# Systematic Evaluation of Causal Discovery in Visual Model Based Reinforcement Learning

**Nan Rosemary Ke** [*,1,2] **Aniket Didolkar**[*,3] **Sarthak Mittal** [3] **Anirudh Goyal** [3]
**Guillaume Lajoie** [3] **Stefan Bauer** [6] **Danilo Rezende** [2]
**Yoshua Bengio** [3,†] **Michael Mozer** [5] **Christopher Pal** [1,4]

## Abstract

Inducing causal relationships from observations is a classic problem in machine learning. Most work in causality starts from the premise that the causal variables themselves are observed. However, for AI agents such as robots trying to make sense of their environment, the only observables are low-level variables like pixels in images. To generalize well, an agent must induce high-level variables, particularly those which are causal or are affected by causal variables. A central goal for AI and causality is thus the joint discovery of abstract representations and causal structure. However, we note that existing environments for studying causal induction are poorly suited for this objective because they have complicated task-specific causal graphs which are impossible to manipulate parametrically (e.g., number of nodes, sparsity, causal chain length, etc.). In this work, our goal is to facilitate research in learning representations of high-level variables as well as causal structures among them. In order to systematically probe the ability of methods to identify these variables and structures, we design a suite of benchmarking RL environments. We evaluate various representation learning algorithms from the literature and find that explicitly incorporating structure and modularity in models can help causal induction in model-based reinforcement learning.

## 1 Introduction

Deep learning methods have made immense progress on many reinforcement learning (RL) tasks in recent years. However, the performance of these methods still pales in comparison to human abilities in many cases. Contemporary deep reinforcement learning models have a ways to go to achieve robust generalization [Nichol et al., 2018], efficient planning over flexible timescales [Silver and Ciosek, 2012], and long-term credit assignment [Osband et al., 2019]. Model-based methods in RL (MBRL) can potentially mitigate this issue [Schrittwieser et al., 2019]. These methods observe sequences of state-action pairs, and from these observations are able to learn a self-supervised model of the environment. With a well-trained world model, these algorithms can then simulate the environment and look ahead to future events to establish better value estimates, without requiring expensive interactions with the environment [Sutton, 1991]. Model-based methods can thus be far more sample-efficient than their model-free counterparts when multiple objectives are to be achieved in the same environment. However, for model-based approaches to be successful, the learned models must capture relevant mechanisms that guide the world, i.e., they must discover the right causal variables and structure. Indeed, models sensitive to causality have been shown to be robust and

[*] Authors contributed equally, [1] Mila, Polytechnique Montréal, [2] Deepmind, [3] Mila, Polytechnique Montréal, [4] Element AI, [5] Google AI, [6]Max Planck Institute for Intelligent Systems, [†] CIFAR Senior Fellow Corresponding authors: rosemary.nan.ke@gmail.com

Submitted to the 35th Conference on Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks. Do not distribute.

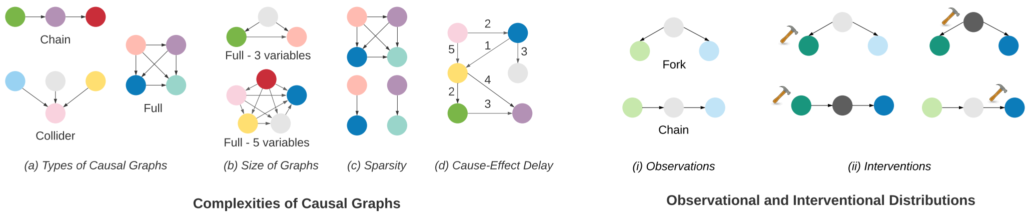

Figure 1: (a)-(d): Different aspects contributing to the complexity of causal graphs. (i), (ii): Difference between observational and interventional data. In RL setting, actions are interventions in the environment. The hammer denotes an intervention. Intervention on a variable not only affects its direct children, but also all reachable variables. Variables impacted by the intervention have a darker shade.

easily transferable [Bengio et al., 2019, Ke et al., 2019]. As a result, there has been a recent surge of interest in learning causal models for deep reinforcement learning [de Haan et al., 2019, Dasgupta et al., 2019, Nair et al., 2019, Goyal et al., 2019, Rezende et al., 2020, Wang et al., 2021]. Yet, many challenges remain, and a systematic framework to modulate environment causality structure and evaluate models' capacity to capture it is currently lacking, which motivates this paper.

What limits the use of causal modeling approaches in many AI tasks and realistic RL settings is that most of the current causal learning literature presumes abstract domain representations in which the cause and effect variables are explicit and given [Pearl, 2009]. Methods are needed to automate the inference and identification of such causal variables (i.e. *causal induction*) from low-level state representations (like images). Although one solution is manual labeling, it is often impractical and in some cases impossible to manually label all the causal variables. In some domains, the causal structure may not be known. Further, critical causal variables may change from one task to another, or from one environment to another. And in unknown environments, one ideally aims for an RL agent that could induce the causal structure of the environment from observations and interventions.

In this work, we seek to evaluate various model-based approaches parameterized to exploit structure of environments purposfully designed to modulate causal relations. We find that modular network architectures appear particularly well suited for causal learning. Our conjecture is that causality can provide a useful source of inductive bias to improve the learning of world models.

***Shortcomings of current RL development environments, and a path forward.*** Most existing RL environments are not a good fit for investigating causal induction in MBRL, as they have a single fixed causal graph, lack proper evaluation and have entangled aspects of causal learning. For instance, many tasks have complicated causal structures as well as unobserved confounders. These issues make it difficult to measure progress for causal learning. As we look towards the next great challenges for RL and AI, there is a need to better understand the implications of varying different aspects of the underlying causal graph for various learning procedures.

Hence, to systematically study various aspects of causal induction (i.e., learning the right causal graph from pixel data), we propose a new suite of environments as a platform for investigating inductive biases, causal representations, and learning algorithms. The goal is to disentangle distinct aspects of causal learning by allowing the user to choose and modulate various properties of the ground truth causal graph, such as the structure and size of the graph, the sparsity of the graph and whether variables are observed or not (see Figure 1 (a)-(d)). We also provide evaluation criteria for measuring causal induction in MBRL that we argue help measure progress and facilitate further research in these directions. We believe that the availability of standard experiments and a platform that can easily be extended to test different aspects of causal modeling will play a significant role in speeding up progress in MBRL.

***Insights and causally sufficient inductive biases.*** Using our platform, we investigate the impact of explicit structure and modularity for causal induction in MBRL. We evaluated two typical of monolithic models (autoencoders and variational autoencoders) and two typical models with explicit structure: graph neural networks (GNNs) and modular models (shown in Figure 5). Graph neural networks (GNNs) have a factorized representation of variables and can model undirected relationships between variables. Modular models also have a factorized representation of variables, along with directed edges between variables which can model directed relationship such as $A$ causing $B$, but not the other way around. We investigated the performance of such structured approaches on learning from causal graphs with varying complexity, such as the size of the graph, the sparsity of the graph and the length of cause-effect chains (Figure 1 (a) - (d)).

The proposed environment gives novel insights in a number of settings. Especially, we found that even our naive implementation of modular networks can scale significantly better compared to other

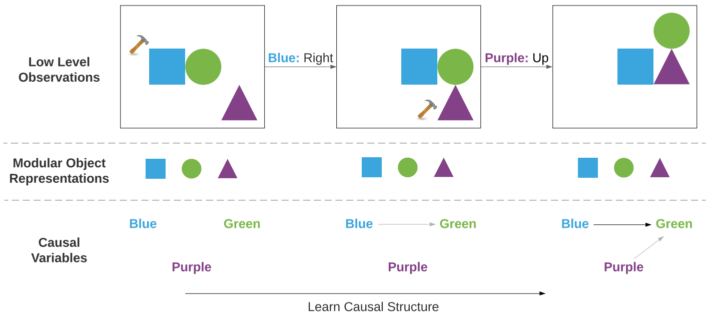

Figure 2: Illustration of the key features of the suite. Environments have objects that interact according to the underlying causal graph which can be based on a subset of objects' properties. An efficient model should be able to infer the high level causal variables from raw pixel data and learn the underlying causal graph through interactions between these high level causal variables.

models (including graph neural networks). This suggests that explicit structure and modularity such as factorized representations and directed edges between variables help with causal induction in MBRL. We also found that graph neural networks, such as the ones from Kipf et al. [2019] are good at modeling pairwise interactions and significantly outperform monolithic models under this setting. However, they have difficulty modeling complex causal graphs with long cause-effect chains, such as the chain graph (demonstration of chain graphs are found in Figure 1 (i)). Another finding is that evaluation metrics such as likelihood and ranking loss do not always correspond to the performance of these models in downstream RL tasks.

## 2 Environments for causal induction in model-based RL

Causal models are frequently described using graphs in which the edges represent causal relationships. In these *structural causal models*, the existence of a directed edge from $A$ to $B$ indicates that intervening on $A$ directly impacts $B$, and the absence of an edge indicates no direct interventional impact (see Appendix B for formal definitions).

In parallel, world models in MBRL describe the underlying data generating process of the environment by modeling the next state given the current state-action pair, where the actions are interventions in the environment. Hence, learning world models in MBRL can be seen as a causal induction problem. Below, we first outline how a collection of simple causal structures can capture real-world MBRL cases, and we propose a set of elemental environments to express them for training. Second, we describe precise ways to evaluate models in these environments.

### 2.1 Mini-environments: explicit cases for causal modulation in RL

The ease with which an agent learns a task greatly depends on the structure of the environment's underlying causal graph. For example, it might be easier to learn causal relationships in a collider graph ( see Figure 1(a)) where all interactions are pairwise, meaning that an intervention on one variable $X_i$ impacts no more than one other variable $X_j$, hence the cause-effect chain has a length of at most 1. However, causal graphs such as full graphs (see Figure 1 (a)) can have more complex causal interactions, where intervening on one variable impacts can impact up to $n - 1$ variables for graphs of size $n$ (see Figure 1). Therefore, one important aspect of understanding a model's performance on causal induction in MBRL is to analyze how well the model performs on causal graphs of varying complexity.

Impotant factors that contribute to the complexity of discovering the causal graph are the *structure*, *size*, *sparsity of edges* and *length of cause-effect* chains of the causal graph (Figure 1). Presence of *unobserved variables* also adds to the complexity. The size of the graph increases complexity because the number of possible graphs grows super-exponentially with the *size of the graph* [Eaton and Murphy, 2007, Peters et al., 2016, Ke et al., 2019]. The *sparsity of graphs* also impacts the difficulty of learning, as observed in [Ke et al., 2019]. Given graphs of the same size, denser graphs are often more challenging to learn. Futhermore, the *length of the cause-effect* chains can also impact learning. We have observed in our experiments, that graphs with shorter cause-effect lengths such as colliders (Figure 1 (a)) can be easier to model as compared to chain graphs with longer cause-effect chains. Finally, *unobserved variables* which commonly exist in the real-world can greatly impact learning, especially if they are confounding causes (shared causes of observed variables).

Taking these factors into account, we designed two suites of (toy) environments: the *physics environment* and the *chemistry environment*, which we discuss in more detail in the fol-

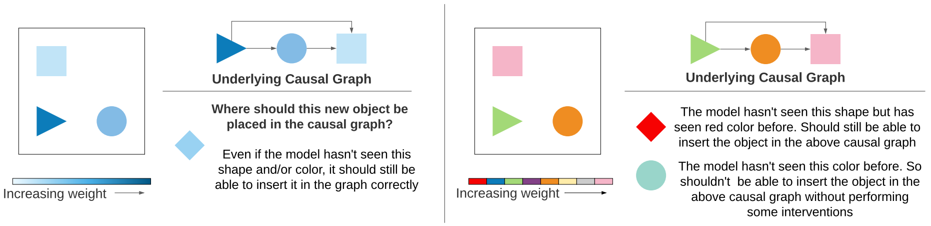
Underlying Causal Graph
Where should this new object be placed in the causal graph?
Even if the model hasn't seen this shape and/or color, it should still be able to insert it in the graph correctly
Increasing weight
Underlying Causal Graph
The model hasn't seen this shape but has seen red color before. Should still be able to insert the object in the above causal graph
The model hasn't seen this color before. So shouldn't be able to insert the object in the above causal graph without performing some interventions
Increasing weight

Figure 3: Demonstration of the weighted-block pushing environment (left: observed, right: unobserved) along with the feasible generalizations that the setup provides.

lowing section. They are designed with a focus on the underlying causal graph and thus have a minimalist design that is easy to visualize.

### 2.1.1 Physics environment: Weighted-block pushing

The physics environment simulates very simple physics in the world. It consists of blocks of different, unique weights. The rule for interaction between blocks is that heavier objects can push lighter ones. Interventions amount to move a particular block, and the consequence depends on whether the block next to it (if present) is heavier or lighter. For an accurate world model, inferring the weights becomes essential. Additionally, one can allow the weight of the objects to be either observed through the intensity of the color, or unobserved, leading to two environment settings described below. The underlying causal graph is an acyclic tournament, shown in Figure 3.

The Physics environment consists of 50 x 50 RGB pixels of renderings of visual scenes in 2D; examples are shown in Figures 2-3. Each episode consists of a fixed set of $k$ objects, drawn without replacement; each object is defined by shape. The initial configuration of objects in the scene is random. Objects reside on a 5x5 grid of cells; each grid cell is rendered as a 10x10 pixel array, giving rise to the 50x50 RGB images. All objects are visible at every time, so the state is Markovian. The action space of the agent is a discrete pair (x,y), where x is the index of the object to intervene on and y is a discrete value that sets the value of the intervention. The index-to-object mapping is fixed across episodes. The intervention involves pushing the object in a given direction (up, down, left, right). The dynamics of that object and others depends on the physics of the domain (e.g., a heavier object pushes an adjacent lighter object in the same direction). For more details about the setup, please refer to Appendix G.

*Fully observed setting*. In the fully observed setting, all objects are given a particular color and the weight of each block is represented by the intensity of the color. Once the agent learns this underlying causal structure, it does not have to perform interventions on new objects in order to infer they will interact with the others.

*Unobserved setting*. In this setting, the weight of each object is not directly observable by its color. The agent thus needs to interact with the object in order to understand the order of weights associated with the blocks. In this case, the weight of objects needs to be inferred through interventions. We consider two sub-divisions of this setting - *FixedUnobserved* where there is a fixed assignment between the shapes of the objects and their weights and *Unobserved* where there is no fixed assignment between the shape and the weight, hence making it a more challenging environment. We refer the reader to Appendix G.2 for details.

### 2.1.2 Chemistry environment

The chemistry environment enables more complexity in the causal structure of the world by allowing arbitrary causal graphs. This is depicted by simple chemical reactions, where the state of an element can cause changes to another variable's state. The environment consists of a number of objects whose positions are kept fixed and thus, uniquely identifiable.

The interactions between different objects take place according to the underlying causal graph which can either be a randomly generated DAG, or specified by the user. An interaction consists of changing the color (state) of a variable. At this point, the color of all variables affected by this variable (according to the causal graph) can change. Interventions change a block's color unconditionally, thus cutting the graph edge linking it with its parents in the graph. All transitions are probabilistic and defined by conditional probability tables (CPTs). A visualization of the environment can be found in Figure 4.

The Chemistry environment (see Figure 4 for examples) also consists of 50 x 50 RGB pixels of renderings of visual scenes in 2D. Each episode also consists of a fixed set of $k$ objects, drawn without replacement; each object is defined by shape. The objects does not move within an episode, instead the colors of the object can change due to an intervention. The action space of the agent is still a discrete pair $(x, y)$, where $x$ is the index of the object to intervene on and $y$ is a discrete value that sets the the color that the object is changed to.

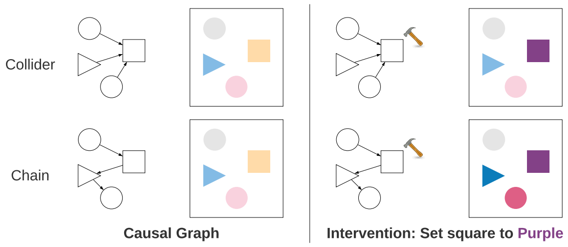

Figure 4: Demonstration of the vanilla chemistry environment (left: ground truth causal graph and a sample from it - same sample shown to demonstrate the affect of interventions, right: the affect of interventions and how far they affect based on underlying causal graph)

This environment allows for a complete and thorough testing of causal models as there are various degrees of complexities which can be easily tuned such as: (1) Complexity of the graph: We can test any model on many different graphs thus ensuring that a models performance is not only limited to a few select graphs. (2) Stochasticity: By tuning the skewness of the probability distribution of each object we can test how good is a given model in modelling data uncertainty. In addition to this we can also tune the number of object or the number of colors to test whether the model generalizes to larger graphs and more colors. A causally correct model should be able to infer the causal relationships between observed objects, as well as their respective color distribution and its dependence on a causal parent's distribution.

### 2.2 Evaluating causal models

In much of the existing literature, evaluation of learned causal models is based on the structural difference between the learned graph and the ground-truth graph [Peters et al., 2016, Zheng et al., 2018]. However, this may not be applicable for most deep RL algorithms, as they do not necessarily learn an explicit causal structure [Dasgupta et al., 2019, Ke et al., 2020]. Even if a structure is learned, it may not be unique as several variable permutations can be equivalent, introducing an additional evaluation burden.

Another possibility is to exhaustively evaluate models on all possible intervention predictions and all environment states, a process that quickly becomes intractable even for small environments. We therefore propose a few evaluation methods that can be used as a surrogate metrics to measure the model's performance on recovering the correct causal structure.

*Predicting Intervention Outcomes.* While it may not be feasible to predict all intervention outcomes in an RL environment, we propose that evaluating predictions on a subset of interventions provides an informative evaluation. Here, the test data is collected from the same environment used in training, ensuring a single underlying causal graph. Test data is generated from new episodes that are unseen during training. All interventions (actions) in the test episodes are randomly sampled and we evaluate the model's performance on this test set.

*Zero Shot Transfer.* Here, we test the model's ability to generalize to unseen test environments, where the environment does not have exactly the same causal graph as training, but training and test causal graphs share some similarity.

For example, in the *observed* Physics environment, a model that has learned the underlying causal relationship between color intensity and weight would be able to generalize to new variables with a novel color intensity.

*Downstream RL Tasks.* Downstream RL tasks that require a good understanding of the underlying causal graph of the environment are also good metrics for measuring the model's performance. For example, in the *physics environment*, we can provide the model with a target configuration in the form of some specific arrangement of blocks on a grid and the model needs to perform actions in the environment to reach the target configuration. Models that capture causal relationships between objects should achieve the target configuration more easily (as it is can predict intervention outcomes). For more details about this setup, please refer to Appendix E.

*Metrics.* We also evaluate the learned models on ranking metrics in the latent space as well as reconstruction-based metrics in the observation space [Kipf et al., 2019]. In particular we measure and report Hits at Rank 1 (H@1), Mean Reciprocal Rank (MRR) and Reconstruction loss for evaluation in standard as well as transfer testing settings. We report these metrics for 1, 5 and 10 steps of prediction in the latent space (refer Appendix C).

# 3 Models

A large variety of neural network models have been proposed as world models in MBRL. These models can roughly be divided into two categories: *monolithic models* and models that have *structure* and *modularity*. *Monolithic models* typically have no explicit structure (other than layers). Some typical monolithic models are Autoencoders and Variational Autoencoders [Kingma and Welling, 2013, Rezende et al., 2014]. Conversely, *structured* models have explicit architecture built into (or learned by) the model. Examples of such models are ones based on graph neural networks [Battaglia et al., 2016, Van Steenkiste et al., 2018, Kipf et al., 2019, Veerapaneni et al., 2020] and modular models [Ke et al., 2020, Goyal et al., 2019, Mittal et al., 2020, Goyal et al., 2020]. We picked some commonly used models from these categories and evaluated their performance to understand their ability for causal induction in MBRL.

To disentangle the architectural biases and effects of different training methodologies, we trained all the models on both likelihood based and contrastive losses, respectively. All models share three common components: *encoder*, *decoder* and *transition model*. We follow a similar training procedure as in Ha and Schmidhuber [2018], Kipf et al. [2019]. Details of the architectures as well as the training protocols and losses can be found in Appendix F.

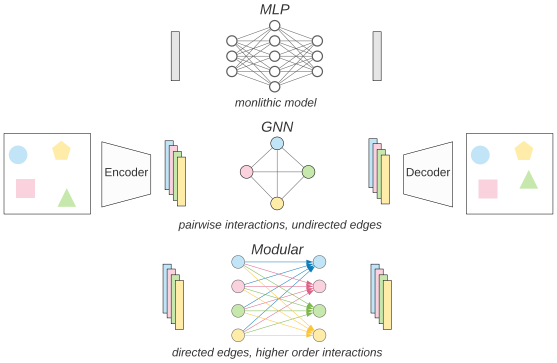
MLP
monlithic model
GNN
Encoder
Decoder
pairwise interactions, undirected edges
Modular
directed edges, higher order interactions

Figure 5: All models have 3 components: *encoder*, *decoder* and *transition model*. The transition models can either be monolithic, modular model or graph neural networks (GNNs). Monothlic models don't have explicit structure. GNNs have factorized representation of variables. Modular models have factorized representation of both variables and directed edges to potentially model causal relationships, e.g. $A$ causing $B$.

## 3.1 Monolithic Models

We evaluate causal induction on two commonly used monolithic models: multilayered autoencoders and variational autoencoders. We follow a similar setup as in Ha and Schmidhuber [2018]. These models do not have strong inductive biases other than the number of layers used.

## 3.2 Modular and Structured Models

Several forms of structure can be included in neural networks, including *modularity*, *factorized variables*, and *directed rules*.

Taking the three factors into account, we consider two types of structured models in our paper, *graph neural networks* (GNN) and so called *modular networks*. Graph neural networks (GNN) [Gilmer et al., 2017, Tacchetti et al., 2018, Battaglia et al., 2018, Kipf et al., 2019] is a widely adopted relational model that have a factorized representation of variables and models pairwise interactions between objects while being permutation invariant. In particular, we consider the C-SWM model [Kipf et al., 2019], which is a state-of-art GNN used for modeling object interactions. Similar to most GNNs, the C-SWM model learns factorized representations of different objects but for modelling dynamics it considers all possible pairwise interactions, and hence the transition model is monolithic (i.e., not a modular transition model).

Modular networks on the other hand are composed of an initial encoder that factorizes inputs (images), and then a *modular transition model* (MTM) - $M$. This internal model is tasked to create separate factored representations for each objects in the environment, while taking into account all other objects' representations. This model also learns interactions between objects. The rules learned here are *directed rules*.

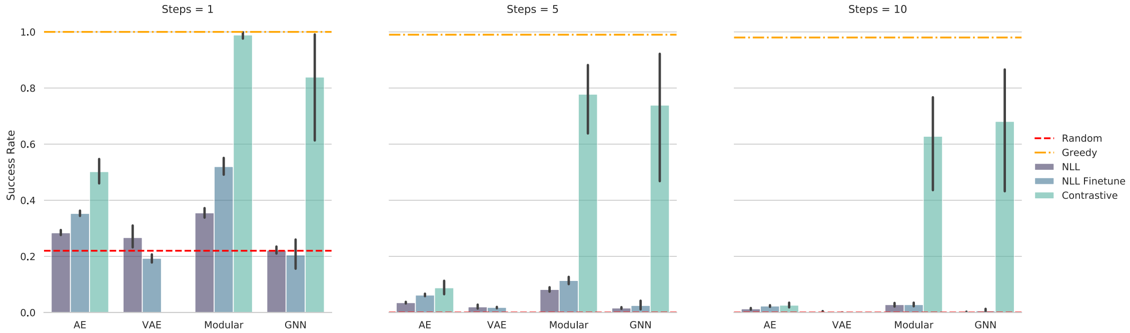

Figure 6: Success Rate *(higher is better)* for different models and training losses for 1, 5 and 10 step prediction for the Fixed Unobserved Physics environment setting with 5 objects. Here, (a) Random stands for a random policy, (b) greedy is the policy with best greedy actions, (c) NLL are models trained in 2 stages: pretraining the encoder/ decoder, following by only training the transition model, (d) NLL with finetune are models in 3 stages: pretraining the encoder/ decoder, following by only training the transition model and then finetuning the encoder, decoder and transition models together. (e) Contrastive are models trained using a contrastive loss. The GNN and Modular models trained on constrastive loss significantly outperform the monolithic models (autoencoders and VAE). The margin significantly increases as the number of steps to reach the goal increase, suggesting that models with explicit structure and modularity have a much better understanding of the world.

# 4 Experiments

Our experiments seak to answer the following questions: (a) Does explicit structure and modularity help for causal induction in MBRL? If so, then what type of structures provide good inductive bias for causal induction in MBRL? (b) How do different objective functions (likelihood or contrastive) impact learning? (c) How do different models scale to complex causal graphs? (d) Do prediction metrics (likelihood and ranking metrics) correspond to better downstream RL performance? (e) What are good evaluation criteria for causal induction in MBRL?

We report the performance of our models on both the Physics and the Chemistry environments, and refer the readers to Appendix F for implementation details.. All models are trained using the procedure described in Appendix F.2 and are evaluated based on *ranking* and *likelihood metrics* on $1, 5$ and 10 step predictions. For the Chemistry environment, we evaluate the models on causal graphs with varying complexity, namely - *chain*, *collider* and *full* graphs. These graphs vary in *the sparsity of edges* and the *length of cause-effect chains*. For the Physics environment, we evaluate the model in the fully observed setting as well as the unobserved setting.

## 4.1 Data

The autoencoder,VAE, modularand GNN models are trained on sequences generated by an agent following a random policy. The training data consists of 1,000 sequences consisting of 100 frames per sequence. The validation data consists of 1,000 sequences with 100 frames per sequence. The test data consists of 10,000 sequences with 10 frames per sequence.

## 4.2 Explicit structure and causal induction

We found that for both the Physics and the Chemistry environments, models with explicit structure outperform monolithic models on both prediction metrics and downstream RL performances. In particular, models with explicit structure (GNNs and modular models) scale better to graphs of *larger size* and *longer cause-effect chains*.

The Physics environment has a complex underlying causal graph (full graph: refer Figure 1 (a)). We found that GNNs performed well in this environment with 3 variables. They achieved good prediction metrics (Figure 8) and high RL performance (Figure 14) even at longer timescales. However, their performance drops significantly on environments with 5 objects both in terms of prediction metrics (Figure 9) and RL performance (Figure 15). We also see in Figures 9 and 15 that modular models scale much better compared to all other models, suggesting that they hold an advantage for *larger* causal graphs. Further, modular models and GNNs when evaluated on zero shot settings outperform monolithic models by a significant margin (Figures 20 and 21 and Tables 15 and 16).

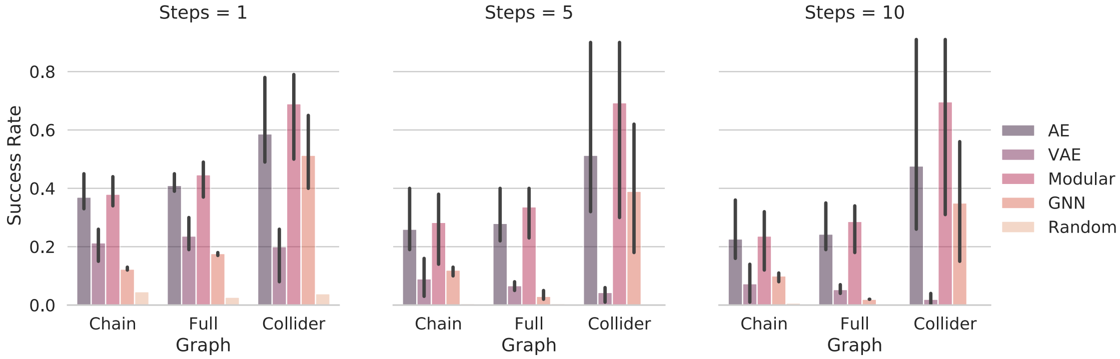

Figure 7: Success rate (higher is better) for different models evaluated on 1, 5 and 10 step predictions for the static chemistry environment with 5 objects and 5 colors. The results are grouped in types of causal graphs for the environment, refer to section 1(a) for illustrations of different types of causal graphs. Chain and full graphs are significantly more challenging compared to collider graphs. This suggests that causal relationships in chain and full graphs with longer cause and effect chains are more challenging to learn compared to the collider graphs, which has only pairwise interactions. Modular models outperform all other models in almost all cases, this is an indication that introducing structure in the form of modularity is an important inductive bias for learning causal models.

For the chemistry environment, we find that modular models outperform all other models for almost all causal graphs in terms of both prediction metrics (Figure 24) and RL performance (Figures 7 and 26). This is especially true on more complex causal graphs, such as *chain* and *full* graphs which have long cause-effect chains. This suggests that modular models scales better to more complex causal graphs.

Overall, these results suggest that structure, and in particular modularity, help causal induction in MBRL when scaling up to larger and more complex causal graphs. The performance comparisons on modular networks and C-SWM [Kipf et al., 2019] suggest that both factorized representation of variables and directed edges between variables can help for causal induction in MBRL.

### 4.3 Complexity of the Underlying Causal Graph

There are several ways to vary complexity in a causal graph: *size of the graph*, *sparsity of edges* and *length of cause-effect chain* (Figure 1). Increasing the size of the graph significantly impacts all models' performances. We evaluate models on the Physics environments with 3 objects (Figure 8) and 5 objects (Figure 9) and find that increasing the number of objects from 3 to 5 has a significant impact on performance. Modular models achieve over 90 on ranking metrics over 10-step prediction for 3 objects while for 5 objects, they achieve only 50 (almost half the performance on 3 objects). A similar pattern is found in almost all models. Another factor impacting complexity of the graph is the *length of cause-effect chain*.We see that collider graphs are the easiest to learn, with modular models and autoencoders significantly outpeforming all other models (Figure 24). This is because the collider graph has short pair-wise interactions, i.e, intervention on any node in a collider graph can impact at most one other node. Chain and full graphs are significantly more challenging because of longer cause-effect chains. For a chain or a full graph of $n$ nodes, an intervention on the $k^{th}$ node can impact all the subsequent $(n-k)$ nodes. Modeling interventions on chain and full graphs require modeling more than pairwise relationships, hence, making it much more challenging. We find that modular models slightly outperform all other models on these graphs.

### 4.4 Prediction Metrics and RL Performance

As discussed in Section 2.2, there are multiple evaluation metrics based on either prediction metrics or RL performance. The performance of the model on one metric may not necessarily transfer to another. We would like to analyze if this is the case for the models trained under various environments. We first note that while the ranking metrics were relatively good for most models on physics environments, most of them only did slightly better than a random policy on downstream RL, especially on larger graphs (Figures Figure 8 - 13 and Table 3 - 8 for ranking metrics; Figure 14 - 19 and Table 9 - 14 for downstream RL). Figures 22, 23 and 28 show scatter plots for each pair of losses, with one loss on each axis. While there is some correlation between ranking metric and RL performance (Modular and GNN; Figure 22), we did not find this trend to be consistent across models and environment settings. We feel that these results give further evidence of need to evaluate on RL performance.

### 4.5 Training objectives and learning

Likelihood loss and contrastive loss [Oord et al., 2018, Kipf et al., 2019] are two frequently used objectives for training world models in MBRL. We trained the models under each of these objective functions to understand how they impact learning. In almost all cases, models with explicit structure (modular models and GNNs) trained on contrastive loss perform better in terms of ranking loss compared to those trained on likelihood loss (refer to Figure 8 - 13). We don't see a very clear trend between training objective and downstream RL performance but we do see a few cases where contrastively trained models performed much better than others (refer to Figures 6, 14, 18 and 19 and Tables 9, 13 and 14). For other key insights and experimental conclusions on different environments, we refer the readers to Appendix G.6 for the physics environment and Appendix H.3 for the chemistry environment.

## 5 Related work

*Video Prediction and Visual Question Answering.* There exist a number of video prediction [Yi et al., 2019, Baradel et al., 2019] and visual question answering [Johnson et al., 2017] datasets that also make use of a blocks world for visual representation. Though these datasets can appear visually similar to ours at first glance, they lack two essential ingredients for systematically evaluating models for causal induction in MBRL. The first is that they do not allow active interventions and hence make it challenging for evaluating model-based reinforcement learning algorithms. Another key point is that these environments do not allow one to systematically perturb different aspects of causal graphs, hence, preventing to systematically study the performances of models for causal induction.

*RL Environments.* There exist several benchmarks for multi-task learning for robotics (Meta-World [Yu et al., 2019] and RLBench [James et al., 2020]), for Physical reasoning Bakhtin et al. [2019] and for video gaming domain (Arcade Learning Environment, CoinRun [Cobbe et al., 2018], Sonic Benchmark [Machado et al., 2018], MazeBase [Nichol et al., 2018] and BabyAI [Chevalier-Boisvert et al., 2018]). However, as mentioned earlier, these benchmarks do not allow one to systematically control different aspects of causal models (such as the structure, the sparsity of edges and the size of the graph), hence making it difficult to systematically study causal induction in MBRL. The Alchemy [Wang et al., 2021] environment, which was released earlier this year, moves a step towards causal induction for meta-RL. Though the environment allows for some level of control of the underlying causal structures of the environment, it still does so in a limited way.

*Block World.* The AI community has been using the "blocks world" for decades as a testbed for various AI problems, including learning theory [Winston, 1970], natural language [Winograd, 1972], and planning [Fahlman, 1974]. Block world allows to easily vary different aspects of the underlying causal structure, and also allow interventions to be performed on many high level variables of the environment giving rise to a large space of tasks which have well-defined relations between them.

## 6 Discussions and conclusions

In our work, we focus on studying various model-based approaches for causal induction in model-based RL. We highlighted the limitations of existing benchmarks and introduced a novel suite of environments that can help measure progress and facilitate research in this direction. We evaluated various models under many different settings and discuss the essential problems and challenges in combining both fields i.e ingredients, that we believe are common in the real world, such as modular factorization of the objects and interactions of objects governed by some unknown rules. Using a proposed evaluation framework, we demonstrate that structural inductive biases are beneficial to learning causal relationships and yield significantly improved performances in learning world models.

**Limitations and Future Work**. There are some limitations of this work that can be explored in interesting directions in the future. One direction is extending the environments to settings such as meta-learning, where different causal graphs are set for each episode of training. Another limitation of our work is that in the environments which we propose the effect occurs immediately after the cause, but in real world settings the effect may sometimes be delayed. For example, if a person smokes, it can take variable amount of time until they get cancer. This is very relevant for reinforcement learning, as this is tightly related to credit assignment in RL. Future works could explore environments where the relation between cause and effect does not occur at fixed time-scales.

**Social Impact**. The authors do not foresee negative social impact of this work beyond that which could arise from general improvements in ML.

# References

Anton Bakhtin, Laurens van der Maaten, Justin Johnson, Laura Gustafson, and Ross Girshick. Phyre: A new benchmark for physical reasoning. In *Advances in Neural Information Processing Systems*, pages 5082–5093, 2019.

Fabien Baradel, Natalia Neverova, Julien Mille, Greg Mori, and Christian Wolf. Cophy: Counterfactual learning of physical dynamics. *arXiv preprint arXiv:1909.12000*, 2019.

Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. In *Advances in neural information processing systems*, pages 4502–4510, 2016.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

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.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. Babyai: First steps towards grounded language learning with a human in the loop. *arXiv preprint arXiv:1810.08272*, 2018.

Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. *arXiv preprint arXiv:1812.02341*, 2018.

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.

Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. In *Advances in Neural Information Processing Systems*, pages 11698–11709, 2019.

Daniel Eaton and Kevin Murphy. Exact bayesian structure learning from uncertain interventions. In *Artificial Intelligence and Statistics*, pages 107–114, 2007.

Scott Elliott Fahlman. A planning system for robot construction tasks. *Artificial intelligence*, 5(1): 1–49, 1974.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.

Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. Recurrent independent mechanisms. *arXiv preprint arXiv:1909.10893*, 2019.

Anirudh Goyal, Alex Lamb, Phanideep Gampa, Philippe Beaudoin, Sergey Levine, Charles Blundell, Yoshua Bengio, and Michael Mozer. Object files and schemata: Factorizing declarative and procedural knowledge in dynamical systems. *arXiv preprint arXiv:2006.16225*, 2020.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910, 2017.

Nan Rosemary Ke, Olexa Bilaniuk, Anirudh Goyal, Stefan Bauer, Hugo Larochelle, Chris Pal, and Yoshua Bengio. Learning neural causal models from unknown interventions. *arXiv preprint arXiv:1910.01075*, 2019.

Nan Rosemary Ke, Jane Wang, Jovana Mitrovic, Martin Szummer, Danilo J Rezende, et al. Amortized learning of neural causal representations. *arXiv preprint arXiv:2008.09301*, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. *arXiv preprint arXiv:1911.12247*, 2019.

Marlos C Machado, Marc G Bellemare, Erik Talvitie, Joel Veness, Matthew Hausknecht, and Michael Bowling. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research*, 61:523–562, 2018.

Sarthak Mittal, Alex Lamb, Anirudh Goyal, Vikram Voleti, Murray Shanahan, Guillaume Lajoie, Michael Mozer, and Yoshua Bengio. Learning to combine top-down and bottom-up signals in recurrent neural networks with attention over modules. *arXiv preprint arXiv:2006.16981*, 2020.

Suraj Nair, Yuke Zhu, Silvio Savarese, and Li Fei-Fei. Causal induction from visual observations for goal directed tasks. *arXiv preprint arXiv:1910.01751*, 2019.

Alex Nichol, Vicki Pfau, Christopher Hesse, Oleg Klimov, and John Schulman. Gotta learn fast: A new benchmark for generalization in rl. *arXiv preprint arXiv:1804.03720*, 2018.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Ian Osband, Yotam Doron, Matteo Hessel, John Aslanides, Eren Sezener, Andre Saraiva, Katrina McKinney, Tor Lattimore, Csaba Szepezvari, Satinder Singh, et al. Behaviour suite for reinforcement learning. *arXiv preprint arXiv:1908.03568*, 2019.

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

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. MIT press, 2017.

Danilo J Rezende, Ivo Danihelka, George Papamakarios, Nan Rosemary Ke, Ray Jiang, Theophane Weber, Karol Gregor, Hamza Merzic, Fabio Viola, Jane Wang, et al. Causally correct partial models for reinforcement learning. *arXiv preprint arXiv:2002.02836*, 2020.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of The 31st International Conference on Machine Learning*, pages 1278–1286, 2014.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *arXiv preprint arXiv:1911.08265*, 2019.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

David Silver and Kamil Ciosek. Compositional planning using optimal option models. *arXiv preprint arXiv:1206.6473*, 2012.

Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

Andrea Tacchetti, H Francis Song, Pedro AM Mediano, Vinicius Zambaldi, Neil C Rabinowitz, Thore Graepel, Matthew Botvinick, and Peter W Battaglia. Relational forward models for multi-agent learning. *arXiv preprint arXiv:1809.11044*, 2018.

Sjoerd Van Steenkiste, Michael Chang, Klaus Greff, and Jürgen Schmidhuber. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. *arXiv preprint arXiv:1802.10353*, 2018.

Rishi Veerapaneni, John D Co-Reyes, Michael Chang, Michael Janner, Chelsea Finn, Jiajun Wu, Joshua Tenenbaum, and Sergey Levine. Entity abstraction in visual model-based reinforcement learning. In *Conference on Robot Learning*, pages 1439–1456. PMLR, 2020.

Jane X Wang, Michael King, Nicolas Porcel, Zeb Kurth-Nelson, Tina Zhu, Charlie Deck, Peter Choy, Mary Cassin, Malcolm Reynolds, Francis Song, et al. Alchemy: A structured task distribution for meta-reinforcement learning. *arXiv preprint arXiv:2102.02926*, 2021.

Nicholas Watters, Loic Matthey, Matko Bosnjak, Christopher P Burgess, and Alexander Lerchner. Cobra: Data-efficient model-based rl through unsupervised object discovery and curiosity-driven exploration. *arXiv preprint arXiv:1905.09275*, 2019.

Terry Winograd. Understanding natural language. *Cognitive psychology*, 3(1):1–191, 1972.

Patrick H Winston. Learning structural descriptions from examples. 1970.

Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B Tenenbaum. Clevrer: Collision events for video representation and reasoning. *arXiv preprint arXiv:1910.01442*, 2019.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. *arXiv preprint arXiv:1910.10897*, 2019.

Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. DAGs with NO TEARS: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems*, pages 9472–9483, 2018.

# Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We do not use data from other people.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] All our data is created using simulation and does not include any personal information
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]