MODEL-INVARIANT STATE ABSTRACTIONS FOR MODEL-BASED REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Accuracy and generalization of dynamics models is key to the success of model-based reinforcement learning (MBRL). As the complexity of tasks increases, learning dynamics models becomes increasingly sample inefficient for MBRL methods. However, many tasks also exhibit sparsity in the dynamics, i.e., actions have only a local effect on the system dynamics. In this paper, we exploit this property with a causal invariance perspective in the single-task setting, introducing a new type of state abstraction called *model-invariance*. Unlike previous forms of state abstractions, a model-invariance state abstraction leverages causal sparsity over state variables. This allows for generalization to novel combinations of unseen values of state variables, something that non-factored forms of state abstractions cannot do. We prove that an optimal policy can be learned over this model-invariance state abstraction. Next, we propose a practical method to approximately learn a model-invariant representation for complex domains. We validate our approach by showing improved modeling performance over standard maximum likelihood approaches on challenging tasks, such as the MuJoCo-based Humanoid. Furthermore, within the MBRL setting we show strong performance gains w.r.t. sample efficiency across a host of other continuous control tasks.

1 INTRODUCTION

Model-based reinforcement learning (MBRL) is a popular framework for data-efficient learning of control policies. At the core of MBRL is learning an environmental dynamics model and using it to: 1) fully plan Deisenroth & Rasmussen (2011); Chua et al. (2018), 2) augment the data used by a model-free solver Sutton (1991), or 3) be used as an auxiliary task while training Lee et al. (2019); Zhang et al. (2021). However, learning a dynamics model — similar to other supervised learning problems — suffers from the issue of generalization since the data we train on is not necessarily the data we test on. This is a persisting issue that is worsened in MBRL as even a small inaccuracy in the dynamics model or changes in the control policy can result in visiting completely unexplored parts of the state space. Thus, it is generally considered beneficial to learn models capable of generalizing well. Various workarounds for this issue have been explored in the past; for example coupling the model and policy learning processes (Lambert et al., 2020) so that the model is always accurate to a certain threshold, or using an ensemble of models to handle the uncertainty in each estimate (Chua et al., 2018). However these approaches are unnecessarily pessimistic, and do not leverage structure in factored dynamics for better generalization.

In this paper, we study how to improve generalization capabilities through careful state abstraction. In particular, we leverage two existing concepts to produce a novel combination in MBRL that yields improved generalization performance. The first concept is the principle of causal invariance, which dictates that given a set of features, we should aim to build representations that comprise *only* those features that are consistently necessary for predicting the target variable of interest across different interventions (Peters et al., 2015). The intuition is that a predictor built only over such invariant features should generalize well for all possible shifts in the data distribution. The second concept is that many real world problems exhibit sparsity in the local dynamics — given a set of state variables, each variable only depends on a subset of those variables in the previous timestep. The two concepts of sparsity and causality are intertwined, in that they both are a form of inductive biases that surround the agent dynamics Goyal & Bengio (2020). The policy of a continuously improving learner is crucial, as it allows for the realization of both causal invariance and sparsity.

We focus on the prediction problem corresponding to learning a model of a Contextual Decision Process (CDP) Krishnamurthy et al. (2016), a generalization of the typical Markov decision process that also encompasses rich and partial observability settings (see Section 2.1 for details). Causal invariance in the CDP setting can be considered a supervised learning problem where the features are the state and action variables (the probable set of causal predictors for the target) and the target variables are the state variables of the next state. In this context, we ask the question, *can we exploit the idea of causal invariance to learn a model with improved generalization ability to unseen parts of the state-action space?* Ultimately, based on experimental results we will show that the answer is "*yes*." Given basic exploratory assumptions, we show both theoretically and empirically that we can learn a model that generalizes well on state distributions induced by any policy distinct from the ones used while learning it.

The contributions of this paper are as follows. 1) We highlight an important concept required to answer this question, that of independence between state variables in a dynamics model. We leverage this observation to propose a new kind of state abstraction, *model-invariance*. model-invariance is similar in flavour to model irrelevance (Li, 2009) but applies to individual state variables instead of the full state as a whole. This leverages natural sparsity over state variables by constructing coarser state abstractions on a per-variable level, also allowing for new generalization capabilities over novel compositions of state variable values. 2) We show that a representation that only uses the causal parents of each state variable is, in fact, a model-invariant representation. 3) We show that learning a model over such an abstraction, and then planning using this model, is optimal, given certain exploratory assumptions on the CDP. 4) We perform a proof-of-concept experiment in the batch setting to show that such a model learning approach always leads to better generalization in unseen parts of the state space for this CDP. 5) We then introduce a practical method which approximates learning a model-invariant representation for more complex domains. 6) We empirically show that our approach results in better model generalization for domains such as the MuJoCo-based Humanoid and follow this by combining our model learning scheme with a policy optimization framework which leads to improvements in sample efficiency.

We believe that the proposed algorithm is an important step towards leveraging sparsity in complex environments and to improve generalization in MBRL methods.

2 Preliminaries

We now formalize and discuss the foundational concepts used in our work.

2.1 Problem Formulation

We consider the agent's interaction with the environment as a discrete time γ-discounted Contextual Decision Process (CDP), a term recently proposed by Krishnamurthy et al. (2016) to broadly model sequential decision processes which require the policy to be based on rich features (context). A CDP is defined as $\mathcal{M} = (\mathcal{X}, \mathcal{A}, P, R, \gamma, \mu)$, where $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{A}$ are the state and action spaces; $P \equiv P(x'|x, a)$ is the transition kernel; $R \equiv r(x, a)$ is the reward function with the maximum value of R_{max}; $\gamma \in (0, 1)$ is the discount factor; and μ is the initial state distribution. CDPs generalize MDPs by unifying decision problems that depend on rich context. Let $\pi : \mathcal{X} \to \Delta_{\mathcal{A}}$ be a stationary Markovian policy, where $\Delta_{\mathcal{A}}$ is the set of probability distributions on $\mathcal{A}$. The discounted frequency of visiting a state s by following a policy π is defined as $\rho_\pi(x) \equiv (1-\gamma)\mathbb{E}[\sum_{t\geq 0} \gamma^t \mathbb{I}\{x_t = x\} \mid \mu, \pi]$. The value function of a policy π at a context $x \in \mathcal{X}$ is defined as $V^\pi(x) \equiv \mathbb{E}[\sum_{t\geq 0} \gamma^t r(x_t, a_t)|x_0 = x, \pi]$. Similarly, the action-value function of π is defined as $Q^\pi(x, a) = \mathbb{E}[\sum_{t\geq 0} \gamma^t r(x_t, a_t)|x_0 = x, a_0 = a, \pi]$. The CDP definition also assumes that there exists a set of latent states $\mathcal{S}$, finite in number, which are latent. If we pose further structural assumptions, such as that of a Block MDP Du et al. (2019); Zhang et al. (2020), then the notion of S becomes more apparent.

There are two important cases we can consider with CDPs. We explore these with simple examples:

Case 1: Large state space or full state input: Consider $\mathcal{X}$ as the proprioceptive states of a robot. In this case, $\mathcal{X}$ is not a rich observation, but rather an arbitrarily large set of state variables $\{x^1, x^2, ..., x^p\}$. There is likely to be little irrelevant information present w.r.t. the downstream task in such a case, i.e., the latent state space and observation space are the same, $\mathcal{S} := \mathcal{X}$. Here, the model-

invariant abstraction $S_i \in \mathcal{S}$, conditioned on a specific state variable X_i, corresponds to some coarser abstraction of the given full state, learning and planning over which can still be optimal.

Case 2: Rich observation or pixel based input: Consider $\mathcal{X}$ to be a set of images, for example, each being a front view of a robot. There is irrelevant information present in the form of background pixels. Nevertheless, the latent state set $\mathcal{S}$ is still the same as in the previous case, a coarse representation of the rich observation space $\mathcal{X}$. Our task here is more challenging, in that we first have to compress a low-dimensional state of the robot from the image that exhibits sparsity (equivalent to what is given in case 1) and then learn a model-invariant representation. Also note that, for us to consider case 2 as tractable, at least theoretically, we would have to assume a block MDP structure, since otherwise having access to just the observations can lead to a POMDP setting.

In this work, we focus on case 1 and from now one use the term CDP and MDP interchangeably throughout the paper. However, we remain general in our setup description since case 2 becomes immediately relevant if we have a method of learning a compressed representation with sparseness properties, which makes our method applicable. In both cases, we assume that the transition dynamics over the full state are factorized. More formally:

Assumption 1. *(Transition Factorization) For given full state vectors $x_t, x_{t+1} \in \mathcal{X}$, action $a \in \mathcal{A}$, and x_i denoting the ith dimension of state x we have $P(x_{t+1}|x_t, a) = \prod_i P(x^i_{t+1}|x_t, a)$.*

Note that this is a weaker assumption than assuming factored MDPs (Kearns & Koller, 1999; Guestrin et al., 2001) since we do not assume a corresponding factorization of the reward function.

2.2 Invariant Causal Prediction

Invariant causal prediction (ICP) (Peters et al., 2015) considers learning an invariant representation w.r.t. spurious correlations that arise due to noise in the underlying (unknown) causal model which describes a given system. The key idea is that across different environments (generally defined by interventions on the data), the response variable Y remains the same given the variables X_i that directly cause the response variable, i.e., its parents $\mathbf{PA}(Y)$.

2.3 Model-based Reinforcement Learning

Model-based reinforcement learning typically involves learning a dynamics model of the environment by fitting it using a maximum-likelihood estimate of the trajectory-based data collected by running some exploratory policy. Such a learned model can then be used with various control methods. Specifically, some popular approaches include using the model 1) to plan for the policy using techniques such as model predictive control (MPC) Williams et al. (2017); Chua et al. (2018); Nagabandi et al. (2018), 2) to improve estimates of the Q value by rolling out the model for a small number of steps Feinberg et al. (2018); Amos et al. (2020) and 3) to provide synthetic data samples for a model-free learner Janner et al. (2019); Kurutach et al. (2018). In the offline/batch RL setting, where we only have access to the data collected by multiple policies, recent techniques build on the idea of pessimism (regularizing the original problem based on how confident the agent is about the learned model) and have resulted in better sample complexity over model-free methods on benchmark domains (Kidambi et al., 2020; Yu et al., 2020).

2.4 State Abstractions and Model Irrelevance

State abstractions allow us to map behaviorally equivalent states into a single abstract state, thus simplifying the learning problem which then makes use of the (potentially much smaller set of) abstract states instead of the original states (Bertsekas & Castanon, 1989). In theory, any function approximation architecture can act as an abstraction, since it attempts to group similar states together. Therefore, it is worth exploring the properties of a representation learning scheme as a state abstraction. In the rest of the paper, we build our theory based on this connection.

We are interested in a specific kind of state abstraction called model irrelevance state abstraction or bisimulation (Even-Dar & Mansour, 2003; Ravindran & Barto, 2004; Li, 2009). An abstraction $\phi : \mathcal{X} \mapsto \mathcal{S}$ is model irrelevant if for any two states $x, x' \in \mathcal{X}$, abstract state $s \in \mathcal{S}$, $a \in \mathcal{A}$ where

$\phi(x) = \phi(x')$,

$$R(x, a) = R(x', a),$$

$$\sum_{x'' \in \phi^{-1}(s)} P(x''|x, a) = \sum_{x'' \in \phi^{-1}(s)} P(x''|x', a)\,.$$

Since an exact equivalence is not practical, prior work deals with approximate variants through the notion of ϵ-closeness (Jiang, 2018). The main difference between a model irrelevance state abstraction and our proposed model-invariance state abstraction is that the model irrelevance abstraction does not leverage sparsity in factored dynamics. Our model-invariance state abstraction is variable specific, assuming the state space consists of a set of state variables. We formally define our model-invariance state abstraction in Section 3.

3 Casual Invariance in Model Learning

In this section, we build towards our goal of learning a generalizable transition model, given limited environment data. We first highlight how the independence assumption (Assumption 1) connects to this central goal by introducing a new kind of state abstractions called model-invariance.

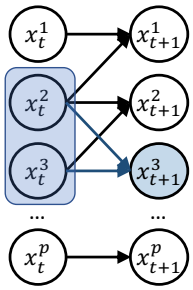


Figure 1: Graphical model of sparsity across state variables. Sparsity example: The dimension x_{t+1}^3 (shaded in blue) only depends on two dimensions x_t^3 and x_t^2 (in the blue box).

3.1 Model Invariant Abstractions

Given conditional independence over state variables, we define model-invariance as an abstraction that preserves transition behavior for each state variable. Formally, we define a reward-free version as follows:

Definition 1. *(Model Invariant Abstraction) ϕ_i is model-invariant if for any $x, x', x'' \in \mathcal{X}, a \in \mathcal{A}$, $\phi_i(x) = \phi_i(x')$ if and only if*

$$P(x_i''|x, a) = P(x_i''|x', a), \tag{1}$$

where x_i'' denotes the value of state variable i in state x''.

In words, an invariant abstraction is one which has the same transition probability to next state for any two given states x and x', in the ith index. If we assume factored rewards, we can define a corresponding reward-based invariant abstraction that parallels the model-irrelevance abstraction more closely, but we focus here on the reward-free setting.

Since it is impractical to ensure this equivalence exactly, we introduce an approximate definition which ensures an ϵ-closeness.

Definition 2. *(Approximate Model Invariant Abstraction) ϕ is $\epsilon_{i,P}$-model-invariant if for each index i,*

$$\sup_{\substack{a \in \mathcal{A}, \\ x, x' \in \mathcal{X}, \phi(x) := \phi(x')}} \|\Phi_i P(x''|x, a) - \Phi_i P(x''|x', a)\| \leq \epsilon_{i,P}.$$

ϕ is ϵ_R-model-invariant if

$$\epsilon_R := \sup_{\substack{a \in \mathcal{A}, \\ x, x' \in \mathcal{X}, \phi(x) = \phi(x')}} \left|R(x, a) - R(x', a)\right|.$$

$\Phi_i P$ denotes the *lifted* version of P, where we take the next-step transition distribution from observation space $\mathcal{X}$ and lift it to latent space $\mathcal{S}$.

Lemma 1. *(Model Error Bound) Let ϕ be an $\epsilon_{i,P}$-approximate model-invariant abstraction on CDP M. Given any distributions $p_{x_i} : x_i \in \phi_i(\mathcal{X})$ where $p_x = \prod_{i=1}^p p_{x_i}$ is supported on $\phi^{-1}(x_i)$, we define $M_\phi = (\phi_i(\mathcal{X}), \mathcal{A}, P_\phi, R_\phi, \gamma)$ where $P_\phi(x, a) = \prod_{i=1}^p P_{\phi_i}(x, a)$. Then for any $x \in \mathcal{X}$, $a \in \mathcal{A}$,*

$$\|P_\phi(x, a) - \Phi P(x, a)\| \leq \sum_{i=1}^p \epsilon_{i,P}.$$

Proof in Section B. Lemma 1 provides a bound on the modelling error when the individual errors for an approximate model-invariant abstraction are compounded. Specifically, P_ϕ refers to the transition probability of a CDP which acts on the states $\Phi(\mathcal{X})$, rather than the original CDP which acts on the original states. Note that we are particularly concerned with the case where each x_i is atomic in nature, i.e., it is not divisible further. Such a property ensures that model-invariance does not collapse to model irrelevance.

4 Theoretical Results

We now move on to providing a connection between causal invariance and model-invariant abstractions. First, we describe the causal setup below:

Definition 3. *(Causal Setup) For each future state variable indexed by i, x^i_{t+1}, there exists a linear structural equation model consisting of state dimensions and actions, $(x^i_{t+1}, x^1_t, ..., x^p_t, a_t)$ with coefficients $(\beta_{jk})_{j,k=1,...,p+2}$, given by a directed acyclic graph. An experimental setting $e \in \mathcal{E}$ arises due to one or more interventions on the variable set $\{x^1_t, ..., x^p_t, a_t\}$, with the exception of X^i_{t+1}.*

Assumption 2. *(Invariant Prediction Peters et al. (2015)) For each $e \in \mathcal{E}$: the experimental setting e arises due to one or several interventions on variables from $(x^1_t, ..., x^p_t, a_t)$ but not on x^i_{t+1}; here, we allow for do-interventions Pearl (2009) or soft-interventions Eberhardt & Scheines (2007).*

For our purposes, each intervention corresponds to a change in the action distribution, i.e., policy. Thus, in turn, each policy π_i defines an environment e.

Proposition 1. *(Causal Feature Set Existence) Under Assumption 2 the direct causes, i.e., parents of x^i_{t+1} define a valid support over invariant predictors, namely $S^* =$ **PA**(x^i_{t+1}).*

The proof follows directly by applying Proposition 1 of Peters et al. (2015) (which itself follows from construction) to each dimension i.

Now that we consider each state variable individually, we wish to incorporate the causal invariance idea into the model prediction problem for each state variable. The key idea is to make sure that in predicting each state variable we use only its set of invariant predictors and not all state variables and actions (see Figure 1).

With this intuition, it becomes clearer why our original model learning problem is inherently tied with learning better representations, in that having access to a representation which discards excess information for each state variable (more formally, a casually invariant representation), would be more suited to learning an accurate model over and thus, at least in principle, lead to improved generalization performance across different parts of the state space. We now show that such a casually invariant representation is in fact a model-invariant abstraction.

Theorem 1. *For the abstraction $\phi_i(x) = [x]_{S_i}$, where $S_i =$ **PA**(x^i_{t+1}), ϕ_i is model-invariant.*

Proof in Appendix B. Next, we show that learning a transition model over a model-invariant abstraction ϕ and then planning over this model is optimal.

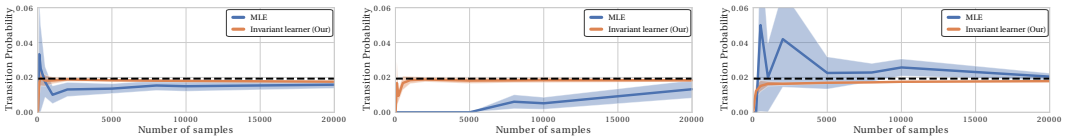
Assumption 3. *(Concentratability Coefficient, Chen & Jiang (2019)) There exists $C < \infty$ such that for any admissible distribution ν,*

$$\forall (x, a) \in \mathcal{X} \times \mathcal{A}, \ \frac{\nu(x, a)}{\mu(x, a)} < C \, .$$

Here, an admissible distribution refers to any distribution that can be realized in the given CDP by following a policy for some timesteps. μ refers to the distribution the data is generated from.

Theorem 2. *(Value bound) If ϕ is an $\epsilon_R, \epsilon_{i,P}$ approximate model-invariant abstraction on CDP M, and M_ϕ is the abstract CDP formed using ϕ, then we can bound the loss in the optimal state action value function in both the CDPs as:*

$$\|[Q^*_{M_\phi}]_M - Q^*_M\|_{2,\nu} \leq \frac{\sqrt{C}}{1-\gamma} \|[Q^*_{M_\phi}]_M - \mathcal{T}[Q^*_{M_\phi}]_M\|_{2,\mu}$$



(a) π_1 (b) π_2 (c) π_3

Figure 2: Consider the network topology CDP (Guestrin et al., 2001). We compare the mean and standard error over 10 random seeds of the estimated transition probability of our invariant learner (orange curve) and MLE (blue curve). π_1 is a policy that restarts whichever machine (based on index order) is not working and does nothing if all machines are working. π_2 is a random policy. π_3 restarts the middle machine most of the times, while acting randomly otherwise. We can see how our invariant learner converges faster and more stably to the common solution (dashed black curve).

$$\|[Q^*_{M_\phi}]_M - \mathcal{T}[Q^*_{M_\phi}]_M\|_{2,\mu} \leq \epsilon_R + \gamma\Big(\sum_{i=1}^{p} \epsilon_{i,P}\Big)\frac{R_{max}}{(2(1-\gamma))}$$

Proof and all details surrounding the theoretical results are provided in Appendix B.

5 Proof of Concept Experiment: Certainty Equivalence

In the tabular case, estimating the model using transition samples and then planning over the learned model is referred to as certainty equivalence (Bertsekas, 1995). Particularly for estimating the transition model, it considers the case where we are provided with n transition samples per state-action pair, (x_t, a_t) in the dataset $D_{x,a}$, and estimate the model as

$$P(x_{t+1}|x_t, a_t) = \frac{1}{n}\sum_{\bar{x}\in D_{x,a}} \mathbb{I}(\bar{x} = x_{t+1})\,. \tag{2}$$

If we assume that the next state components do not depend on each other given the previous state and action (i.e., Assumption 1), we can re-write $P(x_{t+1}|x_t, a_t)$ as $\prod_i P(x^i_{t+1}|x_t, a_t)$. Assuming we know the parents of x^i_{t+1}, we can instead empirically estimate the true transition probabilities as

$$\begin{aligned} P(x^i_{t+1}|x_t, a_t) &= P(x^i_{t+1}|\mathbf{PA}(x^i_{t+1}), a_t) \\ &= \frac{1}{nk}\sum_{\bar{x}\in D} \mathbb{I}(\bar{x}^i = x^i_{t+1})\,, \end{aligned} \tag{3}$$

where $D = \bigcup_{i=1}^{k} D_{x,a}, x \in \phi_i^{-1}(\bar{x})$. In the tabular case, Eq. 2 corresponds to a solution obtained by a standard maximum likelihood learner. On the other hand, Eq. 3 corresponds to a solution obtained by an invariant model learner. Proposition 1 showed that such an invariant solution exists for the given causal abstraction definition. Here, assuming we have access to such an abstraction (i.e. access to parent information for each state variable), we aim to show on a simple MDP that the invariance based solution performs zero shot generalization to unseen parts of the state space while the standard model learner does not.

We consider the simple network topology domain introduced in Guestrin et al. (2001). The setup involves a star based topology comprising five machines. At each step, a machine fails randomly, increasing the chances of the adjacent machine failing. Moreover, at each step, a single machine can be restarted, resulting in a lower chance of it failing in the subsequent time step. Our objective here is to estimate the transition probability for a given (x_t, a_t, x_{t+1}) tuple using the two methods in Eq. 2 and Eq. 3. In Figure 2, we compare our invariant learner (orange curve) against a standard MLE learner (blue curve) and study for three different policies how their estimate varies as the number of samples grows.

Note that Figure 2 is specified by a fixed policy that is used for data collection. If the policy changes, it would result in a different environment as described in Section 3. Our ideal scenario is to find a predictive model that is optimal for all environments. To show this generalization, we find that the invariant learner quickly converges to approximately the same solution across all training environments, in just few data samples. The solution for any test environment is therefore this common solution. As can be seen, this common solution (i.e., 0.02) also coincides with the true probability we are trying to estimate. On the other hand, the standard MLE learner results in different

Algorithm 1 Model-Invariant MBRL

Initialize Replay buffer $\mathcal{D} = \emptyset$; Value and policy network parameters θ_Q, θ_π corresponding to any model based RL algorithm;
for environment steps $t = 1, \ldots, T$ **do**
  Take action $a_t \sim \pi(\cdot|x_t)$, observe r_t and x_{t+1}, and add to the replay buffer $\mathcal{D}$;
  for $M_{\text{model-free}}$ updates **do**
    Sample a batch $\{(x_j, a_j, r_j, x_{j+1})\}_{j=1}^{N}$ from $\mathcal{D}$;
    Run gradient update for the model free components of the algorithm (e.g. θ_π, θ_Q etc.)
  end for
  for M_{model} updates **do**
    Sample a batch $\{(x_j, a_j, r_j, x_{j+1})\}_{j=1}^{N}$ from $\mathcal{D}$;
    Run gradient update for reward model (θ_r)
    Run gradient update for invariant dynamics model: $\theta_f \leftarrow$ invariant_update$(\theta_f, \nabla_{\theta_f} L_f)$ (Pseudocode C.1)
  end for
end for

solutions for each training environment in the low data regime. The solution provided at test time in such a case is an average of all such solutions found during training, which is clearly off the true probability.

It is worth noting that this example is only a proof of concept and that in more complex domains, we do not assume access to the causal parents of each state variable. To that end, in the next section we will describe a practical method that leverages the ideas presented until now.

6 Towards Learning Practical Model-Invariant Representations

We now introduce a practical algorithm for learning model-invariant representations. The main idea is to use two (or more) independent models for dynamics prediction and constraining their predictions to be close to each other for individual state variables (see Figure 3). Specifically, we instantiate two identical models at the start of training. At each optimization step, a model is sampled randomly and is used for minimizing the standard MLE model predictive loss. Simultaneously, an invariance loss defined over the predictions of both models is attached to the main objective. The role of the invariance loss is essentially to *minimize the difference in similarity between the prediction of one model w.r.t. the predictions of the second model and vice versa* (Eq. 4).

An important detail to note is that this similarity is computed for a single state variable (randomly selected) at each training step. The overall rationale is that the invariance loss would implicitly force each model to only depend on the causal parents of each state variable. We borrow the specifics of the similarity definition from Mitrovic et al. (2020) and detail out our exact implementation of the invariance loss in pseudocode form in Appendix C.1.

The overall loss used to learn the dynamics model is thus

$$\mathcal{L}_f = \mathbb{E}_{x\sim\mathcal{D}}\Big[\underbrace{\Big(f(x_t, a_t) - x_{t+1}\Big)^2}_{\text{Standard MLE Loss}} + \underbrace{\text{KL}\big(\psi^i(f,h),\ \psi^i(h,f)\big)}_{\text{Invariance Loss}}\Big] \quad (4)$$

where $\psi^i(f,h) = \Big\langle g(f^i(x_t,a_t)), g(h^i(x_t,a_t))\Big\rangle$ is the similarity between the predictions for the models f and h for the state variable indexed by i. The function g is popularly known as the critic in self-supervised learning losses Chen et al. (2020).

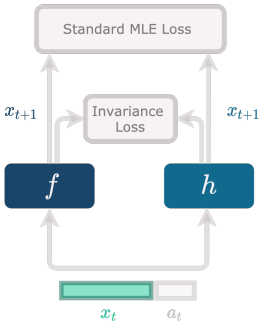


Figure 3: Architecture for learning model-invariant representations.

Eventually, we wish to use the invariant model learner described above within a model based policy optimization algorithm and check for how the policy performance varies as compared to a standard MLE based model learner. There are multiple ways of incorporating a model for policy optimization in RL. A general framework that utilizes an invariant model learner is outlined in Algorithm 1. For the purposes of this paper, we employ a simple actor-critic setup where the model is used to compute multi-step estimates of the Q value used by the actor learner. A specific instantiation of this idea of model value expansion is the SAC-SVG algorithm proposed in Amos et al. (2020). It is important

to note that the proposed version of model-invariance can be used in combination with any MBRL method, and with any type of model architecture, such as ensembles or recurrent architectures.

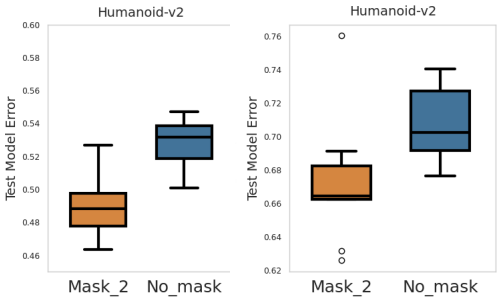
7 EXPERIMENTS

Our experiments address the following questions:

- Moving to more complex control tasks, can we visibly show the adverse effects of spurious correlations arising due to learning the model as the policy distribution, and thus the state distribution changes during learning (Section 7.1)?
- How does our invariant model learning scheme performs in comparison to a standard MLE based model learner on more challenging tasks? Does the performance gain, if any, has any correlation with the number of samples, i.e., amount of data available (Section 7.2)?
- How does learning an invariant model affect performance in a model based policy optimization algorithm? Does learning a more accurate model results in more sample efficient algorithms (Section 7.3)?

7.1 PRESENCE OF SPURIOUS CORRELATIONS

To test the presence of spurious correlations when learning the dynamics model, we present three particular cases. For the Humanoid-v2 domain, we choose to predict a single dimension (the knee joint) when 1) **No Mask:** the entire current observation and action are provided as input, 2) **Mask_1:** when the dimensions that are likely to be useful in predicting the knee joint are masked and 3) **Mask_2:** when the dimensions that seem uncorrelated to the knee joint are masked. Having trained different models for all three cases, we observe that the model error, i.e., loss for case 2) is the most, as would be expected. Furthermore, we see that 1) performs worse than 3), for both horizon values in $\{3, 5\}$ (see Figure 4). This indicates that there indeed is an invariant, casual set of parents among the observation dimensions and that there could be some interference due to spurious correlations in 1) and thus it performs worse than case 3).



(a) Horizon=3 (b) Horizon=5

Figure 4: Effect of spurious correlation on the model learning test loss of Humanoid-v2. We compare model loss of predicting a single dimension (the knee joint) for two schemes: Mask_2 and No_mask. No_mask performs worse, thus supporting the claim that spurious correlations do exist per state variable. Each curve is run for 10 seeds, with the standard deviation shaded. Y-axis magnitude order is 1e-3.

7.2 INVARIANT MODEL LEARNING ON HUMANOID-V2

We compare the invariant model learner to a standard model learner for the Humanoid-v2 task. To observe the effect of the invariance loss clearly, we decouple the model learning component from the policy optimization component by testing the model on data coming from a replay buffer a pre-trained model-free SAC agent.

Such a setup ensures that the change in state distribution according to changes in policy is still present, which is necessary to test the generalization performance of a learned model. We observe that our invariant model learner performs much better than the standard model learner, especially when the number of samples available is low, i.e., around the 200k to 500k mark (see

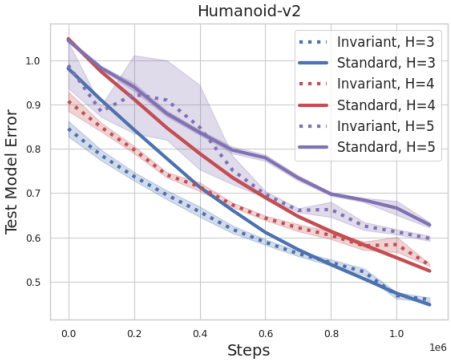


Figure 5: Test model learning error on Humanoid-v2 for different horizon values. We see that the invariant learner consistently generalizes better than the standard model learner. Each curve is the mean and standard error over 10 random seeds.

	POPLIN Cheetah	POPLIN Walker	POPLIN Hopper	POPLIN Ant	MBPO Cheetah	MBPO Walker	MBPO Hopper	MBPO Ant	MBPO Humanoid
*PETS	2288 ±510	282 ± 250	114 ± 311	1165 ± 113	-	-	-	-	-
*POPLIN-A	1562 ±568	-105 ± 125	202 ± 481	1148 ± 219	-	-	-	-	-
*POPLIN-P	4235 ±566	**597** ± 239	**2055** ± 206	2330 ± 160	-	-	-	-	-
*METRPO	2283 ± 450	-1609 ±328	1272 ± 250	282 ± 9	-	-	-	-	-
*SAC	4035 ± 134	-382 ± 424	2020 ± 346	836 ± 34	-	-	-	-	-
SAC-SVG H-3	8211 ±408	-242 ± 606	1869 ±389	3977 ± 357	**7296** ±462	**3274** ±364	3055 ± 91	3090 ± 160	441 ± 33
Ours H-3	**8509** ±470	-768 ± 427	1801 ± 355	**4521** ± 307	7253 ±395	2882 ±416	**3090** ± 109	**3424** ± 320	447 ± 9
SAC-SVG H-4	-	-	-	-	6917 ±564	3190 ±374	3109 ±1126	828 ± 435	538 ± 64
Ours H-4	-	-	-	-	**7206** ± 327	**3392** ±407	**3204** ± 115	**2222** ±383	494 ± 43
SAC-SVG H-5	-	-	-	-	4305 ±1025	2538 ±492	2820 ± 316	**2440** ±479	576 ± 63
Ours H-5	-	-	-	-	**6602** ± 345	**2916** ±442	**3009** ± 280	2162 ±490	463 ± 50
Timesteps	200000	200000	200000	200000	200000	200000	200000	200000	200000

Table 1: Invariant MBRL performance on four MuJoCo based domains from POPLIN Wang & Ba (2019) (left) and five MuJoCo based domains from MBPO Janner et al. (2019) (right). * represents performance reported by POPLIN. We run our method for 10 seeds and report the standard error for all methods.

Figure 5). As the number of samples increases, the performance between both models converges, just as observed in the tabular case. This is expected since in the infinite data regime, both solutions (MLE and invariance based) approach the optimal/true model. Furthermore, we observe that the number of samples it takes for convergence of between the standard and the invariant model learners increases as the rollout horizon (H in Figure 5) of the model learner is increased.

7.3 Invariant Model-based Reinforcement Learning

Finally, we evaluate the invariant model learner within the the policy optimization setting of SAC-SVG (Amos et al., 2020). We compare the difference in performance to SAC-SVG when the horizon length is varied (see MBPO environments in Table 1) and then compare the performance of our method against multiple model based methods including PETS Chua et al. (2018), POPLIN Wang & Ba (2019), METRPO Kurutach et al. (2018), and the model free SAC Haarnoja et al. (2018) algorithm (see POPLIN environments in Table 1). The results show improved performance when the invariant model learner is used instead of the standard model learner across most tasks. Interestingly, the improvement we see in modelling performance is not translated as well in policy optimization performance for the Humanoid-v2 task. It is worth noting that recently Lambert et al. (2020) point out that in some RL tasks, modelling performance could actually be uncorrelated to the policy's performance. Combining our invariant model learner with other policy optimization algorithms is therefore a promising direction for future investigation.

8 Conclusion and Future Directions

This paper introduced a new type of state abstraction for MBRL that exploits the inherent sparsity present in many complex tasks. We first showed that a representation which only depends on the causal parents of each state variable follows this definition and is provably optimal. Following, we introduced a novel approach for learning model-invariant abstractions in practice, which can plugged in any given MBRL method. Experimental results show that this approach measurably improves the generalization ability of the learnt models. This stands as an important first step to building more advanced algorithms with improved generalization for systems that possess sparse dynamics.

In terms of future work, there remain multiple exciting directions and open questions. First, to enable model-invariance, we could also look at other kind of approaches proposed recently such as the AND mask Parascandolo et al. (2020). The AND mask specifically requires the data separated into multiple environments, and thus looks much more suited for offline RL where we have data collected based on multiple policies available. Second, moving to pixel based input, the representation learning task becomes two-fold, including learning to abstract away the irrelevant information present in the pixels and then learning a model-invariant representation. Third, note that our theoretical results do not involve an explicit dependence on a sparsity measure, for example, the maximum number of parents any state variable could have. Including such a dependence would ensure tighter bounds. Fourth, it is worth asking how such an explicit constraint on model-invariance can perform as a standalone representation learning objective, considering the strong progress made by self-supervised RL.

REFERENCES

Brandon Amos, Samuel Stanton, Denis Yarats, and Andrew Gordon Wilson. On the model-based stochastic value gradient for continuous reinforcement learning. *arXiv preprint arXiv:2008.12775*, 2020.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

D. P. Bertsekas and D. A. Castanon. Adaptive aggregation methods for infinite horizon dynamic programming. *IEEE Transactions on Automatic Control*, 34(6):589–598, 1989. doi: 10.1109/9.24227.

Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 1st edition, 1995. ISBN 1886529124.

Craig Boutilier, Thomas Dean, and Steve Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning*, pp. 1042–1051. PMLR, 2019.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pp. 1597–1607, 2020.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 4754–4765, 2018.

Marc Deisenroth and Carl E Rasmussen. PILCO: A model-based and data-efficient approach to policy search. In *International Conference on Machine Learning (ICML)*, pp. 465–472, 2011.

Simon Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudik, and John Langford. Provably efficient RL with rich observations via latent state decoding. In *International Conference on Machine Learning (ICML)*, pp. 1665–1674, 2019.

Frederick Eberhardt and Richard Scheines. Interventions and causal inference. *Philosophy of science*, 74(5):981–995, 2007.

Eyal Even-Dar and Yishay Mansour. Approximate equivalence of markov decision processes. In *Learning Theory and Kernel Machines*, pp. 581–594. Springer, 2003.

Vladimir Feinberg, Alvin Wan, Ion Stoica, Michael I Jordan, Joseph E Gonzalez, and Sergey Levine. Model-based value expansion for efficient model-free reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2018.

Anirudh Goyal and Yoshua Bengio. Inductive biases for deep learning of higher-level cognition. *arXiv preprint arXiv:2011.15091*, 2020.

Carlos Guestrin, Daphne Koller, and Ronald Parr. Max-norm projections for factored MDPs. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, volume 1, pp. 673–682, 2001.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pp. 1861–1870. PMLR, 2018.

Assaf Hallak, François Schnitzler, Timothy Mann, and Shie Mannor. Off-policy model-based learning under unknown factored dynamics. In *International Conference on Machine Learning (ICML)*, pp. 711–719, 2015.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, pp. 12519–12530, 2019.

Nan Jiang. Notes on state abstractions, 2018. URL `http://nanjiang.cs.illinois.edu/files/cs598/note4.pdf`.

Anders Jonsson and Andrew Barto. Causal graph based decomposition of factored mdps. *J. Mach. Learn. Res.*, 7:2259–2301, December 2006. ISSN 1532-4435.

Michael Kearns and Daphne Koller. Efficient reinforcement learning in factored MDPs. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, volume 16, pp. 740–747, 1999.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. *arXiv preprint arXiv:2005.05951*, 2020.

Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Pac reinforcement learning with rich observations. In *Advances in Neural Information Processing Systems*, pp. 1840–1848, 2016.

Thanard Kurutach, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. Model-ensemble trust-region policy optimization. *arXiv preprint arXiv:1802.10592*, 2018.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. *Learning for Dynamics and Control (L4DC)*, pp. 761–770, 2020. URL `http://arxiv.org/abs/2002.04523`.

Alex X Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *arXiv preprint arXiv:1907.00953*, 2019.

Lihong Li. *A unifying framework for computational reinforcement learning theory*. PhD thesis, Rutgers University-Graduate School-New Brunswick, 2009.

Dipendra Misra, Mikael Henaff, Akshay Krishnamurthy, and John Langford. Kinematic state abstraction and provably efficient rich-observation reinforcement learning. In *International Conference on Machine Learning (ICML)*, pp. 6961–6971, 2020.

Dipendra Misra, Qinghua Liu, Chi Jin, and John Langford. Provable rich observation reinforcement learning with combinatorial latent states. In *International Conference on Learning Representations (ICLR)*, 2021. URL `https://openreview.net/forum?id=hx1IXFHAw7R`.

Jovana Mitrovic, Brian McWilliams, Jacob Walker, Lars Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms. *arXiv preprint arXiv:2010.07922*, 2020.

Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7559–7566, 2018.

Ian Osband and Benjamin Van Roy. Near-optimal reinforcement learning in factored mdps. In *Advances in Neural Information Processing Systems*, pp. 604–612, 2014.

Giambattista Parascandolo, Alexander Neitz, Antonio Orvieto, Luigi Gresele, and Bernhard Schölkopf. Learning explanations that are hard to vary. *arXiv preprint arXiv:2009.00329*, 2020.

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

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference using invariant prediction: identification and confidence intervals. *arXiv preprint arXiv:1501.01332*, 2015.

Balaraman Ravindran and Andrew G Barto. *An algebraic approach to abstraction in reinforcement learning*. PhD thesis, University of Massachusetts at Amherst, 2004.

Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. *arXiv preprint arXiv:2010.05761*, 2020.

Alexander L Strehl, Carlos Diuk, and Michael L Littman. Efficient structure learning in factored-state mdps. In *AAAI*, volume 7, pp. 645–650, 2007.

Wen Sun, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Model-based rl in contextual decision processes: Pac bounds and exponential improvements over model-free approaches. In *Conference on Learning Theory*, pp. 2898–2933. PMLR, 2019.

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

Tingwu Wang and Jimmy Ba. Exploring model-based planning with policy networks. *arXiv preprint arXiv:1906.08649*, 2019.

Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M Rehg, Byron Boots, and Evangelos A Theodorou. Information theoretic MPC for model-based reinforcement learning. In *International Conference on Robotics and Automation (ICRA)*, pp. 1714–1721, 2017. doi: 10.1109/ICRA.2017.7989202.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *arXiv preprint arXiv:2005.13239*, 2020.

Amy Zhang, Clare Lyle, Shagun Sodhani, Angelos Filos, Marta Kwiatkowska, Joelle Pineau, Yarin Gal, and Doina Precup. Invariant causal prediction for block MDPs. In *International Conference on Machine Learning (ICML)*, pp. 11214–11224, 13–18 Jul 2020.

Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations (ICLR)*, 2021. URL `https://openreview.net/forum?id=-2FCwDKRREu`.

Appendix

A

WHY CAUSAL INVARIANCE?

Out of distribution (OOD) generalization has been attributed to learnt correlations that do not follow the underlying casual structure of the system. These are referred to as spurious correlations. With the use of deep neural networks, spurious correlations can arise due to 1) the way we collect data, or selection bias, 2) overparameterization of the neural networks, and 3) presence of irrelevant information in the data (ex. the background might be irrelevant for an object classification task). For the setting in this paper, such issues are relevant since we use NNs to learn the dynamics model of the RL environment. Even if these issues are attended to, spurious correlation could still arise. However, this time it would be due to the causal structure assumed and not the modelling technique (NNs) we use over it. Two such causes are 4) hidden confounders in the causal graph and 5) conditioning on anti-causal parts of input x. For our case, 4) could correspond to a hidden non-stationarity in the system such as the friction coefficient between the robot and the floor. Since we are only concerned with the x_t to x_{t+1} causal diagram, 5) may not be as apparent. Nevertheless, we include it for completeness. Therefore, in principle, choosing the right variables and deploying techniques that discover an invariant Y conditioned on a given X helps us avoid spurious correlations. This in turn leads to better OOD generalization.

NOTES ON ASSUMPTIONS

- There is a linearity assumption on the dynamics that is implicitly placed when we borrow the generalization results of Peters et al. (2015). These ensure that given data divided into multiple environments (minimum 2) (in our case that refers to data from multiple single policies), the causal representation results in a model that generalizes over all environments. When the dynamics are non-linear, Arjovsky et al. (2019) showed that a similar argument toward generalization can still be made, with the added requirement of having data from at least a fixed amount ($n_e \geq 2$) of environments. However, recent work (Rosenfeld et al., 2020) has argued that such an analysis is not accurate and thus more investigation is required to ensure OOD generalization. For the proof of concept experiment in Section 5, the dynamics are linear and thus we can deploy ICP for learning the causal parents of each state variable and ensure that the zero-shot generalization shown actually persists for any arbitrarily different policy from the ones used for training the invariant learner. When we move to Section 6 we do away with this approximation since the dynamics are no longer linear. Moreover, we do not restrict ourselves to a multiple environment based regime, the likes of which are required by Peters et al. (2015).
- The transition factorization assumption, i.e. Assumption 1, seems like a strict condition in theory when we move to complex domains, however, it is in fact a natural outcome of how we model the agent dynamics in practice. In practice, each state variable of the next state x_{t+1} is set to only be dependent on the previous state x_t and action a_t. We can see this for example in neural network based dynamics models where the next state as a whole (all state variables simultaneously) is predicted given the previous state and action. Therefore, even though it may seem as an over constraining assumption, in practice it is present by default. In fact, this shows that we should focus more on theoretical results that build on assumptions like transition factorization.
- A constraint on the exploration issue is usually dealt with by the concentratability assumption (Assumption 3) in literature. A recent method to get around such an assumption is by coupling the policy optimization algorithm with a exploration algorithm that maintains a set of exploratory policies (*policy cover* in Misra et al. (2020)) which slowly keeps expanding.
- When describing the practical invariant model learner (Section 6), we do not explicitly focus on finding the exact causal parents for each state variable. On the other hand, we resort to forcing such a constraint implicitly by describing a direct, differentiable invariance-based loss. One benefit of this approach is that the overall method remains end-to-end. The

downside of course is that we do not always ensure that the right set of causal parents is found.

RELATED WORK

On Factored MDPs: Planning based on structural assumptions on the underlying MDP have been explored in significant detail in the past (Boutilier et al., 1999). The most closely related setting is of factored MDPs, but learning based approaches that build on the factored MDP assumption have predominantly also assumed a known graph structure for the transition factorization (Kearns & Koller, 1999; Strehl et al., 2007; Osband & Van Roy, 2014).

On the theory side, most prior works on factored MDPs also do not learn and leverage state abstractions (Kearns & Koller, 1999; Strehl et al., 2007). Jonsson & Barto (2006) draw connections to causal inference, but do so explicitly with dynamic Bayesian networks, as opposed to learning approximate abstractions — and assume knowledge of the model. Most recently, Misra et al. (2021) also tackle the rich observation factored MDP setting, but consider each pixel an atom that belongs to a single factor.

On the algorithmic side, there have been only a few works that discuss learning the graph or DBN structure alongside the factored MDP assumption, e.g., (Hallak et al., 2015). We differ from these in that we only learn the partial graph structure (not explicitly), i.e., only the direct parents of each state variable. Moreover, we achieve this using the invariance principle, which has not been explored in prior work. A major reason for adopting the invariance principle is that it naturally allows us to work in the multiple environment setting, where an environment is characterized by the different state distributions induced by different policies during training, a necessary component for learning an invariant representation. This is an important distinction from the supervised learning setting, one where other graph structure learning methods have been shown to work well. There is little reason to believe that such approaches extend to the RL case as well, particularly because the data distribution is not fixed in general in RL.

On CDPs: There has been a lot of recent work around the newly proposed CDP setting. Our work has overlapping ideas with two specific works — model based learning in CDPs (Misra et al., 2020) and learning efficient abstractions over them (Sun et al., 2019). Besides the more algorithmic and empirically focused nature of this work, there remain several considerable distinctions. Firstly, we focus on abstraction-based learning, whereas Sun et al. (2019) rely on the concept of *witness misfit* to learn efficiently over the original CDP states. Secondly, we are focused on learning abstract states that are a coarser representation of the true full state of the CDP, whereas Misra et al. (2020) deal with the case where the abstract states correspond to the full state/latent states of the CDP. In that sense, the framework adopted here is a blend of that presented in these two works. Ideally, we would like to show that the class of problems where the number of model-invariant abstract states is low, also have a low *witness rank*.

B PROOFS

Theorem 1. *For the abstraction $\phi_i(x) = [x]_{S_i}$, where S_i = **PA**(x^i_{t+1}), ϕ_i is model-invariant. Furthermore, if ϕ follows such a definition for all state variables indexed by i, ϕ is a reward free model irrelevant state abstraction.*

Proof. We first prove that ϕ_i is model-invariant. In the case where $\phi_i(x) = \phi_i(x')$ for some state variable indexed by i, we have:

$$\begin{aligned} P(x''_i|x,a) &= P(x''_i|[x]_{S_i},a) \\ &= P(x''_i|\phi_i(x),a) \\ &= P(x''_i|\phi_i(x'),a). \end{aligned}$$

Following the same steps backwards for $\phi_i(x')$ concludes the proof.

We now prove the latter statement in the theorem. We note that for such a statement to be meaningful, we require that the state space $\mathcal{X}$ includes some irrelevant state variables for the downstream task in hand. For example, we could have some unnecessary noise variables appended to the full state variables. In such a case, the full state variables are relevant for the downstream task whereas the noise variables are irrelevant for the downstream task. Now, if $\phi(x) = \phi(x')$, i.e., $\phi_i(x) = \phi_i(x')$ for all relevant state variables indexed by i, ϕ is a reward free model irrelevant state abstraction, i.e.,

$$\sum_{x'' \in \phi^{-1}(\bar{x})} P(x''|x,a) = \sum_{x'' \in \phi^{-1}(\bar{x})} P(x''|x',a), \tag{5}$$

where $\bar{x}$ is the abstract state that ϕ maps to. With this note, the proof for the latter statement follows directly from Theorem 1 in Zhang et al. (2020).

On the absence of irrelevant state variables: The condition $\phi(x^1) = \phi(x^2)$ is quite strict if we assume the absence of irrelevant state variables (if no such variables are present, then x^1 has to be equal to x^2 for this condition to be met, which is not meaningful).

Extending to model-invariance grounded in reward: Notice that Definition 1 is reward free, and is grounded in the next state x''. We could instead extend this to a definition which is grounded in the reward. Particularly,

Definition 4. *(Reward Grounded Model Invariant Abstraction) ϕ_i is reward grounded model-invariant if for any $x, x', x'' \in \mathcal{X}, a \in \mathcal{A}$, $\phi_i(x) = \phi_i(x')$ if and only if*

$$\begin{aligned} R_i(x,a) &= R_i(x',a) \\ \sum_{x'' \in \phi^{-1}(\bar{x})} P(x''_i|x,a) &= \sum_{x'' \in \phi^{-1}(\bar{x})} P(x''_i|x',a), \end{aligned}$$

We can show that the causal representation of ϕ is a reward free version of the above defined model-invariance abstraction (Definition 4).

Proposition 2. *For the abstraction $\phi_i(x) = [x]_{S_i}$, where S_i = **PA**(x^i_{t+1}), ϕ_i is a reward free version of Definition 4.*

Proof. Now, when $\phi_i(x) = \phi_i(x')$ for a specific state variable indexed by i, we have:

$$
\begin{aligned}
\sum_{x'' \in \phi_i^{-1}(\bar{x})} P(x''|x,a) &= \sum_{x'' \in \phi_i^{-1}(\bar{x})} \prod_{k=0}^{p} P(x''_k|x,a) \\
&= \sum_{x'' \in \phi_i^{-1}(\bar{x})} P(x''_i|[x]_{S_i},a) \prod_{k=0}^{p} P(\{x''\}_{k \neq i,\ i,k \in N}|x,a) \\
&= P(x''_i|\phi_i(x),a) \sum_{x'' \in \phi_i^{-1}(\bar{x})} P(\{x''\}_{k \neq i,\ i,k \in N}|x,a) \\
&= P(x''_i|\phi_i(x),a) \\
&= P(x''_i|\phi_i(x'),a).
\end{aligned}
$$

Following the same steps backwards concludes the proof.

Lemma 1. *(Model Error Bound) Let ϕ be an $\epsilon_{i,P}$-approximate model-invariant abstraction on CDP M. Given any distributions $p_{x_i} : x_i \in \phi_i(\mathcal{X})$ where $p_x = \prod_{i=1}^{p} p_{x_i}$ is supported on $\phi^{-1}(x_i)$, we define $M_\phi = (\phi_i(\mathcal{X}), \mathcal{A}, P_\phi, R_\phi, \gamma)$ where $P_\phi(x,a) = \prod_{i=1}^{p} P_{\phi_i}(x,a)$. Then for any $x \in \mathcal{X}$, $a \in \mathcal{A}$,*

$$
\|P_\phi(x,a) - \Phi P(x,a)\| \leq \sum_{i=1}^{p} \epsilon_{i,P}.
$$

Proof. Consider any x, a and let $q_{x_i} := \Phi_i P(x,a)$, where we have $\|q_{x_i^1} - q_{x_i^2}\| \leq \epsilon_{i,P}$ if $\phi_i(x^1) = \phi_i(x^2)$.

$$
\begin{aligned}
\|P_\phi(x,a) - \Phi P(x,a)\| &= \|\prod_{i=0}^{p} P_{\phi_i}(x,a) - \Phi P(x,a)\| \\
&= \|\prod_{i=0}^{p} P_{\phi_i}(x,a) - \prod_{i=0}^{p} \Phi_i P(x,a)\| \\
&= \|\sum_{\bar{x} \in \phi^{-1}(\{x_i\}_{i \in N})} p_x(\bar{x}) \prod_{i=0}^{p} q_{\bar{x}_i} - \prod_{i=0}^{p} q_{x_i}\| \\
&= \|\sum_{\bar{x} \in \phi^{-1}(\{x_i\}_{i \in N})} p_x(\bar{x}) \Big(\prod_{i=0}^{p} q_{\bar{x}_i} - \prod_{i=0}^{p} q_{x_i} \Big)\| \\
&\leq \sum_{\bar{x} \in \phi^{-1}(\{x_i\}_{i \in N})} p_x(\bar{x}) \| \prod_{i=0}^{p} q_{\bar{x}_i} - \prod_{i=0}^{p} q_{x_i}\|.
\end{aligned}
$$

We now use the following inequality:

$$
\begin{aligned}
\|AB - CD\| &= \|AB - AD + AD - CD\| \\
&= \|A(B-D) + (A-C)D\| \\
&\leq \|A(B-D) + \|(A-C)D\|\| && \text{(Triangle inequality)} \\
&\leq \|A\|_\infty \|B-D\|_1 + \|A-C\|_1 \|D\|_\infty && \text{(Holder's inequality).}
\end{aligned}
$$

The $\infty-$norm of a probability distribution is 1. Apply this result to the above expression p times,

$$\begin{aligned}
\|P_\phi(x,a) - \Phi P(x,a)\| &\leq \sum_{\bar{x}\in\phi^{-1}(\{x_i\}_{i\in N})} p_x(\bar{x}) \| \prod_{i=0}^{p} q_{\bar{x}_i} - \prod_{i=0}^{p} q_{x_i} \| \\
&\leq \sum_{\bar{x}\in\phi^{-1}(\{x_i\}_{i\in N})} p_x(\bar{x}) \| \prod_{i=0}^{p} q_{\bar{x}_i} \|_\infty \| q_{\bar{x}_n} - q_{x_n} \|_1 + \| \prod_{i=0}^{p} q_{\bar{x}_i} - \prod_{i=0}^{p} q_{x_i} \|_1 \| q_{x_p} \|_\infty \\
&\leq \sum_{\bar{x}\in\phi^{-1}(\{x_i\}_{i\in N})} p_x(\bar{x}) \sum_{i=1}^{p} \epsilon_{i,P} \\
&= \sum_{i=1}^{p} \epsilon_{i,P}.
\end{aligned}$$

Theorem 2. *(Value bound) If ϕ is an $\epsilon_R, \epsilon_{i,P}$ approximate model-invariant abstraction on CDP M, and M_ϕ is the abstract CDP formed using ϕ, then we can bound the loss in the optimal state action value function in both the CDPs as:*

$$\|[Q^*_{M_\phi}]_M - Q^*_M\|_{2,\nu} \leq \frac{\sqrt{C}}{1-\gamma} \|[Q^*_{M_\phi}]_M - \mathcal{T}[Q^*_{M_\phi}]_M\|_{2,\mu}$$

$$\|[Q^*_{M_\phi}]_M - \mathcal{T}[Q^*_{M_\phi}]_M\|_{2,\mu} \leq \epsilon_R + \gamma \Big(\sum_{i=1}^{p} \epsilon_{i,P} \Big) R_{max}/(2(1-\gamma))$$

Note that this theorem deals with the batch setting, where we are given a batch of data and are tasked at learning only using this data, without allowing any direct interaction with the CDP. We use the concentratability coefficient as defined in Assumption 3, i.e., there exists a C such that for any admissible distribution ν:

$$\forall (x,a) \in \mathcal{X} \times \mathcal{A}, \;\; \frac{\nu(x,a)}{\mu(x,a)} < C\,.$$

Here, we abuse μ to represent the distribution the data comes from instead of standard notation representing the starting state distribution. Now,

$$\begin{aligned}
\|[Q^*_{M_\phi}]_M - Q^*_M\|_{2,\nu} &= \|[Q^*_{M_\phi}]_M - \mathcal{T}[Q^*_{M_\phi}]_M + \mathcal{T}[Q^*_{M_\phi}]_M - Q^*_M\|_{2,\nu} \\
&\leq \|[Q^*_{M_\phi}]_M - \mathcal{T}[Q^*_{M_\phi}]_M\|_{2,\nu} + \|\mathcal{T}[Q^*_{M_\phi}]_M - \mathcal{T}Q^*_M\|_{2,\nu} \\
&\leq \sqrt{C}\|[Q^*_{M_\phi}]_M - \mathcal{T}[Q^*_{M_\phi}]_M\|_{2,\mu} + \|\mathcal{T}[Q^*_{M_\phi}]_M - \mathcal{T}Q^*_M\|_{2,\nu} \qquad (3)
\end{aligned}$$

Let us consider the second term:

$$\begin{aligned}
\|\mathcal{T}[Q^*_{M_\phi}]_M - \mathcal{T}Q^*_M\|^2_{2,\nu} &= \mathbb{E}_{(x,a)\sim\nu} \Bigg[\Big(\mathcal{T}[Q^*_{M_\phi}]_M(x,a) - \mathcal{T}Q^*_M(s,a) \Big)^2 \Bigg] \\
&= \mathbb{E}_{(x,a)\sim\nu} \Bigg[\Big(\gamma \mathbb{E}_{x'\sim P(x,a)} [\max_a [Q^*_{M_\phi}]_M(x',a) - \max_a Q^*_M(x',a)] \Big)^2 \Bigg] \\
&\leq \mathbb{E}_{(x,a)\sim\nu} \Bigg[\gamma^2 \mathbb{E}_{x'\sim P(x,a)} \Big(\max_a [Q^*_{M_\phi}]_M(x',a) - \max_a Q^*_M(x',a) \Big)^2 \Bigg] \\
&\leq \gamma^2 \mathbb{E}_{(x,a)\sim\nu}\, \mathbb{E}_{x'\sim P(x,a)} \Big[\max_a \Big([Q^*_{M_\phi}]_M(x',a) - Q^*_M(x',a) \Big)^2 \Big] \\
&\leq \max_\nu \Bigg[\gamma^2 \mathbb{E}_{(x,a)\sim\nu}\, \mathbb{E}_{x'\sim P(x,a)} \Big[\max_a \Big([Q^*_{M_\phi}]_M(x',a) - Q^*_M(x',a) \Big)^2 \Big] \Bigg] \\
&\leq \max_\nu \Bigg[\gamma^2 \mathbb{E}_{(x,a)\sim\nu} \Big[\Big([Q^*_{M_\phi}]_M(x',a) - Q^*_M(x',a) \Big)^2 \Big] \Bigg] \\
&= \max_\nu \gamma^2 \|[Q^*_{M_\phi}]_M - Q^*_M\|^2_{2,\nu}
\end{aligned}$$

where the last inequality follows because the two terms inside the expectation only depend on the next state x' and the next action a which can only be less than the value for $x, a \sim \nu$ since we maximize over it.

Plugging this back in (3):

$$\begin{aligned}
\|[Q^*_{M_\phi}]_M - Q^*_M\|_{2,\nu} &\leq \sqrt{C}\|[Q^*_{M_\phi}]_M - \mathcal{T}[Q^*_{M_\phi}]_M\|_{2,\mu} + \max_\nu \gamma\|[Q^*_{M_\phi}]_M - Q^*_M\|_{2,\nu} \\
\max_\nu \|[Q^*_{M_\phi}]_M - Q^*_M\|_{2,\nu} &\leq \sqrt{C}\|[Q^*_{M_\phi}]_M - \mathcal{T}[Q^*_{M_\phi}]_M\|_{2,\mu} + \max_\nu \gamma\|[Q^*_{M_\phi}]_M - Q^*_M\|_{2,\nu} \\
\max_\nu \|[Q^*_{M_\phi}]_M - Q^*_M\|_{2,\nu} &\leq \frac{\sqrt{C}}{1-\gamma}\|[Q^*_{M_\phi}]_M - \mathcal{T}[Q^*_{M_\phi}]_M\|_{2,\mu}
\end{aligned}$$

Since $\|[Q^*_{M_\phi}]_M - Q^*_M\|_{2,\nu} \leq \max_\nu \|[Q^*_{M_\phi}]_M - Q^*_M\|_{2,\nu}$, we have:

$$\|[Q^*_{M_\phi}]_M - Q^*_M\|_{2,\nu} \leq \frac{\sqrt{C}}{1-\gamma}\|[Q^*_{M_\phi}]_M - \mathcal{T}[Q^*_{M_\phi}]_M\|_{2,\mu}$$

Now, we prove the second statement:

$$\begin{aligned}
\|[Q^*_{M_\phi}]_M - \mathcal{T}[Q^*_{M_\phi}]_M\|_{2,\mu} &\leq \|[Q^*_{M_\phi}]_M - \mathcal{T}[Q^*_{M_\phi}]_M\|_\infty \\
&= \|[\mathcal{T}_{M_\phi} Q^*_{M_\phi}]_M - \mathcal{T}[Q^*_{M_\phi}]_M\|_\infty \\
&= \sup_{x,a} |R_\phi(\phi(x), a) + \gamma\langle P_\phi(\phi(x), a), V^*_{M_\phi}\rangle - R(x,a) - \gamma\langle P(x,a), [V^*_{M_\phi}]_M\rangle| \\
&\leq \epsilon_R + \gamma \sup_{x,a} |\langle P_\phi(\phi(x), a), V^*_{M_\phi}\rangle - \langle P(x,a), [V^*_{M_\phi}]_M\rangle| \\
&= \epsilon_R + \gamma \sup_{x,a} |\langle P_\phi(\phi(x), a), V^*_{M_\phi}\rangle - \langle \Phi P(x,a), V^*_{M_\phi}\rangle| \\
&\leq \epsilon_R + \gamma\epsilon_P \|V^*_{M_\phi} - \frac{R_{\text{max}}}{2(1-\gamma)}\mathbf{1}\|_\infty \\
&\leq \epsilon_R + \gamma\epsilon_P R_{\text{max}}/(2(1-\gamma)) \\
&= \epsilon_R + \gamma\Big(\sum_{i=1}^{p} \epsilon_{i,P}\Big) R_{\text{max}}/(2(1-\gamma))
\end{aligned}$$

C IMPLEMENTATION DETAILS

C.1 PYTORCH-LIKE PSEUDOCODE FOR LEARNING MODEL-INVARIANT REPRESENTATIONS

```
for x in loader: # load a minibatch x with n samples
   # independent predictions from two randomly initiated models
   z1, z2 = f(x), h(x) # f: model_1, h: model_2
   # pick random dimension
   dim = rand(z1.shape)

   pred_1 = g(cat(z1[dim], one_hot(dim))) # g: critic
   pred_2 = g(cat(z2[dim], one_hot(dim)))
   p1, p2 = InvLoss(pred_1, pred_2)

   L = KL(p1, p2)

   L.backward()
   update(f, h, g)

def InvLoss(pred_1, pred_2):
   phi_1 = pred_1 * pred_2.T
   phi_2 = pred_2 * pred_1.T

   # matrix of inner product of 2-norm of pred_1 rows with pred_2 columns
   norm_12 = normalize(pred_1, pred_2)
   phi_1 = phi_1 / norm_12
   phi_2 = phi_2 / norm_12.T

   p1 = F.softmax(phi_1, dim=-1)
   p2 = F.softmax(phi_2, dim=-1)
   return p1, p2

def KL(p1, p2):
   p2 = p2.detach()
   return (p1 * (p1 / p2).log()).sum(dim=-1).mean()
```

C.2 SAC-SVG ALGORITHM

The SAC-SVG algorithm is presented in Amos et al. (2020) and is based on the idea of model-based value expansion (MVE) Feinberg et al. (2018). MVE uses the model to expand the value function to compute a multi-step estimate which a model-free base algorithm uses for policy optimization. In SAC-SVG, the model-free base learner is a SAC agent and the multi-step estimates correspond to that of the Q value used by the SAC actor.

$$\mathcal{L}_{\alpha,\pi}^{\text{SAC-SVG}} = \mathbb{E}_{x\sim\mathcal{D},\, a\sim\pi} - Q_{0:H}^{\alpha,\pi}(x,a),$$

where α is the entropy temperature parameter of SAC. Note that for $H = 0$, SAC-SVG is equivalent to SAC, since the model is no longer used for updating the actor. Thus the impact of the model on the final algorithm performance is through the horizon parameter H. Regarding the model learner, SAC-SVG uses a recurrent deterministic model which takes as input the current state and a hidden state to output the next state for a given horizon step H. The other popular alternative is to use an ensemble of probabilistic model learners, as done in Chua et al. (2018).

C.3 MBPO VS POPLIN ENVIRONMENTS

For our MBRL experiments, we used two sets of MuJoCo-based environments, each used before in individual papers. Specifically, the POPLIN based environments were originally used in the paper by Wang & Ba (2019). These refer to the '-v0' versions from OpenAI Gym Brockman et al. (2016) and also includes a separately tweaked Cheetah (called PETS-Cheetah) and Swimmer environments. On the other hand, the MBPO based environments refer to the ones used by the paper Janner et al. (2019) and largely correspond to the '-v2' versions from OpenAI Gym. These include an additional reward for staying alive throughout an episode.

Hyperparameter	**Value**
Replay buffer size	1000000
Initial temperature (α)	0.1
Learning rate	$1e-4$ SAC actor and critic; $1e-3$ Model learner
SAC Critic τ	0.005
Discount γ	0.99
SAC batch size	1024
Model batch size	512
Optimizer	Adam
Model updates per env step	4
Initial steps	1000
Number of encoder hidden layers (Model)	2
Number of decoder hidden layers (Model)	2
Encoder hidden layer size (Model)	512
Decoder hidden layer size (Model)	512
Model critic (g)	Single layer MLP (512)

Table 1: Hyper-parameters used for the Invariant-SAC-SVG algorithm.

C.4 Spurious Correlation

For the experiment in Section 7.1, we used three different input strategies to test for the presence of spurious correlations in model learning. Here, we define the exact masking schemes used. We are interested in only predicting a single dimension here— the left knee joint position. Below are the masking detailed descriptions:

- **No Mask:** None of the observation dimensions are masked.
- **Mask_1:** Dimensions that are seemingly correlated to the left knee joint are masked. Specifically, {left_hip_x, left_hip_y, left_hip_z, left_knee} (qpos and qvel)
- **Mask_2:** Dimensions that are seemingly uncorrelated to the left knee joint are masked. Specifically, {left_shoulder_1, left_shoulder_2, left_elbow} (qpos and qvel)

C.5 Invariant Model Learning

For our invariant model learner, we test on offline data collected in a replay buffer during the first 1M training steps of a model-free SAC agent. We start model training with the initial samples from the replay buffer and continue to add more as the training progresses. Such a scheme ensures that we have access to changing state distributions as the policy changes while remaining isolated from direct policy optimization on the CDP.