Undergraduate Research Opportunity Program

(UROP) Project Report

# BTRL: Barlow Twins for Model-Based Reinforcement Learning

By

New Jun Jie

Department of Computer Science

School of Computing

National University of Singapore

2020/2021

Undergraduate Research Opportunity Program

(UROP) Project Report

# BTRL: Barlow Twins for Model-Based Reinforcement Learning

By

New Jun Jie

Department of Computer Science

School of Computing

National University of Singapore

2020/2021

Project No: U226020

Advisor: Dr. Harold Soh

Deliverables:

UROP Final Report

**Abstract**

Advancements in deep reinforcement learning, while successful on many sequential decision making problems, are still limited in problems such as control tasks with complex visual observations. Representation learning methods such as latent variable models and contrastive learning recently emerged that learn compressed yet informative latent representations of the state space for model-based reinforcement learning. In this report, I propose the application of Barlow Twins for model-based reinforcement learning (BTRL), that learns a state representation in an unsupervised way, maximising the information across learnt state features, and report an improvement in performance and robustness on control tasks with complex visual observations. Through experiments on the DeepMind Control Suite with backgrounds replaced with videos from the ILSVRC dataset, I show empirically that while BTRL strongly outperforms Dreamer on complex visual control tasks, its subpar performance compared to CVRL and its diagnosis is limited by difficulties with the high dimensionality of projections, representational collapse and the variability of design parameters of the BTRL architecture.

Subject Descriptors:

I.2.8 Problem Solving, Control Methods, and Search

I.2.9 Robotics

I.2.10 Vision and Scene Understanding

Keywords:

Model-based Reinforcement Learning, Representation Learning, Contrastive Learning.

Implementation Software and Hardware:

Python, TensorFlow, TensorFlow Probability

**Acknowledgement**

I would like to thank Professor Harold Soh for his invaluable mentorship over the past year on my Undergraduate Research Opportunities Programme (UROP). His guidance and patience was essential to my learning of research skills and methodology. I would also like to thank my seniors Kok Bingcai and Chen Kaiqi for their advice on my work and on research in general. I would also like to express my gratitude to my friends He Shiying, Tan Jun Jie and Ng Wai Ching for their support in my 1-year research journey.

# List of Figures

# Table of Contents

# Chapter 1

# Introduction

"Reinforcement learning is learning what to do–how to map situations to actions–so as to maximize a numerical reward signal" [28]. With the high expressiveness of deep neural networks, reinforcement learning has been proven to be effective across a range of highly complex sequential decision making problems, such as long-horizon control and dexterous manipulation in robotics.

While reinforcement learning faces a problem of high sample complexity where a large number of training samples are required, model-based reinforcement learning learns an explicit model of environment dynamics, reducing the reliance on real-world samples. While model-based reinforcement learning algorithms have been successful in not just sample efficiency and performance [13,22], unresolved challenges remain, one being the high dimensionality of observations that limits training efficiency of both the model and the agent policy, and subsequently performance. Representation learning methods are thus proposed to learn a compressed yet informative latent representation of the state space.

The challenge of high-dimensional observations is prevalent. Traditionally, reinforcement learning algorithms have been trained within simulations where state-based observations are readily available. In domains such as robotics, the goal is to develop algorithms that remain effective in the real world, yet the real world rarely has state-based observations, and the algorithm must necessarily rely on low-dimensional image-based observations. However, images observed in the real world are much noisier and more complex than images obtained from simulations. Reinforcement learning algorithms need to learn the parts of the image observation that are relevant to the task and ignore the rest that are irrelevant. Therefore, the objective is to learn an effective representation of the state from image-based observations that enables high-performing reinforcement learning algorithms.

Barlow Twins is an unsupervised representation learning method designed for ImageNet tasks, that is simple yet competitive relative to state-of-the-art contrastive learning methods. In this report, I study the application of Barlow Twins as a state representation method for model-based reinforcement learning in the presence of visual distractors in image-based observations. I show empirically that while Barlow Twins is able to learn representations that are invariant to visual distractors, its hyperparameters can be highly sensitive and difficult to tune.

# Chapter 2

# Preliminaries

## 2.1 Reinforcement Learning

The reinforcement learning framework consists of an agent learning from its interactions with the environment [28]. With every action $a_t$ taken by the agent, the environment returns a state $s_{t+1}$ and a reward $r_{t+1}$. Reinforcement learning problems can be formally modelled as a Markov Decision Process (MDP), a 4-tuple $(S, A, T_a, R_a)$, where $S$ is a set of states, $A$ is a set of actions where $A_s \subseteq A$ is the set of actions available from state $s$. $T_a$ is the transition function, where $T_a(s, s')$ is the probability that action $a$ in state $s$ at time $t$ will lead to state $s'$ at time $t + 1$. $R_a(s, s')$ is the immediate reward received after transitioning from state $s$ to state $s'$ having taken action $a$.

The goal of reinforcement learning is to find the optimal policy $a = \pi^*(s)$, that gives the best action $a$ in all states $s \in S$ that will maximise the reward. The expected sum of future rewards $V^\pi(s) = E[\sum_{t=0}^{\infty} \gamma^t R_{\pi(s_t)}(s_t, s_{t+1})]$, discounted with parameter $\gamma$ over $t$ time steps, with $s = s_0$. $V^\pi(s)$ is the value function of a state.

## 2.2 Model-based Reinforcement Learning

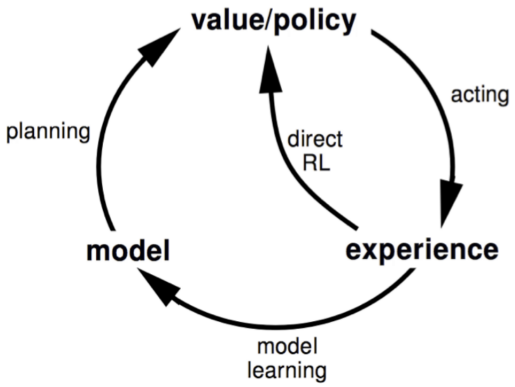
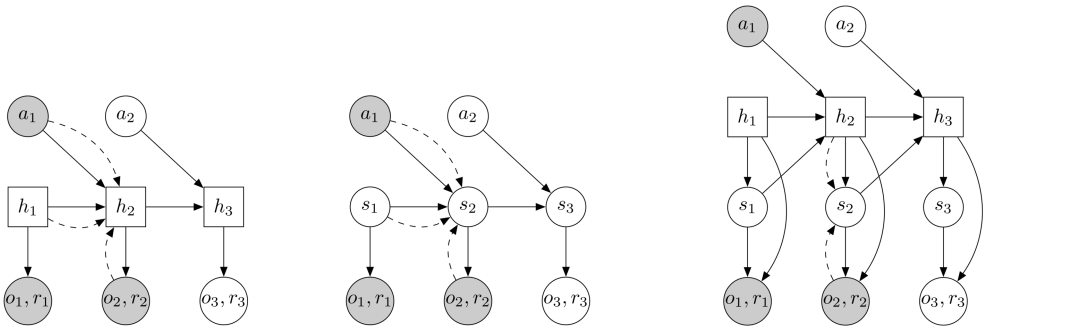



Figure 2.1: Model-Based Reinforcement Learning

Reinforcement learning suffers from the problem of sample inefficiency, and worsened by limited real-world data in many domains such as robotics, motivating the need for model-based reinforcement learning, having a model that models the environment dynamics, allowing sampling from the environment model, improving sample efficiency in terms of real-world data.

In model-based reinforcement learning (MBRL), a transition model $T_a(s, s')$ and optionally a reward model $R_a(s, s')$ are learnt. The models can be learned by sampling the environment and then be used to update the policy and value. When learning the transition/reward model is less complex than learning the policy model, the model-based approach is more sample efficient [24].

## 2.3 Latent State-Space Models




(a) Deterministic model (RNN) (b) Stochastic model (SSM) (c) Recurrent state-space model (RSSM)

Figure 2.2: Latent State-Space Models

State-space models (SSMs) learn a representation of the state and can model complex non-linear transition dynamics for model-based reinforcement learning. Given a probabilistic graphical model, the joint distribution of a SSM can be factorized as:

$$p_\theta(o_{1:T}, r_{1:T}, s_{1:T}|a_{1:T}) = \Pi_{t=1}^{T} p_\theta(o_t|s_t)p_\theta(r_t|s_t)p_\theta(s_t|s_{t-1}, a_{t-1}))$$

where $\theta$ are learnable model parameters, $o_{1:T}$ denotes all observations from $t = 1, ..., T$, and likewise for $r_{1:T}$, $s_{1:T}$ and $a_{1:T}$, and the 3 distributions in the factorization correspond to observations $p_\theta(o_t|s_t)$, rewards $p_\theta(r_t|s_t)$, and transitions $p_\theta(s_t|s_{t-1}, a_{t-1})$.

The state space model can be viewed as a partially-observable Markov decision process (POMDP), where $\theta$ is learned from observed data $D = o_t, a_t, r_t{}_{t=1}^{T}$. Since maximum likelihood estimation is intractable as latent $s_t$'s need to be marginalized out, the evidence lower bound (ELBO) under the data distribution $p_d$ can be optimized, i.e. $E_{p_d}[L_e] \leq E_{p_d}[logp_\theta(s_{1:T}, r_{1:T}|a_{1:T})]$, where

$$L_e = \sum_{t=1}^{T} (E_{q_\phi(s_t)}[logp_\theta(o_t|s_t)] + E_{q_\phi(s_t)}[logp_\theta(r_t|s_t)] - E_{q_\phi(s_{t-1})}[D_{KL}[q_\phi(s_t)\|p_\theta(s_t|s_{t-1}, a_{t-1})]])$$

and $q_\phi$ is a variational distribution parameterized by $\phi$.

# Chapter 3

# Literature Review and Related Work

## 3.1 Learning Latent Representations for MBRL via Reconstruction

World Models was introduced by Ha and Schmidhuber (2018) [12] that captures spatial and temporal information of a state using a variational autoencoder [19] and a mixture density recurrent neural network [5]. The resultant spatio-temporal latent representation is then processed by the agent instead of raw images for learning. World Models further showed that an agent that learns purely within the world model, using only model samples without any real samples, can outperform high-performing model-free reinforcement learning algorithms.

While relatively successful, learning dynamics models that are sufficiently accurate for planning in the latent space, especially in image-based domains, remain a challenge. The limitation of planning accuracy by the compounding error phenomenon, where one-step prediction errors of learnt transition models can accumulate over multiple steps, is a well-studied phenomenon [2, 18, 25, 29]. To improve accurate planning over multiple time-steps, Hafner et al. (2019) proposed the Deep Planning Network (PlaNet) [14], which showed that a dynamics model with deterministic and stochastic components can be trained with latent overshooting to minimise multi-step prediction errors, solving control tasks that exceed the difficulty of tasks previously solved by planning with learned models [3, 33].

Following the effectiveness of latent state representation for learning and applying latent representations for forward planning, Hafner et al. (2020) proposed Dreamer [13], that trains both the dynamics model and the agent policy end-to-end, significantly outperforming all previous model-free and model-based methods to learn long-horizon behaviours purely within its world model, achieving human-level performance on 55 Atari games [4]. Nonetheless, while Dreamer successfully performs accurate long-term predictions up to 45 steps forward in latent space, recent model architecture advancements with transformers can show improvements up to 100 steps in model prediction [17].

Another approach to resolve the compounding error phenomenon is the application of dynamics models that use forward, backward and inverse models. The forward model $s_{t+1} = f(s_t, a_t)$ predicts the next state $s_{t+1}$ given a current state $s_t$ and action $a_t$ at time $t$. The backward model $s_t = f(s_{t+1}, a_t)$ predicts the previous state given the current state and previous action taken. The inverse model $a_t = f(s_t, s_{t+1})$ predicts the action that takes the current state to the next state. To improve planning accuracy, combinations of directional dynamics models were proposed to better capture the context of a state within its latent representation.

Lai et al. (2020) proposed Bidirectional Model-based Policy Optimisation (BMPO), showing that constructing an additional backward dynamics model alongside a forward model lowers reliance on the accuracy of predictions of the forward model [20]. Since BMPO uses both forward and backward models to generate shorter roll-outs from real-world samples, prediction errors have fewer steps to compound, and BMPO outperforms existing model-based methods in sample efficiency and performance. In contrast to implicitly learning the context of a state, Lee et al. (2020) proposed the Context-aware Dynamics Model (CaDM) that learns an explicit context latent vector that captures contextual information useful for both forward and backward prediction.

## 3.2 Contrastive Learning of Latent Representations for MBRL

Latent variable and directional dynamics models mentioned in previous sections require prediction of possibly high-dimensional observations, needing to reconstruct all pixels of an image-based observation. Pixel-based reconstruction is limited, especially when a small pixel change in the environment is crucial to performing well in a task, such as a small bullet on the screen. In the real world, observations also tend to be visually complex, and pixel reconstruction does not generalise well to similar tasks with different appearances. Contrastive representation learning is an info-theoretic approach to model a latent representation based on the mutual information between variables, aiming to capture only informative or task-relevant features of the observation, circumventing the problems with pixel-based reconstruction.

Contrastive learning is a self-supervised learning method that learns useful representations without the need for labels [8, 9, 11, 15, 35], which has shown to close the performance gap between supervised and unsupervised methods of deep image models [6, 7]. Contrastive learning pulls together similar "positive" samples and pushes away different "negative" samples in the embedding space, and the resulting representation vector has shown to be highly performant for downstream tasks.

Srinivas et al. (2020) proposed Contrastive Unsupervised Representations for Reinforcement Learning (CURL) that extracts high-level features from raw pixels using contrastive learning to perform off-policy control on top of the extracted features [21]. CURL first performs data augmentations, such as random cropping, to obtain data-augmented "views" of the same image that are labelled as positive samples and other images are labelled as negative samples for contrastive learning. CURL showed that contrastive learning of augmented views nearly matches the sample efficiency of methods that use state-based features, significantly outperforming PlaNet, Dreamer and other model-free methods.

## 3.3 Learning to Solve Complex Visual Control Tasks

Other than contrasting between augmented views of the same image to represent states, a range of contrastive learning formulations has recently emerged. Ma et al. (2020) contrasts between an image and its learnt latent representation [22], Nguyen et al. (2021) contrasts between images of adjacent timesteps [23], Stooke et al. (2021) contrasts between images in the same trajectory [27] and Yarats et al. (2021) contrasts between data-augmented views of images at adjacent timesteps, assigning representations to prototypes before contrasting the assigned prototypes to accelerate exploration [34]. In contrast, similar to Ma et al. (2020) [22], our work contrasts between an image and its learnt latent representation, but inspired by the sensitivity of the choice negative samples, BTRL employs the Barlow Twins architecture to remove the need to explicitly choose negative samples.

# Chapter 4

# Barlow Twins for Model-Based Reinforcement Learning

I propose Barlow Twins as a state representation method for model-based reinforcement learning (BTRL) with complex visual observations. Only part of the true underlying current state of the environment is available in the image-based observations. The complex visual task is formulated as a partially observable Markov decision process (POMDP) with discrete time $t = 1, 2, ..., T$, continuous action $a_t$, image-based observation $o_t$, and a scalar reward $r_t$ obtained after each action is taken.

BTRL learns a world model in an unsupervised manner, representing the state as a latent vector that can be projected into de-correlated vectors. Unlike contrastive learning, which requires choosing of explicit negative samples, Barlow Twins does not require the choice of negative samples. Barlow Twins is a significantly more robust representation learning method than reconstruction-based representation methods, because it avoids pixel-level reconstruction of complex observations.

## 4.1 Barlow Twins for State Representation

Barlow Twins employs the loss function $L_{BT}$:

$$L_{BT} = \sum_i (1 - C_{ii})^2 + \lambda \sum_i \sum_{j \neq i} C_{ij}^2$$

where $\lambda$ is a positive constant that trades off the first and second terms of the loss that represents the on-diagonals and off-diagonals respectively, and $C$ is the cross-correlation matrix computed between outputs of two networks along the batch dimension:

$$C_{ij} = \frac{\sum_b z_{b,i}^A, z_{b,j}^B}{\sqrt{\sum_b (z_{b,i}^A)^2} \sqrt{\sum_b (z_{b,j}^B)^2}}$$

where $b$ indexes batch samples and $i, j$ index the vector dimension of the networks' outputs. $C$ is a

square matrix with size of the dimensionality of the network's output, with values comprised between -1 (i.e. perfect anti-correlation) and 1 (i.e. perfect correlation).

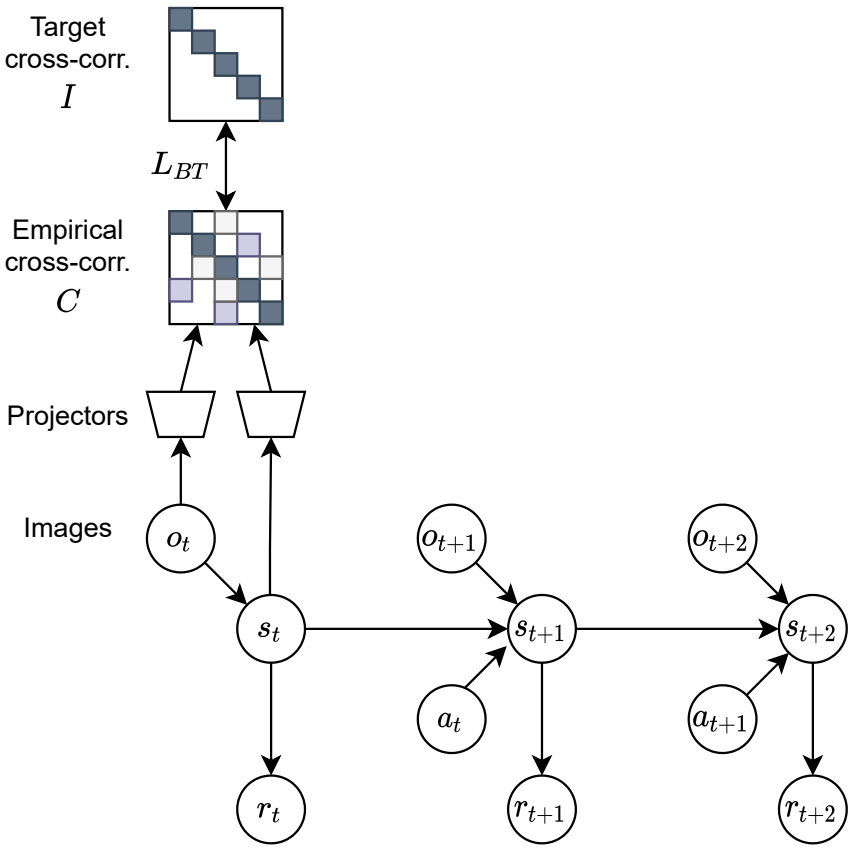



Figure 4.1: BTRL Model Architecture

BTRL applies the Barlow Twins loss between projections of the observations $o_t$ and projections of the states $s_t$ of the same timestep $t$. The design of the Barlow Twins loss naturally reduces redundancy between feature variables in the latent representation of the state.

## 4.2 Model-Based Reinforcement Learning with Decorrelated Latents

BTRL applies the Barlow Twins for model-based reinforcement learning with the following loss:

$$\sum_{t=1}^{T}(E[L_{BT}] + E[logp_\theta(r_t|s_t)] - E[KL[q_\phi(s_t|o_{\leq t}, a_{\leq t})]||p_\theta(s_t|s_{t-1}, a_{t-1})])$$

where the first term is the Barlow Twins loss, the second term is the reconstruction loss for reward prediction, and the third term is the dynamics loss.

# Chapter 5

# Experiments

I empirically evaluate BTRL on various settings of continuous Mujoco control tasks in the DeepMind Control suite. I evaluate its ability to handle a more realistic but more complex scenario of visual tasks, DeepMind Control with natural backgrounds. I discuss how the model is superior compared to existing methods. Finally, I conduct ablation studies to demonstrate the importance of various components of the BTRL model.

## 5.1 Experiment Setup

### 5.1.1 Control Tasks

The DeepMind Control (DMC) Suite is a set of continuous control tasks that serve as performance benchmarks for reinforcement learning agents [30]. The DMC Suite is powered by the MuJoCo physics engine [31]. BTRL is evaluated on 6 DeepMind Control (DMC) tasks: Cartpole Swingup, Cheetah Run, Walker Run, Pendulum Swingup, Hopper Hop and Cup Catch. The tasks' background are replaced with natural videos sampled from the ILSVRC dataset [26]. All experiment runs are each completed over 500,000 training episodes over 3 random seeds.

### 5.1.2 Baseline Methods

I compare BTRL with Dreamer and CVRL. Dreamer is the current state-of-the-art model-based method for planning from pixels [13]. CVRL is a closely-related contrastive-based model that aims to perform well in complex visual tasks [22]. Each method is evaluated by the environment return in 1000 steps. For the baselines, we use the best set of hyperparameters as reported in their paper.

Dreamer consists of a world model and an actor-critic. The world model comprises 3 models: a representation model $p(s_t|s_{t-1}, a_{t-1}, o_t)$ that encodes an image into a latent state, a transition model $q(s_t|s_{t-1}, a_{t-1})$ that predicts forward in latent space, and a reward model $q(r_t|s_t)$ that predicts the reward given a latent state. The actor-critic comprises 2 models: an action model $a_t \sim q_\phi(a_t|s_t)$ that implements the agent policy, and a value model $v_\psi(s_t) \approx E_{q(\cdot|s)}(\sum \gamma^{T-t} r_t)$. The 3 models that comprise the world model are trained jointly, with the combined loss function:

$$E_p[\sum_t (lnq(o_t|s_t) + lnq(r_t|s_t) + -\beta KL(p(s_t|s_{t-1}, a_{t-1}, o_t)||q(s_t|s_{t-1}, a_{t-1}))]$$

where $KL(p, q)$ is the Kullback-Leibler divergence.

CVRL, like Dreamer, consists of a world model and an actor-critic. Unlike Dreamer, CVRL employs a contrastive learning term, $log\frac{f_\theta(o_t,s_t)}{\sum_{o_t'\in O_t} f_\theta(s_t,o_t')}$, in the loss function for joint training of the world model, in replacement of the representation model in Dreamer, with the combined loss function:

$$\sum_{t=1}^{T}(E[log\frac{f_\theta(o_t, s_t)}{\sum_{o_t'\in O_t} f_\theta(s_t, o_t')}] + E[logp_\theta(r_t|s_t)] - E[KL[q_\phi(s_t|o_{\leq t}, a_{\leq t})||p_\theta(s_t|s_{t-1}, a_{t-1})])$$

## 5.2 Comparisons in the Natural Background Setting

The natural background setting where backgrounds of the DMControl tasks are replaced with natural videos from the ILSVRC dataset provides visual distractors that are irrelevant to the task, mimicking the complex visual observation setting in the real world. The natural background setting requires the agents to retrieve information from the image-based observation that are relevant to the task, compressing it into an abstract state representation.

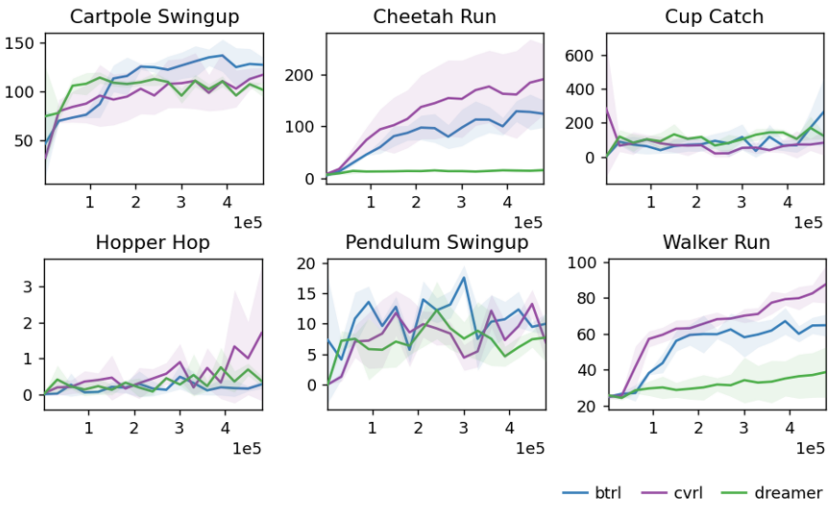



Figure 5.1: Comparison between BTRL, CVRL and Dreamer on Natural MuJoCo

In the natural background setting, BTRL's performance exceeds that of Dreamer for Cartpole Swingup, Cheetah Run and Walker Run. The superior performance of BTRL over Dreamer on Natural MuJoCo shows that the Barlow Twins architecture learns a state representation that is more robust to complex visual observations. However, although the performance of BTRL comes close to CVRL, it underperforms on Cheetah Run, Hopper Hop and Walker Run, and outperforms only on Cartpole Swingup.

## 5.3 Ablations

### 5.3.1 Projection Dimension

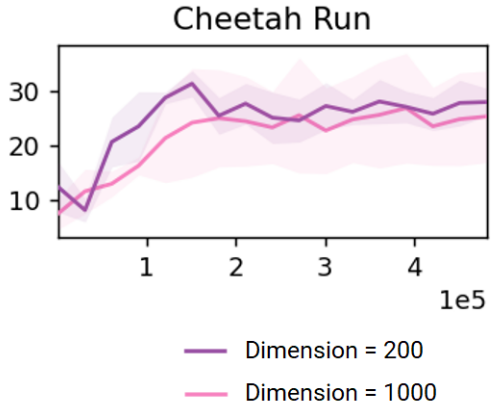



Figure 5.2: Effect of Projection Dimension on Performance

A sufficiently large dimension of the projection vector is shown to be important when the Barlow Twins architecture was used for linear evaluation on ImageNet. [35] showed that Barlow Twins performs better when the dimensionality of the projector network output if very large. However, a comparison between a projection dimension of 200 and 1000 shows that performance does not differ much for BTRL on the Natural Cheetah Run task.

### 5.3.2 Barlow Twins Loss Weights

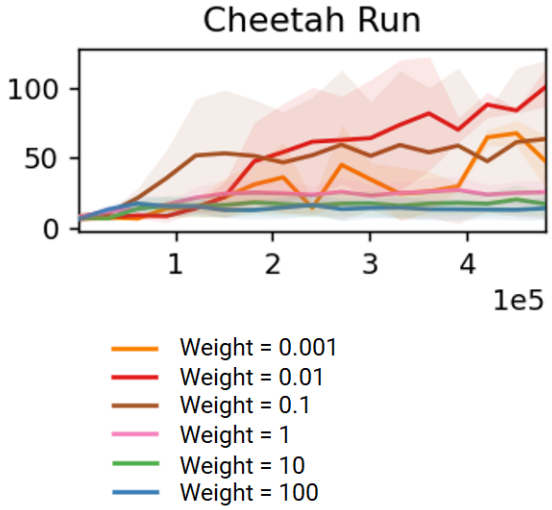



Figure 5.3: Effect of BT Loss Weight on Performance

Since the components of the world model, the representation model, transition model and reward model are trained jointly, the BT loss weight determines the amount of influence the loss has on the

joint training of the world model. The performance of BTRL is empirically shown to be sensitive to the BT loss weight. The optimal BT loss weight is 0.001. When the BT loss weight exceeds 1, training collapses and BTRL fails to learn.

### 5.3.3 Loss Coefficients

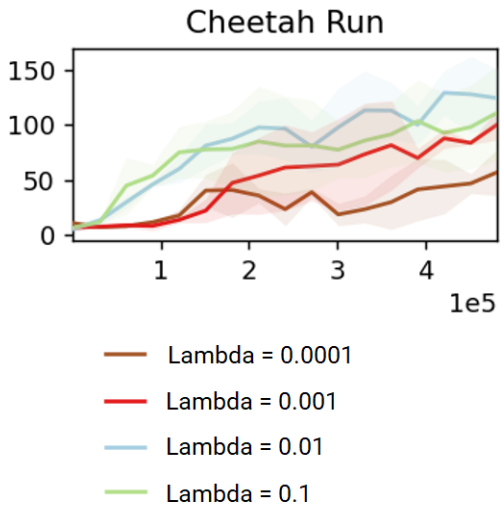



Figure 5.4: Effect of BT Loss Lambda on Performance

The Barlow Twins loss comprises coefficients of the on-diagonals and off-diagonals of the correlation matrix between projection vectors. $\lambda$ is a hyperparameter that controls the trade-off between the off-diagonal and on-diagonal losses. A higher value of $\lambda$ means that off-diagonal coefficients are weighted more heavily. The optimal value of $\lambda$ is 0.01.

# Chapter 6

# Conclusions: Discussion, Limitations and Future Work

## 6.1 Discussion

### 6.1.1 Relation to CVRL

Given that the performance of Barlow Twins exceeds that of methods that employ the InfoNCE loss on linear evaluation on ImageNet [35], it is unexpected that the performance of BTRL only comes close to CVRL, underperforming on Cheetah Run, Hopper Hop and Walker Run, and outperforms only on Cartpole Swingup. Compared to InfoNCE, InfoNCE can be shown to maximize the variability of embedding learnt by maximizing pairwise distance between all pairs of samples while Barlow Twins decorrelates the components of the embedding vectors. Since BTRL requires high-dimensional projection vectors, there are more trainable parameters and a larger search space, increasing the difficulty of training in the reinforcement learning context, since the embeddings learnt need to be adapted over time as the agent collects more diverse observations through its exploration in the environment. One possible future direction is to reduce the reliance on a large-dimension projection vector of informative state representations.

### 6.1.2 Negative Samples and Batch Statistics

CVRL selects negative samples of a state-observation pair from other trajectories in the same batch. While CVRL requires the explicit choice of negative samples, which Ma et al. (2020) states significantly affects the quality of contrastive learning [22], BTRL uses negative samples implicitly from the batch through normalization with batch statistics and thus does not require the explicit choosing of negative samples. The performance of BTRL would be expected to be similarly dependent on the choice of batch samples since it affects batch statistics, and future work should inspect the influence of batch statistics and the choice of batch samples.

### 6.1.3 Relation to Other Contrastive Formulations

Temporal Predictive Coding (TPC) is a similar information-theoretic approach that employs predictive coding to encode elements in the environment that can be predicted across time [23]. TPC, unlike CVRL, employs a reconstruction-free loss function. Unlike BTRL and CVRL which contrasts between image-based observations and learnt state representations, TPC contrasts between an observation in the current time step $o_t$ and an observation-action pair from the previous timestep $(o_t, a_t)$. A range of other contrastive formulations exist, and it may be worthwhile to investigate the application of Barlow Twins in the form of other formulations.

## 6.2 Limitations

Although BTRL shows a significant performance improvement over Dreamer in my experiments, there is an unexpected performance gap between BTRL and CVRL, since Barlow Twins was shown to outperform multiple other methods that employ the InfoNCE loss, such as SimCLR [35]. In this section, I explain difficulties with designing and diagnosing BTRL within the span of time of my research.

### 6.2.1 Representational Collapse

In self-supervised learning, a common idea behind most state-of-the-art approaches is to enforce invariance of representation embeddings to pre-defined augmentations or other other irrelevant differences. One issue is the existence of completely collapsed solutions. Hua et al. (2021) proves the existence of a complete collapse, where all inputs map to the same embedding, and a commonly overlooked dimensional collapse, where axes collapse to strong correlations between axes [16]. Dimensional collapse can be avoided by feature decorrelation methods such as standardizing covariance, potentially explaining the effectiveness of redundancy reduction in Barlow Twins. The importance of batch normalization as one method to curb collapse is also shown empirically in other related work [10]. Nonetheless, representational collapse and how it affects downstream task performance is not fully understood, and further work on representational collapse is necessary to pinpoint problems with BTRL.

### 6.2.2 Variability of Design Parameters

The difficulty of designing architectural changes and tuning hyperparameters, that is already difficult due to challenges in deep reinforcement learning such as high variance in training, hyperparameter sensitivity of results and heavy computational requirements, is exacerbated by the need to tune several components in model-based reinforcement learning – training of a dynamics model, the prediction of trajectories, policy optimization and planning – each component requiring tuning of several design parameters that significantly impact performance [36]. The impact of such difficulties is compounded by discrepancies in the comparison across published methods' performance [32], due to the statistical uncertainty implied by the use of a limited number of training runs [1]. Without more robust benchmarking and comparisons, the relative performance of BTRL compared to related methods such as CVRL and TPC cannot be reasonably determined.

## 6.3 Conclusion

Model-based reinforcement learning has benefited greatly from advancements in unsupervised representation learning, such as Dreamer, CVRL, TPC. In this report, I propose Barlow Twins for model-based reinforcement learning (BTRL). I show that while BTRL outperforms Dreamer on complex visual control tasks, its subpar performance compared to CVRL and its diagnosis is limited by difficulties with the high dimensionality of projections, representational collapse and the variability of design parameters.

# References


[1] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *arXiv preprint arXiv:2108.13264*, 2021.

[2] Kavosh Asadi, Dipendra Misra, and Michael Littman. Lipschitz continuity in model-based reinforcement learning. In *International Conference on Machine Learning*, pages 264–273. PMLR, 2018.

[3] Ershad Banijamali, Rui Shu, Hung Bui, Ali Ghodsi, et al. Robust locally-linear controllable embedding. In *International Conference on Artificial Intelligence and Statistics*, pages 1751–1759. PMLR, 2018.

[4] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

[5] Christopher M Bishop. Mixture density networks. 1994.

[6] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018.

[7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[9] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[10] Abe Fetterman and Joshal Albrecht, 2020. URL: `https://generallyintelligent.ai/blog/2020-08-24-understanding-self-supervised-contrastive-learning/`.

[11] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

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

[13] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

[14] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, pages 2555–2565. PMLR, 2019.

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[16] Tianyu Hua, Wenxiao Wang, Zihui Xue, Sucheng Ren, Yue Wang, and Hang Zhao. On feature decorrelation in self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9598–9608, 2021.

[17] Michael Janner, Qiyang Li, and Sergey Levine. Reinforcement learning as one big sequence modeling problem. *arXiv preprint arXiv:2106.02039*, 2021.

[18] Nan Rosemary Ke, Amanpreet Singh, Ahmed Touati, Anirudh Goyal, Yoshua Bengio, Devi Parikh, and Dhruv Batra. Learning dynamics model in reinforcement learning by incorporating the long term future. *arXiv preprint arXiv:1903.01599*, 2019.

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

[20] Hang Lai, Jian Shen, Weinan Zhang, and Yong Yu. Bidirectional model-based policy optimization. In *International Conference on Machine Learning*, pages 5618–5627. PMLR, 2020.

[21] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, pages 5639–5650. PMLR, 2020.

[22] Xiao Ma, Siwei Chen, David Hsu, and Wee Sun Lee. Contrastive variational model-based reinforcement learning for complex observations. *arXiv preprint arXiv:2008.02430*, 2020.

[23] Tung Nguyen, Rui Shu, Tuan Pham, Hung Bui, and Stefano Ermon. Temporal predictive coding for model-based planning in latent space. *arXiv preprint arXiv:2106.07156*, 2021.

[24] Aske Plaat, Walter Kosters, and Mike Preuss. Deep model-based reinforcement learning for high-dimensional problems, a survey. *arXiv preprint arXiv:2008.05598*, 2020.

[25] Sébastien Racanière, Théophane Weber, David P Reichert, Lars Buesing, Arthur Guez, Danilo Rezende, Adria Puigdomenech Badia, Oriol Vinyals, Nicolas Heess, Yujia Li, et al. Imagination-augmented agents for deep reinforcement learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5694–5705, 2017.

[26] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[27] Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In *International Conference on Machine Learning*, pages 9870–9879. PMLR, 2021.

[28] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[29] Erik Talvitie. Model regularization for stable sample rollouts. In *UAI*, pages 780–789, 2014.

[30] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

[31] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

[32] Tingwu Wang, Xuchan Bao, Ignasi Clavera, Jerrick Hoang, Yeming Wen, Eric Langlois, Shunshi Zhang, Guodong Zhang, Pieter Abbeel, and Jimmy Ba. Benchmarking model-based reinforcement learning. *arXiv preprint arXiv:1907.02057*, 2019.

[33] Manuel Watter, Jost Tobias Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. *arXiv preprint arXiv:1506.07365*, 2015.

[34] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Reinforcement learning with prototypical representations. *arXiv preprint arXiv:2102.11271*, 2021.

[35] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021.

[36] Baohe Zhang, Raghu Rajan, Luis Pineda, Nathan Lambert, André Biedenkapp, Kurtland Chua, Frank Hutter, and Roberto Calandra. On the importance of hyperparameter optimization for model-based reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 4015–4023. PMLR, 2021.