Exploiting Expert-guided Symmetry Detection in Markov Decision Processes

Giorgio Angelotti,1,2 Nicolas Drougard,1,2 Caroline P. C. Chanel1,2

1ISAE-Supaero, University of Toulouse, France
2ANITI, University of Toulouse, France
{name.surname}@isae-supaero.fr

Abstract

Offline estimation of the dynamical model of a Markov Decision Process (MDP) is a non-trivial task that greatly depends on the data available to the learning phase. Sometimes the dynamics of the model is invariant with respect to some transformations of the current state and action. Recent works showed that an expert-guided pipeline relying on Density Estimation methods as Deep Neural Network based Normalizing Flows effectively detects this structure in deterministic environments, both categorical and continuous-valued. The acquired knowledge can be exploited to augment the original data set, leading eventually to a reduction in the distributional shift between the true and the learnt model. In this work we extend the paradigm to also tackle non deterministic MDPs, in particular 1) we propose a detection threshold in categorical environments based on statistical distances, 2) we introduce a benchmark of the distributional shift in continuous environments based on the Wilcoxon signed-rank statistical test and 3) we show that the former results lead to a performance improvement when solving the learnt MDP and then applying the optimal policy in the real environment.

1 Introduction and Related Works

In Offline Model Based Reinforcement Learning (RL) and Offline Learning for Planning the environment dynamics is inferred from a batch of already pre-collected experiences. Wrong previsions lead to bad decisions. The distributional shift, defined as the discrepancy between the learnt model and reality, is the main responsible for the performance deficit of the (sub)optimal policy obtained in the offline setting compared to the true optimal policy (Levine et al. 2020). In recent years (van der Pol et al. 2020a,b) an expert-guided detection of alleged symmetries based on Density Estimation statistical techniques in the context of the offline learning of both continuous and categorical environments was proposed in order to eventually augment the starting data set. The authors showed that correctly detecting a symmetry and data augmenting the starting data set exploiting this information led to a decrease in the distributional shift. Unfortunately, the said work concerned only deterministic MDPs and did not include an analysis of the performance of the policy obtained in the end. In other fields of Machine Learning data augmentation has been extensively exploited to boost the efficiency of the algorithms in data-limited setups (van Dyk and Meng 2001; Shorten and Khoshgoftaar 2019; Park et al. 2019).

In this work we propose an extension of the approach that also tackles stochastic environments. More specifically, the
contribution of this paper are the followings:
1. A refinement of the decision threshold, based on statistical distances, is defined for categorical MDPs;
2. We introduce a benchmark for the distributional shift in the continuous case that is based on statistical tests and that brings a universal metric for comparisons;
3. The improvement of the policy performance obtained by augmenting the data with the symmetric images of the transitions is demonstrated experimentally.

2 Background

In this section, we introduce some necessary base concepts following (Angelotti, Drougard, and Chanel 2021).

**Definition 1 (Markov Decision Process).** An MDP (Bellman 1966) is a tuple $M = (S, A, R, T, \gamma)$. $S$ and $A$ are the set of states and actions, $R : S \times A \rightarrow \mathbb{R}$ is the reward function, $T : S \times A \rightarrow \text{Dist}(S)$ is the transition function, and $\gamma \in [0, 1]$ is the discount factor. Time is discretized and at each step $t \in \mathbb{N}$ the agent observes a system state $s = s_t \in S$, acts with $a = a_t \in A$ drawn from a policy $\pi : S \rightarrow \text{Dist}(A)$, and with probability $T(s, a, s')$ transits to a next state $s' = s_{t+1}$, earning a reward $R(s, a)$. The value function of $\pi$ and $s$ is defined as the expected total discounted reward $V_{\pi}(s) = \mathbb{E}_s[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)|s_0 = s]$. The optimal value function $V^*$ is the maximum of the latter over every policy $\pi$.

**Definition 2 (MDP Symmetry).** Given an MDP $M$, let $k$ be a surjection on $S \times A \times S$ such that $k(s, a, s') = (k_\sigma(s, a, s'), k_c(s, a, s'), k_{\alpha skirts}(s, a, s')) \in S \times A \times S$. $k$ is a symmetry if $\forall (s, s') \in S^2, k \in A$ both $T$ and $R$ are invariant with respect to the image of $k$:

$$\text{(1)} \quad (T \circ k) (s, a, s') = T(s, a, s'),$$

$$\text{(2)} \quad R\left(k_\sigma(s, a, s'), k_c(s, a, s'), k_{\alpha skirts}(s, a, s')\right) = R(s, a).$$

In the continuous case, since the model dynamics was learnt by estimating the frequencies of transition in $T$, it is not as a collection of pdfs but as a regressor. Let $D_{\text{ev}}$ be a very big data set of transitions that we use for evaluation purposes.

**Probability Mass Function Estimation for Discrete MDPs.** Let $D = \{(s_i, a_i, s'_i)\}_{i=1}^n$ be a batch of recorded transitions. Performing mass estimation over $D$ amounts to compute the probabilities that define the categorical distribution $T$ by estimating the frequencies of transition in $D$. In other words:

$$\hat{T}(s, a, s') = \begin{cases} \sum_{n_{s,a,s'}} & \text{if } \sum_{s'} n_{s,a,s'} > 0, \\ \frac{n_{s,a,s'}}{|S|^{-1}} & \text{else;} \end{cases}$$

where $n_{s,a,s'}$ is the number of times the transition $(s_t = s, a_t = a, s_{t+1} = s') \in D$.

**Probability Density Function Estimation for Continuous MDPs.** Performing density estimation over $D$ means obtaining an analytical expression for the probability density function (pdf) of transitions $(s, a, s')$ given $D$. $L(s, a, s'|D)$. Normalizing flows (Dinh, Krueger, and Bengio 2015; Kobyzev, Prince, and Brubaker 2020) allow defining a parametric flow of continuous transformations that reshapes a known initial pdf to one that best fits the data.

**Expert-guided detection of symmetries.** The paradigm described in (Angelotti, Drougard, and Chanel 2021) can be resumed as follows:

1. An expert presumes that a to be learnt model is endowed with the invariance of $T$ with respect to a transformation $k$;
2. He computes $\hat{T}$, an estimate of $T$, using the transitions in a batch $D$:
   - (a) in the categorical case by applying Equation (1);
   - (b) in the continuous case by training a regressor using a Deep Neural Network (DNN);
3. (in the continuous case) He performs Density Estimation over $D$ using Normalizing Flows;
4. He applies $k$ to all transitions $(s, a, s') \in D$ and then checks whether the fraction $\nu_k$:
   - (a) (categorical case) of samples $k(s, a, s') = (k_\sigma(s, a, s'), k_c(s, a, s'), k_{\alpha skirts}(s, a, s')) \in k(D)$ s.t. $T(s, a, s') = (T \circ k)(s, a, s')$ exceeds an expert given threshold $\nu$;
   - (b) (continuous case) of probability values $L$ evaluated on $k(D)$ exceeds a threshold $\theta$ that corresponds to the $q$-order quantile of the distribution of probability values evaluated on the original batch. The quantile order $q$ is given as an input to the procedure by an expert (see Algorithm 2);
5. If the last condition is fulfilled then $D$ is augmented with $k(D)$.

**Evaluation** In the end, the validity of the approach was shown by computing $\Delta$, the discrepancy in the distributional shifts between the true dynamics $T$ and $\hat{T}$ and the one between $T$ and $\hat{T}_k$, the transition functions learnt after the data augmentation exploiting the symmetry. In the discrete case $\Delta$ is defined as follows:

$$\Delta = d(T, \hat{T}) - d(T, \hat{T}_k)$$

where $d(T, \hat{T})$ and $d(T, \hat{T}_k)$ are the sums of each state-action pair Total Variational distances:

$$d(T, \hat{T}) = \frac{1}{2} \sum_{(s, a, s') \in S^2, a \in A} |T(s, a, s') - \hat{T}(s, a, s')|.$$  

In the continuous case, since the model dynamics was learnt not as a collection of pdfs but as a regressor. Let $D_{\text{ev}}$ be a very big data set of transitions that we use for evaluation purposes, then

$$\Delta = \mathbb{E}_{D_{\text{ev}}} \left( s, a, s' \right) - \mathbb{E}_{D_{\text{ev}}} \left( s, a, s' \right)$$

where $\mathbb{E}$ is the average of an error (e.g. the Squared Error) over the set in the subscript and $s'(s, a)$ is the output of the learnt regressor with ($s'_k$) or without ($s'$) augmenting the data set with the symmetry $k$ (point 2.b of the previous list).

3 Algorithmic Contributions

In this section we present the algorithmic contributions proposed in our work that improve the detection of alleged symmetries and its subsequent evaluation in stochastic MDPs.
3.1 \( \nu_k \) in the categorical setting

Our first contribution relies in the improvement of the calculation of \( \nu_k \) in part (4.a) of the previous list. Indeed, that approach does not yield valid results when applied to stochastic environments. In order for the method to work in stochastic environments we need to measure a distance in distribution. The latter somehow was already considered in the version of the approach that took care of continuous environments. However, when dealing with categorical states we can not exploit the notion of distance between features.

We propose to compute the percentage \( \nu_k \) relying on a distance between categorical distributions. Since the transformation \( k \) is a surjection on transition tuples, we do not know a-priori which will be the correct mapping \( k_{s',s} = k_{s,s'} \) for \( \forall s' \in S \). In other words, we can compute \( k_{s,s'} \), the symmetric image of \( s' \), only when we receive as an input the whole tuple \( (s, s', \bar{s}) \) since an inverse mapping might not exist.

Therefore we will resort to compute a pessimistic approximation of the Total Variational Distance (proportional to the \( L^1 \)-norm). In particular, given \( (s, s', a') \), we aim to calculate the Chebyshev distance (the \( L^\infty \)-norm) between \( T(s, a, \cdot) \) and \( T(k_{s,s'}, k_{a, s}, k_{a, s'}) \). Recall that given two vectors of dimension \( d \), \( x \) and \( y \) both in \( \mathbb{R}^d \), \( ||x - y||_\infty \leq ||x - y||_1 \).

Let us then define the following four functions:

\[
\begin{align*}
m(s, a, s') &= \min_{\tau \in S \setminus \{s'\} : \hat{T}(s, a, \tau)}, \\
M(s, a, s') &= \max_{\tau \in S \setminus \{\hat{s}\}} \hat{T}(s, a, \tau), \\
m_k(s, a, s') &= \min_{\tau \notin k_{s,s'}, k_{a, s}, k_{a, s'}} \hat{T}(k_{s,s'}(s, a, s'), k_{a, s}, k_{a, s'}), \\
M_k(s, a, s') &= \max_{\tau \in S \setminus S} \hat{T}(k_{s,s'}(s, a, s'), k_{a, s}, k_{a, s'}),
\end{align*}
\]

where \( m(M) \) and \( m_k(M_k) \) are the minimum (maximum) of the pmf \( \hat{T} \) when evaluated respectively on an initial state and action \( (s, a) \) and \( (k_{s,s'}(s, a, s'), k_{a, s}, k_{a, s'}) \) for which \( \hat{T} \neq 0 \). In order to approximate the Chebyshev distance between \( \hat{T}(s, a, \cdot) \) and \( \hat{T}(k_{s,s'}, k_{a, s}, k_{a, s'}) \) we define a pessimistic approximation \( d_k \) as follows:

\[
d_k(s, a, s') = \max \left\{ \left| M(s, a, s') - m_k(s, a, s') \right|, \left| M_k(s, a, s') - m(s, a, s') \right|, \left| \hat{T}(s, a, s') - (\hat{T} \circ k)(s, a, s') \right| \right\}. \tag{7}
\]

Notice that

\[
0 < d_k(s, a, s') \leq 1 \quad \forall (s, a, s') \in S \times A \times S. \tag{8}
\]

We propose then to estimate \( \nu_k \) as in Line 2 of Algorithm 1:

\[
\nu_k(D) = 1 - \frac{1}{|D|} \sum_{(s, a, s') \in D} d_k(s, a, s'). \tag{9}
\]

From equations (7) and (8) it follows that 1) in deterministic environments \( \nu_k \) (Eq. 9) coincides with the one prescribed in (Angelotti, Drougard, and Chanel 2021) and 2) \( 1 > \nu_k > 0 \), so \( \nu_k \) can be interpreted as a percentage. The last remark allows us to suppose that \( \nu_k \) is an estimate of the probability of \( k \) being a symmetry of the dynamics, and therefore we can relax the necessity of defining an expert-given threshold \( \nu \) as an input in Algorithm 1 by setting \( \nu = 0.5 \) and eventually augmenting the batch if \( \nu_k > 0.5 \) (Lines 3-4).

3.2 \( \Delta \) in the continuous setting

In the original paper a beneficial data augmentation resulted into a negative \( \Delta \) (Equation 6). Only the sign mattered, and a universal scale of comparison for its magnitude was missing. Our second contribution consists in a robust evaluation benchmark of the distributional shift \( \Delta \) in continuous environments that 1) works also in the stochastic setting, 2) it is based on statistical tests and 3) outputs a value between 0 and 1, therefore providing an universal scale of comparison.

Let \( \varepsilon \) be some distance between two \( n \) dimensional vectors (e.g. the Euclidean distance). We perform a Wilcoxon signed-rank test on the error samples \( \varepsilon_{D_{uv}}(s', s'(s, a)) \) and \( \varepsilon_{D_{uv}}(s', s'_k(s, a)) \) paired by \( (s, a, s') \) where \( \varepsilon_{D_{uv}} \) is the set defined in Continuous environment (8).

The null hypothesis is

\[
H_0 : \varepsilon_{D_{uv}}(s', s'(s, a)) \geq \varepsilon_{D_{uv}}(s', s'_k(s, a)) \tag{10}
\]

and the alternative hypothesis is

\[
H_1 : \varepsilon_{D_{uv}}(s', s'(s, a)) < \varepsilon_{D_{uv}}(s', s'_k(s, a)). \tag{11}
\]

Therefore when the resulting \( p \) value will be \( < 0.05 \) we can reject the null hypothesis and conclude that there is a statistically significant difference between the two sets that hints that the distributional shift using the regressor learnt with the augmented batch will be lower. We will set \( \Delta = p \).

Algorithm 1: Symmetry detection and data augmenting in a categorical MDP

Input: Batch of transitions \( D \), \( k \) alleged symmetry
Output: Possibly augmented batch \( D \cup D_k \)
1 \( \hat{T} \leftarrow \) Most Likely Categorical pmf from \( (D) \)
2 \( \nu_k = 1 - \frac{1}{|D|} \sum_{(s, a, s') \in D} d_k(s, a, s') \) (see Eq. 7)
3 if \( \nu_k > 0.5 \) then
4 return \( D \cup D_k \)
5 else
6 return \( D \)
7 end

4 Experiments

In order to show the improvements provided by our contribution we tested the algorithms in a stochastic version of the toroidal Grid environment and two continuous state environments of the OpenAI’s Gym Learning Suite: CartPole and Acrobot. The deterministic version of these environments...
Algorithm 2: Symmetry detection and data augmenting in a continuous MDP with detection threshold $\nu = 0.5$ (Angelotti, Drougard, and Chanel 2021)

**Input:** Batch of transitions $D$, $q \in [0, 1)$ order of the quantile, $k$ alleged symmetry

**Output:** Possibly augmented batch $D \cup D_k$

1. $L \leftarrow$ Density Estimate ($D$)
2. $\Lambda \leftarrow$ Distribution $L(D)$ ($L$ evaluated over $D$)
3. $\theta = q$-order quantile of $\Lambda$
4. $D_k = k(D)$
5. $\nu_k = \frac{1}{|D_k|} \sum_{(s,a,s') \in D_k} 1_{\{L(s,a,s'|D) > \theta\}}$
6. if $\nu_k > 0.5$ then
   7. return $D \cup D_k$
7. else
   8. return $D$
9. end

were used for the experiments in (Angelotti, Drougard, and Chanel 2021) and so they make a valid set of scenarios to assess the validity of our approach.

### 4.1 Setup

We collect a batch of transitions $D$ using a uniform random policy. An expert alleges the presence of a symmetry $k$ and we proceed to its detection using Algorithm 1 (categorical case) or Algorithm 2 (continuous case). In the continuous case Density Estimation is performed by a Masked Autoregressive Flow architecture (Papamakarios, Pavlakou, and Murray 2017) with 3 layers of bijectors. The regressors that mimic the MDP dynamics are Multilayer Perceptrons made of 3 hidden dense layers with 128 (CartPole) or 256 (Acrobot) neurons each trained over the batches splitted in a validation a training set (sizes 10% and 90% respectively) for a maximum of 5000 epochs with an Early Stop Callback with patience = 5. The states and actions were appropriately preprocessed to ease the learning.

**Computation of $\nu_k$ and batch augmentation** We report the $\nu_k$ obtained with an ensemble of $N$ different iterations of the procedure: we generate $z \in \mathbb{N}$ sets of $N$ different batches $D$ of increasing size (in the original paper only one batch size was considered). Remember that since $\nu_k \in [0, 1)$ we can interpret it as the probability of the presence of a symmetry and select a detection threshold $\nu = 0.5$, while in (Angelotti, Drougard, and Chanel 2021) the threshold $\nu$ was expert-given.

**Evaluation of the distributional shift** In the categorical environment $\Delta$ is measured along Equation 4 while in the continuous environments it will be the $p$ value of the Wilcoxon pairwise signed-rank test with hypothesis defined in Equations 10 and 11.

**Evaluation of the performance** In the end, let $\rho$ be the distribution of initial states $s_0 \in S$ and let the performance...
$U^\pi$ of a policy $\pi$ be

$$U^\pi = \mathbb{E}_{s \sim \rho} [V^\pi (s)].$$

(12)

our final contribution is the comparison between the performances obtained by acting in the real environment with $\hat{\pi}$ (the optimal policy obtained by solving the MDP learnt with $\hat{T}$) and $\tilde{\pi}_k$ (the optimal policy obtained with $\tilde{T}_k$).

In particular we consider the quantity $\Delta U = U^\pi - U^\tilde{\pi}_k$. In categorical environments the policies are obtained with Policy Iteration and evaluated with Policy Evaluation. In continuous environments the policies are computed and evaluated using the Proximal Policy Optimization (PPO) algorithm (Schulman et al. 2017) available in the Stable Baselines3 python library (Raffin et al. 2021) trained for 50000 time steps.

4.2 Environments

We proceed by describing the environments used in the simulations along with the tested transformations.

Table 1: Grid. Proposed transformations and label.

| $k$ | Label |
|-----|-------|
| $k_\alpha(s, a, s') = s'$ | TRSAI |
| $k_\alpha(s, a, \uparrow, \downarrow, \leftarrow, \rightarrow, s') = (\downarrow, \uparrow, \rightarrow, \leftarrow)$ | SDAI |
| $k_\alpha(s, a) = s$ | ODAI |
| $k_\alpha(s, a, \uparrow, \downarrow, \leftarrow, \rightarrow) = (\downarrow, \uparrow, \rightarrow, \leftarrow)$ | ODWA |
| $k_\alpha(s, a, s') = s'$ | TI |
| $k_\alpha(s, a, s') = a$ | TIOD |

Stochastic Cartpole (Continuous) In this environment the agent can move along fixed directions over a torus by acting with any $a \in A = \{\uparrow, \downarrow, \leftarrow, \rightarrow\}$. The grid meshing the torus has size $l = 10$. The agent can spawn everywhere on the torus with an uniform probability and must reach a fixed goal.

At every time step he receives a reward $r = -1$ if he does not reach the goal and a reward $r = 1$ once the goal has been reached, terminating the episode. When performing an action the agent has 60% chances of moving to the intended direction, 20% to the opposite one, and 10% along an orthogonal direction. We collect $z = 10$ sets of $M = 100$ batches with respectively $N = 1000 \times i_z$ steps in each batch ($i_z$ going from 1 to $z$). The proposed symmetries for this environment are outlined in Table 2. We check for the invariance of the dynamics with respect to the following six invariant transformations (the valid symmetries are displayed in bold):

1. Time reversal symmetry with action inversion (TRSAI);
2. Same dynamics with action inversion (SDAI);
3. Opposite dynamics and action inversion (ODAI);
4. Opposite dynamics but wrong action (ODWA);
5. Translation invariance (TI);
6. Translation invariance with opposite dynamics (TIOD).

The $N$ dependent average results are reported in Table 3. Their effects on the transition $(s, a, s')$ are listed in Table 2. Average results are displayed in Figure 1.
Figure 2: Stochastic CartPole. \( \nu_k \), \( \Delta \), and \( \Delta U \) for the transformations \( k \) computed over sets of different batches of size \( N \). Points are mean values and are a bit shifted horizontally for the sake of display.

Stochastic Acrobot (Continuous) It is the very same Acrobot of (Brockman et al., 2016) but at every time step a noise \( \epsilon \) is sampled from a uniform distribution on the interval \([-0.5, 0.5]\) and added to the torque. A state is represented by the features \( (s_1, c_1, s_2, c_2, \omega_1, \omega_2) \) where \( s_i \) and \( c_i \) are respectively \( \sin(\alpha_i) \) and \( \cos(\alpha_i) \) in shorthand notation. The action set \( A = \{-1, 0, 1\} \). For the evaluation of \( \nu_k \) we set \( q = 0.1 \). For the detection case we collected \( z = 5 \) sets of \( M = 100 \) batches with \( N = 1000 \times i_z \) steps within each one (\( i_z \) going from 1 to \( z \)). As for Stochastic Cartpole, the evaluation of the performance was carried out with \( z = 4 \) and \( M = 3 \) due to long time needed to train a PPO agent in an environment simulated with a DNN regressor. For the evaluation of \( \Delta \) \( z = 5 \) due to computational necessities. The error function \( \varepsilon \) used was the Logcosh. We allege the following transformations \( k \), as always the valid ones are bolded:

1. Angles and angular velocities inversion (AAVI);
2. Cosines and angular velocities inversion (CAVI);
3. Action inversion (AI);
4. Starting state inversion (SSI).

The images of the transformations are reported in Table 3. The \( N \) dependent average results are reported in Figure 3.

| \( k \)                  | Label          |
|-------------------------|----------------|
| \( k_{sa}(s = (s_1, s_2, \omega_1, \omega_2, \ldots, a, s') \)  | AAVI           |
| \( = (-s_1, -s_2, -\omega_1, -\omega_2, \ldots) \)             |                |
| \( k_{sa}(s, a = (-1, 0, 1), s') = (1, 0, -1) \)               |                |
| \( k_{sr}(s, a, s' = (s_1, s_2, \omega_1, \omega_2, \ldots) \) | CAVI           |
| \( = (-s_1, -s_2, -\omega_1, -\omega_2, \ldots) \)             |                |
| \( k_{sr}(s, a = (-1, 0, 1), s') = (1, 0, -1) \)               |                |
| \( k_{ar}(s, a, s' = (c_1, c_2, \omega_1, \omega_2, \ldots) \) | AI             |
| \( = (-c_1, -c_2, -\omega_1, -\omega_2, \ldots) \)             |                |
| \( k_{ar}(s, a, s' = (-1, 0, 1), s') = (1, 0, -1) \)           |                |
| \( k_{ar}(s, a, s' = (c_1, c_2, \omega_1, \omega_2, \ldots) \) | SSI            |
| \( = (-c_1, -c_2, -\omega_1, -\omega_2, \ldots) \)             |                |
| \( k_{ar}(s, a, s' = s) \)                                      |                |
| \( k_{ar}(s, a, s' = (-1, 0, 1), s') = (1, 0, -1) \)           |                |
| \( k_{ar}(s, a, s' = s' \)                                      |                |
| \( k_{ar}(s, a, s' = \alpha \)                                  |                |
| \( k_{ar}(s, a, s' = s' \)                                      |                |

We now comment on the results of the simulations environment by environment.

5.1 Stochastic Grid (Categorical) Detection phase \( \nu_k \) In Grid the algorithm perfectly manages to identify the real symmetries of the environment: \( \nu_k > 0.5, k \in \{ \text{TRSAI, ODAI, TII} \} \). Moreover, there are no false positives: \( \nu_k < 0.5, k \in \{ \text{SDAI, ODWA, TIOD} \} \). We notice that while in a deterministic environment \( \nu_k = 0 \) \( \forall k \) which is not a symmetry, here the stochasticity makes the detection more complicated since \( \nu_k \approx 0.5^\ast \) for \( N = 2000 \).
Evaluation of distributional shift ($\Delta$) The distributional shift $\Delta$ is always $> 0$ for $k$ = TRSAI and it is inversely proportional to $N$, while it is $> 0$ for $k$ $\in$ {ODAI, TI}, $N \leq 4000$ and $< 0$ for $N > 4000$. The first result is reassuring: it means that when the starting batch $D$ becomes big enough there is no point in augmenting the data set. The second result is a bit strange: when $N > 4000$ augmenting the data set with the symmetric images of the transitions leads to a model that is farther away from the true one according to the used distance (Equation 4). When $k$ is not a symmetry $\Delta$ almost always $< 0$ and decreases with $N$, however for $N \leq 2000$ and $k$ $\in$ {SDAI, TIOD} $\Delta > 0$. This might be both an artifact caused by a batch that is too small or another hint suggesting that maybe the metric adopted does not serve well its purpose.

Evaluation of performance gain ($\Delta U$) The difference in performance of the deployed policies $\Delta U$ perfectly fits the expected behaviour. When $k$ is a symmetry $\Delta U < 0$ and saturates to 0 with $N$ increasing. When $k$ is not a symmetric transformation of the dynamics $\Delta U > 0$ and keeps increasing with $N$ (see Figure 2c).

5.2 Stochastic CartPole (Continuous)

Detection phase ($\nu_k$) In Stochastic CartPole the algorithm fails to detect the symmetry $k$ = TI. This could be due to the fact that the translation invariance symmetry in this case is fixed for a specific value (see TI in Table 2 where the translation is set at 0.3). If the translation is too small the neural network fails to discern the transformation from the noise. Nevertheless, the algorithm classifies correctly as a symmetry $k$ = SAR and the remaining transformations as non symmetries (see Figure 2a).

Evaluation of distributional shift ($\Delta$) The mean over the batches of the distributional shift $\Delta$ (the $p$ value of the Wilcoxon pairwise signed-rank test) is greater than 0.05 for $k$ $\in$ {TRSAI, TI} (see Figure 2b). The latter means that the null hypothesis $H_0$ defined in Equation 10 can’t be rejected because the difference lies within the interval of confidence: the distribution of errors of the regressor learnt using $D$ is greater than the one obtained by using the regressor trained using $D_k$. When $k$ is not a symmetry $\Delta < 0.05$ and therefore there is a statistically significant difference in the error distribution that allows us to reject $H_0$ and conclude that using the augmented batch to learn the dynamics could lead to a greater distributional shift.

Evaluation of performance gain ($\Delta U$) The results are portrayed in Figure 2c. In this case $\Delta U$ is almost always $> 0$ when $k$ is not a symmetry and almost always $< 0$ when $k$ is a symmetry. The curves are not smooth because the statistics are performed over less simulations. Notice that for $N = 1000$ and $k$ = SFI $\Delta U < 0$, this could be an artifact generated by the small size of the data set and the non perfectly fine-tuned hyperparameters of PPO.

5.3 Stochastic Acrobot (Continuous)

Detection phase ($\nu_k$) In this environment the only real symmetry of the dynamics, AAVI, gets successfully detected by the algorithm with $q = 0.1$. The non symmetries all yield a $\nu_k < 0.5$ (Figure 3a).

Evaluation of distributional shift ($\Delta$) Almost always $\Delta > 0.05$ for AAVI, hence confirming that exploiting the symmetry for data augmenting reduces the distributional shift (see Figure 3b). $H_1$ (Equation 11) is accepted for all the other proposed transformations ($\Delta < 0.05$).

Evaluation of performance gain ($\Delta U$) As we could imagine the mean of the performance difference $\Delta U$ for the only valid symmetry $k$ = AAVI is always $< 0$ (Figure 3c). For the others transformation the results are less evident: the mean oscillates around $\Delta U = 0$ for $N = 1000$ and $N = 3000$. This effect could be generated by both the low number of batches used to perform the average and the necessity to fine-tune the hyperparameters of PPO.

6 Conclusions

Data-efficiency in the offline learning of data-driven MDPs is highly coveted. Exploiting the intuition of an expert about the nature of the model can help to learn dynamics that better represent the reality.

In this work we built a semi-automated tool that can aid an expert providing a statistical data-driven validation of his intuition about some properties of the environment. A correct deployment of the tool could improve the performance of the optimal policy obtained by solving the learnt MDP. Indeed, our results suggest that the proposed algorithm can effectively detect a symmetry of the dynamics of an MDP with high accuracy and that exploiting this knowledge can not only reduce the distributional shift, but also provide performance gain in an envisaged optimal control of the system.

Besides its pros, the current work is still constrained by several limitations:

1. The quality of the approach in continuous MDPs is greatly affected by the architecture of the Normalizing Flow used for Density Estimation and, more generally, by the state-action space preprocessing;

2. The metric used to compute the distributional shift in the categorical case sometimes lead to puzzling result.

Further directions can be considered to expand this approach in future work:

1. We count on trying out more recent Normalizing Flow architectures like FFFJORD (Grathwohl et al. 2019);

2. Finding methods to make the approach more stable with respect to the choice of the hyperparameters of the DNN architectures (regressors, density estimators and Deep Reinforcement Learning agents) is a compelling necessity.
(a) Probability of symmetry $\nu_k$. The threshold at $\nu = 0.5$ is displayed as a dashed line. $\nu_k > 0$ means that the transformation is detected as a symmetry.

(b) Distributional shift $\Delta$. The threshold at $\Delta = 0.05$ is displayed as a dashed line. $\Delta > 0.05$ means that data augmenting does not allow to reject the null hypothesis.

(c) Performance difference $\Delta U$. The threshold at $\Delta U = 0$ is displayed as a dashed line. $\Delta U < 0$ means that data augmenting leads to better policies.

Figure 3: Stochastic Acrobot. $\nu_k$, $\Delta$, and $\Delta U$ for the transformations $k$ computed over sets of different batches of size $N$. Points are mean values and are a bit shifted horizontally for the sake of display.

References

Angelotti, G.; Drougard, N.; and Chanel, C. P. C. 2020. Offline Learning for Planning: A Summary. In Proceedings of the 1st Workshop on Bridging the Gap Between AI Planning and Reinforcement Learning (PRL) at the 30th International Conference on Automated Planning and Scheduling, 153–161.

Angelotti, G.; Drougard, N.; and Chanel, C. P. C. 2021. Expert-Guided Symmetry Detection in Markov Decision Processes. arXiv:2111.10297.

Bellman, R. 1966. Dynamic Programming. Science, 153(3731): 34–37.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. OpenAI Gym. arXiv preprint arXiv:1606.01540.

Dean, T.; and Givan, R. 1997. Model Minimization in Markov Decision Processes. In AAAI/IAAI, 106–111.

Dinh, L.; Krueger, D.; and Bengio, Y. 2015. NICE: Non-linear Independent Components Estimation. In Bengio, Y., and LeCun, Y., eds., 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings.

Grathwohl, W.; Chen, R. T. Q.; Bettencourt, J.; Sutskever, I.; and Duvenaud, D. 2019. FFJORD: Free-Form Continuous Dynamics for Scalable Reversible Generative Models. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net.

Gross, D. J. 1996. The role of symmetry in fundamental physics. Proceedings of the National Academy of Sciences, 93(25): 14256–14259.

Kobyzev, I.; Prince, S.; and Brubaker, M. 2020. Normalizing Flows: An Introduction and Review of Current Methods. IEEE Transactions on Pattern Analysis and Machine Intelligence.

Levine, S.; Kumar, A.; Tucker, G.; and Fu, J. 2020. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. ArXiv, abs/2005.01643.

Li, L.; Walsh, T. J.; and Littman, M. L. 2006. Towards a Unified Theory of State Abstraction for MDPs. ISAIM, 4: 5.

Narayananmurthy, S. M.; and Ravindran, B. 2008. On the Hardness of Finding Symmetries in Markov Decision Processes. In Proceedings of the 25th international conference on Machine learning, 688–695.

Papamakarios, G.; Pavlakou, T.; and Murray, I. 2017. Masked Autoregressive Flow for Density Estimation. In Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17, 2335–2344. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781510860964.

Park, D. S.; Chan, W.; Zhang, Y.; Chiu, C.-C.; Zoph, B.; Cubuk, E. D.; and Le, Q. V. 2019. SpecAugment: A simple data augmentation method for automatic speech recognition. arXiv preprint arXiv:1904.08779.

Raffin, A.; Hill, A.; Gleave, A.; Kanervisto, A.; Ernestus, M.; and Dormann, N. 2021. Stable-Baselines3: Reliable Reinforcement Learning Implementations. Journal of Machine Learning Research, 22(268): 1–8.
Ravindran, B.; and Barto, A. G. 2001. Symmetries and Model Minimization in Markov Decision Processes. Technical report, USA.

Ravindran, B.; and Barto, A. G. 2004. Approximate Homomorphisms: A Framework for Non-exact Minimization in Markov Decision Processes.

Schulman, J.; Wolski, F.; Dhariwal, P.;Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

Shorten, C.; and Khoshgoftaar, T. M. 2019. A survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 6(1): 1–48.

van der Pol, E.; Kipf, T.; Oliehoek, F. A.; and Welling, M. 2020a. Plannable Approximations to MDP Homomorphisms: Equivariance under Actions. In Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS ’20, 1431–1439. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450375184.

van der Pol, E.; Worrall, D.; van Hoof, H.; Oliehoek, F.; and Welling, M. 2020b. MDP Homomorphic Networks: Group Symmetries in Reinforcement Learning. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., Advances in Neural Information Processing Systems, volume 33, 4199–4210. Curran Associates, Inc.

van Dyk, D. A.; and Meng, X.-L. 2001. The Art of Data Augmentation. Journal of Computational and Graphical Statistics, 10(1): 1–50.