CAUSAL INDUCTION FROM VISUAL OBSERVATIONS FOR GOAL DIRECTED TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Causal reasoning has been an indispensable capability for humans and other intelligent animals to interact with the physical world. In this work, we propose to endow an artificial agent with the capability of causal reasoning for completing goal-directed tasks. We develop learning-based approaches to inducing causal knowledge in the form of directed acyclic graphs, which can be used to contextualize a learned goal-conditional policy to perform tasks in novel environments with latent causal structures. We leverage attention mechanisms in our causal induction model and goal-conditional policy, enabling us to incrementally generate the causal graph from the agent's visual observations and to selectively use the induced graph for determining actions. Our experiments show that our method effectively generalizes towards completing new tasks in novel environments with previously unseen causal structures.

1 INTRODUCTION

Causal reasoning is an integral part of natural intelligence. The capacity to reason about cause and effect has been observed in humans and other intelligent animals as a means of survival (Blaisdell et al., 2006; Taylor et al., 2008). Such capacity plays a crucial role for young children in their interaction with the physical world. As behavioral psychology studies have shown, young children discover the underlying causal mechanisms from their play with the world (Schulz & Bonawitz, 2007), and their knowledge of causality in turn facilitates their subsequent learning of objects, concepts, languages, and physics (Rehder, 2003; Corrigan & Denton, 1996).

Nowadays, data-centric methods in artificial intelligence, such as deep networks, have achieved tremendous success in learning associations between inputs and outputs from large amounts of data, such as images to class labels (He et al., 2016). However, empirical evidence indicates that the absence of correct causal modeling in these methods has posed a major threat to generalization, causing image captioning models to generate unrealistic captions (Lake et al., 2017), deep reinforcement learning policies to fail in novel problem instances (Edmonds et al., 2018), and transfer learning models to adapt slower to new distributions (Bengio et al., 2019).

In this work, we propose to endow a learning-based interactive agent with the capacity of causal reasoning for completing goal-directed tasks in visual environments. Imagine that a household robot enters a new home for the first time. Without prior knowledge of the wiring configuration, it has to toggle the switches and sort out the correspondences between lights and switches, before it can be commanded to turn on the kitchen light or the bathroom light. We refer to the first phase of toggling switches as **causal induction**, where the agent discovers the latent cause and effect relations via performing actions and observing their outcomes; and we refer to the second phase of turning on specific lights as **causal inference**, where the agent uses the acquired causal relations to guide its actions for the completion of a task. To build an effective computational model for causal induction and inference, we have to address generalization towards novel causal relations and new task goals at the test time, both of which can be unseen during training.

We cast this as a meta-learning problem of two phases following Dasgupta et al. (2019). In the first stage, we use a **causal induction model** to construct a causal structure, i.e., a directed acyclic graph of random variables, from observational data from an agent's interventions. In the second stage, we use the causal structure to contextualize a **goal-conditional policy** to perform the task given a goal. However, in contrast to Dasgupta et al. (2019) we explicitly construct the causal structure instead of a

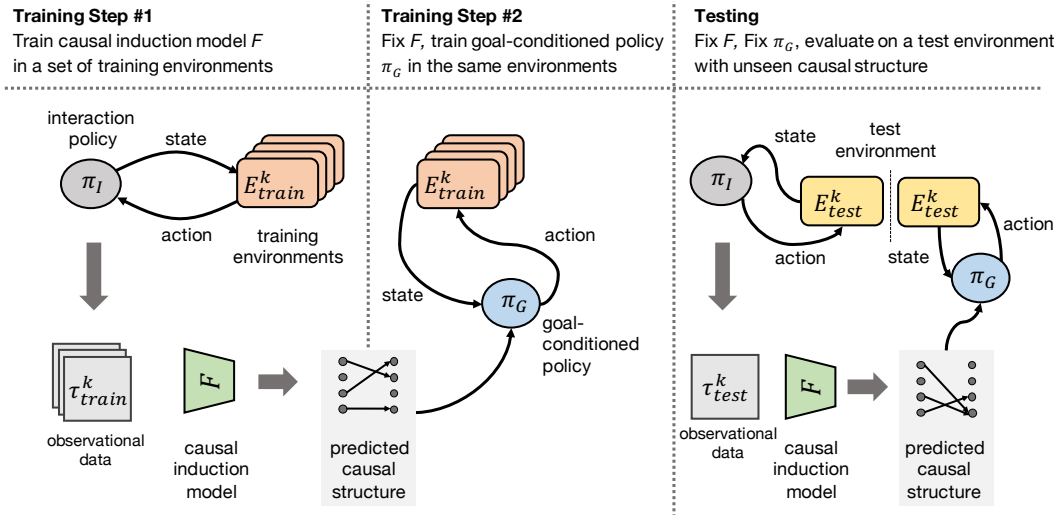

Figure 1: **Overview of Causal Induction and Inference Procedure.** During training each episode samples one of K training environments and uses the interaction policy π_I to probe the environment and collect a trajectory of visual observations. Using supervised learning we train the causal induction model F, which takes as input the trajectory of observational data and constructs $\hat{C}$, the estimate of C^k_{train}, which captures the underlying causal structure. Then, the predicted structure $\hat{C}$ is provided as input to the policy π_G conditioned on goal g, which learns to use the causal model to efficiently complete a specified goal in the training environments. At test time, F and π_G are fixed and the agent is evaluated on new environments with unseen causal structures.

latent feature encoding, leading to substantially better generalization towards new problem instances in long-horizon tasks as opposed to simplistic one-step querying.

To this end, we propose two technical contributions: 1) an iterative causal induction model with attention, which learns to incrementally update the predicted causal graph for each observed interaction in the environment, and 2) a goal-conditioned policy with an attention-based graph encoding, forcing it to focus on the relevant components of the causal graph at each step. We find that by factorizing the induction and inference processes through the use of causal graphs, it generalizes well to unseen causal structures given as few as 50 training causal structures. We compare our approach to using the ground-truth causal structure (which provides oracle performance), a non-iterative architecture which directly predicts the causal structure, and encoding the observation data into the LSTM (Hochreiter & Schmidhuber, 1997) memory of the policy similar to prior work (Dasgupta et al., 2019). We demonstrate that our method outperforms the baselines and achieves close to oracle performance in terms of both recovering the causal graph and success rate of completing goal-conditioned tasks, across several task sizes, types, and number of training causal structures.

2 Problem Statement

We formulate the agent's interaction in a Goal-Conditioned Markov Decision Process (MDP) defined by a six tuple $(\mathcal{S}, \mathcal{A}, p, \mathcal{G}, r, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $p : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ defines the transition dynamics, $\mathcal{G}$ is the goal space, $r : \mathcal{S} \times \mathcal{A} \times \mathcal{G} \rightarrow \mathbb{R}$ is a reward function where $r(s, a, g)$ gives the one-step immediate reward conditioned on the goal $g \in \mathcal{G}$, and γ is the discount factor. Our goal is to learn a goal-conditioned policy $\pi_G : \mathcal{S} \times \mathcal{G} \rightarrow \mathcal{A}$ that maximizes the expected sum of rewards $R_t = \sum_{k=t}^{\infty} \gamma^{k-t} r(s_k, a_k, g)$.

In this work, we not only want π_G to generalize well across goals in $\mathcal{G}$, but consider a more ambitious aim of making π_G generalize across a set of MDPs. We consider $\mathcal{M} = \{M^{(1)}, M^{(2)}, \ldots, M^{(K)}\}$ as the entire set of K MDPs with the same state space and action space but different transition dynamics, where $M^{(k)}$ is defined by $(\mathcal{S}, \mathcal{A}, p^{(k)}, \mathcal{G}, r, \gamma)$. The dynamics $p^{(k)}$ determines underlying causal relations between states and actions. Taking the same action at the same state could lead to a different next state under a different dynamics. We expect our agent to operate on its first-person vision and has no access to the latent causal relations. It receives high-dimensional RGB observations

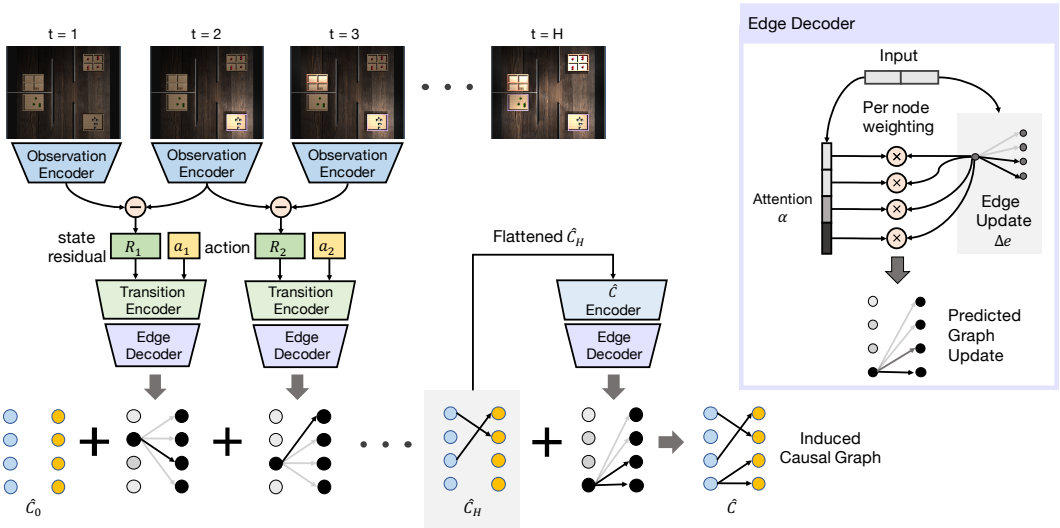

Figure 2: **Iterative Causal Induction Network**. Our iterative network architecture for inducing the causal structure from a visual trajectory of observational data with horizon H. First each frame is encoded into a latent state embedding s. Then the difference between state embeddings across time steps (state residual) is computed, and concatenated with the corresponding action. This is fed into the Edge Decoder module, which predicts an edge update, as well as an attention vector which is used to weight how the edge update is applied to nodes. On the last step one more edge update based on the current graph is applied, and a final predicted graph is outputted.

and has to induce a causal model from observational data. As illustrated in Figure 1, the overall procedure has two stages: 1) we execute an interaction policy π_I to collect a sequence of transitions $\tau = \{(s_1, a_1), (s_2, a_2), \ldots\}$, which is consumed by an induction model F to construct a latent causal model $\hat{C} = F(\tau)$; and 2) we use the causal model $\hat{C}$ to contextualize a goal-conditioned policy π_G such that it can perform tasks in the new MDP with novel causal relations. We formulate this as a meta-learning problem (Dasgupta et al., 2019; Finn et al., 2017). We partition the set of all MDPs $\mathcal{M}$ into two disjoint sets $\mathcal{M}_{train}$ and $\mathcal{M}_{test}$. During training, we learn our induction model F and goal-conditioned policy π_G with $\mathcal{M}_{train}$. During testing, we evaluate whether F is able to learn from the observational data from π_I in a novel MDP from $\mathcal{M}_{test}$ to construct a causal model that can be used by π_G to perform tasks in this new MDP.

Direct modeling of causal relations in raw pixel space is intractable due to the large dimensionality. Following Chalupka et al. (2014), we assume that cause and effect in our problems can be defined on a handful of causal macro-variables. For example, the illuminance of the kitchen (the agent's visual observation) is determined by the on and off of the kitchen light (macro-variable), which is caused by toggling the state of the switch that controls the light (another macro-variable). This assumption enables us to construct a directed acyclic causal model $\hat{C}$ to represent the causal effects of actions on these macro-variables. Given the set of macro-variables, the induction model F predicts directed edges between them from visual observations. A primary challenge here is the confounders raised from partial observability and spurious correlations in the agent's visual perception. For example, illuminance changes in the kitchen might be due to turning on/off the kitchen light or the living room light. Hence, it requires the agent to disentangle the correct causal relations from visual inputs.

3 METHOD

The goal of our method is to enable a policy to complete goal-conditioned vision-based control tasks in environments with unseen causal structures, given only a short trajectory of observational data in the environment. Prior work (Dasgupta et al., 2019) has shown promising results on simplistic one-step querying problems using an LSTM-based policy which encodes the interaction into the policy's memory. Our hypothesis is that to generalize in complex multi-step control problems, a more structured induction and policy scheme will be required. To address this, we propose iterative updates and attention bottlenecks in the induction model $\hat{C} = F(\{(s_1, a_1), (s_2, a_2), \ldots\})$ and in the policy

$a = \pi_G(s, g, \hat{C})$ respectively, which we demonstrate significantly improves generalization to unseen causal structures.

3.1 Iterative Causal Induction Network

Inducing the causal structure from raw sensory observations requires accurately capturing the unique effect of each action on the environment, while accounting for confounding effects of other actions. We hypothesize that the causal induction network that best generalizes will be one which disentangles individual actions and their corresponding effect, and only updates the relevant components of the causal graph.

We implement this idea in our iterative model, where we begin with an initial guess of the causal structure $\hat{C}$ which has all edge weights of 0 (meaning we assume no causal relationships). We then use an Observation Encoder to map each image of the observational data to an encoding s and compute the state residual R between subsequent time steps. This R, which captures the change in state is then concatenated with the corresponding action, and then fed into the Edge Decoder module. The output of the Edge Decoder module is an update to the edge strengths of the causal graph $\Delta\hat{C}$. This update is applied to each observed transition, that is $\hat{C}_{t+1} = \Delta\hat{C} + \hat{C}_t$, and at the final layer the whole graph is encoded to do a final edge update before the causal graph is predicted (see Figure 2). The Edge Decoder takes either the encoded R and a, or the encoded edge matrix of $\hat{C}_H$, and outputs a $1 \times N$ soft attention vector α and a $1 \times N$ change to the edge weights Δe, where N is the number of actions in the environment. The attention vector α is used to weight which nodes in the causal graph the edge update Δe should be applied to. Thus at each iteration the update amounts to:

$$\hat{C}_{t+1} = (\alpha^T \Delta e) + \hat{C}_t, \alpha = \phi(R, a)$$

where ϕ is the Transition Encoder, a fully connected module (see Appendix A for details). Using this attention mechanism further encourages the module to make independent updates, which we observe enables better generalization.

3.2 Learning Goal-Conditioned Policies

The objective of the policy is given an initial image s_0, a goal image g, and the predicted causal structure $\hat{C}$, reach the goal within T time steps. Additionally, the policy $\pi_G(s, g, \hat{C})$ is a reactive one, and thus can only solve the goal-conditioned task if it can learn to use the predicted causal structure $\hat{C}$. That is — since the policy has no memory, it cannot learn to induce the graph internally during inference time, and thus must use the causal graph $\hat{C}$.

We hypothesize that like the causal induction model, the policy which best generalizes is one which learns to focus exclusively on the edges in the causal graph which are relevant to the current step of the task. To that end we propose an attention bottleneck in the graph encoding, which encourages the policy to select edges pertaining to one "effect" at each step, which enables better generalization.

Specifically, the policy encodes the current image s and goal image g. Based on this encoding it outputs an attention vector α of size $1 \times N$ over the "effects" in the causal graph. This vector is used to perform a weighted sum over the outputs of the $N \times N$ causal graph (N causes, N effects, and edges between them), resulting in a size N vector of the selected edges e. The selected edges and visual encodings are used to output the final action:

$$a = \phi_3(E(s, g), \phi_2(e)), e = \hat{C} \cdot \alpha^T, \alpha = \phi_1(E(s, g))$$

where E has identical architecture to the image encoder as F, but which encodes both current and goal image and ϕ_i are all fully connected layers (see Figure 3).

3.3 Model Training

The induction network F is trained using supervised learning in the limited set of training environments, specifically to minimize the ℓ_2 reconstruction loss between the ground-truth causal graph C and the predicted causal graph $\hat{C} = F(\{(s_1, a_1), (s_2, a_2), \ldots\})$.

The policy π_G is trained using the DAgger (Ross et al., 2010) algorithm by imitating a planner using the ground-truth causal graph in the training environments. Then π_G is tested in unseen environments with only visual inputs and goal images. Specifically, in the training environments, the planner uses

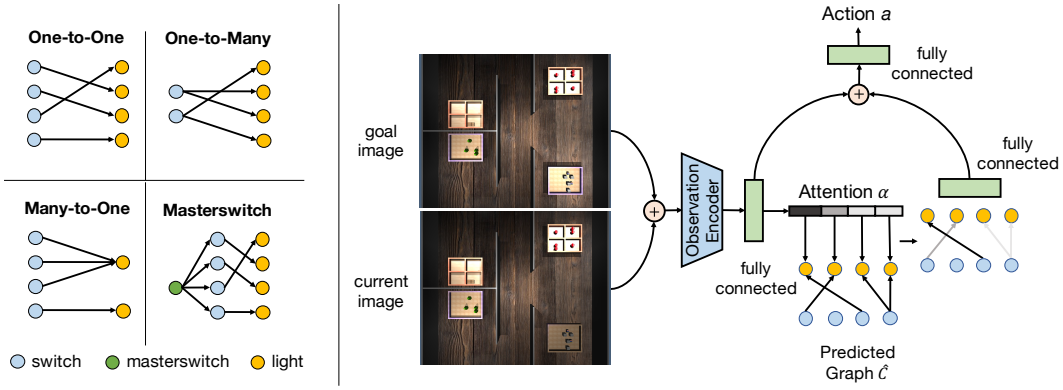

Figure 3: **Types of Causal Structures (Left)** We explore four types of causal structures, ONE-TO-ONE, ONE-TO-MANY, MANY-TO-ONE, and MASTERSWITCH. These cover a standard causal mapping, common cause causal patterns, common effect causal patterns, and causal chains. **Goal-Conditioned Policy (Right)**. The policy takes as input the current image, goal image, and predicted causal graph $\hat{C}$. The current image and goal image are concatenated channel wise and encoded. This encoding is used to predict an attention vector over the "effects" in $\hat{C}$ which extract the relevant edges, which is then concatenated with the image encoding to predict the action.

the ground-truth graph and privileged low dimensional state/goal information to compute an optimal plan to the goal. At each time step, the expert's action is added to the memory of the policy, which is then trained using a standard cross entropy loss to imitate the expert given the current image and goal image. The policy is also injected with ϵ-greedy noise during training, with $\epsilon = 0.3$. Network architectures and additional training details can be found in the appendix.

4 EXPERIMENTS

Through our experiments, we investigate three complementary questions: 1) Does our iterative induction network enable better causal graph induction?, 2) Does our attention bottleneck in the graph encoding in π_G enable the policy to generalize better to unseen causal structures?, and 3) By combining our proposed F and π_G, are we able to outperform the current state-of-the-art Dasgupta et al. (2019) on visual goal-directed tasks?

4.1 EXPERIMENTAL SETUP

Task Definition: We examine the multi-step task of light switch control. In particular an agent has control of N switches (macro-variables) , which have some underlying causal structure of how they control N lights (macro-variables). However, the macro-variables of the lights manifest themselves in noisy visual observations, whose partial observability and overlap result in confounding factors. The objective of the agent is starting from an initial state s_0, to control the switches to reach a specified lighting goal g, where both the state and goal are $32 \times 32 \times 3$ images. However as specified in the problem setup, the underlying causal structure is unknown to the agent, all that is provided is limited observational data from the environment. Thus the agent must (1) induce the causal structure between the switches and lights from the observational data, then (2) use it to reach the goal.

We explore 4 different types of causal patterns between the switches and lights (See Figure 3). The first type of causal structure is ONE-TO-ONE, in which each switch maps to one light. The second type of causal structure is MANY-TO-ONE (Common Effect (Keil, 2006)), where each switch controls one light, but multiple switches may control the same light. The third type is ONE-TO-MANY (Common Cause (Keil, 2006)), where all lights are controlled by at most one switch, but a single switch may control more than one light. Lastly, we also explore the MASTERSWITCH domain in which there is a Causal Chain (Keil, 2006) where only once one master switch is activated can the other switches causal effects be observed, applied on top of a ONE-TO-ONE causal structure. In our method we represent the causal structure C as a graph with directed edges between N switch states and N light states in the environment, with edge strength corresponding to the likelihood that that switch controls the indicated light, with an additional N edges in the MASTERSWITCH setting.

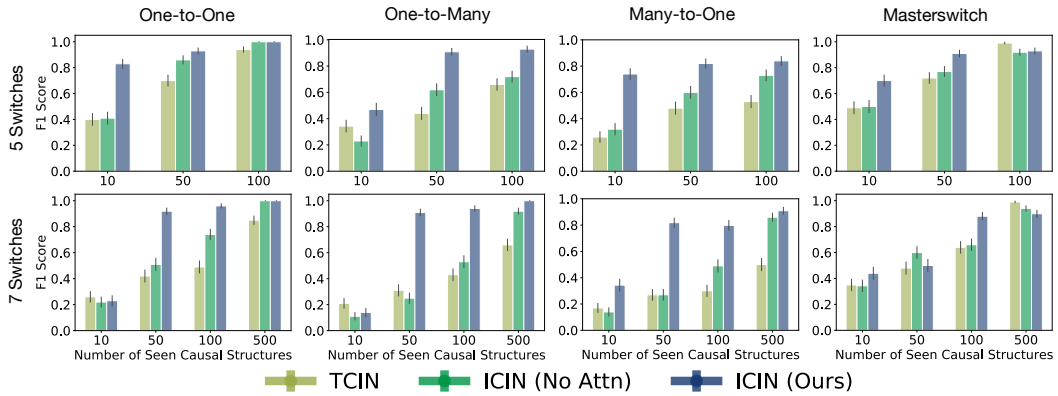

Figure 4: **F1 Score on Unseen Causal Structures**. The F1 Scores of edges on the predicted causal structure compared to the ground-truth on unseen causal structures. We compare across variable numbers of seen structures {10, 50, 100, 500} and problem size {5, 7}. Our iterative approach with attention outperforms the comparisons across almost all settings.

Environment Setup: The trajectory of observational data from which we induce the causal structure is in visual space, consisting of a $32 \times 32 \times 3$ image as well as an size $A = N$ action vector for each timestep of the H timestep trajectory. The visual scene consists of the N lights mounted in a MuJoCo (Todorov et al., 2012) simulated house with 3 rooms, where the lights are mounted in the rooms and hallway, and the effect of each light is rendered onto the floor when they are turned on, which is captured by a top down camera. The illumination from the lights overlap, resulting in confounding factors, which must be disentangled by the model in order to correctly predict the causal structure. The state space (and goal space) of the policy are also $32 \times 32 \times 3$ images of the same environment and camera. The action space A consists of size N, one discrete action for each switch. During policy learning, the goals are sampled uniformly from the space of possible lighting configurations under the environments causal structure, i.e., the set of all reachable states in the environment.

Collecting Observational Data: The observational data is collected using a heuristic interaction agent π_I, which executes a simple policy. In the all but the MASTERSWITCH case, the policy simply takes each action once (horizon $H = N$). In the MASTERSWITCH setting the exploratory agent presses each switch until one has an effect on the environment, and then proceeds to press each of the other switches (horizon up to $H = 2N - 1$).

4.2 EVALUATION METHODS

We evaluate the following methods and baselines to examine the effectiveness of our causal induction model and goal-conditioned policy.

First to examine how much our iterative causal induction network **(ICIN)** improves performance on inducing the causal graph we compare against a non-iterative induction model which uses temporal convolutions **(TCIN)**, as well as an ablation of our method which uses an iterative model without the attention mechanism **(ICIN (No Attn))**. We compare these approaches based on F1 score of recovering the ground-truth causal graph.

Next, we compare the performance of the goal conditioned policy using all variants (ICIN, ICIN (No Attn), TCIN), compared to the previous work of Dasgupta et al. (2019) which induces the graph using the latent memory of the policy (**Memory**). Specifically, we provide the policy π_G with LSTM memory, and before running DAgger, feed the interaction trajectory one step at a time through the policy. This end-to-end approach allows the causal graph to be encoded into the latent memory of the policy, to be used when doing a goal-directed task. We also compare to **Memory (RL/Low Dim)**, a version of Dasgupta et al. (2019) which has access to privileged state information (ground-truth states), and is trained using model-free reinforcement learning with a dense reward. In this setting the same visual interaction trajectory is encoded into the policy's LSTM memory, but the actual policy receives a binary vector for state and goal and is trained using the PPO algorithm (Schulman et al., 2017). We also add a comparison to using the ground-truth causal graph (even at test time) as an Oracle (**Oracle**), which provides an upper bound on performance. All methods are compared based on success rates in unseen environments. Implementation details can be found in Appendix B.

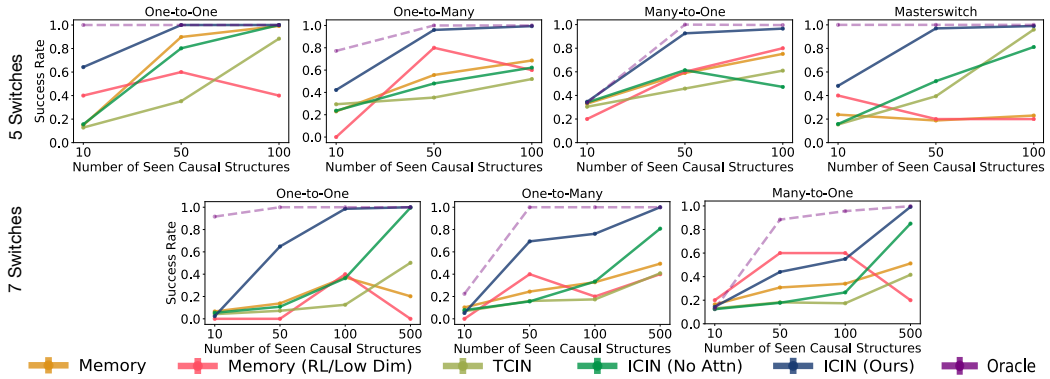

Figure 5: **Policy Success Rates (Unseen Causal Structures)**. The final success rates of the goal-conditioned policy for each method on unseen causal structures for either 10, 50, 100, or 500 seen causal structures for 5 or 7 switches. Our iterative approach achieves the best generalization on unseen tasks across almost all settings.

Lastly, to understand how critical the attention bottleneck in the goal-conditioned policy is for generalization, we compare the success rates in unseen causal structures of the goal-condition policy using a graph induced by ICIN, with and without the attention bottleneck.

4.3 Causal Induction Evaluation

First we examine our approach's ability to induce the causal model from the trajectory of observational data. We report the F1 score (threshold=0.5) between edges of the predicted causal graph and ground-truth causal graph. We compute the F1 scores across up to 100 unseen tasks given 10, 50, 100, or 500 seen structures in the 5 and 7 switch problems (see Figure 4). We observe that across almost all settings, our iterative approach with attention (**ICIN**) dominates. While our method without attention generally outperforms the non-iterative baseline, both fall significantly behind our final approach, suggesting that the modularity that attention provides plays a large role in enabling generalization. Furthermore, we observe that our iterative attention approach especially outperforms the others when there are less training causal structures, which would suggest that by forcing the network to make independent updates, it is able to learn a general method for induction with limited training examples. This is likely because the attention forces the model to learn to update a single edge given a single observation, while being agnostic to the total graph, which is far less likely to overfit to the training structures. Qualitative examples of ICIN can be found in Appendix C.

4.4 Goal-Conditioned Policy Evaluation

We examine the success rate of the converged policy π_G on 500 trials in unseen causal structures with randomized goals in Figure 5. We observe in most settings that the **Memory** based approach of Dasgupta et al. (2019) provides a strong baseline, outperforming the **TCIN** and **ICIN (No Attn)** baselines. We suspect that it learns that to best imitate the expert, it has to encode relevant information from the interaction trajectory into its latent memory, implicitly performing induction. While this works, it generalizes to unseen structures much worse than our proposed method, likely due to the compositional structure of our approach. The memory baseline which uses low dimensional states (**Memory (RL/Low Dim)**), and is trained via model-free RL also performs well, in fact beating our approach on a few cases in the 7 switch, Many-to-One setting. However in general the performance of this approach is much lower than ours, and likely the cases in which it does succeed can be attributed to its use of privileged information (ground-truth states) instead of visual observations. In almost all cases **ICIN** significantly outperforms all baselines. In fact, in the 5 switch case our **ICIN** method nearly matches the Oracle, suggesting that it almost perfectly induces C.

Finally, we study the importance of our proposed attention bottleneck in the graph encoding in $\pi_G(s, g, \hat{C})$, which forces the policy to focus on only relevant edges at each timestep. We examine the success rate of using π_G with the attention bottleneck compared to just flattening $\hat{C}$ given the current image, goal image, and predicted causal graph from ICIN.

We find that using the attention bottleneck in the graph encoder of π_G yields a roughly 10% increase in success rate in the ONE-TO-ONE (1:1) and MASTERSWITCH (MS) cases, and a roughly 40% increase in success rate in the ONE-TO-MANY (1:K) and MANY-TO-ONE (K:1) cases. This is because by encouraging the policy to pick relevant edges, it has led to a modular policy which can 1) identify the changes it wants to make in the environment and 2) predict the necessary action based on the causal graph $\hat{C}$, enabling better generalization.

π_G	1:1	1:K	K:1	MS
No Attn	0.84	0.51	0.59	0.90
Ours	**1.0**	**0.95**	**0.94**	**0.98**

Table 1: **Policy Attention Bottleneck.** We observe in that across all settings using the attention bottleneck in the graph encoder of the policy significantly improves performance. Tested on 5 switches and 50 seen structures.

5 RELATED WORK

Causal reasoning has been extensively studied by a broad range of scientific disciplines, such as social sciences (Yee, 1996), medical sciences (Kuipers & Kassirer, 1984), and econometrics (Zellner, 1979). In causality literature, structural causal model (SCM) (Pearl, 2009) has offered a formal framework of modeling causation from statistical data and counterfactual reasoning. SCMs are a directed graph that represents causal relationships between random variables. Both causal induction (constructing SCMs from observational data) (Shimizu et al., 2006; Hoyer et al., 2009; Peters et al., 2014; Ortega & Braun, 2014) and causal inference (using SCMs to estimate causal effects) (Bareinboim et al., 2015; Bareinboim & Pearl, 2016) algorithms have been developed. Conventional methods have limited applicability in complex domains where observational data is high-dimensional and partially observable. Recent work has shown that causal learning can take advantage of the representational power of deep learning methods for inducing causal relationships from interventions (Dasgupta et al., 2019) and for improving policy learning via counterfactual reasoning (Buesing et al., 2018). However, they have focused on toy-sized problems with low-dimensional states. In contrast, our model induces and makes use of the causal structure for complex interactive tasks from raw image observations.

Lake et al. (2017) are among the first to discuss the limitations of state-of-the-art deep learning models in causal reasoning. There has been a growing amount of efforts in marrying the complementary strengths of deep learning and causal reasoning. Causal modeling has been explored in several contexts, including image classification (Chalupka et al., 2014), generative models (Kocaoglu et al., 2017), robot planning (Kurutach et al., 2018), policy learning (Buesing et al., 2018), and transfer learning (Bengio et al., 2019). Pioneer work on causal discovery with deep networks has applied to time series data in healthcare domains (Kale et al., 2015; Nauta et al., 2019). In addition, adversarial learning (Kalainathan et al., 2018), graph neural networks (Yu et al., 2019), and gradient-based DAG learning (Lachapelle et al., 2019) have been recently introduced to causal discovery, but largely focusing on small synthetic datasets. Most relevant to ours are Dasgupta et al. (2019) and Edmonds et al. (2018), which investigated causal reasoning in deep reinforcement learning agents. In contrast to them, our method directly learns an explicit causal structure from raw observations to solve multi-step, goal-conditioned tasks.

Generalization to new environments and new goals has been a central challenge for learning-based interactive agents. This problem has been previously studied in in the context of domain adaptation (Tzeng et al., 2015; Peng et al., 2017), system identification (Yu et al., 2017; Zhou et al., 2019), meta-learning (Finn et al., 2017; Sæmundsson et al., 2018), and multi-task learning (Andrychowicz et al., 2017; Schaul et al., 2015). These works have addressed variations in dynamics, visual appearances, and task rewards, while assuming fixed causal structures. Instead, we focus on changes in the latent causal relationships that determine the preconditions and effects of actions.

6 CONCLUSIONS

We have proposed novel techniques for 1) causal induction from raw visual observations and 2) causal graph encoding for goal-conditioned policies, both of which lead to better generalization to unseen causal structures. Our key insight is that by leveraging iterative predictions and attention bottlenecks, it facilitates our causal induction model and goal-conditioned policy to focus on the relevant part of the causal graph. Using this approach we show better generalization towards novel problem instances than previous works with limited training causal structures.

In this work, we induce the causal structure from observational data collected by a heuristic policy. We plan to explore more complex tasks where probing the environment to discover the causal structure requires more sophisticated strategies, and develop algorithms that jointly learn the interaction policy.

REFERENCES

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *CoRR*, abs/1707.01495, 2017.

Elias Bareinboim and Judea Pearl. Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences*, 113(27):7345–7352, 2016.

Elias Bareinboim, Andrew Forney, and Judea Pearl. Bandits with unobserved confounders: A causal approach. In *Advances in Neural Information Processing Systems*, pp. 1342–1350, 2015.

Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher Pal. A meta-transfer objective for learning to disentangle causal mechanisms. *arXiv preprint arXiv:1901.10912*, 2019.

Aaron P Blaisdell, Kosuke Sawa, Kenneth J Leising, and Michael R Waldmann. Causal reasoning in rats. *Science*, 311(5763):1020–1022, 2006.

Lars Buesing, Theophane Weber, Yori Zwols, Sebastien Racaniere, Arthur Guez, Jean-Baptiste Lespiau, and Nicolas Heess. Woulda, coulda, shoulda: Counterfactually-guided policy search. *arXiv preprint arXiv:1811.06272*, 2018.

Krzysztof Chalupka, Pietro Perona, and Frederick Eberhardt. Visual causal feature learning. *arXiv preprint arXiv:1412.2309*, 2014.

Roberta Corrigan and Peggy Denton. Causal understanding as a developmental primitive. *Developmental review*, 16(2):162–202, 1996.

Ishita Dasgupta, Jane Wang, Silvia Chiappa, Jovana Mitrovic, Pedro Ortega, David Raposo, Edward Hughes, Peter Battaglia, Matthew Botvinick, and Zeb Kurth-Nelson. Causal reasoning from meta-reinforcement learning. *arXiv preprint arXiv:1901.08162*, 2019.

M. Edmonds, J. Kubricht, C. Summers, Y. Zhu, B. Rothrock, S.C. Zhu, and H. Lu. Human causal transfer: Challenges for deep reinforcement learning. *Annual Meeting of the Cognitive Science Society*, 2018.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *CoRR*, abs/1703.03400, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Conference on Computer Vision and Pattern Recognition*, 2016.

Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. Stable baselines. `https://github.com/hill-a/stable-baselines`, 2018.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL `http://dx.doi.org/10.1162/neco.1997.9.8.1735`.

Patrik O Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. In *Advances in neural information processing systems*, pp. 689–696, 2009.

Diviyan Kalainathan, Olivier Goudet, Isabelle Guyon, David Lopez-Paz, and Michèle Sebag. Sam: Structural agnostic model, causal discovery and penalized adversarial learning. *arXiv preprint arXiv:1803.04929*, 2018.

David C Kale, Zhengping Che, Mohammad Taha Bahadori, Wenzhe Li, Yan Liu, and Randall Wetzel. Causal phenotype discovery via deep networks. In *AMIA Annual Symposium Proceedings*, volume 2015, pp. 677. American Medical Informatics Association, 2015.

Frank C. Keil. Explanation and understanding. *Annual Review of Psychology*, 57(1):227–254, 2006. doi: 10.1146/annurev.psych.57.102904.190100. URL https://doi.org/10.1146/annurev.psych.57.102904.190100. PMID: 16318595.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Murat Kocaoglu, Christopher Snyder, Alexandros G Dimakis, and Sriram Vishwanath. CausalGAN: Learning causal implicit generative models with adversarial training. *arXiv preprint arXiv:1709.02023*, 2017.

Benjamin Kuipers and Jerome P Kassirer. Causal reasoning in medicine: analysis of a protocol. *Cognitive Science*, 8(4):363–385, 1984.

Thanard Kurutach, Aviv Tamar, Ge Yang, Stuart J Russell, and Pieter Abbeel. Learning plannable representations with Causal InfoGAN. In *Advances in Neural Information Processing Systems*, pp. 8733–8744, 2018.

Sébastien Lachapelle, Philippe Brouillard, Tristan Deleu, and Simon Lacoste-Julien. Gradient-based neural dag learning. *arXiv preprint arXiv:1906.02226*, 2019.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

Meike Nauta, Doina Bucur, and Christin Seifert. Causal discovery with attention-based convolutional neural networks. *Machine Learning and Knowledge Extraction*, 1(1):312–340, 2019.

Pedro A Ortega and Daniel A Braun. Generalized thompson sampling for sequential decision-making and causal inference. *Complex Adaptive Systems Modeling*, 2(1):2, 2014.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. *CoRR*, abs/1710.06537, 2017.

Jonas Peters, Joris M Mooij, Dominik Janzing, and Bernhard Schölkopf. Causal discovery with continuous additive noise models. *The Journal of Machine Learning Research*, 15(1):2009–2053, 2014.

Bob Rehder. Categorization as causal reasoning. *Cognitive Science*, 27(5):709–748, 2003.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. No-regret reductions for imitation learning and structured prediction. *CoRR*, abs/1011.0686, 2010. URL http://arxiv.org/abs/1011.0686.

Steindór Sæmundsson, Katja Hofmann, and Marc Peter Deisenroth. Meta reinforcement learning with latent variable gaussian processes. *arXiv preprint arXiv:1803.07551*, 2018.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1312–1320, Lille, France, 07–09 Jul 2015. PMLR. URL http://proceedings.mlr.press/v37/schaul15.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

Laura E Schulz and Elizabeth Baraff Bonawitz. Serious fun: preschoolers engage in more exploratory play when evidence is confounded. *Developmental psychology*, 43(4):1045, 2007.

Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, and Antti Kerminen. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(Oct):2003–2030, 2006.

Alex H Taylor, Gavin R Hunt, Felipe S Medina, and Russell D Gray. Do new caledonian crows solve physical problems through causal reasoning? *Proceedings of the Royal Society B: Biological Sciences*, 276(1655):247–254, 2008.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IROS*, pp. 5026–5033. IEEE, 2012.

Eric Tzeng, Coline Devin, Judy Hoffman, Chelsea Finn, Xingchao Peng, Sergey Levine, Kate Saenko, and Trevor Darrell. Towards adapting deep visuomotor representations from simulated to real environments. *arXiv preprint arXiv:1511.07111*, 2(3), 2015.

Albert S Yee. The causal effects of ideas on policies. *International Organization*, 50(1):69–108, 1996.

Wenhao Yu, C. Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. *CoRR*, abs/1702.02453, 2017.

Yue Yu, Jie Chen, Tian Gao, and Mo Yu. Dag-gnn: Dag structure learning with graph neural networks. *arXiv preprint arXiv:1904.10098*, 2019.

Arnold Zellner. Causality and econometrics. In *Carnegie-Rochester conference series on public policy*, volume 10, pp. 9–54. Elsevier, 1979.

Wenxuan Zhou, Lerrel Pinto, and Abhinav Gupta. Environment probing interaction policies. *ICLR*, 2019.

A ARCHITECTURE DETAILS

A.1 INDUCTION MODELS

A.1.1 OBSERVATION ENCODER

The image encoder used in all models takes as input a $32 \times 32 \times 3$ image of the scene, feeds it through 3 convolutional layers with each followed by ReLU activation and 2×2 Max Pooling. The output filters of the convolutions are 8, 16, and 32 respectively, and the resulting $4 \times 4 \times 32$ tensor is mapped to latent vector of size N using a single fully connected layer, where N is the number of switches/lights.

A.1.2 ICIN TRANSITION ENCODER

The transition encoder used in our iterative model takes as input a state residual of dimension N and an action of size $N + 1$ concatenated together, and feeds it through fully connected layers of size 1024 and 512 each with ReLU activation, followed by another layer which outputs an attention vector of size N (or $N + 1$ in the MASTERSWITCH case) with SoftMax activation and an edge update of size N with Sigmoid activation. The first two layers are trained with dropout of 30%.

A.1.3 ICIN $\hat{C}$ ENCODER

The causal graph encoder used in the last step our iterative model takes as input the $N \times N$ (or $N + N \times N$ in the MASTERSWITCH Case) flattened edge weights of the current graph $\hat{C}_H$ and feeds them though a single fully connected layer, which outputs an attention vector of size N (or $N + 1$ in the MASTERSWITCH Case) with SoftMax activation and an edge update of size N with Sigmoid activation.

A.1.4 ICIN (NO ATTN)

The non-iterative ablation of our method has an identical architecture, but instead of the third fully connected layer outputting an attention vector of size N with SoftMax activation and an edge update of size N with Sigmoid activation, it instead updates a full set of $N \times N$ edge weights with Sigmoid activation.

A.1.5 TCIN

The temporal convolution induction network uses the same image encoder as our approach. However the size N state encodings are then concatenated with the size $N + 1$ action labels, and then are passed through three layers of temporal convolutions, with filter size [256, 128, 128] and a size 3 kernel with stride 1. The horizon H by 128 dimensional output is then flattened and fed through fully connected layers of size 1024 and 512, each with 30% dropout. Finally, the size $N \times N$ causal graph is outputted with Sigmoid activation.

A.2 POLICY ARCHITECTURE

A.2.1 ATTENTION BASED π_G

The same image encoder as the induction models is used for the policy, except as input it takes a $32 \times 32 \times 6$ image which contains the current image and goal image, concatenated channel wise. The encoded image is then flattened and fed through a fully connected layer of size 128, which then outputs an attention vector of size N, which is used to do a weighted sum over the edges of the $N \times N$ causal graph, producing a vector of size N. This is then encoded to size 128, concatenated with the 128 dim image encoding, and def through 2 more fully connected layers of size 64 and ultimately outputting the final action prediction of size N.

In the non attention version the architecture is identical except the full graph is flattened, then encoded and concatenated directly with the image encoding.

A.3 MEMORY BASELINE

In this baseline we use an image encoder as above, except there is an additional input for action. There is also an LSTM Cell of hidden dimension 256 which the image encoding and action encoding are fed into, which is then fed through fully connected layers of size 256, 64 which output the action.

A.4 MEMORY (RL/LOW DIM) BASELINE

The policy is a MLP-LSTM policy as implemented in Hill et al. (2018), with two fully connected layers of size 64, and an LSTM layer with 256 hidden units. It is augmented with additional input heads for each step of the observational data, namely $32 \times 32 \times 3$ images and size N actions.

B TRAINING DETAILS

B.1 CAUSAL INDUCTION MODEL (F) TRAINING

Each causal induction model is trained for each split of seen/unseen causal structures as described in the experiments. The F is trained offline on all τ^k_{train} and corresponding C^k_{train}, for 60000 training iterations using Adam optimizer (Kingma & Ba, 2015) with learning rate 0.0001 and batch size 512. They are implemented using PyTorch (Paszke et al., 2017) and trained on an NVIDIA Titan X GPU.

B.2 DAGGER POLICY TRAINING

The policies trained with DAgger (Ross et al., 2010) are trained in the training environments with episode horizon $T = 2N$. The policy takes actions in the environment, and at each step an expert action is appended to the policy's memory buffer. The policy is then trained to imitate the experience in the memory. The expert uses the ground-truth causal graph and ground-truth low dimensional states to compute the difference between the goal and current state, and based on the graph what action needs to be taken. Each policy is trained for 100000 episodes, with learning rate 0.0001 and batch size from the memory of 32.

B.3 RL POLICY TRAINING

The Memory (RL/Low Dim) baseline is trained using the Proximal Policy Optimization algorithm (Schulman et al., 2017) as implemented in Stable-Baselines (Hill et al., 2018). They use hyper-parameters $\gamma = 0.99$, 128 steps per update, entropy coefficient 0.01, learning rate 0.00025, value function coefficient 0.0001 and $\lambda = 0.95$. The policy itself consists of two fully connected layers of size 64, as well as an LSTM layer consisting of a 256 size hidden state. The policy is trained until the policy performance converges on unseen causal structures, and capped at a max of 9 million episodes. In this experiment, we set the horizon T of each episode equal to the number of switches/lights N.

C CAUSAL INDUCTION QUALITATIVE EXAMPLES

Here we demonstrate an example trajectory and how the causal induction model iteratively builds the causal graph.

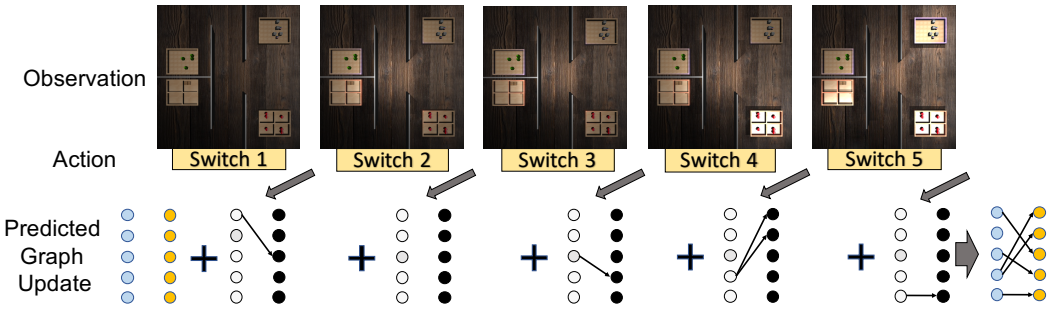

Figure 6: **Sample of Causal Induction**. Here we show an example of our Iterative Causal Induction Model for 5 switches, in the "One-to-Many" case. Given the trajectory of actions and images of the scene, the model needs to reason about which lights were turned on, and how what update this implies in the graph. In this example, the first observed action turns on one of the switches, and the model makes the corresponding update to the graph. The next switch does not change the lighting so the model outputs no update to the graph. The next action sees one light go on, and updates the corresponding switch. The next action turns on two lights, and the graph is updated to reflect this. Lastly, since one light remains unaccounted for, the model knows to add that edge to the graph. Note: The edges and updates are soft updates, but the model learns to predict close to exactly 1 for edges and exactly 0 for non-edges.