and hinders the exploration of the states. How to prevent this problem is one of the tasks of Reinforcement Learning. Also, we

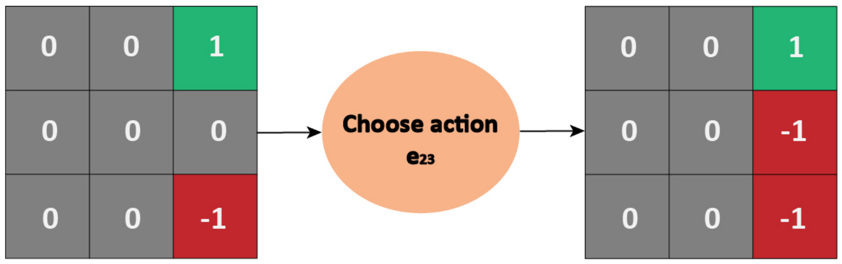


Figure 4: An example of the MDP dynamic illustrated in Section 5

limit our game modes to one of CC3, CE3, CC4, and CE4, all of whose 1 elements are in a row or in a column. This pattern can also be learned by Reinforcement Learning.

We design a simple reward function to encode the information of the environment. Basically, a reward of 10 is given when the agent finds a new solution that opens the door. 0 is given if the agent chooses a new action that fails to open the lock. We put a penalty on repeated actions that -10 is given when the agent chooses a visited position in the grid.

5.2 Model Details

A deep Q-learning model [3] is used as the underlying Q-learning algorithm. In each trial, the grid is initialized to all zeros. The model starts at a random position. After each step, a neural network takes the current state and action and outputs the Q values (as shown in Figure 5). The model has a chance of 0.85 to select the sequence with the highest value as the action in the next step (It randomly selects a position in the grid with a probability of 0.15 to avoid overfitting).

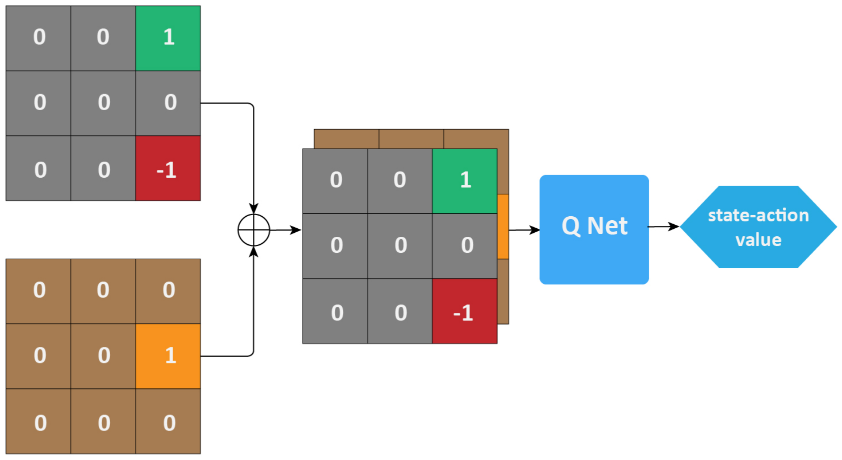


Figure 5: How our Q function works.

5.2.1 Spatial pyramid pooling

Because in OpenLock Games, we can get either 3 or 4 active layers (for some more general settings we may also get 5 or more active layers), we can not limit the order of our model's inputs. As a result, we use the SPP-net [2] to construct our Q function, enabling it to handle arbitrary order states inputs. Please refer to Appendix A for a detailed introduction.

5.2.2 Prioritized experience replay

We notice that an untrained agent tends to keep choosing the visited states and always gets a reward -10, failing to utilize the occasional success experiences. This phenomenon means that the OpenLock Game has a sparse reward. To solve this problem, we use Prioritized experience replay [5] to give those successful experiences more chances to be selected into the replay buffer.

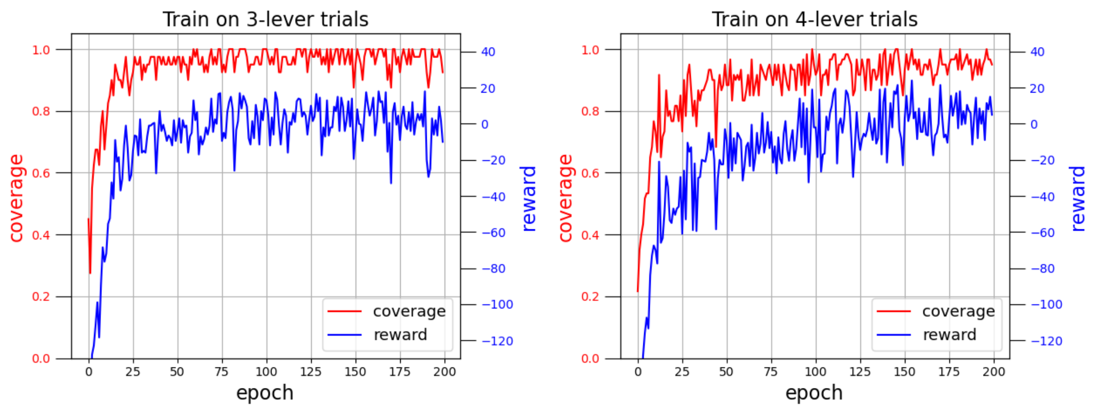


Figure 6: The training curve in different environments. The red line stands for the ratio of explored solutions to total solutions in 20-time steps per epoch. The blue line stands for the total reward collected in 20-time steps per epoch. The left figure represents the agent's performance when the training environment only provides 3 active layers settings. The right figure represents the agent's performance when the training environment only provides 4 active layers settings. Notice that our model structure keeps unchanged no matter how many active levers there are.

6 Experiments

We train our model on both 3-lever trials setting and 4-lever trials setting and test their performances crossly. After averaging 20 experiments' results, Figure 6 shows that our Reinforcement Learning algorithm converges to nearly optimal performance.

Particularly, Table 1 shows that our model has great generalization ability. The model trained purely on 3-lever trials also has a preferable performance when tested by pure 4-lever trials and mixed trials; the model trained purely on 4-lever trials also has a preferable performance when tested by pure 3-lever trials and mixed trials. These results indicate that Reinforcement Learning enables the agents to deduce basic knowledge for solving novel (or more complex) problems by training them on simple tasks.

Test setting	Model trained on 3-lever trials		Model trained on 4-lever trials		Random policy	
	Coverage	Reward	Coverage	Reward	Coverage	Reward
3 active levers	1.0000	20.0000	0.5867	-76.5333	0.1817	-192.7333
4 active levers	0.5733	-129.1000	0.9544	12.2333	0.1222	-193.6667
3/4 active levers	0.7889	-55.1667	0.7578	-32.5667	0.2811	-164.1000

Table 1: The performance of different models in different settings

7 Conclusion

To understand the ability of the Reinforcement Learning in causal reasoning, we developed a Deep RL algorithm to solve the OpenLock Task. To make our research realizable and specific, we give the agent certain prior knowledge by simplifying it to an OpenLock Grid Game. After the simplification, the agent has already known some basic common sense such as the buttons can be pressed to unlock the door and only dark levers are active. What agents still do not know are things like repeating the similar failed pressing order will not unlock the door and the inner structures are only one of CE and CC. Our deep Q-learning model performs perfectly on the OpenLock Grid Game, indicating that the Reinforcement Learning can learn the higher causal structure of the OpenLock Task. What's more, our method also shows a great generalization ability and can transfer the knowledge trained in certain conditions to some novel and more complex tasks, indicating that it can deduce and conclude some principles from simple lessons and generalize them into more challenging scenes.

Our method already does well in solving the OpenLock problem, yet it doesn't fully display the advantage of modeled RL over vanilla RL, as the OpenLock problem is too simple. Our next goal is to measure the performance of our method in a more complex task setting where there are more than 4 active levers. We may also introduce more randomness to evaluate the model capacity. For example, the correct movement sequences will only open the door with a certain probability rather than certainty.

References

[1] Mark Edmonds, James Kubricht, Colin Summers, Yixin Zhu, Brandon Rothrock, Song-Chun Zhu, and Hongjing Lu. Human causal transfer: Challenges for deep reinforcement learning. In *40th Annual Meeting of the Cognitive Science Society*, 2018. 2, 3, 4

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37, 06 2014. doi: 10.1109/TPAMI.2015.2389824. 6, 8

[3] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013. 6

[4] A. Robb. Optical geometry of motion: a new view of the theory of relativity. 01 2023. 1

[5] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015. 6

[6] Tingwu Wang, Xuchan Bao, Ignasi Clavera, Jerrick Hoang, Yeming Wen, Eric Langlois, Shunshi Zhang, Guodong Zhang, Pieter Abbeel, and Jimmy Ba. Benchmarking model-based reinforcement learning, 07 2019. 3

[7] Yixin Zhu, Tao Gao, Lifeng Fan, Siyuan Huang, Mark Edmonds, Hangxin Liu, Feng Gao, Chi Zhang, Siyuan Qi, Ying Nian Wu, et al. Dark, beyond deep: A paradigm shift to cognitive ai with humanlike common sense. *Engineering*, 6(3):310–345, 2020. 1, 2

A Spatial pyramid pooling

In order to solve the problem that existing deep convolutional neural networks (CNNs) require a fixed-size input image and this requirement may hurt the recognition accuracy for the images or sub-images of an arbitrary size/scale, He et al. [2] proposed SPP-net, where they equipped the networks with a more principled pooling strategy, "spatial pyramid pooling", which can generate a fixed-length representation regardless of image size/scale. The power of SPP-net is more significant in object detection. SPP-net computes the feature maps from the entire image only once, and then pools features in arbitrary regions (sub-images) to generate fixed-length representations for training the detectors.

Spatial pyramid pooling (spatial pyramid matching or SPM), as an extension of the Bag-of-Words (BoW) model, is one of the most successful methods in computer vision. It partitions the image into divisions from finer to coarser levels and aggregates local features in them. In He et al. [2], they added an SPP layer on top of the last convolutional layer, which pools the features and generates fixed-length outputs, which are then fed into the fully-connected layers (or other classifiers). In other words, spp performs some information "aggregation" at a deeper stage of the network hierarchy (between convolutional layers and fully connected layers) to avoid the need for cropping or warping at the beginning. Aggregation at a deeper stage is more physiologically sound and more compatible with the hierarchical information processing in our brains since it's likely that our brains handle arbitrarily-shaped objects at some deeper layers by aggregating the already deeply processed information from the previous layers.

Optionally include extra information (complete proofs, additional experiments, and plots) in the appendix. This section will often be part of the supplemental material.