Generalisation Ability of Proper Value Equivalence Models in Model-Based Reinforcement Learning

Severin Bratus[1]

Supervisor(s): Frans Oliehoek[1]**, Jinke He**[1]

[1]**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2024

Name of the student: Severin Bratus
Final project course: CSE3000 Research Project
Thesis committee: Frans Oliehoek, Jinke He, Mathijs de Weerdt

Generalisation Ability of Proper Value Equivalence Models in Model-Based Reinforcement Learning

Abstract

We investigate the generalization performance of predictive models in model-based reinforcement learning when trained using maximum likelihood estimation (MLE) versus proper value equivalence (PVE) loss functions. While the more conventional MLE loss aims to fit models to predict state transitions and rewards as accurately as possible, value-equivalent methods (e.g. PVE) prioritize value-relevant features. We show that in a tabular setting, MLE-based models generalize better than their PVE counterparts when fit to a small number of training policies, whereas PVE-based models perform better as the number of policies increases. With increasing model rank, generalisation error tends to improve for MLE and PVE, and the two become closer in generalisation ability.

1 Introduction

Reinforcement learning (RL) is broadly divided into model-free and model-based methods, where model-based methods (MBRL) equip the agent with an internal model of the environment dynamics and rewards, which can be used for planning to make better decisions based on the anticipated consequences of agent's actions [1].

In standard model learning, models are commonly fit on sampled state transitions and rewards from the environment, to predict the next state and reward based on a given state, with a loss based on the maximum likelihood estimation principle (MLE). However, in a complex environment with some features irrelevant for the ultimate objective of the agent (which is to maximise the value function), not all aspects of the environment need to be accurately modelled for optimal planning behaviour. This is addressed by value-equivalent (VE) methods, which enforce that the dynamics model most closely captures value-relevant features of the environment. [2]

Recent work has introduced the *value equivalence principle*: two models are value-equivalent with respect to a set of functions and a set of policies if they result equivalent updates under the corresponding induced Bellman operators for any function and policy in the sets [3]. Further work has introduced proper value equivalence (PVE), which no longer requires a set of functions to be specified [4]. Often in MBRL one is interested in models that are value-equivalent to a model that describes the environment perfectly [3]. A model which is PVE to the environment wrt all possible policies, is sufficient for planning – a policy optimal in PVE will also be optimal in the modeled environment [4].

Some empirically successful RL methods, e.g. MuZero [5], have been shown to approximately optimize the PVE loss (wrt to the behaviour policy) [4]. However, MuZero's predictive model fails to generalize in policy evaluation for unseen or unfamiliar policies [6]. Generalization ability of models fit by PVE loss has not yet been systematically explored, though it may offer insights on applicability of VE methods.

In this work, we aim to answer the following question:

> How do predictive models based on MLE and PVE loss functions compare in evaluation of unseen policies?

We hypothesize in a low-data regime with relatively large models MLE-based models will generalize better, as they do not prioritise any particular set of policies. In a high-data regime with smaller models we hypothesize VE-based methods will have an advantage, since they are able to use the limited model capacity to best represent value-relevant features.

Based on the above we pose the following sub-questions:

1. How does the number of policies included in PVE loss affect the generalisation ability of the resulting model?
2. How do MLE-based and PVE-based models trained on the same data compare in test set value prediction error, as (a) model size varies, (b) training dataset size varies?

To answer the above questions, we focus on a tabular domain, where it is tractable to define (deterministic) policies as matrices, and provide policy values for the PVE loss. We measure *generalisation ability* as the value prediction error on the test set. The experimental setup is as follows: we define two sets of policies Π_{train}, Π_{test}, train models with PVE or MLE loss wrt the policies in Π_{test}, and evaluate the resulting models on the policies in Π_{test}.

We find that, as expected, models trained on PVE loss improve in generalisation ability as the number of policies used in their training is increased. When only a few policies are used (4–16), PVE models generalise much worse than those of MLE, and when many are used (32–256), PVE models outperforms MLE models.

2 Background

2.1 General background for dynamics' modeling

The agent's interaction with the environment will be modeled as a *Markov decision process* (MDP) $\mathcal{M} \equiv \langle \mathcal{S}, \mathcal{A}, r, p, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $r(s,a) \equiv \mathbb{E}_{s' \in \mathcal{S}}\, r(s,a,s')$ is the expected reward when taking action a from state s, $p(s'|a,s)$ is the transition kernel defining the distribution of the next state s' when taking action a from state s, and $\gamma \in [0,1)$ is a discount factor [7].

A *policy* $\pi(a|s)$ maps a state s to a distribution over actions. A policy π is *deterministic*, if for every state s it selects some single action a with probability 1.

A policy's *value function* (or state-value function) is defined as

$$v_\pi(s) \equiv \mathop{\mathbb{E}}_{\pi}\Big[\sum_{i=0}^{\infty}\gamma^i r(S_{t+i}, A_{t+i}) \,|\, S_t = s\Big] \tag{1}$$

where $\mathbb{E}_\pi[\cdot]$ is the expectation over the trajectories induced by policy π and transition kernel p, and the random variables S_t, A_t are the state and the action at time t. The problem of computing a policy's state-value function is called *policy evaluation*, or *value prediction* [1].

The objective of the agent is to seek a policy that maximises the value function of every state. In order to evaluate a candidate policy in this search, one can repeatedly apply the policy's Bellman operator to an arbitrary function $v : \mathcal{S} \mapsto \mathbb{R}$. The Bellman operator $\mathcal{T}_\pi$ for a policy pi is defined as follows:

$$\mathcal{T}_\pi[v](s) \equiv \mathop{\mathbb{E}}_{\substack{A\sim\pi(\cdot|s)\\ S'\sim p(\cdot|s,A)}}\Big[r(s,A) + \gamma v(S')\Big] \tag{2}$$

It is known that this procedure converges to the policy's true value function, for any initial v – formally $\lim_{k\to\infty}\mathcal{T}_\pi^k v = v_\pi$. However, the environment dynamics r, p are not known to the agent, so it cannot apply the Bellman operator directly. Instead, the agent can learn a *model* $\tilde{m} \equiv (\tilde{r}, \tilde{p})$ from experience, and use an approximate Bellman operator $\tilde{\mathcal{T}_\pi}$ induced by the model, replacing r, p by the modeled $\tilde{r}, \tilde{p}$, resp.

2.2 Loss functions for learning models

To find $\tilde{r}$, it is common to minimise the following MSE loss:

$$\ell_{r,\mathcal{D}}(r, \tilde{r}) \equiv \mathop{\mathbb{E}}_{(S,A)\sim\mathcal{D}}\left[(r(S,A) - \tilde{r}(S,A))^2\right] \tag{3}$$

where $\mathcal{D}$ is a distribution over the space of state-action pairs $\mathcal{S} \times \mathcal{A}$.

The loss to learn $\tilde{p}$ is typically formulated based on the maximum likelihood estimation principle (MLE):

$$\ell_{p,\mathcal{D}}(p, \tilde{p}) \equiv \mathop{\mathbb{E}}_{(S,A)\sim\mathcal{D}}\left[\mathrm{D_{KL}}(p(\cdot|S,A) \,||\, \tilde{p}(\cdot|S,A))\right] \tag{4}$$

where $\mathrm{D_{KL}}$ is the Kullback-Leibler divergence [3]. Minimising the KL-divergence between the actual data probability and the approximate probability also maximises the likelihood of the data, given the sample size is large enough [8], [9].

Models based on MLE loss will learn to predict aspects of the state transition dynamics not relevant to the value, when a model with a smaller capacity would be able to plan better by focusing on dynamics' value-relevant features [10]. Motivated by this, recent work [4] has introduced the notion *proper value equivalence*: given a set of policies Π, and a set of models $\mathcal{M}$, let

$$\mathcal{M}^\infty(\Pi) = \{\tilde{m} \in \mathcal{M} : \tilde{v}_\pi = v_\pi \;\forall\pi \in \Pi\} \tag{5}$$

where v_π is the true value function of π (in the environment), and $\tilde{v}_\pi$ is the value function of π as evaluated by the model $\tilde{m}$. Each $\tilde{m} \in \mathcal{M}^\infty(\Pi)$ is then said to be **proper value equivalent** to the environment with respect to Π.

Importantly, if a policy is optimal when evaluated by a model $\tilde{m} \in \mathcal{M}^\infty(\mathbb{\Pi})$, the policy is also optimal in the environment. Here $\mathbb{\Pi}$ denotes the set of all possible policies. The property also holds for $\mathbb{\Pi}_{\text{det}}$, the set of all deterministic policies. This implies that it is sufficient for an agent to have a model that belongs to $\mathcal{M}^\infty(\mathbb{\Pi})$ or $\mathcal{M}^\infty(\mathbb{\Pi}_{\text{det}})$ to find an optimal policy.

The PVE loss can be formulated as follows (for any $k \in \mathbb{N}$):

$$\ell^k_\Pi(m^*, \tilde{m}) \equiv \sum_{\pi\in\Pi}\|v_\pi - \tilde{\mathcal{T}}^k_\pi v_\pi\| \tag{6}$$

where $\|\cdot\|$ is any vector norm. Here m^* denotes the exact model of the environment, which in this loss formulation is required to obtain true values of v_π. The loss is valid for any value of $k \geq 1$. Note that v_π is here represented as a vector, where each entry holds the value of a state by policy π. Note also that the PVE loss requires a set of policies with their true state-value functions to be specified, whereas the MLE loss requires a distribution.

3 Related work

Our work builds upon previous research on value equivalence [3], [4], [11], which introduces a theoretic framework with model equivalence classes based on the future use of the model in planning. The VE theory is motivated by the so-called *objective mismatch* problem – dynamics' models fit to conventional approaches, like maximum likelihood estimation (MLE), maximum entropy estimation, the *maximum a posteriori* estimation, or Bayesian posterior inference, have an objective distinct from the planning objective – in other words, models that achieve better likelihood will not necessarily produce high-value policies [10], [12]–[14]. Furthermore, when using an imperfect model for sample rollouts, small errors in value estimation tend to compound, magnifying this discrepancy [15].

Grimm, Barreto, Singh, *et al.* [3] formulated the value equivalence principle – two models are said to be value-equivalent wrt to a set of functions and a set of policies if if they yield the same Bellman updates of the functions on the policies [3]. In particular, one is typically interested in finding an approximate model that belongs to the same equivalence class as the true model of the environment – since such an approximate model may require less capacity than the exact one. To illustrate their claims, Grimm, Barreto, Singh, *et al.* include experiments on simple problems, i.e. Catch, Four Rooms (tabular), and Cart-Pole (non-tabular), showing that the VE models achieve better planning performance than their MLE counterparts.

In a later work, Grimm, Barreto, Farquhar, *et al.* [4] generalise the notion of VE to order-k counterparts, defined wrt k applications of the Bellman operator. Proper value equivalence (PVE) is the limit of k-VE, as $k \to \infty$. As in the limit, after infinitely many applications of the Bellman operator, all functions become value functions, the PVE equivalence class does not require a set of functions to be specified. Importantly, Grimm, Barreto, Farquhar, *et al.* also show PVE models are sufficient for optimal planning. The work also details a series of experiments on a stochastic version of Four Rooms,

in particular, showing that performance improves with increasing model capacity. The work omits any theoretical or experimental comparison to MLE, or other reconstruction-based losses.

Grimm, Barreto, and Singh [11] expand previous theoretic results on VE to the approximate setting [11]. Their experiments in the Four Rooms domain explore the effect of the minimum tolerated estimation error ϵ (which tends to increase with the number of functions in the VE specification), and model capacity. The results show that good performance is achieved when, first, enough value functions are used in fitting the model (which relates to generalisation ability), and model capacity is large enough.

In none of the three works by Grimm is value-equivalent models' generalisation ability in value prediction directly evaluated. This is, however, an interesting niche to explore, as, for instance, MuZero, which is PVE wrt its behaviour policy, has been shown to perform poorly in value prediction of unseen policies different from its behaviour policy [6].

Value-equivalent models can be seen as Q^π-irrelevance abstractions (wrt a set of policies Π) within the unified theory of state abstractions for MDPs proposed by Li, Walsh, and Littman [16] [6]. In other words, a value-equivalence model may internally represent some states in a way such that they will have the same action-value function for all actions and policies. It is then reasonable to suppose generalisation can be achieved when such a model identifies symmetries within the environment wrt the value.

There exists a complementary framework to VE based on the same premise of models that are fit to be accurate wrt value, *value-aware model learning* (VAML) [13], [17], [18]. VAML focuses on the optimization problems induced by the premise, while VE is concerned with model classes. In a similar vein, *policy-aware model learning methods* [9] apply the principle of value-awareness (or more broadly decision-awareness) to policy gradient methods, as VAML originally has focused only on value-based methods. However, these and similar works [19] focus on measuring performance, and not directly the generalisation error.

Some notable empirical successes in MBRL may be conceptualised as applications of the VE principle, including the Predictron [20], Value Iteration Networks [21], Value Prediction Networks [22], MuZero [5], and others [2], [23]. In particular, minimizing MuZero's loss minimizes a squared PVE loss with respect to a single policy – the current behaviour policy [4]. The ability of MuZero in evaluating unseen policies has been previously studied, demonstrating that MuZero fails predict values accurately for unseen or unfamiliar policies [6].

The generalisation ability of value-equivalent models may be a valuable area of study, since it relates to the potential use of the model in policy exploration, and may possibly provide insight to failures of value-equivalent or value-aware methods in some applications [24].

4 Experimental Setup

In Section 1 we have posed the following research questions:

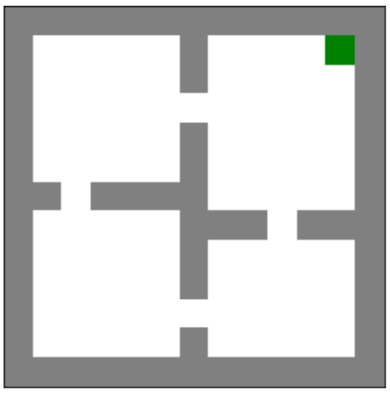

Figure 1: The Four Rooms environment. Source: [4]

1. How does the number of policies included in PVE loss affect the generalisation ability of the resulting model?
2. How do MLE-based and PVE-based models trained on the same data compare in test set value prediction error, as (a) model size varies, (b) training dataset size varies?

To answer both questions, we analyse the collection of models obtained as described further in this section. In brief, we have trained 2 models (MLE and PVE) in each experimental run, varying model rank and training policy set size, for 10 instances of random seeds (in total $8 \cdot 7 \cdot 2 \cdot 10 = 1120$ models).

We measure generalisation ability by the average absolute policy evaluation error (also further referred to as prediction error) across all states with respect to the test policy set.

Policies as data In order to evaluate the models' generalisation ability in value prediction, we sample a subset of distinct deterministic policies $\Pi \in \mathbb{\Pi}_{\text{det}}$, and split it into two equally-sized sets Π_{train} and Π_{test}, one to train the models, the other to test their value prediction error (on policies unseen in training). We decided to focus on deterministic policies, since they are easier to analyse, yet still sufficient to obtain a PVE model [4].

Environment The experiments were performed in a stochastic version of the Four Rooms environment [25] (see Figure 1). The size of the state space is $|\mathcal{S}| = 104$, corresponding to the number of free tiles without walls. The action space $\mathcal{A}$ consists of four actions corresponding to the cardinal directions. As the agent takes an action, it will move in the intended direction with probability .8, and otherwise move in a random direction. If the agent attempts to move into a wall, it will remain in place. Moving to the upper-right tile yields a reward of 1, and all other moves result in a reward of 0. This environment was selected in order to compare our results to prior work.

Model representation and initialisation Models are represented tabularly as two components: a matrix $\tilde{R} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$, and a tensor $\tilde{P} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|\times|\mathcal{S}|}$, which can also be thought of as a set of matrices $\tilde{P}^a \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ for $a \in \mathcal{A}$. Then $\tilde{R}_{s,a} = \tilde{r}(s,a)$ and $\tilde{P}^a_{s,s'} = \tilde{p}(s'|s,a)$.

To compare PVE and MLE in settings with various degrees of limited model capacity, we ensure that each $\tilde{P}^a$ has a rank of at most k by factoring as follows: $\tilde{P}^a = D^a K^a$ where $D^a \in \mathbb{R}^{|\mathcal{S}| \times k}$, $K^a \in \mathbb{R}^{k \times |\mathcal{S}|}$, and D^a, K^a are both constrained to be *row-stochastic*.

Note, a matrix is row-stochastic if its rows sum to 1. A matrix can be constrained to be row-stochastic by parameterising it by a matrix of the same shape, and applying softmax on the rows. Note that the product of row-stochastic matrices is also row-stochastic. Thus each $\tilde{P}^a$ is parameterised by the two unconstrained matrices that parameterise D^a and K^a. In model initialisation, the entries of these two parameterising matrices are sampled from a uniform distribution. The rewards' model $\tilde{R}$ is also initialised from a uniform distribution, as it is unconstrained.

Loss functions The PVE-loss based model is fit on the loss given in (6), with $k = 1$ and the L^2 norm, on policies π from Π_{train} and their exact values v_π. The MLE-loss based model is fit on the following expected loss wrt a set of policies (compare with [13], [26]):

$$\ell_\Pi = \sum_{\pi \in \Pi} \left[\mathop{\mathbb{E}}_{\substack{(S,A) \sim \mathcal{D}_\pi \\ S' \sim p(\cdot|S,A)}} \left[-\log \tilde{P}^A_{S,S'} \right] \right] \tag{7}$$

where $\mathcal{D}_\pi$ is a distribution derived from policy π. For the derivation of the loss, refer to Appendix A. We use the stationary distribution derived from the Markov chain induced by the true environment dynamics P and policy π, however, other choices for the distribution are possible in principle. The MLE loss is expected, not empirical. This decision was made to make the MLE loss comparable to the PVE loss, by eliminating statistical sampling error, which would result from fitting on episodes sampled from policies $\pi \in \Pi_{\text{train}}$.

In the MLE model, R is provided to the agent directly, without learning. This simplification is motivated by the fact that with a distribution that visits all transitions with some non-zero probability, in a tabular setting one can learn the reward $r(s, a)$ by keeping an average of the rewards after taking action a in state s.

Training details All models were fit with mini-batch gradient descent (batch size of 50) using the Adam optimizer with default hyperparameters $\beta_1 = 0.99$, $\beta_2 = 0.999$, $\epsilon =$1e-8, and a learning rate of 5e-4. We apply gradient descent as no analytical solution was found to exist. Note, we do not control for overfitting, as we want the models to be fit as best as possible to the loss objectives, so that they can be compared at their minima. We train each model for 10,000 epochs, as we have observed this was sufficient for convergence in all scenarios.

Range of settings To assess how MLE and PVE compare in different settings, we train the both, varying two factors: number of distinct policies included in training, ranging over values $\{4, 8, 16, 32, 64, 128, 256\}$, and model capacity (constrained by rank, as described above), ranging over $\{10, 30, 50, 70, 90, 104, 130, 150\}$. Note, we have placed 104 instead of 110, as this is exactly the state space size. Also note that the sequence of policy set sizes was chosen to be exponentially increasing, to study the models at different magnitudes of training data size. Each of these 56 combinations was tested ten times, with seeds ranging from 0 to 9.

Evaluation Finally, we evaluate and compare the models by the average absolute value prediction error $\|v_\pi - \tilde{v}_\pi\|$ for policies in the test set Π_{test}, as a measurement of their generalisation ability.

Discussion on alternative methodology choices We could have used the empirical versions of the loss functions instead of the expected forms, however we have decided to focus on comparing the behaviour of the losses at their simplest, without accounting for the factor of statistical error, which we leave for future work.

We have chosen to study tabular environments because it allows us to exactly compute the value functions of policies in closed form.

The central method of our study is experimental, which limits the extent of the conclusions we can make, as we might expect that in another, possibly more complex, environment results would be unlike ours.

5 Results

5.1 Effect of training set size

To study the effect of the training policy set size on the PVE models' generalisation ability, we fix model rank to 104 (also the size of the state space). These results are illustrated by Figure 2. For other settings of model rank the trends are similar to those in the case we have selected (see Figure 3, heatmap on the top right).

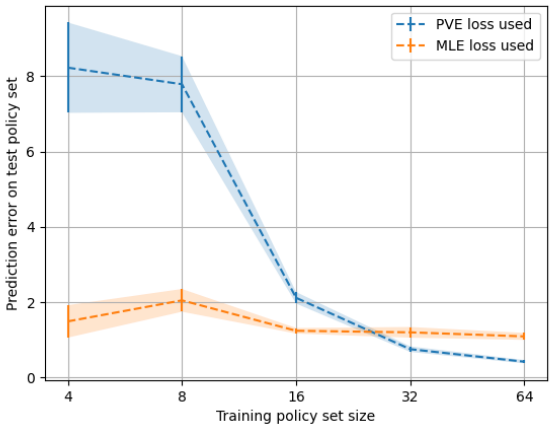

Figure 2: Effect of the number of policies used to train the model on the model's value prediction error (on the test set of policies), when either MLE or PVE loss is used. Model capacity is here unconstrained ($k = |\mathcal{S}| = 104$). Note the x-axis is in log-scale. Also note that in this, and the following figures, the error bars represent the standard error of the mean (SEM), across the ten seed instances. We see that, as compared to MLE, performance of PVE loss based model is more affected by the training set size. In both cases, more policies in the test set lead to better generalisation ability.

As expected, we find that the value prediction error on the hold-out test policy set decreases with increasing the size of the training policy set. On Figure 2, we see a sharp drop as the size increases from 8 to 16, and then a slowing decrease, implying diminishing returns from adding new policies to the training set. If we compare the trends in PVE and MLE generalisation performance for reference, we see that the MLE models' error in stays approximately on the same level for all training set sizes. The explanation for this may be that once enough policies are included in training, the MLE model is fit to a transitions' distribution with sufficient coverage over all possible state-action pairs to estimate the next states and rewards as accurately as MLE loss can allow. On the other hand, in the case of PVE, including a new random policy (along with its true value function) in the loss makes the model evaluate an unseen policy significantly better than before. We observe that 4–8 policies are not enough for PVE to reach performance of MLE, yet 32–64 policies are – we may explain this by the fact PVE loss takes policies' value into account, while MLE does not, hence PVE achieves an higher test accuracy in policy evaluation.

Our observations confirm our intuitive expectations, as we have anticipated that with more policies' value functions included in the model, it becomes able to generalise to a broader set of policies.

5.2 Comparison of PVE and MLE in different settings

In Figure 3 we see PVE gives extremely low error on the training set, and high error on the test set. This may indicate the model is overfit to its set of policies, as compared to MLE, which has comparable levels of error on both training and test sets. Along the x-axis, we observe that test error generally decreases as model capacity increases.

Figure 4 contains a compact comparison of MLE versus PVE, across varied settings of training set size and model capacity. When few policies are used in training, PVE models are markedly worse, though their performance improves with training set size, and at size 64 and onwards, PVE outperforms MLE. The disadvantage of PVE is the greatest in the low-data low-capacity setting ($n = 10, k = 8$). We can also observe that with increase in the model rank, the magnitude of the difference decreases – in other words, generalisation performance of MLE and PVE tends to equalise as model capacity grows larger.

Firstly, we have hypothesized that in a setting with few training policies and large model capacity, MLE-based models would generalize better than PVE. This has proven partially correct, since though MLE models generalise better with few training policies, they in fact show a higher advantage over PVE when model representational capacity is large, not low. The effect of training policy set size has been discussed in the previous subsection (5.1). Wrt model capacity, we may hypothesise that when the PVE model capacity is all too low, the effect of insufficient policies is exacerbated, whereas MLE is able to maintain good generalisation ability via keeping a sufficient estimate of transition probabilities, even though given a limited range in data due to the small number of deterministic policies used.

Secondly, we have hypothesized that PVE would generalise better in a setting with many training policies and small model capacity. This hypothesis holds generally true, though model capacity has a less pronounced effect on generalisation ability than the number of policies used in training. The results seem reasonable, if we see the amount of training data as the main driving factor behind generalisation, and model capacity being a factor limiting model's expressive ability, and hence generalisation.

6 Discussion

6.1 Relation to known results

We have demonstrated that, at least in some tabular settings, there exists a boundary that divides contexts where MLE is preferable to PVE in policy evaluation generalisation ability, and vice versa.

Our results are consistent with previous work on the subject. For instance, it has been previously shown that planning performance (here the performance of the model-optimal policy in the environment) of MLE models improves with increasing model rank [4]. This coincides with our observations, as model rank positively affects the ability to evaluate unseen policies, which is helpful in planning, since in planning with policy iteration the model evaluates multiple models until finding the optimal one. A non-generalisable model will evaluate unseen policies incorrectly, leading to a worse estimate of the optimal policy.

However, we also see that superiority in planning performance may not always correspond to superiority in generalisation ability. Prior work [3] has, similarly to ours, compared MLE and 1-VE (which is a smaller class than PVE) models on Four Rooms in a series of bivariate experiments, varying model-rank and number of policies, with notable differences: (1) measuring planning performance, (2) fitting an empirical loss from offline sample transitions. Their results show PVE outperforms MLE across all scenarios, especially with lower model ranks, illustrating "value equivalence principle can yield a better allocation of the limited resources of model-based agents" [3]. In contrast, we show MLE outperforms PVE in generalisation with small policy set sizes, and especially with low model ranks.

6.2 Limitations

We can identify multiple limitations in the experimental setup. The experiments were made in a simple tabular grid-world setting, without function approximation. Results may be different in a more complex setting with function approximation. In training the MLE model we used distributions which, while valid, may not coincide with the normalized occupancy measure for the MDP [27]. Note that we have deliberately decided to remove the factor of statistical error when comparing the losses, and hence used expected non-empirical formulations.

7 Conclusion and future work

In this work we have shown experimentally in a tabular, grid-world setting, that (1) number of policies used in training a

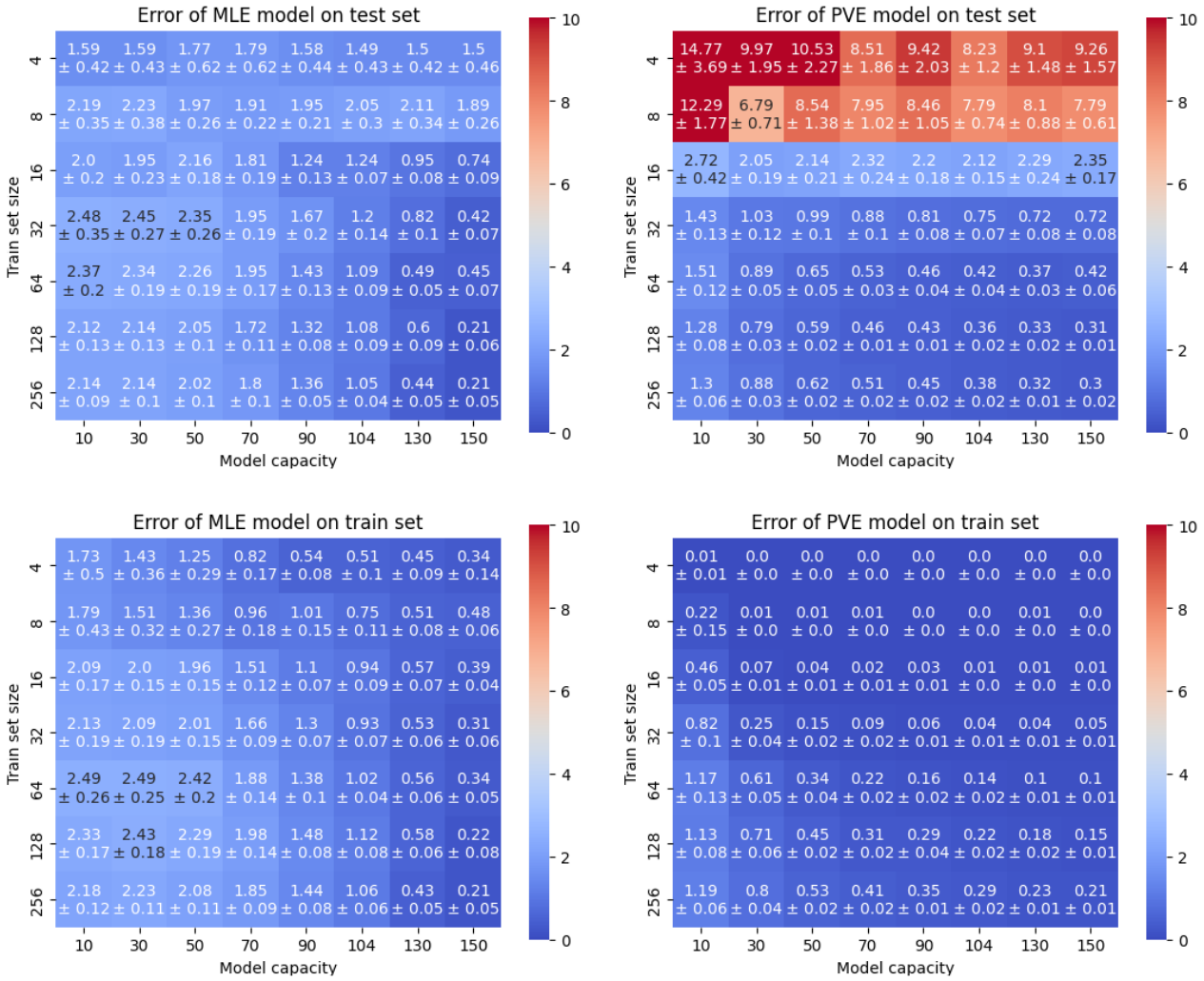

Figure 3: Value prediction errors for PVE and MLE models, on training and test policy sets. We have varied the policy set size used in training (y-axis of the heatmaps), and the model capacity, which is the rank of the tabular model parameters (x-axis of the heatmaps). Note, the color scale begins at 0 (dark blue for the lowest error), and is capped at 10 (dark red for the highest error). We can see the range of MLE models is more moderate than that of PVE in both the training and test sets. PVE models show a high test set error when trained on few policies. The training set error is extremely low for PVE, implying overfitting. In general, errors tend to decrease with model capacity, but not necessarily with training policy set size (except in PVE models' prediction error on the test set).

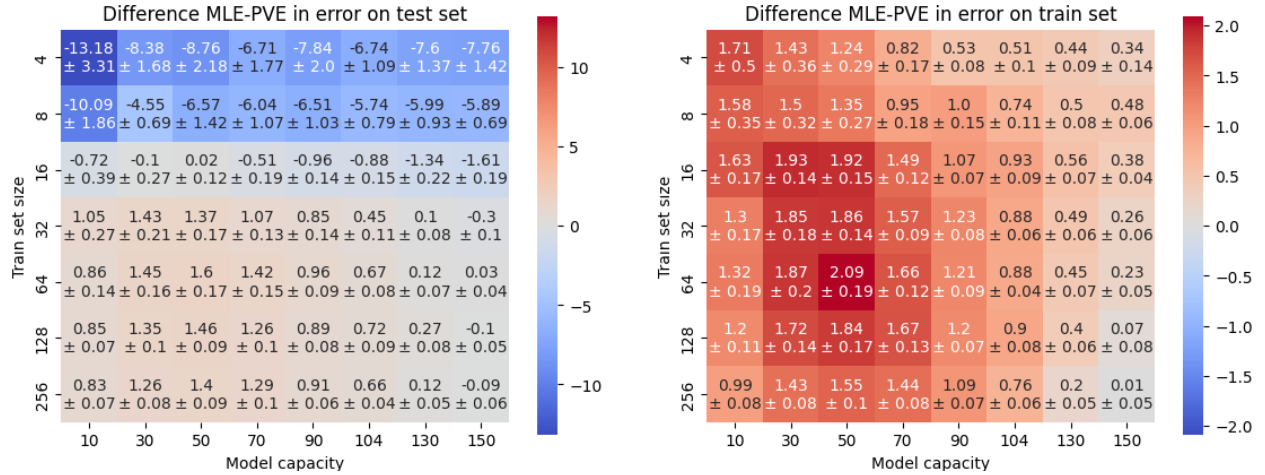

Figure 4: Difference between MLE and PVE value prediction errors, for varied settings of training policy set size, and model capacity. The visualised matrices were produced by taking the element-wise difference between the corresponding matrices on the right-hand side and left-hand side of Figure 3. Shades of red indicate error of PVE is better, and blue – that that of MLE is better. We observe that although PVE behaves significantly worse when trained on few policies, it improves as the training set grows, outperforming MLE in settings with larger sizes.

PVE loss based model will improve its ability to evaluate unseen policies, and (2) models trained on MLE loss generalise better than those trained with PVE loss when few policies are used in training, and the opposite is the case when many policies are used.

There exist multiple potential directions for future work. For instance, it would be interesting to formally show the existence of the boundary identified in Section 6, and study its properties. One could also compare MLE and PVE in a more realistic setting, with a large state and action space requiring function approximation, since in a non-tabular setting, there will always be some generalisation error between the model and the environment [26]. Additionally, this work has not examined how well PVE models generalise to policies similar to those seen in training, for instance wrt policies from the same trajectory of policy iteration.

8 Responsible research

This work is concerned with specific theoretic questions in machine learning, and as thus we believe the work does not possess any tangible potential for negative societal impact. The limitations and scope of the work are discussed in Subsection 6.2.

This work includes instructions sufficient to reproduce the main experimental results, see Section 4. The code repository is publicly available at https://github.com/severinbratus/compare_pve_mle. It is an extension of a repository by Christopher Grimm [4], and uses several of their functions. In particular, we have used their implementation of the grid-world environment, and the PVE models' training. Other logic, e.g. MLE model training, was implemented by the author. The original repository is available at https://github.com/chrisgrimm/proper_value_equivalence.

All experimental data was generated pseudo-randomly, with multiple seeds documented in the paper for reproducibility. Thus the FAIR Data Principles [28] do not directly apply, as the input data to the training were not persistently stored, but rather (re)generated at run-time – the data is however perfectly reproducible. Error bars were reported to truthfully represent the uncertainty observed in our experimental data. Training details (data splits, hyperparameters) are specified in the codebase. All experiments were performed on the Delft-Blue Supercomputer [29], with approx. 560 jobs in total, approx. 30 minutes each, on Intel Xeon E5-6448Y Processors.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT press, 2018.

[2] T. M. Moerland, J. Broekens, A. Plaat, C. M. Jonker, *et al.*, "Model-based reinforcement learning: A survey," *Foundations and Trends® in Machine Learning*, vol. 16, no. 1, pp. 1–118, 2023.

[3] C. Grimm, A. Barreto, S. Singh, and D. Silver, "The value equivalence principle for model-based reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5541–5552, 2020.

[4] C. Grimm, A. Barreto, G. Farquhar, D. Silver, and S. Singh, "Proper value equivalence," *Advances in Neural Information Processing Systems*, vol. 34, pp. 7773–7786, 2021.

[5] J. Schrittwieser, I. Antonoglou, T. Hubert, *et al.*, "Mastering atari, go, chess and shogi by planning with a learned model," *Nature*, vol. 588, no. 7839, pp. 604–609, 2020.

[6] J. He, T. M. Moerland, and F. A. Oliehoek, *What model does MuZero learn?* Oct. 2023. DOI: 10.48550/arXiv.2306.00840. arXiv: 2306.00840 [cs]. (visited on 05/03/2024).

[7] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Aug. 2014, ISBN: 978-1-118-62587-3.

[8] M. Arjovsky, S. Chintala, and L. Bottou, *Wasserstein GAN*, Dec. 2017. DOI: 10.48550/arXiv.1701.07875. arXiv: 1701.07875 [cs, stat]. (visited on 06/16/2024).

[9] R. Abachi, M. Ghavamzadeh, and A.-M. Farahmand, *Policy-Aware Model Learning for Policy Gradient Methods*, Jan. 2021. DOI: 10.48550/arXiv.2003.00030. arXiv: 2003.00030 [cs]. (visited on 05/03/2024).

[10] J. Joseph, A. Geramifard, J. W. Roberts, J. P. How, and N. Roy, "Reinforcement learning with misspecified model classes," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 939–946. DOI: 10.1109/ICRA.2013.6630686. (visited on 06/16/2024).

[11] C. Grimm, A. Barreto, and S. Singh, "Approximate value equivalence," *Advances in Neural Information Processing Systems*, vol. 35, pp. 33 029–33 040, 2022.

[12] N. Lambert, B. Amos, O. Yadan, and R. Calandra, *Objective Mismatch in Model-based Reinforcement Learning*, Apr. 2021. DOI: 10.48550/arXiv.2002.04523. arXiv: 2002.04523 [cs, stat]. (visited on 06/14/2024).

[13] A.-M. Farahmand, A. Barreto, and D. Nikovski, "Value-Aware Loss Function for Model-based Reinforcement Learning," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, PMLR, Apr. 2017, pp. 1486–1494. (visited on 05/03/2024).

[14] B. Eysenbach, A. Khazatsky, S. Levine, and R. R. Salakhutdinov, "Mismatched No More: Joint Model-Policy Optimization for Model-Based RL," *Advances in Neural Information Processing Systems*, vol. 35, pp. 23 230–23 243, Dec. 2022. (visited on 06/18/2024).

[15] E. Talvitie, "Model Regularization for Stable Sample Rollouts," in *Conference on Uncertainty in Artificial Intelligence*, Jul. 2014. (visited on 06/19/2024).

[16] L. Li, T. Walsh, and M. Littman, "Towards a Unified Theory of State Abstraction for MDPs.," in *Proceedings of the Ninth International Symposium on Artificial Intelligence and Mathematics*, Jan. 2006.

[17] A.-M. Farahmand, A. M. Barreto, and D. N. Nikovski, "Value-aware loss function for model learning in reinforcement learning," in *13th European Workshop on Reinforcement Learning (EWRL)*, vol. 1, 2016, p. 36.

[18] A.-M. Farahmand, "Iterative Value-Aware Model Learning," in *Advances in Neural Information Processing Systems*, vol. 31, Curran Associates, Inc., 2018. (visited on 05/03/2024).

[19] C. Voelcker, V. Liao, A. Garg, and A.-M. Farahmand, *Value Gradient weighted Model-Based Reinforcement Learning*, Jun. 2023. DOI: 10.48550/arXiv.2204.01464. arXiv: 2204.01464 [cs]. (visited on 06/14/2024).

[20] D. Silver, H. Hasselt, M. Hessel, *et al.*, "The Predictron: End-To-End Learning and Planning," in *Proceedings of the 34th International Conference on Machine Learning*, PMLR, Jul. 2017, pp. 3191–3199. (visited on 05/03/2024).

[21] A. Tamar, Y. Wu, G. Thomas, S. Levine, and P. Abbeel, "Value Iteration Networks," in *Advances in Neural Information Processing Systems*, vol. 29, Curran Associates, Inc., 2016. (visited on 05/03/2024).

[22] J. Oh, S. Singh, and H. Lee, "Value Prediction Network," in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017. (visited on 05/03/2024).

[23] A. Plaat, W. Kosters, and M. Preuss, *High-Accuracy Model-Based Reinforcement Learning, a Survey*, Jul. 2021. DOI: 10.48550/arXiv.2107.08241. arXiv: 2107.08241 [cs]. (visited on 06/14/2024).

[24] Â. G. Lovatto, T. P. Bueno, D. D. Mauá, and L. N. Barros, "Decision-Aware Model Learning for Actor-Critic Methods: When Theory Does Not Meet Practice," PMLR, Feb. 2020, pp. 76–86. (visited on 06/20/2024).

[25] R. S. Sutton, D. Precup, and S. Singh, "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning," *Artificial Intelligence*, vol. 112, no. 1, pp. 181–211, Aug. 1999, ISSN: 0004-3702. DOI: 10.1016/S0004-3702(99)00052-1. (visited on 06/19/2024).

[26] F.-M. Luo, T. Xu, H. Lai, X.-H. Chen, W. Zhang, and Y. Yu, *A Survey on Model-based Reinforcement Learning*, Jun. 2022. DOI: 10.48550/arXiv.2206.09328. arXiv: 2206.09328 [cs]. (visited on 06/14/2024).

[27] R. Jiang, J. Tavakoli, and Y. Zhao, *Hitting time for Markov decision process*, May 2022. arXiv: 2205.03476 [cs]. (visited on 05/27/2024).

[28] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, *et al.*, "The FAIR Guiding Principles for scientific data management and stewardship," *Scientific Data*, vol. 3, p. 160018, Mar. 2016, ISSN: 2052-4463. DOI: 10.1038/sdata.2016.18. (visited on 06/23/2024).

[29] Delft High Performance Computing Centre (DHPC), *DelftBlue Supercomputer (Phase 2)*, 2024.

Appendix

A Derivation of a MLE loss wrt a set of policies

Here we show the derivation of the MLE loss used in experiments (7) from its KL-divergence formulation (4).

$$\begin{aligned}
\ell_{p,\mathcal{D}}(p,\tilde{p}) &= \mathop{\mathbb{E}}_{(S,A)\sim\mathcal{D}}\left[\mathrm{D}_{\mathrm{KL}}(p(\cdot|S,A)\,||\,\tilde{p}(\cdot|S,A))\right] \\
&= \mathop{\mathbb{E}}_{(S,A)\sim\mathcal{D}}\left[\sum_{s'} p(s'|S,A)\log\frac{p(s'|S,A)}{\tilde{p}(s'|S,A)}\right] \\
&= \mathop{\mathbb{E}}_{(S,A)\sim\mathcal{D}}\left[\mathop{\mathbb{E}}_{S'\sim p(\cdot|S,A)}\left[\log\frac{p(s'|S,A)}{\tilde{p}(s'|S,A)}\right]\right] \\
&= \mathop{\mathbb{E}}_{\substack{(S,A)\sim\mathcal{D}_\pi \\ S'\sim p(\cdot|S,A)}}\Big[\log p(s'|S,A) - \log\tilde{p}(s'|S,A)\Big]
\end{aligned}$$

As the first term in the expectation $p(s'|S,A)$ is not dependent on the model being optimised, we may omit it from the loss. Then, if we sum of this quantity over some set of policies Π, adjusting $\mathcal{D}$ to be a policy-specific distribution $\mathcal{D}_\pi$, we obtain:

$$\begin{aligned}
\ell_\Pi &= \sum_{\pi\in\Pi}\left[\mathop{\mathbb{E}}_{\substack{(S,A)\sim\mathcal{D}_\pi \\ S'\sim p(\cdot|S,A)}}\Big[-\log\tilde{p}(S'|S,A)\Big]\right] \\
&= \sum_{\pi\in\Pi}\left[\mathop{\mathbb{E}}_{\substack{(S,A)\sim\mathcal{D}_\pi \\ S'\sim p(\cdot|S,A)}}\Big[-\log\tilde{P}^A_{S,S'}\Big]\right]
\end{aligned}$$