Dimensionless Bayesian Model-Based Reinforcement Learning

Valentin Charvet

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Computing Science
College of Science and Engineering
University of Glasgow

Abstract

This work explores an approach for improving the robustness of Model-Based Reinforcement Learning algorithms by transforming the observation and decision spaces with the Buckingham-Π theorem. This theorem is part of the field of Dimensional Analysis (DA) which studies the link between physical measurements and the units they are expressed in. The Buckingham-Π theorem provides a dimensionality reduction technique through a power law between the variables. The transformation can be applied on inputs and outputs of statistical learning models to increase their robustness. We extend prior work to study the impact of that procedure, called non-dimensionalization, through its equivariance properties on stationary dynamic systems. Our method stems from increasing the level of a priori physics knowledge within the Machine Learning models. That additional knowledge is brought implicitly by the constraints implied by the non-dimensionalization procedure into Machine Learning models. The results in this thesis suggest this approach is well suited for zero-shot transfer learning without data augmentation.

Throughout this thesis, we conduct the experiments on pendulum and cartpole environments within numerical simulations. First, we propose a framework for applying the Buckingham theorem to dynamic systems. We showed that under a full-rank assumption, we can transform the state variables as a function of the static variables. This transformation in turn yields estimators that are resilient to perturbations of the underlying dynamics. We included comparisons between Gaussian Process and Multi-Layer Perceptron for the regression task. The estimators are able to make maintain good predictive performance in the presence of distribution shift. Second, we propose a method to circumvent the need to measure all the variables for the transformation. With a probabilistic approach, we infer the hidden variables and constrain their dimensions.We expose two cases for this latent variables model, one that requires observations of the hidden variables during training and one that does not. Finally, we apply the previous findings to a Reinforcement Learning problem. To do so, we modify the *Contextual Markov Decision Process* (MDP) and non-dimensionalize the state and action spaces. Subsequently, we propose a generic model-based policy search algorithm within the dimensionless Π-MDP and demonstrate results with Gaussian Process dynamics models. We showed that within the evaluated environments, the dimensionless controller is more robust than its natural counterpart.

We showed the benefits of the transformation for generalizing predictions under distribution shift. The simplicity of the approach allows it to be applied to different domains such as regression and sequential decision-making. Our experiments suggest the Buckingham transformation is a promising avenue for statistical modelling under distribution shift.

Declaration

I declare that, except where explicit reference is made to the contribution of others, that this dissertation is the result of my own work and has not been submitted for any other degree at the University of Glasgow or any other institution

Acknowledgements

First, I would like to thank my supervisors Rod Murray-Smith and Sebastian Stein for giving me the change to start this degree and helping me during during all this time. I am also very thankful to Bjørn Sand Jensen who suported me during the first two years and introduced me to the marvels of probabilistic models and Gaussian Processes. This whole thesis could not have been made possible without you three.

Thank you to everyone in the department: Aurélien, Marco, Carlos, Rory, Chaitanya, Songpei, Anders, Josh, Jonathan. All of our discussions and your feedbacks have been so helpful to make this thesis a reality, and the journey so pleasant.

Finally, to my family and friends in France I would like to thank you for your support during these difficults years.

Contents

List of Figures

Notation

We let scalar variables be written by italic lowercase letters as $x \in \mathbb{R}$. Vectors are denoted with bold lowercase and indexed as $\boldsymbol{x} = [x_1, \dots, x_d]^T \in \mathbb{R}^d$. Matrices are bold lowercase letters $\boldsymbol{A} \in \mathbb{R}^{n \times d}$. The identity matrix with 1 on the diagonal and 0 everywhere else is writen $\boldsymbol{I}$.

Multivariate normal distribution with mean vector $\boldsymbol{m}$ and covariance matrix $\boldsymbol{K}$ is written as $\mathcal{N}(\boldsymbol{m}, \boldsymbol{K})$ The joint probability of X and Y and conditional probability of X given Y are $p(X, Y)$ and $p(X|Y)$ respectively. A Gaussian Process with mean function μ and covariance function k is written as $\mathcal{GP}(\mu(\cdot), k(\cdot, \cdot))$. D_{KL} is the symbol for the Kullback-Leibler divergence between two probability distributions.

In probabilistic graphical models, we write observed random variables with white circles, and hidden ones with grey circles as on figure 1.

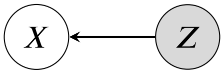


Figure 1: Graphical model

Chapter 1.

Introduction

In science, statistic inference is often used to make predictions about the outcome of experiments or the future state of a system. To do so, scientists rely on models that are built on top of a set of hypotheses and assumptions. Such models usually support reasoning outside the scope of observations. For instance, the laws of classical mechanics may be deduced from the falling movement of an apple and applied to predict the movement of planets around the sun. This capacity for counterfactual induction is what makes causal models so precious for understanding the world around us.

On the other hand, *Machine Learning* is the process of constructing a model from data alone. This aspect guarantees such models to be flexible and do not require extensive a priori domain knowledge to function. By means of minimizing a measure of empirical risk on a set of observations, ML models are able to deduce the latent mechanism that generated the data. They can then later be used to make predictions or generate new realistic observations. However, because such models are constructed from data alone, they can only represent the system accurately within the limits of what they have been exposed to during training. As a consequence, such models generalize poorly when the experimental conditions of the system change. This phenomenon, called *Distribution Shift* denotes the capacity of a system to generate different data under different conditions [Quiñonero-Candela et al., 2008]. The specificity of statistical models to represent only their training data is well known and the reason why cross-validation is used so often in predictive models to make sure they do not overfit. A model is called *robust* if it is able to maintain accurate predictions on a system even when it is subjected to distribution shift. In other words, robustness denotes the resilience of the model to perturbations of the systems it is deployed in. In order to increase model robustness, augmenting the size and diversity of the training data is a popular solution, especially in the current era where huge datasets are readily available. Acquiring such large databases can nevertheless be expensive because of the absence of appropriate sensors to the measures variables of interest as well as the cost of remote and local storage.

An orthogonal direction to data augmentation is the modification of the predictive model itself. While ML algorithms are data-driven, they are also constituted of assumptions and inductive biases. Those task-dependent inductive biases rely on expert knowledge and are a key component for increasing generalization [Mitchell, 1980]. For instance, Convolution Neural Networks assume translation invariance, meaning the detection of an object should not depend on where this object is located. Such additional hypotheses allow statistical models to be more data-efficient and less prone to overfitting. In general, the stronger the hypothesis, the better the generalization of the model [Botev et al., 2021]. This thesis investigates how to bring together both data and hypotheses in ML pipelines to increase their generalization capabilities while maintaining their flexibility.

Reinforcement Learning is a paradigm for solving sequential decision-making problems by trial-and-error. It emerged as a subfield of computer science to solve tasks with little to no prior knowledge of the system they interact with [Richard S. Sutton, 2018]. To do so, the learning agent interacts in a dynamic environment and receives a reward signal after each action that indicates how close it is to the solution. Using this signal, the agent balances the exploration of potentially high-gain actions with the exploitation of its knowledge so far. However, myopic rewards are not sufficient because they do not carry any information about the future. One must instead consider how the agent performs along all the duration of its deployment and offset the short-term with long-term benefits of its decisions. During its deployment, the environment in which the agent evolves changes continuously meaning it should be equipped with sensors in order to perceive its current state. Overall, the learning process is slow and requires a large amount of interaction time before converging to an optimal solution. Within Reinforcement Learning, robustness to distribution shift is a crucial aspect for the development of such algorithms in real-world applications [Dulac-Arnold et al., 2021a; Zhao et al., 2020]. In this work, we focus our investigation on Model-Based algorithms.

Statistical predictions made from data usually suffer from *model bias*, meaning they can only be as good as the model itself and by extension its training data. *Bayesian statistics* propose a solution to approach this problem. In this framework, probabilities are interpreted as a degree of belief about random variables. Consequently, it can be used as a principled method for measuring the uncertainty associated with the prediction of a probabilistic model. A probabilistic model is constructed upon the combination of initial hypotheses in the form of a *prior distribution* with actual data called *likelihood*. Using Bayes' rule, these two distributions form the basis of a *posterior distribution* that represent the potential values of the outcome as well as their levels of confidence. Our work follows the footsteps of the application of Bayesian inference for solving sequential-decision problems with Reinforcement Learning [Ghavamzadeh et al., 2015, 2016].

Dimensional Analysis, while being crucial in physics and engineering, is often ignored within the ML and statistics communities. It consists of the analysis of the relationship between the measurement of physical quantities and their systems of units. In other words, measurements of physical quantities are made in specific units (meters, seconds etc...) and operations between them must respect consistency of the units system. It would not make sense to add together a speed with an electric voltage for instance. Moreover, two lengths expressed in feet and meters would need to be converted into the same unit for them to appear in the terms of an equation. Since 1914 when the Buckingham theorem was presented [Buckingham, 1914], we know that the knowledge of units within an equation may allow rewriting that equation with a reduced number of variables called Π-*groups*. This idea has been recently introduced in ML prediction tasks to demonstrate that models acting in Π-*groups* generalize better that their counterparts in natural feature spaces. This trait makes this theorem an efficient way to increase model robustness as it only requires knowing the measurement units of a system to transform the observations accordingly and obtain robust predictive models. This process of removing the units of a measurement to build equivariant features is called *non-dimensionalization*, and the Π-groups are called *dimensionless features*. Throughout this thesis, we will investigate the ability of such features to allow generalization in actuated dynamic systems.

1.1 Outline of this thesis

The research contributions of this thesis are contained in chapters 3, 4, 5.

- Chapter 2 is a review of the literature for the concepts we are working with throughout the thesis. We first explain why the problem of distribution shift is crucial for the deployment of Reinforcement Learning systems in the real world and how the problem can be formulated as a specific instance of a Partially Observed Markov Decision Process. Then, we review how the Buckingham theorem is used for dimensionality reduction in Machine Learning. Finally, we introduce the basics of probabilistic modelling with Bayesian inference and how those principles are used within the framework of Gaussian Process Regression.
- In chapter 3, we evaluate the robustness of the Buckingham-Π transformation to uncertainty. We propose a reformulation of the theorem for second-order systems with hidden static variables and demonstrate. We demonstrated empirically the generalization properties of the dimensionless models and their ability to cope with uncertain parameters..
- In chapter 4, we relax the need to observe all the variables required for constructing the dimensionless variables. We propose a model with dimensional latent variables where the physical constraint is imposed by the Buckingham theorem. We test the model in a few-shot learning setting on a simple pendulum and demonstrate its ability to adapt to new data.
- In chapter 5, we apply the findings of the previous chapters to a Model-Based Reinforcement Learning algorithm. We demonstrate empirically that controllers acting in the dimensionless state space are able to generalize far outside the bounds of the training distribution support.
- Last, we summarize our findings, discuss their limitations and the future investigation directions they open in chapter 6.

Chapter 2.

Background

2.1 Distribution Shift in Machine Learning

2.1.1 Distribution Shift in Supervised Learning

Supervised Learning [Hastie et al., 2009] is a framework for training algorithms to map a set of inputs $x_i \in \mathbb{R}^p$ to outputs y_i. In the remainder of this thesis we will only focus on regression, meaning the targets y are multidimensional real-valued vectors.

Given a set of pairs (x_i, y_i) samples from an unknown training distribution $\mathcal{D}_{\text{train}}$, the *learning process* consists in minimizing the discrepancy between the predicted and true targets. This is commonly done by choosing a function class optimizing a loss function with respect to the approximator parameters:

$$\boldsymbol{\theta}^* = \underset{x,y \in \mathcal{D}_{\text{train}}}{\arg\min} \mathcal{L} \ (\hat{y}(\boldsymbol{\theta}), y), \tag{2.1}$$

where $\hat{y}(\theta)$ is the prediction given by the regressor with parameters $\boldsymbol{\theta}$.

The *validation* or *test* measures the quality of the optimal parameters found during training on previously unseen samples. We call $\mathcal{L}_{test}(\theta^*)$ the loss function evaluated on the test data samples from distribution $\mathcal{D}_{\text{test}}$.

If the distance between the training and testing distribution is sufficiently small, that means the two distribution are similar. When the training is done correctly to prevent overfitting, the algorithm will be able to make as good predictions on the test set as the train set. This is the most classical setting, which we usually perform with cross-validation to make sure predictions are stable.

distance	generalization
$d(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}}) < \epsilon$	easy
$d(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}}) > \epsilon$	difficult

Table 2.1: Generalization difficulty given distance between sampling distributions.

If on the other hand, the training and testing distribution do not match, making accurate predictions on the test set is more challenging. This phenomenon is called *distribution shift* and is the source of great difficulty to deploy Machine-Learning systems in the real world. [Quiñonero-Candela et al., 2008] proposes a taxonomy of the different types of distributions shits that may arise depending on which random variables change between training and testing environments. The most common that they call *source component shift*, is caused by data generated from many different sources. In this specific case, suppose the relationship between x and y is expressed as $y = f(x, c)$. Source component shift happens when the hidden c changes between samples of (x, y), the joint distribution of the variables will be impacted even though the underlying relationship f remains the same. The source may be a hidden variable that is not part of the measurement process and acts as a confounder for both $\boldsymbol{X}$ and $\boldsymbol{y}$. We illustrate this type of drift on figure 2.1 where a context variable $\boldsymbol{C}$ changes between training and testing, causing a shift in the data distribution.

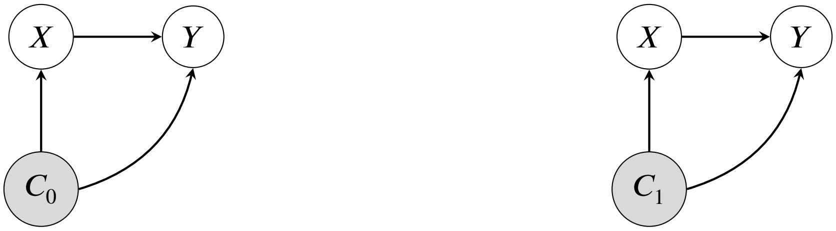


Figure 2.1: Graphical model illustrating source component shift. The random variables X and Y are both caused by a third hidden one $\boldsymbol{C}$. When C changes from training (left) to testing (right) environment, it causes a shift in the joint distribution of (X, Y).

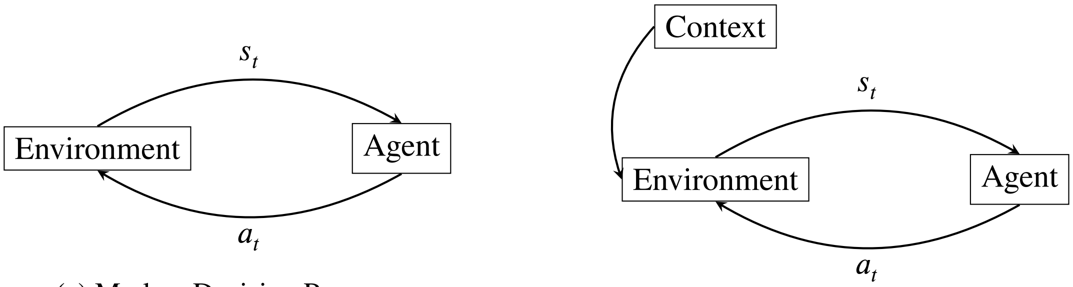
The ability of an estimator to make good prediction in the presence of distribution shift is called *generalization* [Arjovsky, 2020]. Alternatively, we may call *robustness* the resilience of the estimator to drift, meaning the quality of its predictions are little impacted by a change in the test distribution.

2.1.2 Distribution Shift in Reinforcement Learning

Markov Decision Process

Sequential decision-making problems are found in many scientific, industrial and economic fields. All the domain-specific settings share, however, common structures that can be theoretically represented by a *Markov Decision Process (MDP)* [Richard S. Sutton, 2018]. They represent an agent interacting with an environment that aims to solve a set of predefined tasks.

The interaction between an agent and its environment is illustrated from a high level on figure 2.2. At a given time step t the agent perceives the state of the environment s_t through its sensors



(a) Markov Decision Process.

(b) Contextual MDP

Figure 2.2: High level view of an agent interacting with its environment.

and sends an action (or control) signal a_t as a response. The procedure by which the agent selects actions is called a *policy*, a mapping we write $a_t \sim \pi(s_t)$. As a consequence of this signal, the environment transitions into a next state according to a transition kernel $s_{t+1} \sim f(s_t, a_t)$. This transition follows the *Markov Property*, meaning the future state at time $t+1$ solely depends on the state of the system at time t and not the past $t-1, ..., t_0$ and gives its name to MDPs. In addition to perceiving the state of the environment, the agent receives a reward signal $r_t \sim \mathcal{R}(s_t, a_t)$ informing it of the quality of the chosen action.

We can summarize this process as a tuple

$$\mathcal{M} = \left(\mathcal{S}, \mathcal{A}, f, \mathcal{R}, \rho_0\right), \tag{2.2}$$

with

- $\mathcal{S} \in \mathbb{R}^d$ is the *state space*.
- $\mathcal{A} \in \mathbb{R}^f$ is the continuous *action space*.
- f is the *transition kernel* or simply *transition function*.
- ρ_0 is a distribution of initial state (ie where the agents starts).
- $\mathcal{R}$, the reward function that we assume to be known. It is not a strong assumption, as the reward is often decided by the programmer or engineer and a distance between current and desired state-action vectors.

This formulation of the sequential decision-making process defines the long-term performance of the agent, thus turning the problem into an optimization one. We call *return* (2.3) the discounted sum of rewards

$$R(\boldsymbol{s}, \pi) = \mathop{\mathbb{E}}_{\substack{s \sim f \\ a \sim \pi}} \left[\sum_{t=0}^{T} \gamma^t r_t | \boldsymbol{s}_0 = s \right]. \tag{2.3}$$

From equation 2.3 stems an ordering over policies: a policy is better than another if it yields a higher return for all states s. An optimal policy is one that is better than any other and is given by

$$\pi^* = \arg\max_{\pi} \ \{R(\boldsymbol{s}, \pi), \forall s\} \,. \tag{2.4}$$

We only consider discrete time processes with finite horizon with $T < \infty$.

Finding the optimal policy is a NP-hard problem [Papadimitriou and Tsitsiklis, 1987]. Therefore, all the methods that tackle the MDP problem are merely concerned with *approximating a solution*, which is enough for most use cases. Classical methods for solving the MDP were first designed as early as the 1950s [Bellman and Kalaba, 1965] for stabilizing systems at a set of predefined equilibrium points. The advances in this field have consistently been driven by requirements for safety and robustness because of their deployment on critical systems such as aircraft and nuclear plants. In order to satisfy this, *Proportional Integral Derivative (PID)*, *Linear Quadratic Regulator (LQR)* and variations thereof, have been developed and successfully deployed in the real-world. This type of controller rely on analytical tractability to ensure safe deployment. It can however, only be achieved through strong assumptions such as linearity. In more recent years, data-driven methods based on Reinforcement Learning allow the relaxation of such hypotheses on the system, thus allowing solving complex tasks with little prior knowledge [Arulkumaran et al., 2017; Degrave et al., 2022].

Contextual Markov Decision Processes

The evaluation of controllers trained with RL is too often done in the same environment they have been trained on. While this consists a good test-bed for designing and comparing algorithms, it tends to oversimplify what would actually happen in the real world, where dynamics can be non-stationary [Dulac-Arnold et al., 2021a]. Physical wear-and-tear or hidden feedback loops [Sculley et al., 2015] can cause significant distribution shift which hinders the ability of a controller to stabilize the system at its equilibrium. Though it is not the only approach to illustrate this drift, we assume the dynamics of the MDP are subjected to a set of hidden variables that impact its one-step transitions. We follow the notations from [Kirk et al., 2023] and call this set of variables the *context*.

From this follows the definition of *Contextual Markov Decision Process (C-MDP)* [Hallak et al., 2015; Doshi-Velez and Konidaris, 2016; Ghosh et al., 2021], characterized by the following transition kernel

$$\boldsymbol{s}_{t+1} \sim f_{|c}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t; \boldsymbol{c}). \tag{2.5}$$

This new transition kernel yields a context-specific return function, which is the expected sum of rewards of the policy in the specific C-MDP. We can write it as

$$R(\pi, \boldsymbol{s}_0 f_{|\boldsymbol{c}}) = \mathop{\mathbb{E}}_{\substack{s \sim f_{|c} \\ \boldsymbol{a} \sim \pi}} \left[\sum_{t=0}^{T} \gamma^t r_t | \boldsymbol{s}_0 = s \right], \tag{2.6}$$

which we will often write $R(\pi, f_{|\boldsymbol{c}})$ where we omit the initial state for clarity. This value is a way to measure the quality of a policy π from a context to another.

Remark 1
C-MDPs can alternatively be viewed ad Partially Observed MDP with an emission function that constantly return the observed state $\mathcal{O}(\boldsymbol{s}_t, \boldsymbol{c}) = \boldsymbol{s}$. They are also in close connection with Latent MDPs [Kwon et al., 2021] where the context is sampled at random at the beginning of each episode.

Remark 2
What we call the context here, is a set of confounding variables that impact the dynamics of the dynamic systems. In that sense, they are similar as the variables described on figure 2.1 since any change in context will affect the next-state sampling distribution.

Because we assume that the context is slowly evolving, we assume in all the following analysis that it is sampled from an unknown distribution $p(c)$ at the beginning of an episode and remains static along its duration. The control objective in this setting can then be extended as

$$\max_{\pi} \left\{ \mathop{\mathbb{E}}_{c \sim p(\boldsymbol{c})} \left[R(\pi, f_{|\boldsymbol{c}}) \right] \right\}. \tag{2.7}$$

Similarly, as in supervised learning, we can define the *Generalization Gap* [Kirk et al., 2023] as the discrepancy between returns obtained in the training environment and the testing one,

$$\text{GenGap}\left(\pi, \boldsymbol{c}_{train}, \boldsymbol{c}_{test}\right) = R\left(\pi, f_{|\boldsymbol{c}_{train}}\right) - R\left(\pi, f_{|\boldsymbol{c}_{test}}\right). \tag{2.8}$$

This metric returns a scalar value, which is lower if a controller generalizes well. It can take negative values, in the case where the policy is not optimal in the training environment, but it is in the testing one. A robust controller will be able to achieve a low generalization gap for a wide set of testing context. This may, however, come at the cost of being overly conservative, meaning the controller will not be optimal even on the training environment. Trading-off optimal performance and robustness is at the core of robust RL research, as we will see in the next section.

2.1.3 Robust Reinforcement Learning Litterature

Robustness can be achieved by optimizing a *pessimistic objective*. This is often referred to as the *Robust Markov Decision Process* (MDP) framework [Wiesemann et al., 2013; Eysenbach and Levine, 2021], which can be solved by approximate dynamics programming [Mankowitz et al., 2018; Tamar et al., 2014] or within Maximum a Posteriori Policy Optimization [Mankowitz et al., 2019]. Such methods go as back to 2005 [Morimoto and Doya, 2005] where the authors apply an actor-critic where the controller attempts to correct for disturbances generated by an internal agent. More recently [Pinto et al., 2017] apply a similar method with neural networks. In essence, these methods solve a minimax optimization problem to account for worst-case scenarios. [Derman et al., 2020] defines an Uncertainty-Robust Bellman Equation and derive a robust TD error from it. This general framework was empirically verified in both discrete and high-dimensional continuous domains. Other types of methods inject noise in the policy or the model in order to prevent overfitting [Charvet et al., 2021; Igl et al., 2019]. These optimization procedures tend to yield controllers that are overly conservative, they generalize quite well at the cost of loosing optimality even on IID data.

Some other meta learning approaches rely on *domain randomization*. These consist in training from multiple version of the environment (ie several contexts) so as to disentangle local and global properties of the task [Sæmundsson et al., 2018; Kupcsik et al., 2013; Akkaya et al., 2019]. All of these approaches however require access to a white-box simulator, on which we can intervene to change its properties.

On the other hand, augmenting the set of initial hypotheses may increase the model and policy ability to learn and generalize with no additional data [van der Pol et al., 2021; Muglich et al., 2022]. Successes on zero-shot transfer have been increased with causal models [Kansky et al., 2017; Huang et al., 2023]. There are also recent works that studied the generalization problem but in the visual domains [Yang et al., 2023; Zhu et al., 2023]

The issue of distribution shift is also a concern for Offline RL [Levine et al., 2020]. In that specific setting however, it is not caused by non-stationarity but by the lack of training data in regions the offline-optimal policy visits. Several model-based methods propose to bypass it by means of regularization. MOReL and variants [Kidambi et al., 2020; Kim and Oh, 2023] construct a pessimistic MDP and uses a mechanism to detect unknown state-actions in order to split the space between regions of low and high uncertainty. MOPO [Yu et al., 2020] also optimize the policy in a surrogate MDP, where the reward in penalized by the model error. Both maximize a lower bound of the true objective. While both method are conceptually similar MOPO resorts to a softer penalty than MOReL. Other methods rely on Importance-Sampling schemes such as [Yuan et al., 2023; Hishinuma and Senda, 2021; Hong et al., 2023]

Like [Derman et al., 2020], we believe Bayesian models are well-fitted for the generalization task in RL. In the domain of classical methods, *Dual Control* [Unbehauen, 2000] maintains a probabilistic estimation of the plant parameters to derive robust adaptive controllers. This is due to the way Bayesian can reason about an infinite number of models with means of a distribution and integrate over all the possibilities weighted by how likely they are. In opposition, worst-case approaches only consider a subset of models that include the most pessimistic realizations.

2.2 Statistical Invariance

2.2.1 Invariance in Decision-Making

Group Theory

In order to provide a consistent definition of invariance, we recall some definitions from group theory.

Definition 2.2.1 (Group)
A group is an algebric structure $(G, \star)$ consisting of a set G and operation $\star$ that satisfy the following properties,

- (Closure) $\forall g, h \in G^2,\ g \star h \in G$
- (Associativity) $\forall g, h, k \in G^3,\ g \star (h \star k) = (g \star h) \star k$
- (Identity) $\exists e \in G,\ \forall g \in G, e \star g = g \star e = g,$ *e is called the* identify *of the group G.*
- (Inverse) $\forall g \in G\ \exists h, g \star h = h \star g = e$*, the element that satisfies this property is called the inverse of g.*

Example 2.2.1 (Common groups)
We here give two examples of groups to highlight they are strucure that depend on both a set and binary operation.

1. *The set of integers $\mathbb{Z}$ equipped with the addition is a group with identity $e = 0$ and where any integer's inverse is its opposite value ($\forall z \in \mathbb{Z}, z - z = 0$). However, $(\mathbb{Z}, \times)$ is not a group because the inverse element of an integer $1/z \notin \mathbb{Z}$*

2. *The set of real numbers $\mathbb{R}$ is a group for both additive and multiplivative operations.*

We follow the definitions of invariance and equivariance from [Villar et al., 2021].

Definition 2.2.2 (Equivariance and Invariance)
Suppose a function $f : \mathcal{X} \to \mathcal{Y}$ and a group G acting on $\mathcal{X}$ and $\mathcal{Y}$ as $\star$. f is:

Description	Symbol	Unit (International System)
Time	t	second
Length	L	meter
Mass	M	kilogram
Electric Current	I	Ampere
Temperature	T	Kelvin
Mole (quantity of matter)	n	mol
Candela (light intensity)	I_v	cd

Table 2.2: Elementary physical dimensions and their respective units.

- *G-invariant if* $f(g \star x) = f(x), \; \forall (g, x) \in G \times \mathcal{X}$
- *G-equivariant if* $f(g \star x) = g \star f(x), \; \forall (g, x) \in G \times \mathcal{X}$

Equivariance is a property of physical laws that means any transformation of the input incurred by a group action g will affect the output in the same way. On the other hand, an invariant function will not be affected by a transformation of the input. Recent years have seen a growing literature on invariance in machine-learning. With means of inductive biases such as convolution layers in neural network, we are able to enforce translation and scale rotation invariance [Mitton, 2023; Villar et al., 2023]. Such symmetries are also a key component of graph neural networks to express invariance with respect to permutation transformation on graphs.

2.2.2 Dimensional Analysis

Units and Equation Homogeneity

Before jumping into the details of the main theorem, we need to explain what a physical measurement is and what a dimension is. The measure of a physical quantity comprises both a magnitude and a dimension as

$$X = \{X\} \, [X] \, . \tag{2.9}$$

The measure of a distance for example will have the dimension of a length $[L]$, and acceleration a length per time squared $\left[LT^{-2}\right]$. Any physical variable can be expressed as a product of integer exponents of the 7 elementary units written in table 2.2. The dimension is the actual object that is measured and that is not impacted by a change of units.

Among the quantities listed in table 2.2, the first three are most important to our subsequent work as they suffice to express all quantities present in mechanical systems. Before going further, it is worth keeping in mind that units and dimensions are not the same thing. Any system of units is an affine transformation of the other [Lee et al., 2021] and unit homogeneity is crucial to ensure correct analysis.

Operations on Dimensioned Quantities

Following the bracket notation from [Sonin, 2001], $[X]$ denotes the dimension of variable X.

- Two quantities X and Y can be added provided $[X] = [Y]$ and the resulting quantity has magnitude $\{X + Y\} = \{X\} + \{Y\}$ and dimension $[X + Y] = [X] = [Y]$.
- Two quantities can be multiplied whatever their dimensions and $\{X \times Y\} = \{X\} \times \{Y\}$, $[X \times Y] = [X] \times [Y]$.
- A quantity can be raised to the power of a rational fraction $\gamma \in \mathbb{Q}$ with $X^\gamma = \{X\}^\gamma [X]^\gamma$.

It is worth noting that for these operations to be properly defined, the quantities must be expressed in a consistent set of units [Villar et al., 2023; Shen, 2015]. In 1999, the Mars Climate Orbiter crashed because a software module designed to compute trajectories was returning Imperial rather than metric units [Board, 1999].

Dimension Homogeneity in Machine Learning Models

These remarks shed a new light unto the interpretation of machine learning models. Let us take the example of a linear regression problem

$$y_i = \beta \times x_i + \epsilon_i, \text{with } \epsilon_i \sim \mathcal{N}(0, \sigma^2), y_i \in \mathbb{R}, x_i \in \mathbb{R}^d, \tag{2.10}$$

where the outputs y_i are noisy realizations of a linear process and β is a vector of free parameters of size d. From a physicist point of view, this equation only makes sense if

$$\forall j = 1 \dots d,\ [\beta_j] = [y]/[x_j]. \tag{2.11}$$

Because this estimator is built on simple algebraic operations, this does not pose any major theoretical problem. One can simply assume equation 2.11 is verified and use the model as is.

What about kernel-based methods? There is debate among mathematicians about whether applying the exponential function to a dimensioned quantity makes sense. As we know, the exponential function can be written as the infinite sum

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}. \tag{2.12}$$

In general, the Taylor expansion (2.12) would be composed of heterogenous terms because $i \neq j \implies [x]^i \neq [x]^j$. As we saw previously, adding quantities only makes sense if they share the same dimension, which occurs if and only if x is dimensionless ($[x] = \mathbf{0}$). The question of applying transcendental functions to dimensional quantities is still an open question [Lee et al., 2021; Villar et al., 2023], so we here give the reader an intuition of where the problem is coming from, and why it might be ignored all together.

Let us take the simple case of the Squared Exponential kernel from equation 2.13.

$$k_{SE}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 e^{-(\boldsymbol{x}-\boldsymbol{x}')L^{-2}(\boldsymbol{x}-\boldsymbol{x}')}, \tag{2.13}$$

As we stated, the term $(\boldsymbol{x} - \boldsymbol{x}')L^{-2}(\boldsymbol{x} - \boldsymbol{x}')$ should be dimensionless to be passed into the exponential function. In the particular Automatic Relevance Detection case, we may assume

$$\forall j = 1 \dots d,\ [L_j] = [x_j]. \tag{2.14}$$

The equation (2.14) ensures the distance term is dimensionless and can be exponentiated. If the lengthscale is shared across all input dimensions, however, one might advocate it does not make mathematical sense since it computes heterogenous quantities. If the elements $\boldsymbol{x}$ are already dimensionless, the homogeneity is respected.

As the exponential function, its inverse the logarithm may be subjected to homogeneity issues [Molyneux, 1991]. The Taylor expansion for the logarithm can be written as,

$$\ln(x) = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{(x-1)^k}{k}. \tag{2.15}$$

Once again, each term in the sum will have a different dimension ($[x]^k$) unless x is dimensionless. This argument can however be countered since logarithms are usually applied to ratio of physical quantities. In such cases, $\ln(u/v) = \ln(u) - \ln(v)$ where $[u] = [v]$ and so the operation is legitimate. We point out the absence of scientific consensus on this matter to this day and refer the reader to [Lee et al., 2021] for additional discussion on homogeneity in transcendental functions.

Historically, this homogeneity problem has not appeared when GP models where used for kriging. Because the input x had the dimension of a length, dividing by the lengthscale naturally gave rise to a dimensionless exponent. One strength of considering the homogeneity of ML models is the increase in interpretability as was showed in [Kumar et al., 2018; Chandra et al.]. While we do not advocate for the theoretical invalidity of using dimensioned quantities with commonly used kernels, we believe the non-dimensionalization technique described in the next section gives a more theoretically sound application of the model.

Units-Equivariant Functions

Definition 2.2.3 (Units-typed space [Villar et al., 2023])
Random variables that are expressed as a magnitude and unit as in equation (2.9) take values in a ***units-typed space****:*

$$\mathcal{X} = \prod_{i=1}^{d} \mathcal{X}_{[x_i]}. \tag{2.16}$$

Formally we write an element of a units-typed space $\boldsymbol{x} = (\boldsymbol{x}, \bar{\boldsymbol{x}})$ *for making explicit its magnitude and dimension.*

A variable is called *dimensionless* if the vector $\bar{\boldsymbol{x}}$ is $\mathbf{0}$.

Example 2.2.2 (Unit-typed spaces in mechanics)
Suppose a mechanical equation is expressed in the based units kg, m, s the SI units for mass, length and time. Measurement can be expressed in those units by means of a vector $\bar{\boldsymbol{x}}_i$*:*

- *a mass of* $m_1 = 2\ kg$ *can be expressed as* $m_1 = 2\ [1, 0, 0]$*,*
- *an acceleration of* $a_1 = 10\ m.s^{-2}$ *expressed as* $a = 10\ [0, 1, -2]$*.*

Definition 2.2.4 (Rescaling Group [Villar et al., 2023])
Let us consider a unit-typed space $\mathcal{X}$ *of dimension* d *with* k *base units. Those units impose a* ***rescaling group*** $G = (\mathbb{R}_+^k, \times)$ *of which an element* $(g_1, \dots, g_k)$ *rescales the units of each element of* $\mathcal{X}$ *pointwise.*

$$\boldsymbol{g} \cdot \boldsymbol{x} = \left(\prod_{j=1}^{k} \boldsymbol{g}_j^{-\bar{\boldsymbol{x}}_j} \right) \cdot \boldsymbol{x} \tag{2.17}$$

Example 2.2.3 (Rescaling Unit)
In the second example above, should we wish to change the the acceleration in $km.h^-2$*, we can rescale the measurement with the element* $g = (1, 1000, 3600)$*. The new acceleration will thus be* $a_1 = 1 \times (1000)^{-1} \times (3600)^2 \times 10 = 129600\ km.h^{-2}$

Definition 2.2.5 (Units-typed function [Villar et al., 2023])
A function $f : \mathcal{X}_{[x]} \to \mathcal{Y}_{[y]}$ *is called a* ***units-typed function*** *if both spaces* $\mathcal{X}_{[x]}$ *and* $\mathcal{Y}_{[y]}$ *are unit-typed spaces.*

Definition 2.2.6 (Units-equivariant function [Villar et al., 2023])
A units-typed function is called ***units-equivariant*** *if it satisfies the following property,*

$$\forall g \in G, \forall x \in \mathcal{X}_{x_{[]}} \qquad f(g \cdot x) = g \cdot f(x). \tag{2.18}$$

In other words, that definition states that a units-equivariant function preserves the coherence of the equation by scaling its inputs and outputs appropriately.

Theorem 2.2.1 (Buckingham)
Assuming a physical system is described as a function of d independent variables as

$$f(x_1 \dots x_d) = 0. \tag{2.19}$$

If k elementary dimensions suffice to describe the system i.e.:

$$\forall i \in \{1 \dots d\}, \bar{x}_i = \prod_{j=1}^{k} \bar{d}_j^{\,\gamma_{i,j}}. \tag{2.20}$$

Then, the system can be equivalently described by $k-r$ dimensionless variables, called Π groups:

$$\forall j \in \{1, \dots, (k-r)\} \; \Pi_j = \prod_{i=1}^{d} x_i^{z_{i,j}}, \; z_{i,j} \in \mathbb{Z}. \tag{2.21}$$

The Π groups satisfy the equation

$$f_{\Pi}(\Pi_1, \dots, \Pi_{d-k}) = 0. \tag{2.22}$$

It is worth noting that the Π-groups are not unique and will condition the form of f_{Π}. The point of this theorem is to create a feature space that will be independent of the choice of units. This theorem had been developed to reduce the number of variables to control for collecting experimental data. It was instrumental in the discovery of instrumental quantities such as the Reynolds number in fluid dynamics [Lee et al., 2021]. More importantly, [Shen and Lin, 2019, 2018] demonstrate that dimensionless variables are maximal invariant statistics to scale transformation in fundamental dimensions.

Application of the Theorem

Now that we considered the theoretical aspects and benefit of the dimensionless variables with respect to statistical invariance, we present how to find the dimensionless variables. As it

Variable	Dimension
Diameter of the cylinder (D)	L
Density of the fluid (ρ)	ML^{-3}
Velocity (v)	LT^{-1}
Viscosity (μ)	$ML^{-1}T^{-1}$
Drag (F)	MLT^{-2}

Table 2.3: Physical variables and their dimensions for a fluid going through a cylinder.

turns out, that problem can be reduced to that of solving a system of linear diophantine equations. The equation 2.21 is called a power law and constitutes the basis for removing the dimension of the variables. To ensure that, the coefficients $z_{i,j}$ must satisfy the constraints

$$\sum_{i=1}^{d} z_{i,j}\bar{x}_j = 0 \tag{2.23}$$

for each of the $k - r$ Π-groups. In other words, that equation renders the variables dimensionless through the power law. A detailed example for solving such a system on a pendulum is available in Appendix A.1.2 The direct consequence is the non-uniqueness of the Π-groups which may follow if the rank of the system is not full.

Example 2.2.4 (Reynolds Number)
Fluid mechanics is the study of the behaviour of fluids under different environmental conditions. The Buckingham theorem has been used to construct Pi-groups to reduce the number of variables during experiments. Let us consider the problem of predicting the pressure in a cylinder. The relevant variables and their dimensions are summarized in table 2.3 As we can see, all the variables can be described by the 3 dimensions of mass, length and time. The system can therefore be described by $5 - 3 = 2$ dimensionless variables. The dimensional matrix for this system writes down as

$$\bar{\boldsymbol{X}} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & -3 & 0 \\ 0 & 1 & -1 \\ 1 & -1 & -1 \\ 1 & 1 & -2 \end{bmatrix}. \tag{2.24}$$

Each column represent the dimensions (M, L, T) in that order and each row is a variable. The Reynolds number is generally defined [Lee et al., 2021] as

$$\Pi_1 = Re = \frac{\rho v D}{\mu}. \tag{2.25}$$

, It can be found numerically by solving , which yields the second Π*-group* $\Pi_2 = \dfrac{F}{\rho D^2 V^2}$.

General Consequences

Training machine learning models in dimensionless spaces presents multiple advantages. The first is an improvement in prediction performance. [Villar et al., 2023; Oppenheimer et al., 2023] demonstrated that estimators trained in that space are able to make accurate predictions on test data with out-of-distribution properties. This is partly due to the dimensionality reduction property of the transformation and the deletion of colinearities between the features. Moreover, the constraints induced by the physics prior enforce appropriate scaling relationships between the inputs and outputs. Additionally, the normalization induced by the nondimensionalization procedure imposes a better conditionning of data which will improve the stability of models trained on them.

An issue raised by this method is caused by the method used for finding appropriate Π-groups. They come from solving the system (2.23) which, in general, admits non-unique solutions. In fact, the set of solutions forms a lattice of the space which may be infinite. As a consequence, the one solution used for transforming the space should be carefully considered. It will necesarily require domain-specific knowledge to ensure the physical validity and coherence of that specific Π-group. This may be increasingly difficult as the dimension of the systems at hand grow. When, several masses and lengths are available to non-dimensionalize a velocity to instance, one should make sure that the variables present a causal link. Otherwise the transformation will not allow meaningful equivariance properties. This raises a question of trade-off between preserving the flexibility of a statistical learning method and modeling from first principles. The more prior knowledge is needed, the less benefit we pull from data-driven approaches.

Finally the main drawback of this approach is the strict requirement for dimensional measurements. Text or image data for instance do not present this property as these modalities do not take values in units-types spaces. The theorem could in theory be extended to other scientific fields such as economics. The step for doing so is adapting the dimensions of governing equations such as monetary value, volatility, man-hours and so forth [Barnett, 2004; Texocotitla et al., 2020]. However, we emphasize that in economics sciences the dimensions are not as well-defined as in physics. Therefore, applying the principles of DA to other fields should be made with careful attention.

2.3 Probabilistic Modeling

2.3.1 Bayesian Inference

Bayesian statistics are a mathematical paradigm that interpret probabilities as a measure of uncertainty. Inference in this view is done by combining a priori knowledge with actual data. This process is enlightened by Bayes' rule of conditional probabilities

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}. \tag{2.26}$$

These methods are used in machine-learning to train classifiers and regressors that are aware of their own uncertainty. On figure 2.3, we illustrate this principle on a linear model. A distribution is associated with both weights of the model and additive noise. This trait is particularly appealing to remove the model bias caused by lack of training data and for designing exploration strategies in Bayesian Optimization or dynamic systems [Barto, 2013; Shahriari et al., 2016]. Bayesian inference however suffers from the need computing the evidence, an integral that is intractable except with strong limiting assumptions on the family distributions of the prior and likelihood. The application of this framework therefore consists chiefly in finding good approximations of the posterior distribution.

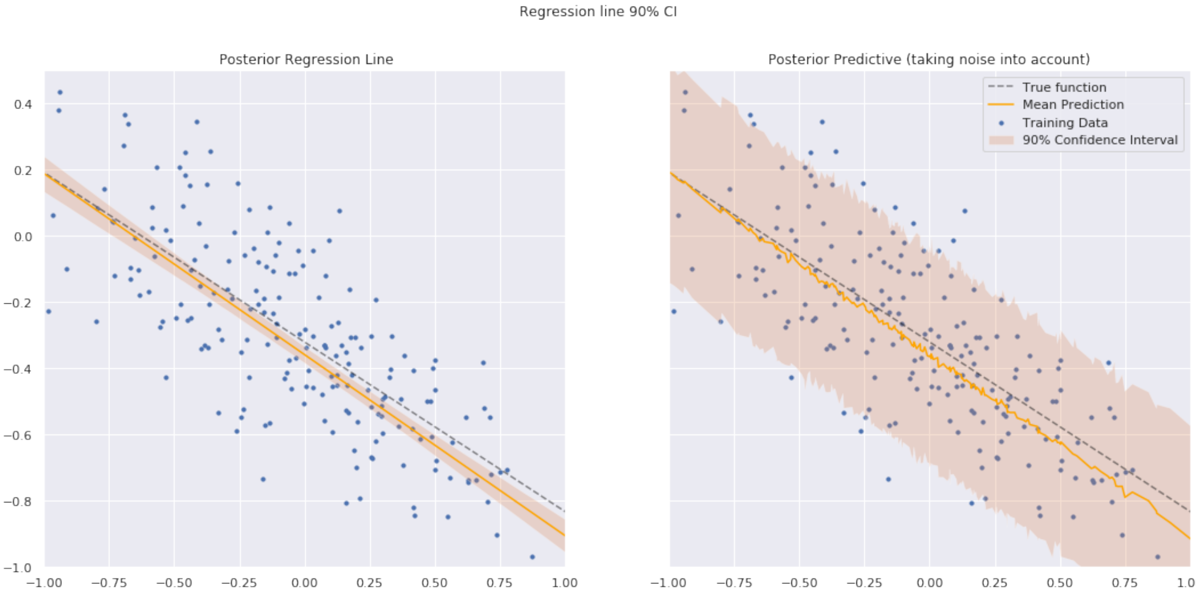


Figure 2.3: Bayesian linear regression. On the left is the distribution of the latent linear function and the right the predictive likelihood.

For decades, the major algorithm used for approximating posteriors has been Monte Carlo methods and their variants [Neal, 1993; Homan and Gelman, 2014]. It consists chiefly of approximating an integral as the expected value of its density using the law of Large Numbers (2.27),

$$\frac{1}{N}\sum_{i=1}^{N} f(x_i),\ x_i \sim p(x_i) \underset{N\to\infty}{\to} \mathbb{E}_p\left[f(X)\right]. \tag{2.27}$$

In order to reduce the sample complexity of the algorithm, it draws samples that are not independent but from a Markov Chain which stationary distribution is that of the posterior, thus gave the name *Markov Chain Monte Carlo* (MCMC). Provided an infinite number of samples is drawn, this algorithm will converge to the exact posterior. This comes at the cost of the sampling time, which can be prohibitively expensive. Moreover, this family of algorithms suffers from the *curse of dimensionality* which states that distances increase exponentially with space dimension [Bellman, 1962] and causes data sparsity.

The last 15 years have in turn, seen the advent of a new family of posterior approximations methods allowed by progress in both parallel hardware architectures and stochastic optimization software. This approach, called *Variational Inference* (VI) provides a consistent framework for the posterior approximation problem [Hoffman et al., 2013; Blei et al., 2017]. Rather than approximating the integral directly as MCMC does, VI maximizes a parametric lower bound of it. By doing so, the problem of computing an intractable integral is converted into an continuous optimization one. As such, the practitioner can resort to the modern stochastic optimization toolkit with gradient descent on huge datasets using Graphical Computing Units (GPU). In the next section, we explain on a concrete example of Gaussian Process Regression how these schemes work and what their respective strength and weaknesses are.

2.3.2 Gaussian Process Regression

Gaussian Processes are distribution over functions with multivariate normal finite-dimensional distribution [Rasmussen and Williams, 2005]. Gaussian processes are uniquely defined by their mean and covariance function, the class of which conditions the characteristics of their samples paths. Because many stochastic processes are Gaussian, using them as priors for Bayesian inference allows for estimating a wide variety of functions. For instance, any smooth real-valued function can be viewed as the realization of a GP with squared-exponential kernel. In the following, we derive the most common approximation used for GP regression, namely the *type-II Marginal Likelihood Estimation*. We focus here on the derivation using the squared exponential covariance function

$$k_{SE}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 e^{-(\boldsymbol{x}-\boldsymbol{x}')L^{-2}(\boldsymbol{x}-\boldsymbol{x}')}, \tag{2.28}$$

with $\lambda = (\sigma, L)$ the set of hyperparameters. σ is the signal variance and $L = (l_1, \dots, l_d)$ are the lengthscales.

In the case of regression with white noise, we write the relation between inputs and observations as

$$\boldsymbol{y} = \boldsymbol{f} + \boldsymbol{\epsilon}, \epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2_{obs} \boldsymbol{I}). \tag{2.29}$$

$$\begin{cases} p(\boldsymbol{y}|f) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{f}, \Sigma_y) \\ p(\boldsymbol{y}|X) = \mathcal{N}(\boldsymbol{y}, \mathbf{0}, \boldsymbol{K} + \Sigma_y) \end{cases} \tag{2.30}$$

A Gaussian likelihood is conjugate to the prior, therefore the posterior distribution is also a multivariate Gaussian.

$$p(\boldsymbol{f}|\boldsymbol{y}, \boldsymbol{X}) = \mathcal{N}\left(\boldsymbol{f}|\boldsymbol{K}\left(\boldsymbol{K} + \Sigma_y\right)^{-1} \boldsymbol{y}, \boldsymbol{K}\left(\boldsymbol{K} + \Sigma_y\right)^{-1} \Sigma_y\right) \tag{2.31}$$

For prediction at a new test point $\boldsymbol{x}^*$,

$$p(\boldsymbol{f}^*, \boldsymbol{y}|x^*, \boldsymbol{X}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{f}^* \\ \boldsymbol{y} \end{bmatrix} |\mathbf{0}, \begin{bmatrix} \boldsymbol{k}_{**} & \boldsymbol{k}_*^T \\ \boldsymbol{k}_* & \boldsymbol{K} + \Sigma_y \end{bmatrix}\right) \tag{2.32}$$

Conditional posterior over $\boldsymbol{f}^*$,

$$p(\boldsymbol{f}^*|\boldsymbol{x}^*, \boldsymbol{y}, \boldsymbol{X}) = \mathcal{N}\left(\boldsymbol{f}^*|\boldsymbol{k}_*^T\left(\boldsymbol{K} + \Sigma_y\right)^{-1} \boldsymbol{y}, \boldsymbol{k}_{**} - \boldsymbol{k}_*^T\left(\boldsymbol{K} + \Sigma_y\right)^{-1} \boldsymbol{k}_*\right) \tag{2.33}$$

Model selection is done by optimizing $p(\boldsymbol{y}|\boldsymbol{X}, \lambda)$ with respect to λ with gradient-based or Newton methods,

$$\log\left(p(\boldsymbol{y}|\boldsymbol{X}, \lambda)\right) = -\frac{1}{2}\boldsymbol{y}^T(\boldsymbol{K} + \Sigma_y)^{-1}\boldsymbol{y} - \frac{1}{2}\log\left|\boldsymbol{K} + \Sigma_y\right| - \frac{N}{2}\log 2\pi. \tag{2.34}$$

The derivations above rely on estimating the hyperparameters pointwise hence removing any uncertainty associated with them.

2.3.3 The Fully Bayesian GP Model

In this section, we consider the full probabilistic model for Gaussian Process Regression. Inference in such a model follows the graphical model described in 2.4.

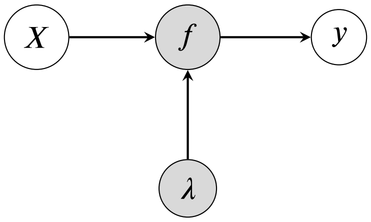


Figure 2.4: Graphical Model for Gaussian Process regression. The node λ represents the free parameters and f the latent hidden function.

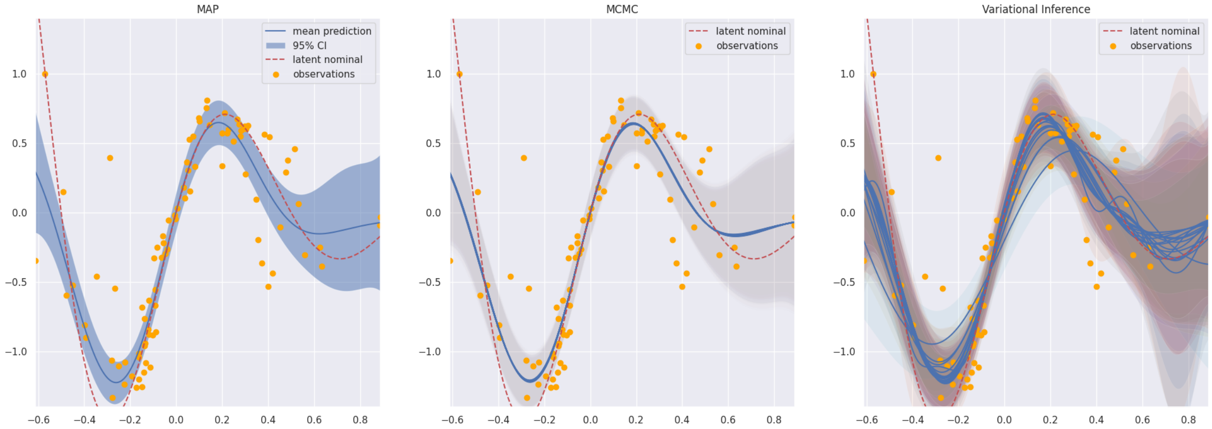


Figure 2.5: Comparison of MAP (left), MCMC (center) and VI (right) inference for a Gaussian Process regression on toy data.

$$\begin{aligned}
\text{Prior over Hyperparameters} \qquad & \lambda\sigma_n \sim p(\lambda) \\
\text{Prior over parameters} \qquad & \boldsymbol{f}|\boldsymbol{X}, \lambda \sim \mathcal{N}\left(0, K_\lambda\right) \\
\text{Data Likelihood} \qquad & \boldsymbol{y}|\boldsymbol{f} \sim \mathcal{N}\left(\boldsymbol{f}, \sigma_n^2 \boldsymbol{I}\right).
\end{aligned} \tag{2.35}$$

The drawback of deriving the full posterior over the model and hyperparameters is that it renders its computation, as well as that of the marginal likelihood, intractable. Therefore, inference in that model relies on sampling or variational approximations as was studied in [Lalchand et al., 2022; Lalchand and Rasmussen, 2020; Rossi et al., 2021; Yu et al., 2019]. The prior on the kernel hyperparameters was demonstrated to have only little influence on the end results [Chen and Wang, 2018] and can be leveraged to estimate non-stationary kernels [Burnaev et al., 2016].

On figure 2.5, we plot the posterior distributions of the 3 inference schemes on a simple dataset. The toy data is generated as noisy samples of a one-dimensional damped harmonic oscillator. On the leftmost plot, a single mean function is plotted since it is a deterministic function given in equation (2.33). On the center and right plots however, the means are conditioned by the hyperparameter samples and therefore non-deterministic. This is why several means in blue are showed on the graph.

2.3.4 Sparse Gaussian Process

The main impediment for scaling GP models to large dataset is the requirement for inverting the covariance matrix, which is a cubic operation in the number of samples. To circumvent this issue, various methods have been developed to relax that dependency on the training points. Early works such as [Lawrence et al., 2003; Csató and Opper, 2002] proposed to condition the posterior only on a subset of observations. This however led to the combinatorial problem of identifying the most significant of these subsets to represent all the data.

An orthogonal approach by [Snelson and Ghahramani, 2005] then paved the way to the *Sparse Gaussian Process*, which instead conditions the posterior on a set of pseudo-points that are not part of the dataset but instead aim to summarize the observations. The number of pseudo-points M is chosen such that $M \ll N$ such that the posterior and likelihood computations are less heavy than on the actual data. Much of the work in sparse approximation hence consist in finding good low-rank approximations of the covariance matrix. The variational sparse GP [Titsias, 2009; Hensman et al., 2013] then allowed optimization with stochastic variational inference, making such models applicable to large datasets up to millions of samples in both regression and classification tasks [Hensman et al., 2015]. This approach does not modify does not modify the GP prior and leaves the generative model unchanged and augment it with the inducing points $\boldsymbol{Z}$ and their evaluations $\boldsymbol{u}$,

$$p(\boldsymbol{y}, \boldsymbol{f}, \boldsymbol{u} | \boldsymbol{X}, \boldsymbol{Z}) = p(\boldsymbol{y}|\boldsymbol{f}) p(\boldsymbol{f}, \boldsymbol{u} | \boldsymbol{X}, \boldsymbol{Z}). \tag{2.36}$$

We introduce a proposal distribution over the latent variables $q(\boldsymbol{f}, \boldsymbol{u})$ that will approximate the true posterior $p(\boldsymbol{f}, \boldsymbol{u} | \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{Z})$ with

$$q(\boldsymbol{f}, \boldsymbol{u}) = p(\boldsymbol{f} | \boldsymbol{u}, \boldsymbol{X}, \boldsymbol{Z}) q(\boldsymbol{u} | \boldsymbol{Z}), \tag{2.37}$$

Where the distribution $p(\boldsymbol{f} | \boldsymbol{u}, \boldsymbol{X}, \boldsymbol{Z}) = \mathcal{N}(\boldsymbol{f} | \boldsymbol{K}_{NM} \boldsymbol{K}_{MM}^{-1} \boldsymbol{u}, \boldsymbol{K}_{NN} - \boldsymbol{K}_{NM} \boldsymbol{K}_{MM}^{-1} \boldsymbol{K}_{MN})$ is the conditional for $\boldsymbol{f}$ in the standard sparse approximation [Snelson and Ghahramani, 2005].

This gives us the following ELBO (using Jensen inequality),

$$\begin{aligned}
\log p(\boldsymbol{y}|\boldsymbol{X}) &= \log \int \frac{q(\boldsymbol{f}, \boldsymbol{u})}{q(\boldsymbol{f}, \boldsymbol{u})} p(\boldsymbol{y}|\boldsymbol{f}) p(\boldsymbol{f}, \boldsymbol{u} | \boldsymbol{X}, \boldsymbol{Z}) d[\boldsymbol{f}, \boldsymbol{u}] && (2.38) \\
&\geq \int q(\boldsymbol{f}, \boldsymbol{u}) \log \frac{p(\boldsymbol{y}|\boldsymbol{f}) p(\boldsymbol{f}, \boldsymbol{u} | \boldsymbol{X}, \boldsymbol{Z})}{q(\boldsymbol{f}, \boldsymbol{u})} d[\boldsymbol{f}, \boldsymbol{u}] && (2.39) \\
&= \int q(\boldsymbol{f}, \boldsymbol{u}) \log \frac{p(\boldsymbol{y}|\boldsymbol{f}) \cancel{p(\boldsymbol{f} | \boldsymbol{u}, \boldsymbol{X}, \boldsymbol{Z})} p(\boldsymbol{u}|\boldsymbol{Z})}{\cancel{p(\boldsymbol{f} | \boldsymbol{u}, \boldsymbol{X}, \boldsymbol{Z})} q(\boldsymbol{u}|\boldsymbol{Z})} d[\boldsymbol{f}, \boldsymbol{u}] && (2.40)
\end{aligned}$$

$$= \mathbb{E}_{q(\boldsymbol{f}|\boldsymbol{X},\boldsymbol{Z})}\left[\log p(\boldsymbol{y}|\boldsymbol{f})\right] - D_{\mathrm{KL}}\left[q(\boldsymbol{u}|\boldsymbol{Z})||p(\boldsymbol{u}|\boldsymbol{Z})\right] \tag{2.41}$$

where

$$q(\boldsymbol{f}|\boldsymbol{X},\boldsymbol{Z}) = \int p(\boldsymbol{f}|\boldsymbol{u},\boldsymbol{X},\boldsymbol{Z})q(\boldsymbol{u}|\boldsymbol{Z})d\boldsymbol{u}. \tag{2.42}$$

The latest expression is analytically tractable when the distribution $q(\boldsymbol{u}|\boldsymbol{X})$.

Finally in order to apply that framework to large datasets, [Hensman et al., 2015] introduced the distribution $q(\boldsymbol{u}|\boldsymbol{X}) = \mathcal{N}(\boldsymbol{u}|\boldsymbol{m},\boldsymbol{S})$ where $\boldsymbol{m},\boldsymbol{S}$ are variational parameters. This allows the decomposition of the expectation of the likelihood in equation (2.41) across batches of data to apply stochastic optimization routines.

Gaussian Process regression is the gold-standard method for Bayesian non-parametric inference, it is both sample efficient and able to return well calibrated uncertainty estimates. In this section, it served us to explain how to do inference in a probabilistic model with MAP, MCMC or VI. The key component of probabilistic modelling that matter for the goal of robustness is the ability to consider an infinity of realizations with the means of a probability distribution and integrate them.

Chapter 3.

Dimensional Analysis and Context Drift

In this chapter, we study a transformation of the feature space based on the Buckingham-Pi theorem introduced to determine the number of dimensional groups required to describe a physical phenomenon. We evaluate the invariance properties of the transformation in the context of regression on second-order systems, specifically applied to the actuated pendulum. We demonstrate that estimators trained on this state space are able to make accurate predictions outside of the training distribution support and that this transformation is robust to uncertainty about the system variables.

Dimensional Analysis is the study of the interplay between the measure of physical quantities and the units they are defined in. It allows reasoning about equation homogeneity and is ubiquitous in physical sciences. As taught in undergraduate physics and engineering courses one can add quantities if and only if they are expressed in the same unit, thus the same dimension. For example, it does not make sense to add a length quantity to a speed, as one does not compare apples and oranges.

In the statistical learning world however, things are a bit different [Lee et al., 2021]. Dependent variables or features are assumed to take values in Euclidean space that ignore their units. This flaw is highlighted by Cox in [Blitzstein, 2023] when asked about the stability of statistical models. He states that statistical analysis should not be disconnected from the relevant scientific field such that representations should satisfy the constraints it imposes.

While this does not pose problem from a practical standpoint, the knowledge of the dimension in which measurements are expressed can be leveraged to reduce the dimension of the problem with the Buckingham-Pi theorem [Buckingham, 1914]. This theorem exploits the symmetries of a dimensioned physical equation to reduce the number of variables required to express it. Moreover, the embedding transforms the natural space into equivariant features that lead to better out-of-distribution generalization. While this transformation requires additional knowledge compared to state-space view, it is resilient to uncertain measurements of the variables required for non-dimensionalization. Our study focuses on dynamic systems, whose transition function are conditioned on static variables.

This chapter comprises three main sections. First, we introduce how statistical shift prevents identification of real-world dynamic systems. In the second part, we explain how dimensional analysis and the Buckingham-Pi theorem are used to build invariant predictors. Last, we demonstrate empirically on a second order system how that dimensionality reduction technique improves the generalization capabilities of statistical estimators.

3.1 Motivation: Context Drift in Dynamic Systems

In the Machine-Learning literature, distribution shift [Quiñonero-Candela et al., 2008] is the phenomenon that occurs when the data-generating distribution of an observable changes between experiments. For our study, we focus on shifts that occur in second-order dynamic systems. In the general settings, let us assume observations of the position x at any time take value in a space $\mathcal{X}$. The evolution of the system is characterized by its second-order derivatives with respect to time,

$$\ddot{\boldsymbol{x}}_t = f(\boldsymbol{x}_t, \dot{\boldsymbol{x}}_t). \tag{3.1}$$

Ordinary Differential Equations (ODE) such as (3.1) are found in most scientific areas from biology to economics. They are a generic tool to describe the evolution of continuous variables with respect to time. Consequently, they can be used for predicting the future state of the system at any given point in time given an initial condition $\boldsymbol{x}_0, \dot{\boldsymbol{x}}_0$. This is called an *Initial Value Problem*, and can be solved in two ways. If the equation has an *analytic solution*, we can directly write the value of the system state for any time provided the initial values or boundary condition can constrain the solution. Analytic solutions are, in general, difficult to find or non-existent if the equation is not linear. In such cases, scientists and practitioners resort to *numerical integration*. That second approach involves integrating the equation step by step until the desired time is reached.

The aforementioned methods however, require sufficient domain knowledge for writing the equation (3.1) in the first place. This is often done from first principles, which are not necessarily available given the complexity of dimensionality of the system. The alternative then, is to infer the equation from observational data instead. This approach is grounded in supervised learning, as it aims to learn a mapping from measurements $\boldsymbol{x}_t, \dot{\boldsymbol{x}}_t$ to $y_t = f(\boldsymbol{x}_t, \dot{\boldsymbol{x}}_t) + \epsilon$. Where ϵ is a standard Gaussian noise. From an estimation $\hat{f}$ trained on trajectories of the system, we can then predict the future from any initial value $\boldsymbol{x}_0, \dot{\boldsymbol{x}}_0$ with a stepwise integrator. If the discrepancy between the model $\hat{f}$ and the true dynamics f is small, the predicted trajectories will simulate the true system accurately up to that difference.

In practice, however, external perturbations or hidden feedback loops can cause the system dynamics to change after repeated experiments. If the new dynamics f_{shift} differ too much from the initial ones, then the model $\hat{f}$ will no longer be able to simulate the system accurately. Such change will cause a *shift* in the distribution of observed trajectories. This modification of the dynamics does not however come from nowhere and is a matter of perspective. While we wrote the dynamics as a function of the state, they in fact also depend on additional context variables. This context is not included in equation 3.1 as it remains static and does not depend on time. More precisely, it depends on time but moves on a slower timescale than the state variables and is thus omitted. Assuming full observation of the dynamic and static system variables, the evolution of the system is entirely described as

$$\ddot{\boldsymbol{x}}_t = f_c(\boldsymbol{x}_t, \dot{\boldsymbol{x}}_t) = f(\boldsymbol{x}_t, \dot{\boldsymbol{x}}_t, \boldsymbol{c}). \tag{3.2}$$

In a pendulum for instance, the state variables are the angle position and velocities while the context includes its mass and length.

Given that perspective, it is clear that changing from a context $\boldsymbol{c}_0$ to $\boldsymbol{c}_*$ will modify the dynamics of the system. The function governing the dynamics does not change itself, but the parameters on which it depends do. A robust model of the system is one that is able to reduce the prediction error on a range of context C_{test} as

$$\hat{f}^* = \arg\min_{\hat{f}} \int_{c \in C_{test}} \|f_c - \hat{f}\|, \tag{3.3}$$

for a given functional norm $\|.\|$.

We tackle this problem from the standpoint of zero-shot transfer learning. In that specific case, one aims to estimate a model given training data from an atomic context $\boldsymbol{c}_0 = \left\{\boldsymbol{c}_0\right\}$ to a compact sub-ensemble C_{test}. In the next section, we describe how this can be achieved using the Buckingham-Pi transformation.

3.2 Dimensionality Reduction with Buckingham Pi Theorem

3.2.1 Buckingham-Pi Theorem for Dynamic Systems

Context-Dependent Non-Dimensionalization

The Buckingham theorem can be applied in machine learning tasks as a dimensionality reduction technique. More importantly than that, the dimensionless features lead to equivariant features [Villar et al., 2023]. For simplicity, we rewrite the equation (3.2) more generally as

$$f(\boldsymbol{x};\boldsymbol{c}) = \boldsymbol{y}, \tag{3.4}$$

with the dimensions of $\boldsymbol{x}$, $\boldsymbol{c}$ and $\boldsymbol{y}$ are d, r and p respectively. In that formulation, $\boldsymbol{x}$ includes all the time-dependent variables of the system and $\boldsymbol{y}$ is the dependent observable of the system.

Assumption 1 (Full-rank context)
We assume the rank of unit-typed set matrix $[\bar{\boldsymbol{c}}_i]_{i=1,\dots,r}$ is full. This means the span of the context contains all the k elementary dimensions in which the ODE is expressed

This assumption implies that the position and velocity variables can be made dimensionless by multiplying each with a product of the context ones. A rescaling group that non-dimensionalize the state variables is expressed as

$$\forall i = 1,\dots,d \,:\, g_i = \prod_{j=1}^{k} c_j^{\alpha_{i,j}}, \qquad s.t. \sum_{j=1}^{d} \alpha_{i,j}\bar{x}_{ij} = 0, \tag{3.5}$$

where $\alpha_{i,j} \in \mathbb{Z} \forall i, j$. We call the action of a group that satisfies (3.5) a **Π-group** in reference to the Buckingham theorem.

Finding the elements of that group therefore consists in finding the coefficients $\alpha_{i,j}$, which comes down to solving a linear system of equations. Under the full rank hypothesis 1, a unique solution exists. That solution is unique if $d > k$ and non-unique if $d \leq k$ In section 3.2.2, we demonstrate on an example how to solve the system for a simple pendulum.

Regression in Dimensionless State-Space

Let us assume we have found the coefficients $\boldsymbol{\alpha}$ that satisfy equation (3.5), we write the corresponding group action as

$$G_{\Pi_x}(\boldsymbol{c}) = \left(g_1,\dots,g_d\right) \tag{3.6}$$

$$G_{\Pi_y}(\boldsymbol{c}) = \left(h_1,\dots,h_p\right). \tag{3.7}$$

We emphasize here that the group actions are dependant of the context vector. Equation (3.4) can then be rewritten as follows:

$$f(G_{\Pi_x}(\boldsymbol{c})\cdot\boldsymbol{x}) = G_{\Pi_y}(\boldsymbol{c})\cdot f(\boldsymbol{x}). \tag{3.8}$$

The resulting equation (3.8) is a direct application of the Buckingham theorem with the appropriate Π-groups and their equivariant property 2.18. Suppose we have trained a model on a singleton nominal context $\boldsymbol{c}_0$. Any perturbation on it can be written as $\boldsymbol{c} = \boldsymbol{\kappa c}_0$ with $\boldsymbol{\kappa}$ composed of strictly positive elements. Finally, using the fact that $G_{\Pi_x}(\boldsymbol{\kappa c}_0) = G_{\Pi_x}(\boldsymbol{\kappa})G_{\Pi_x}(\boldsymbol{c}_0)$ We can then rewrite equation (3.8) as

$$f(G_{\Pi_x}(\boldsymbol{\kappa})G_{\Pi_x}(\boldsymbol{c}_0) \cdot \boldsymbol{x}) = G_{\Pi_y}(\boldsymbol{\kappa})G_{\Pi_x}(\boldsymbol{c}_0) \cdot f(\boldsymbol{x}). \tag{3.9}$$

This definition for the model corresponds to the form of an equivariant function 2.2.2. The consequence of equation (3.9) is that we can estimate a single model for the latent function f that will be valid across a range of domains through the rescaling of its inputs and outputs. Because of the equivariance of the transformation, that means a singleton context $\{\boldsymbol{c}_0\}$ should in theory be enough to construct a model that can generalize to any $\boldsymbol{c} \in \mathcal{C}$.

3.2.2 Application to Actuated Pendulum

In all the following, we call Φ_Π the context-dependent mapping from state variables into dimensionless variables,

$$\Phi_\Pi : (\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^d \times \mathbb{R}^p \mapsto \left(G_{\Pi_x} \cdot \boldsymbol{x}, G_{\Pi_y} \cdot y\right). \tag{3.10}$$

We now apply the theorem to the case of an actuated frictionless pendulum to discover dimensionless state-space features of the dynamic system. The physical variables that describe this system are summarized in table 3.1. The output variable, angular acceleration is linked to the input variables with an unknown ODE as

$$\ddot{\theta} = f\left(M, g, L, c; u, \theta, \dot{\theta}\right). \tag{3.11}$$

The system is entirely described by 8 variables and the 3 elementary dimensions of mass, length and time. Therefore, according to Buckingham-Pi theorem, it can be reduced to 4 dimensionless variables. Following the derivation described in appendix A.1.2, we can describe the frictionless

[1] Dimension is actually $[L]^0$ as an angle is a ratio of lengths, what matters to the analysis here is that they are dimensionless. [Lee et al., 2021]

Variable	Dimension
mass m	$[M]$
Earth gravitational constant g	$[L][t]^{-2}$
length l	$[L]$
friction coefficient	$[\mathrm{M}][\mathrm{t}]^{-1}$
torque u	$[M][L]^2[t]^{-2}$
angle [1] θ	1
angular velocity $\dot{\theta}$	$[t]^{-1}$
angular acceleration $\ddot{\theta}$	$[t]^{-2}$

Table 3.1: Physical variables of the frictionless pendulum and their dimensions. The first 4 are static variables, while the last 4 depend on time.

pendulum exactly with the Π-groups in equation (3.12).

$$\begin{aligned}\Pi_u &= \frac{u}{MgL}\\ \Pi_{\cos(\theta)} &= \cos(\theta)\\ \Pi_{\sin(\theta)} &= \sin(\theta)\\ \Pi_{\dot{\theta}} &= \dot{\theta}\sqrt{\frac{L}{g}}\\ \Pi_{\ddot{\theta}} &= \frac{L\ddot{\theta}}{g}\end{aligned} \tag{3.12}$$

Note that we additionally transform the raw angle θ into $(\cos(\theta), \sin(\theta))$. The quantity $\frac{L}{g}$ is well known by physicists and often written ω_0^2. It is proportional to the period of isochronous free oscillations in the small-angle approximation. These transformations are not affected by non-dimensionalization because angles, and their sine and cosine are all dimensionless quantities.

Let us note the dimensionless variable for $\dot{\theta}$ is not exactly a power law. The variable we found analytically in A.1.2 is $\Pi'_{\ddot{\theta}} = \frac{\ddot{\theta}}{\omega_0}^{2}$. However, this transformation looses the information of the sign of the angular speed. To circumvent that problem, we consider the square root instead for the group in 3.12. We write the mapping from natural to dimensionless space as

$$\phi_\Pi(x, y) = (\Pi_u, \Pi_{\cos(\theta)}, \Pi_{\sin(\theta)}, \Pi_{\dot{\theta}}, \Pi_{\ddot{\theta}}). \tag{3.13}$$

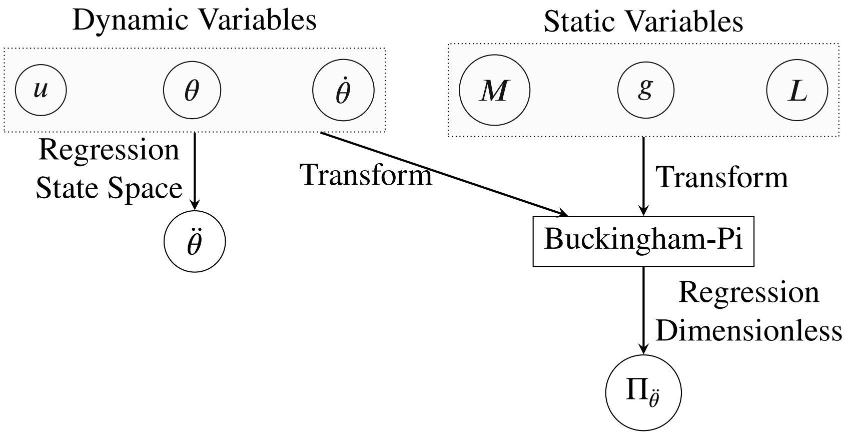


Figure 3.1: Regression task on angular velocity for the pendulum.

Pendulum Equations

The equations of motion of the damped pendulum writes down as

$$\ddot{\theta} + \frac{c}{M}\dot{\theta} - \frac{g}{L}\sin(\theta) - \frac{1}{ML^2}u = 0. \tag{3.14}$$

We multiply by $\frac{L}{g}$ to obtain the corresponding Π-groups.

$$\Pi_{\ddot{\theta}} + \frac{c}{M}\sqrt{(}\frac{L}{g})\Pi_{\dot{\theta}} - \Pi_{\sin(\theta)} - \Pi_u = 0 \tag{3.15}$$

The equation (3.15) is hence much more simple than (3.14). In the absence of friction if $c \ll 1$, is is invariant all the parameters. However in the case where friction is not negligible, the trajectories in dimensionless space will be sensitive to changes in parameters. In the following experimental section, we aim to demonstrate empirically that the discovered Π-groups lead to an invariant predictor across perturbations of the context, as per definition 2.2.2.

3.3 Application to Robust System Identification

In this section, we study the ability of Buckingham-Pi theorem to generate an invariant feature space that can be used for robust statistical estimation. Specifically, how well does an estimator trained on features from the dimensionless space make predictions out of its training distribution. Secondly, we study the impact of uncertainty associated with the physical static variables used for non-dimensionalizing the state-space features. The experiments aim to answer the following questions:

- Is adding context information to natural space enough for increasing robustness?
- How much does non-dimensionalization increase generalization?

Data Generation and Models

In the reminder of this section, we illustrate the application of the Buckingham-Pi theorem on the frictionless pendulum, as an example of second-order dynamic system. We generate data by sampling trajectories with random initial states and integrate the equations of motion with Runge-Kutta scheme of order 5 given by Scipy package [Virtanen et al., 2020]. At each time step, the system is excited with a signal sampled uniformly over the permissible action space between $[-1, +1]Nm$. For each version of the environment, we use 10000 samples for training and testing. For each dataset, we train and evaluate three different models:

- Multi-Layer Perceptron
- Variationally Sparse Gaussian Proces with Type-II MLE
- VSGP with fully Bayesian inference, using mean-field stochastic variational inference [Ranganath et al., 2016].

MLP are universal function approximators and Gaussian Process can be viewed as infinitely wide single-layer Bayesian networks [Neal, 1994; MacKay, 1991]. These models are ubiquitous in Machine-Learning hence it makes sense to compare them for evaluating the impact of a transformation of feature spaces for probabilistic and non-probabilistic models. The hyperparameters used for training the models are detailed in section 3.3.2.

The graph in figure 3.1 illustrates the regression procedure in the natural state-space view and on the dimensionless equivalent.

The metric we used is Symmetric Mean Absolute Percentage Error (sMAPE) equation 3.16. It has often been used for problems of time series forecasting [Chen and Yang, 2004] as an alternative to Mean Absolute Percentage Error (MAPE) which values may diverge to infinity if target values are too close to 0. The reason we turn to this metric rather than mean squared error is that it is insensitive to multiplicative factors, meaning we do not have to transform the dimensionless target before computing it. In other words, the error is the same in both natural and dimensionless spaces.

$$sMAPE = \frac{100}{N}\sum_{i=1}^{N}\frac{|\hat{y}_i - y_i|}{|\hat{y}_i| + |y_i|}. \tag{3.16}$$

3.3.1 Generalization in Dimensionless Space

We generate 3 regression datasets by sampling $x = (M, g, L; u, \theta, \dot{\theta})$ according to the values in table 3.2 and the target $y = \ddot{\theta}$ according to the system ODE conditioned by the values of x. The first dataset is used for training while the remaining two are for testing. Each of these datasets is

Parameter	Training	Test 1	Test 2
M	1	.	.
g	9.81	.	.
L	1	$\mathcal{U}(0.75, 1.25)$	$\mathcal{U}(0.5, 1.5)$
u	$\mathcal{U}(-1, 1)$	.	.
θ	$\mathcal{U}(0, \pi)$	.	.
$\dot{\theta}$	$\mathcal{U}(-2, 2)$	.	.

Table 3.2: Sampling distributions of the input variables for generating the training and test distributions.

then non-dimensionalized according to equation (3.12).

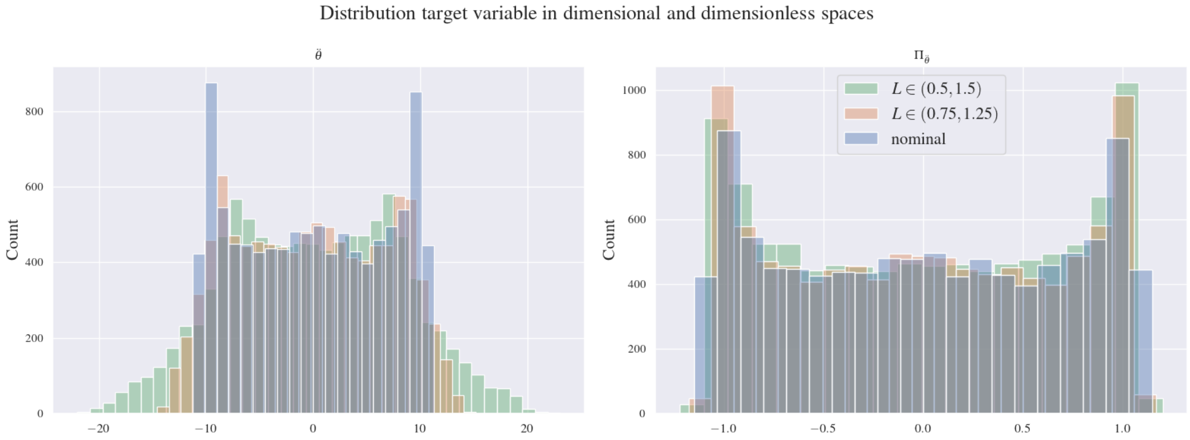


Figure 3.2: Distribution of the target variable in natural space (left) and dimensionless space (right) for the training and test distributions. In the dimensionless space, the transformation causes a normalization of the supports whereas they do not overlap in the natural space.

Firstly, we analyse the distribution of the target variable, as shown in figure 3.2. Of interest to is the support of distribution, and how it varies when the pole length shifts. As we can see on the left-hand side, the absolute value of $\ddot{\theta}$ increases as the sampling support of L widens. On the other hand, the support of the target variable in dimensionless space remain similar. This causes the generalization problem to be turned from extrapolation to interpolation as was observed in [Oppenheimer et al., 2023]. We summarize the link between the width of the pole length support and the maximum value of angular acceleration in table 3.3. Figure 3.3 shows the pairwise distribution of the dimensionless variables.

What we observe here, is that the dimensionless target variable support is only marginally impacted by the pole length shift. A consequence of the collapse of supports highlighted in [Oppenheimer et al., 2023] is that models do not need to extrapolate beyond training data.

We now study the impact of distribution shift on our 3 regression models. Each of them is

Pole length support	$\lvert\ddot{\theta}\rvert_{max}$	$\lvert\Pi_{\ddot{\theta}}\rvert_{max}$
$L = 1m$	11.3	1.2
$L \in [0.75, 1.25]m$	14.8	1.2
$L \in [0.5, 1.5]m$	22.1	1.2

Table 3.3: Relation between pole length support and magnitude of angular acceleration in natural space (middle column) and dimensionless one (right column).

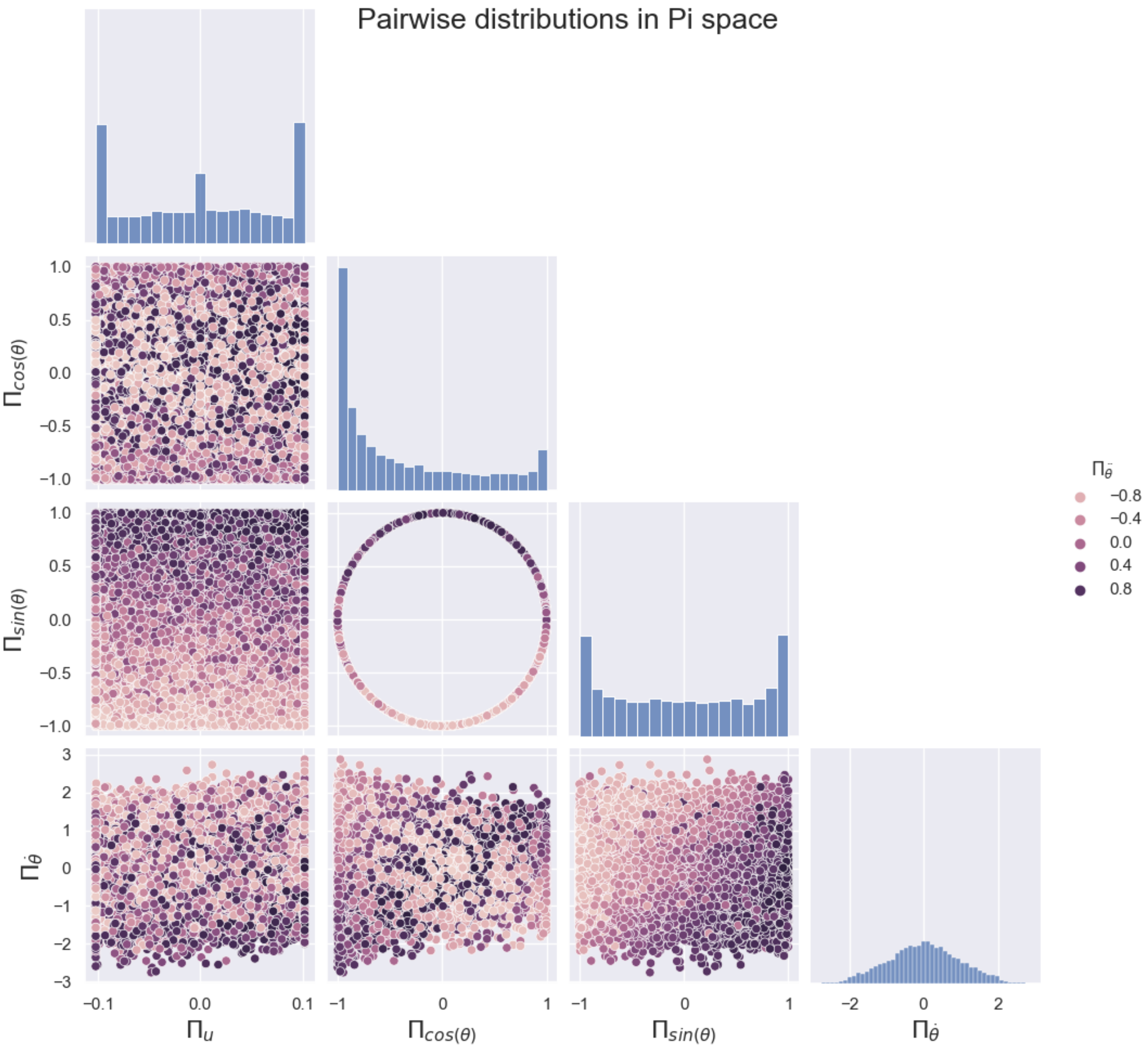


Figure 3.3: Pairplot of state variables dimensionless space

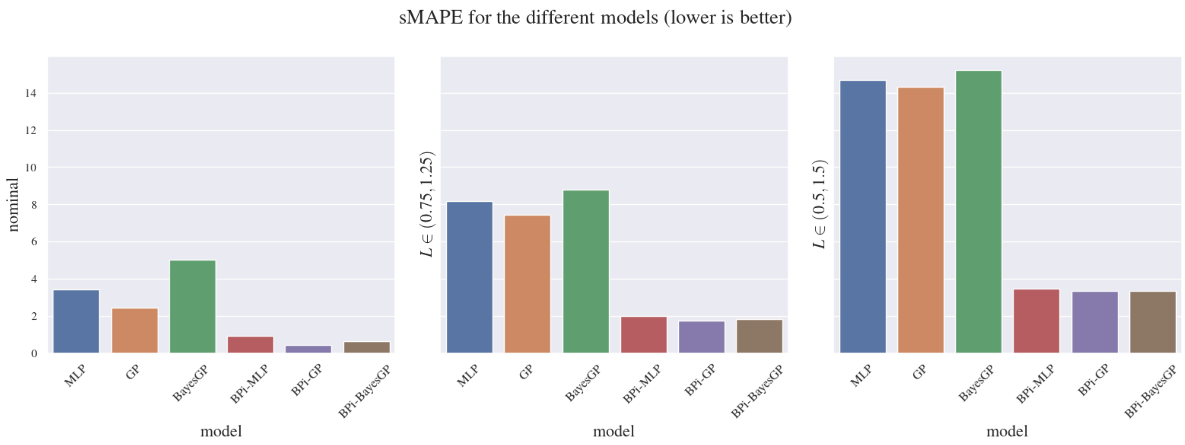


Figure 3.4: sMAPE score of the different models on training and testing datasets. The first 3 bars represent the models trained on the natural space features and the last 3 on the dimensionless space. A lower sMAPE indicates a lower prediction error.

trained on the same training dataset with a nominal pole length of 1. The training performance, measured with sMAPE (3.16) is displayed on the left plot of figure 3.4. While the Bayesian GP model shows slightly worse training performance, the MLP and MAP-GP are able to fit the data as well on both natural and dimensionless space.

However, when the pole length distribution shifts (middle and right plot on figure 3.16), the performance drops significantly more for the models trained in the natural space. This trend is also confirmed when we plot the model predictions against their true values on figure 3.5. This graph also shows that the models are not able to predict values outside the training target support, causing an underestimation of the large absolute values of $\ddot{\theta}$.

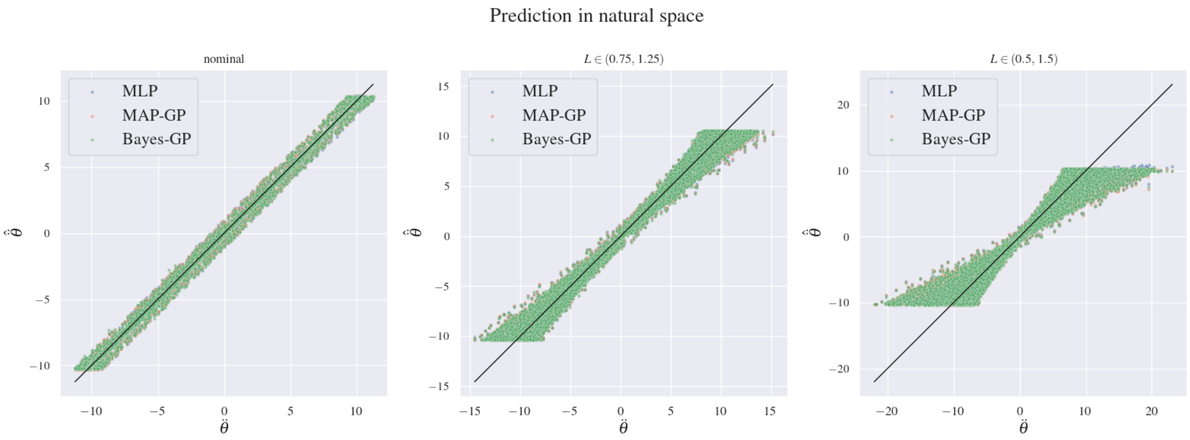


Figure 3.5: Predictions against true values for models trained using natural features. The y-axis is for predictions and the x-axis for true values. Perfect predictions would yield points exactly on the identity line plotted in black. We can see that all models stuggle to make predicitons outside of the training support.

On the other hand, the models trained with dimensionless features are able to predict targets much more accurately in the presence of distribution shift. We can see on figure 3.6 that the predictions in middle and right plot do not suffer from the same bias as on figure 3.5. Looking at the middle and right plots on figure 3.6, we observe that the predictions have more variance than on the left. Nevertheless, the predictions do not seem to suffer from the same bias at the extreme values of the domain of $\ddot{\theta}$. This suggests that the benefits of non-dimensionalization can not be explained only through the lens of the reduction of target distribution support. Rather, that the transformation leads to an invariant predictor across pole length changes.

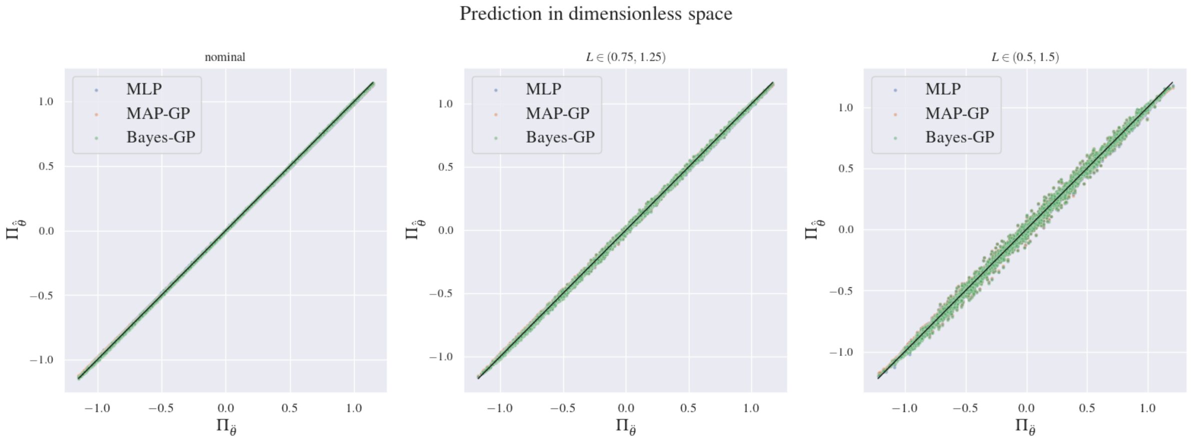


Figure 3.6: Predictions against true values for models trained in dimensionless space. While the test predictions (middle and right plot) get slightly worse than in the nominal context (left), they remain close to accurate within all the support of the points.

In addition, we also plot the generalization gap (2.8) on figure 3.7 for models on both natural and dimensionless spaces. This metric measures the performance drop between training and testing distributions. It confirms the findings from above that non-dimensionalization improves generalization for all 3 tested models. We note that the Bayesian GP model's better generalization is partly due to its worse nominal performance, meaning it is a more conservative estimator than the other two.

While we only consider a shift in the pole length, we saw it is enough to incur a significant modification of the latent dynamics. That shift is enough to decrease the performance of the models trained on the dimensional space.

3.3.2 Influence of Uncertainty on Context Variables

In the previous section, we assumed perfect knowledge of the physical variables. Here, we study the impact of uncertainty in those variables. Namely, how much it affects model training and weather dimensionless models trained under high uncertainty are still more robust than the

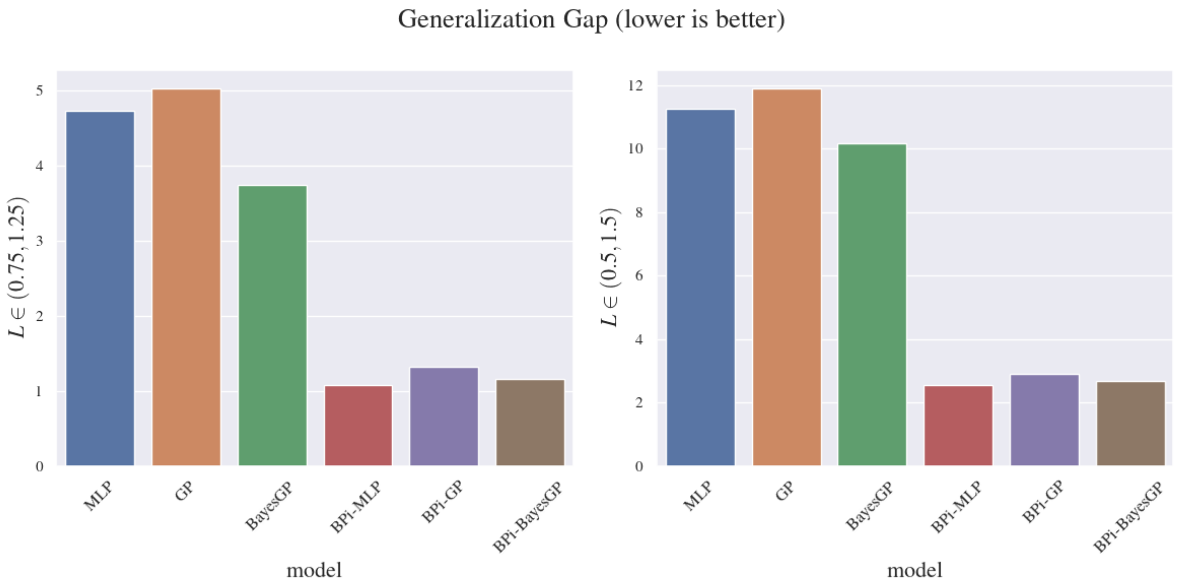


Figure 3.7: Generalization gap for the natural and dimensionless models on the pendulum. In both cases the models trained on features given by the Π groups show better generalization performance by a factor of 4 to 5.

natural ones. Because the feature map (3.13) depends on those variables, we expect the in-distribution performance of the models trained on them to drop. This decreased performance however, may not be so significant that it prevents the models to be robust to context drift.

Questions:

- How much does noise in context variables affect model predictions?
- Can models trained on noisy dimensionless data still generalize to new contexts?

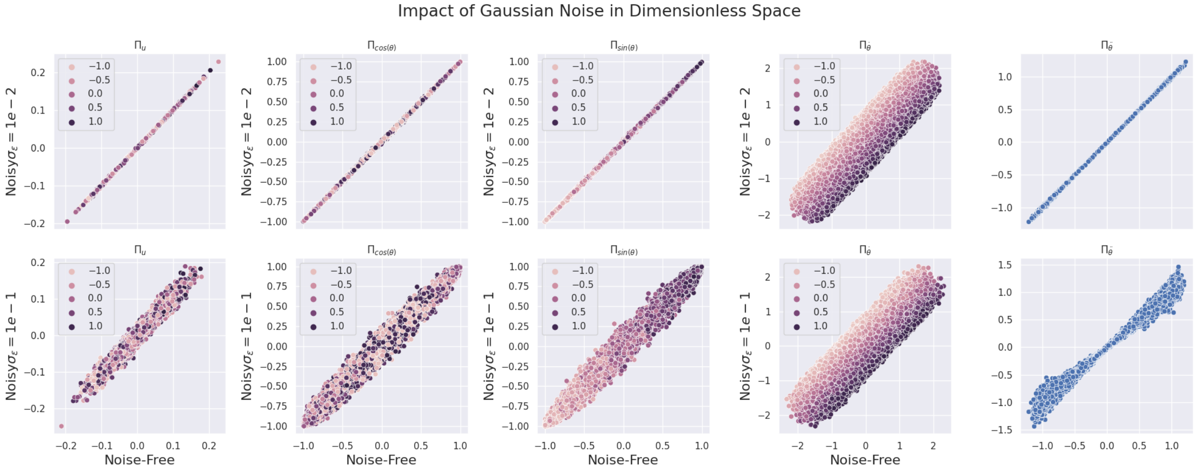


Figure 3.8: Plot of noisy against noise-free dimensionless variables. The rightmost plot corresponds to the target $\Pi_{\ddot{\theta}}$. The top plot corresponds to $\sigma = 0.01$, the bottom plot $\sigma = 0.1$. Because of the power-law transformation, the uncertainty is not homogeneous across the samples.

As a preliminary step, we aim to visualize the impact of uncertainty on the dimensionless variables compared to if we had exact knowledge of them. We plot the impact of uncertainty on dimensionless features in figure 3.8. Because θ is already dimensionless, it is not impacted by uncertainty in M, g, L hence we see a straight line. On the other dimensionless variables though, because of compounding effect of uncertainty, they appear as if corrupted by heteroskedastic noise. Therefore, without any further prior assumption on the model and with noisy observations of the physical variables, we expect the dimensionless models to perform worse than their natural counterparts.

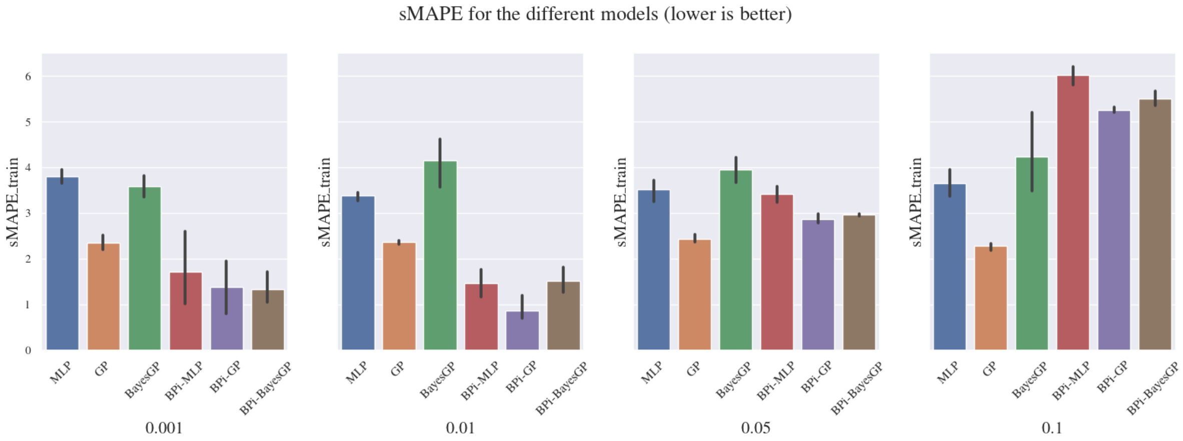


Figure 3.9: Training metric on the natural and dimensionless spaces for increasing uncertainty. The models in natural space are not affected by uncertainty associated with the state and velocity variables.

Here, we repeat the experiments from the previous section where we corrupt the measurements of M, g, L with Gaussian noise. Figure 3.10 and 3.9 show the impact of uncertainty on training performance with increasing standard deviations. We display the score for the MLP and GP trained in natural space for comparison (solid lines), but since they do not observe the noisy static variables, the variation in performance is only due to variability of the optimization and data-generation procedures. The dashed lines represent the models trained in dimensionless space. We can see that their performance does not suffer up to standard deviations of 0.01. Above this level, the model fit becomes increasingly bad. That being said, we are still interested in measuring the robustness of dimensionless models with uncertain physical variables.

Figure 3.11 summarizes the findings of this chapter. The leftmost bar plots show the training performance, with increasing uncertainty going from top to bottom. The middle and right columns report the test performance for increasing pole length shifts. Interestingly, while the in-sample performance of dimensionless models for $\sigma = 0.05$ and $\sigma = 0.1$ is significantly worse than their natural equivalents, they are still more robust to a shift in the pole length distribution.

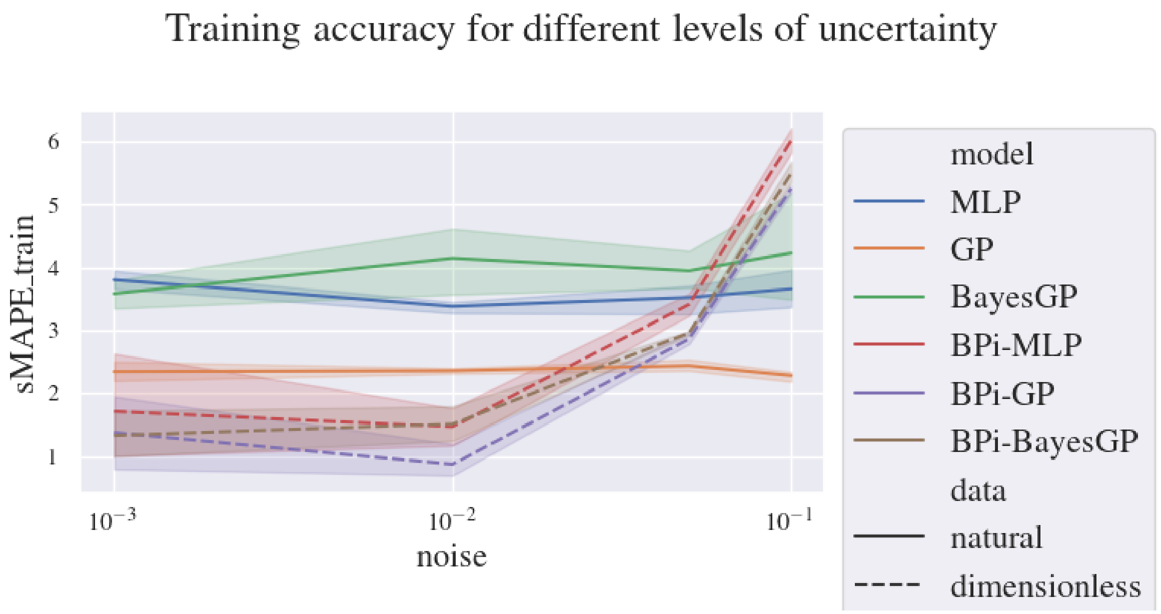


Figure 3.10: Training metric on the natural and dimensionless spaces for increasing values of uncertainty on the pendulum. Shaded areas represent 95% confidence intervals over 5 random seeds. High uncertainty levels above 0.05 cause higher prediction errors on dimensionless models.

This finding is even more striking looking at the corresponding generalization gaps on figure 3.12. We can see the dimensionless models generalize much better even when the physical parameters are uncertain.

Hyperparameter	Value (pendulum)
Number of hidden layers	2
Units per layer	32
Activation	ReLU
Optimization	Adam
Learning Rate	0.01
Batch Size	500
Training Epochs	200

Table 3.4: Hyperparameters for Multi-Layer Perceptron.

Model Hyperparameters for the experiments

Hyperparameters for training MLP are summarized in table 3.4 and GP in table 3.5.

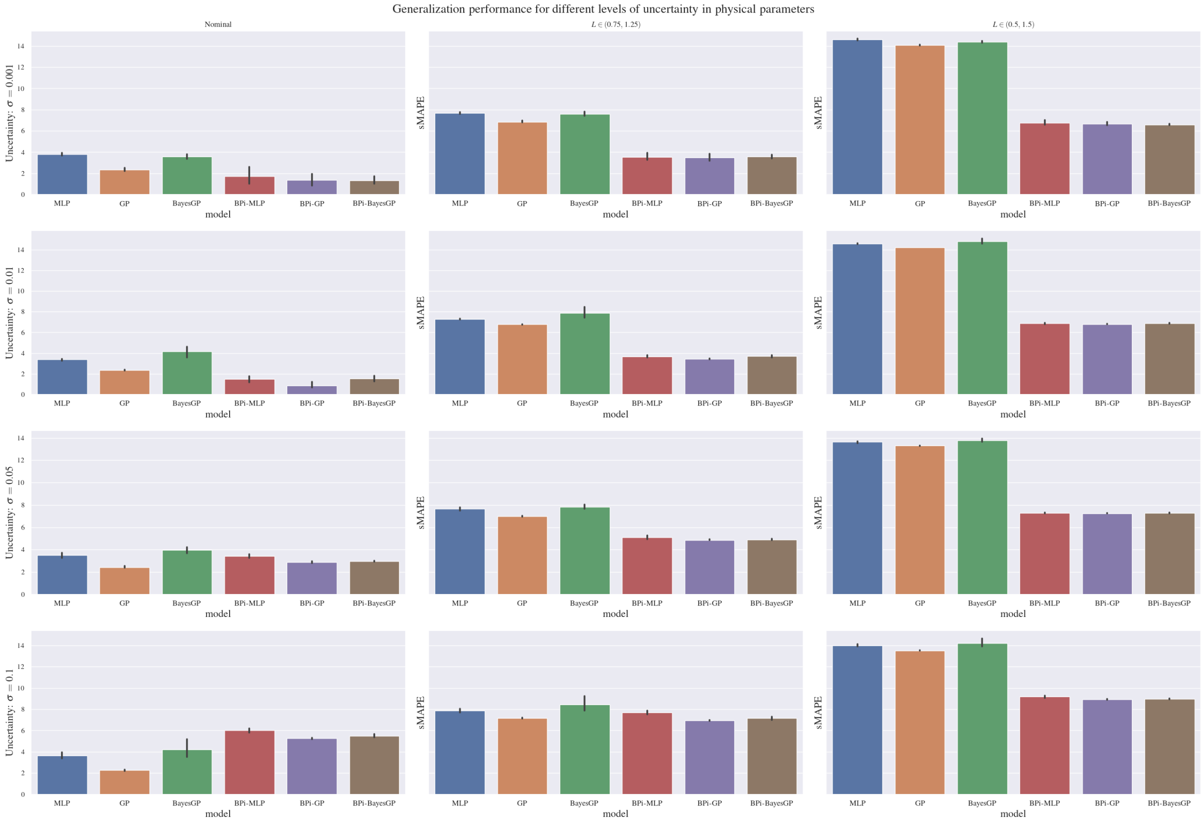


Figure 3.11: sMAPE metric with noise and context shift. From left to right the pole length is shifting. From top to bottom the uncertainty of pole length measurement increases. In the most extreme context shift (right column), even the models trained and evaluated with uncertainty predict better than their natural counterparts.

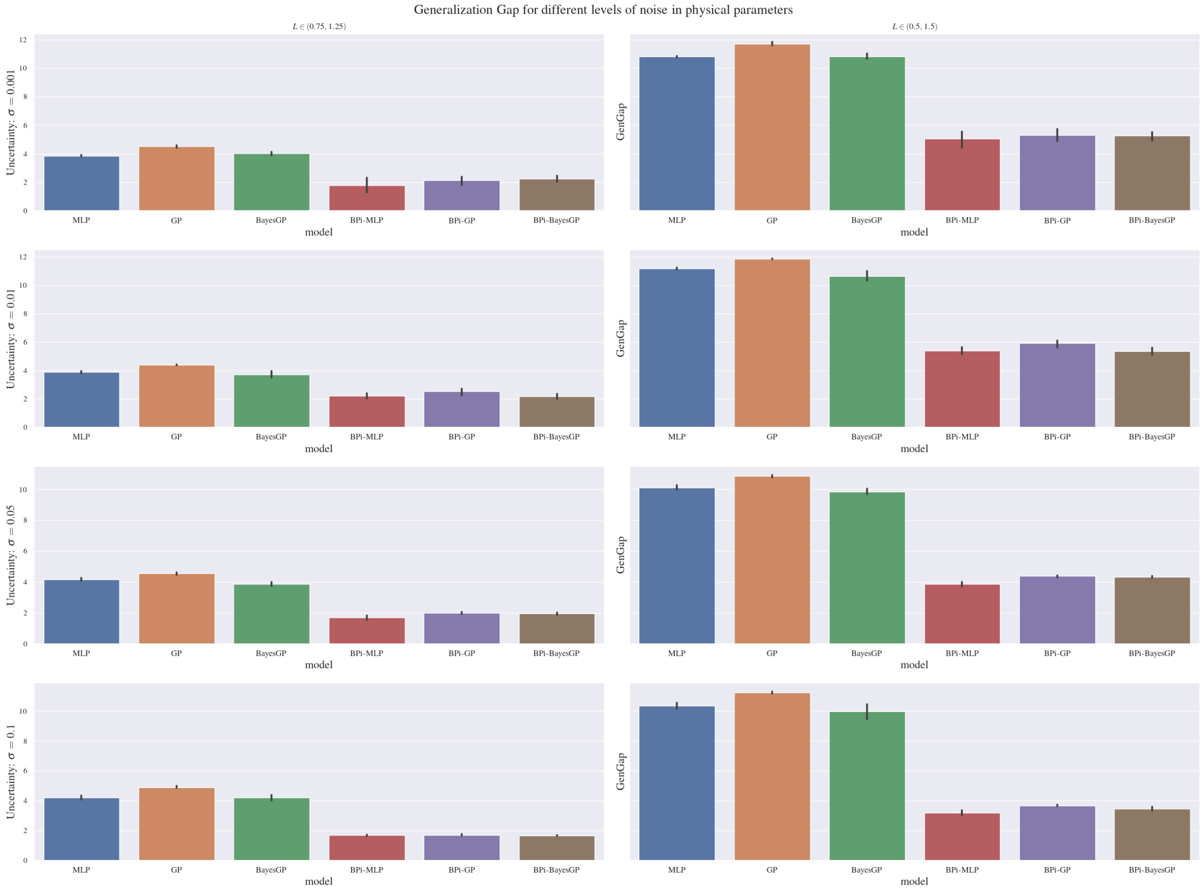


Figure 3.12: Generalization gap with noise and context shift. From left to right the pole length is shifting. From top to bottom the uncertainty of pole length measurement increases. In all cases, the dimensionless models' generalization error is lower than natural ones.

3.4 Conclusion

The Buckingham-Pi theorem is more than a century old and has only been recently applied to machine learning algorithms. From the theorem follows a method for non-dimensionalizing physical state variables. More than a dimensionality reduction technique, it allows building an equivariant feature map by exploiting knowledge from the variables dimensions without knowledge of the exact equation governing the system. The invariance of the feature map allows models to generalize beyond the support of the training data.

Going further than previous work, we investigated the sensitivity of that transformation to uncertainty associated with the state variables. We demonstrated the benefits of non-dimensionalization on the actuated pendulum dynamics with MultiLayer Perceptron and Gaussian Process models on simulated data. The first benefit of the transformation is the reduction of the size of the test distribution support (figure 3.2, table 3.3). The second one is the invariance of the target prediction with respect to the context change. When we use it to make predictions, the models in dimensionless space generalize better than those trained on natural features (figure 3.4). The presence of noise in those measurements has a negative impact on the models training accuracy as shown on figure 3.10. It does not prevent the models from generalizing better than the same models trained in the natural space, meaning the presence of uncertainty associated with the context does not impede good model generalization (figure 3.11, 3.12). However, compared to state-space models, it requires additional measurement of the static parameters of the dynamic system. In the next chapter, we address this shortcoming and study how to alleviate the need for observing the static variables.

Hyperparameter	MAP-GP Value (pendulum)	Bayesian GP (pendulum)
Inducing Points	100	100
Kernel	RBF	RBF
Optimization	Adam	Adam
Learning Rate	5.10^{-3}	5.10^{-3}
Lengthscale Prior	LogNormal(0, 1)	LogNormal(0, 1)
Variance Prior	LogNormal(0, 1)	LogNormal(0, 1)
Noise Prior	LogNormal(−1, 1)	LogNormal(−1, 1)
Batch Size	1000	1000
Training Epochs	400	500

Table 3.5: Hyperparameters for Gaussian Processeses.

Chapter 4.

Dimensionless Latent Variable Inference

Dimensionless feature spaces produced with Buckingham-Pi theory lead to robust estimators, but require access to observation of additional variables. Any change in these variables will incur a shift in the system dynamics to which the dimensionless space will be invariant. On the other hand, if such context variables are hidden one can not know when the distribution shift occurs neither how to transform the space to reflect those changes. In this chapter, we propose to tackle the problem of partial observations with the aim of building robust estimators. We train a model with data generated from a single environment and then evaluate the ability of the model to detect a shift so as to quickly adapt to best fit a new one. We propose to tackle that problem by augmenting the model with latent variables that are assigned the physical dimensions of the unobserved parameters. When the variables are observed during training,the model is able to re-estimate them accurately after they have changed. If they are hidden during training as well, we found that the learned latent variables correlate strongly with the true parameters and can be used to make accurate predictions with little retraining.

4.1 Motivation

In the course of their deployment, dynamic systems may be subjected to perturbations that will modify their intrinsic properties and thus their observable behaviour. Such perturbations may be slow and gradual in the case of erosion or sudden and brutal because of discrete events. Because the models of such systems are designed and trained on a specific context or set of contexts, they are likely to enter modes of failures after such perturbations. Provided perfect knowledge of the parameters subjected to these changes, dimensionless embeddings provide estimators that are robust to these perturbations, even if they have not been encountered during training.

Let us consider the case of an autonomous driving vehicle. A model of how the vehicle reacts to steering, acceleration and brakes is trained from realistic simulations and then fine-tuned on real data. This model could be built upon features that account for the car's physical parameters like friction between the road and the tires. This specific parameter is likely to decrease after prolonged usage, as the tires become more and more smooth. Such change is even more abrupt and dramatic when climate conditions change, if there is heavy rain for instance. The difficulty lies in that measuring this exact coefficient requires precise interventional experiments, which are not feasible while the vehicle is driving. As such, it is crucial to be able to detect when such parameters shift occur and estimate their values such that the model of the system best fits its new environmental configuration.

Usual approaches to tackle that problem rely on meta learning. They require training a model a several different versions of the environment to allow the model to separate local and global task properties. When access to a simulator is available, generating training data from various contexts can easily be done. This however is not possible if the context variables are fixed and can not be intervened upon. For training a robotic arm for instance, one can not change the length of each component or the friction between joints because they are physically fixed by construction. In such cases, all the training data is conditioned on a single nominal context, given by the robot properties. If that context is known, meaning we can measure the physical variables on which it depends, then we can leverage that information in a semi-supervised approach we describe in section 4.2.2. If, however, we have no knowledge of the context at all, we propose a latent variable model that can estimate it in section 4.2.3. Our work develops a similar model as [Sæmundsson et al., 2018] but with a key difference. While we also train a Gaussian Process model with latent variables, we constrain them to a specific dimension imposed with the Buckingham theorem. Because that transformation is equivariant, we do not need several versions of the environment for training.

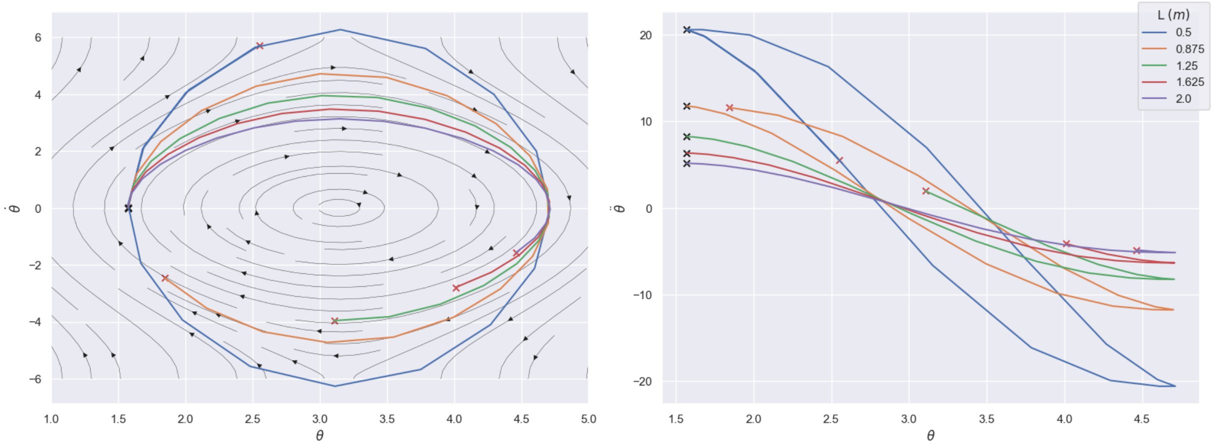


Figure 4.1: Phase plane of the pendulum for different L values. The initial and final point of the trajectories are represented by black and red crosses respectively

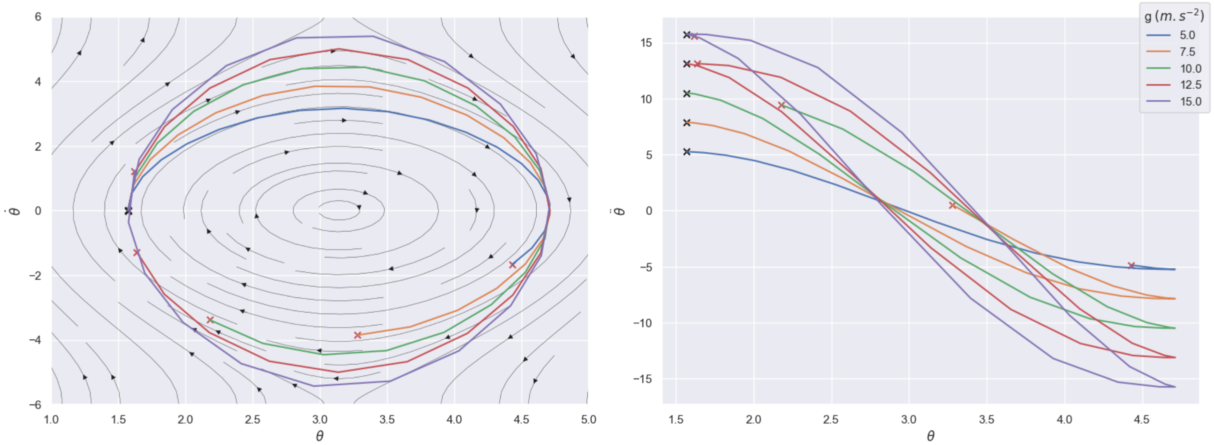


Figure 4.2: Phase plane of the pendulum for different values of g.

4.2 Methods and Inference

4.2.1 Formulation of the problem

Let us consider a general second-order dynamic system

$$\ddot{\boldsymbol{x}} = f(\boldsymbol{x}; \boldsymbol{c}), \tag{4.1}$$

where

- $\boldsymbol{x} \in \mathbb{R}^d$ are the positions and velocities of the state variables,
- $\boldsymbol{c} \in \mathbb{R}^k$ are static context variables.

In the subsequent pendulum example, $\boldsymbol{x}$ is the angle position and velocity and $\boldsymbol{c}$ comprises the pendulum length, mass and gravity field magnitude. The context variables are hidden in usual state-space models and model parameters are inferred only from the dynamic ones. However, when the context variables change they can significantly impact the shape of the dynamics and produce a shift in the trajectories sampling distribution as is illustrated on figures 4.1 and 4.2.

As we saw in the previous chapter, we can create a dimensionless feature mapping of the dynamic variables with the Π-groups 2.22, so that a model of equation 4.1 is invariant to the context. Considering some of the context variables are not observed, we propose to estimate their distribution which we will then use for transforming the feature space.

Formally, we assume an emission function o that hides some elements of the context as

$$o(\boldsymbol{c}) = \boldsymbol{c}_o \in \mathbb{R}^{k-h}, \tag{4.2}$$

where h is the number of hidden dimensions of the context. Conversely, we write $\boldsymbol{c}_h \in \mathbb{R}^h$ the missing hidden variables that are discarded by the emission function. We augment the generative model with a set of latent variables $\boldsymbol{H} \in \mathbb{R}^h$, each element of which is assigned the dimension of a missing physical parameter such that,

$$\forall j \in 1 \dots h, \left[h_j\right] = \left[c_{k-j}\right]. \tag{4.3}$$

Assigning a dimension to the latent variable now allows us to use the Buckingham-Pi theorem again to transform the state variables into dimensionless ones using

$$\Pi = \phi_\Pi(\ddot{\boldsymbol{x}}, \boldsymbol{x}, \boldsymbol{c}_o, \boldsymbol{h}), \tag{4.4}$$

where ϕ_Π is the dimensionless feature map, this time applied to the concatenation of observables and latent variables.

Comparison with Meta-Learning

Before we go into the details of inference in that model, we point out the main difference of that approach compared to other meta-learning approaches. Because we are creating an invariant model of the observed context and latent, we do not need to observe several versions of the environment with domain randomization procedures. This would be redundant to our model since the model is by construction, invariant with respect to the domain context. As a consequence, the latent variables will naturally allow the model to separate the global from the local properties of the task at hand.

4.2.2 Semi-Supervised Approach

We first place ourselves in the setting where the full state of the system can be observed during the initial training phase. This corresponds, for example, to a case where one has access to a simulator. This means we are able to use all the dependent variables to construct a dimensionless feature space. We can therefore train a model in this space, that will be invariant to the context variables that are susceptible to change in the course of the system deployment. Once a shift as occurred, we can then use Monte Carlo sampling to infer the posterior distribution over the context.

Training Model on Fully Observed Data

The training data consists in N samples from trajectories generated by the system from equation 4.1. The observations y may be corrupted by Gaussian noise, hence the data we have is

$$y_i = f(\boldsymbol{x}; \boldsymbol{c}) + \epsilon,\ i = 1 \dots N,\ \epsilon \sim \mathcal{N}(0, \sigma_n). \tag{4.5}$$

Using the Buckingham theorem, we perform the regression task in the dimensionless space using the feature map ϕ_Π,

$$\Pi_{y_i} = f_\Pi(\phi_\Pi(\boldsymbol{x}, \boldsymbol{c}) + \epsilon,\ i = 1 \dots N. \tag{4.6}$$

We place a Gaussian Process prior on the latent f_Π and train it with maximum likelihood estimation.

Adaptation

So far, we have remained in the same training setting as in the previous chapter. We now aim to use that model, to make predictions when some of the context is only partially observed as per equation 4.2. Because we use a probabilistic model (Gaussian Process), we can detect when a distribution shift occurs using newly observed trajectories. Suppose we observe one or several new trajectories $\tau = \{X_{test}, y_{test}\}$, their likelihood under the model will indicate whether substantial changes have occurred in the hidden variables. If

$$\log p(y_{test} | \hat{f}, X_{test}) < \alpha \log p(y_{train} | \hat{f}, X_{train}), \tag{4.7}$$

for a given threshold α (say 0.95 for instance), then we can start the adaptation procedure. It consists in placing a prior on the latent variables $p(H)$ and assign a physical dimension to each of its elements following equation 4.3. We then use Monte Carlo sampling for estimating the posterior of the latent given the model and new data as

$$p(H | \hat{f}, \tau) \sim p(\tau | \hat{f}, H) p(H). \tag{4.8}$$

The posterior can then be used to make accurate probabilistic predictions about the new data by computing

$$p(y_{test} | \hat{f}, X_{test}, H) \sim \hat{f}\left(.|\phi_\Pi(X_{test}, H)\right). \tag{4.9}$$

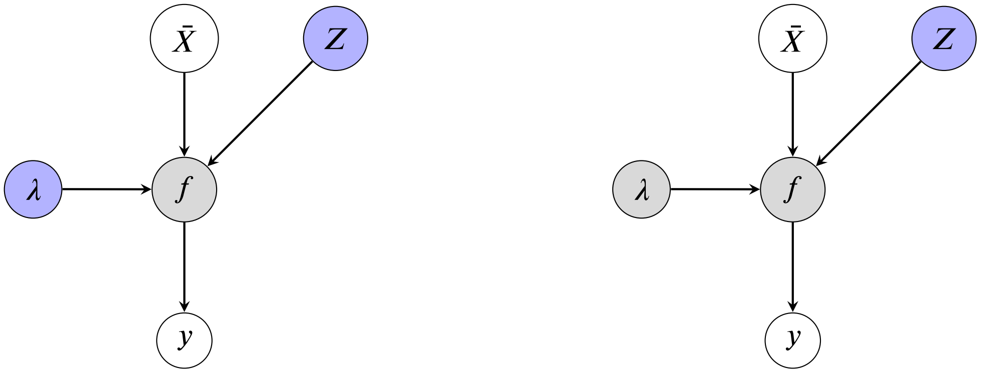


Figure 4.3: Latent Variable model during training (left) and adaptation (right). The blue nodes indicate the free parameters.

4.2.3 Latent Variable Model

Invariant Meta-Learning Model

The approach we present here is a form of meta-learning. This form of learning consists of learning a hierarchical model to disentangle the local and global properties of the learning task. It usually functions by exposing the learning model to data coming from various different versions of the same environment. A standard supervised learning model will have difficulty to learn a single model for all of them.

For simplicity, let us write the full observations $\bar{\boldsymbol{x}} \in \mathbb{R}^{d+k-h}$. We augment the observations with a vector of latent variables $\boldsymbol{h} \in \mathbb{R}^{h}$. Each dimension of the latent vector will represent a physical variable that has been lost by the emission function following equation 4.3. Therefore, we can use the Buckingham-Pi transformation to create an invariant embedding, combining the observed and latent variables. The new features will be $\boldsymbol{x}_{\Pi} = f_{\Pi}(\tilde{\boldsymbol{x}})$, where $\tilde{\boldsymbol{x}}$ represents the concatenation of $\bar{\boldsymbol{x}}$ and $\boldsymbol{h}$, and ϕ_{Π} is from equation 3.13.

We place a Gaussian Process prior on the new features for estimating the latent function f,

$$\boldsymbol{f} \sim \mathcal{GP}(\phi_{\Pi}(\bar{\boldsymbol{x}})). \tag{4.10}$$

We assume Gaussian noise, so the likelihood given functionals $\boldsymbol{f}$ writes as

$$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{y}|\boldsymbol{f}(\phi_{\Pi}(\bar{\boldsymbol{x}}), \boldsymbol{\Sigma})), \tag{4.11}$$

and prior on latent variable is a standard multivariate normal prior $\boldsymbol{h} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.

The joint distribution of the model is

$$p(\boldsymbol{y}, \boldsymbol{f}, \boldsymbol{h}) = p\left(\boldsymbol{y} | \boldsymbol{f}(\phi_{\Pi}(\bar{\boldsymbol{x}}\right), \boldsymbol{\Sigma}) p(\boldsymbol{f}) p(\boldsymbol{h}). \tag{4.12}$$

This generative model closely resembles that of [Sæmundsson et al., 2018], except we add another step for transforming the physical variables into dimensionless ones.

Inference

Because Gaussian Processes do not scale well with the number of samples, we use the variational sparse variant described in section 2.3.4. In order to include the estimation of the latent H into the inference procedure with a single optimization objective, we place a variational distribution on them. For simplicity, we assume a Gaussian distribution

$$q(H) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}_H), \tag{4.13}$$

where $\boldsymbol{\phi} = (\boldsymbol{\mu}, \boldsymbol{\Sigma}_H)$ are the variational parameters. In practice, we found that a diagonal covariance was enough to approximate the posterior of the latent variables, which corresponds to the mean-field approximation.

The set of parameters $\boldsymbol{\theta} = (\Lambda, \boldsymbol{\phi}, \Sigma)$ includes the GP, likelihood and latent variational parameters. They are optimized using the Evidence Lower Bound (ELBO),

$$ELBO(\boldsymbol{\theta}) = \mathbb{E}_{q(\boldsymbol{f} | \phi_{\Pi}(\bar{\boldsymbol{x}})}\left[\log p(\boldsymbol{y} | \boldsymbol{f})\right] - D_{\mathrm{KL}}(q(\boldsymbol{h}) || p(\boldsymbol{h})) - D_{\mathrm{KL}}(q(U) || p(U)). \tag{4.14}$$

It is worth noting at this point, the difference of this approach in comparison to other meta-learning approaches. We learn a single variational posterior of H, since we are focused on the zero-shot transfer. During evaluation, when facing a new task the variational parameters of the latent will be relearned whilst retaining the model parameters (Λ, Σ) to a fixed value. The pseudo-code for training this model is depicted in algorithm 1.

Algorithm 1 Dimensionless Latent Variable Regression

```
1: Input model $\hat{f} = \mathcal{GP}$, observations $\bar{X}$, output $y$, prior $p(Z)$, variational family $q_\phi(Z)$
2: for $i = 1, \dots, n_{epochs}$ do
3:     $\boldsymbol{h} \sim q_\phi(\boldsymbol{h})$
4:     $\boldsymbol{X} = [\bar{\boldsymbol{x}}, \boldsymbol{h}]$        ▷ concatenate latent and observations
5:     $\boldsymbol{X}_\Pi = \phi_\Pi(\boldsymbol{X})$        ▷ Buckingham-Pi transform
6:     $\hat{\boldsymbol{y}}_\Pi \sim \hat{f}(\boldsymbol{X}_\Pi)$
7:     $\hat{\boldsymbol{y}} = \phi_\Pi^{-1}(\hat{\boldsymbol{y}}_\Pi)$        ▷ Transform dimensionless prediction into natural space
8:     compute $ELBO(\boldsymbol{\theta})$
9:     $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \times \nabla_\theta(ELBO)$        ▷ Gradient step
   return $\hat{\boldsymbol{f}}, q_{\boldsymbol{\phi}}$
```

We now highlight an important point for computing the ELBO (4.14) on algorithm 1[line 8]. It is essential that the likelihood is computed with the true targets and so the dimensionless predictions $\hat{y}_\Pi$ must be transformed back into natural space. If this step is ignored, it could be possible to compare the the predictions in dimensionless space, but in that case the gradients of the cost are no longer unbiased estimates of the ELBO. Therefore, the condition for stochastic gradient descent to converge is no longer met and the algorithm will not be able to converge and find suitable parameters. In practice, we use a different set of learning rates for the model and the latent variables.

When the model is exposed to a new task, the procedure is repeated, but we only optimize the parameters $\boldsymbol{\phi}$ of the latent variable variational distribution. We summarize the inference procedure for training and adaptation on the graphics of figure 4.3.

4.3 Experiments

In the following, we aim to give empirical answers for the following questions:

1. Are the models augmented with latent variables able to make accurate predictions when confronted to data from a different context?
2. Are the latent variables physically meaningful?

The first question aims to answer about the few-shot transfer abilities of the models. Meaning adapting the model with little data such that predictions on a new task are accurate. The second

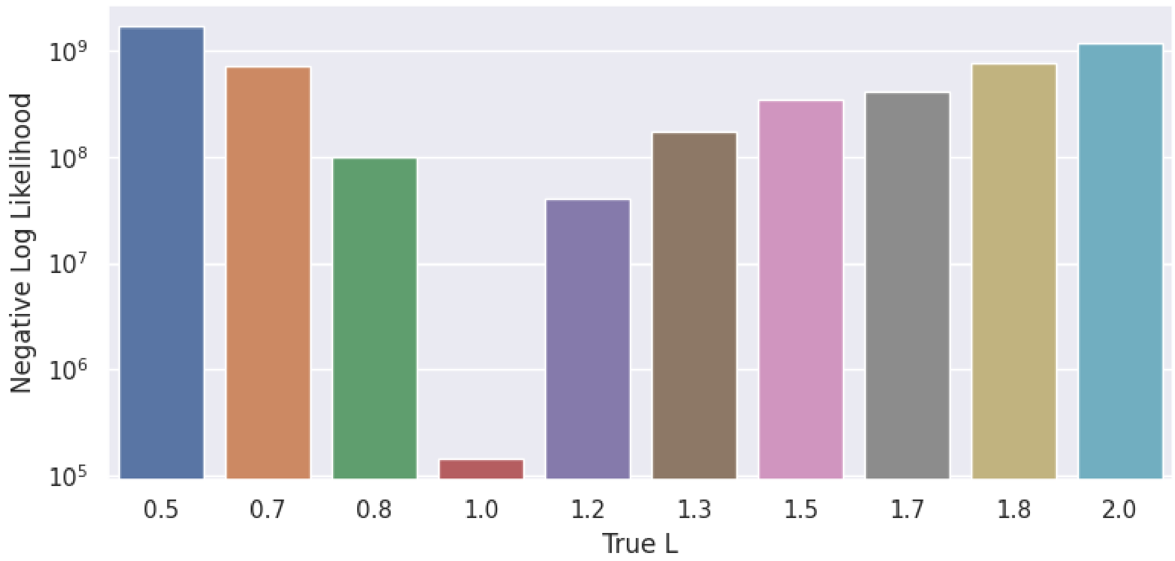


Figure 4.4: Evaluation of the model on several datasets assuming $L = 1$. The high NLL values indicate that a shift in the underlying dynamics has occured.

one aims to bring up whether the information carried by the latent space can be used to estimate what the true hidden parameters are, given they are observed during training or not.

The experiments are performed on the simple actuated pendulum and the model we use for predictions is a Gaussian Process with type-II marginal likelihood inference. For all these experiments we used Pyro [Bingham et al., 2019], a high-level probabilistic language built on top of Pytorch which benefits from GPU acceleration.

4.3.1 Semi-Supervised

This section relates experiments with the model that is trained on full observation of both dynamic and static variables. We collect 5000 training samples that are generated by 2s trajectories with random initial points, Gaussian actions and 10Hz sampling frequency. The nominal pole length is $L = 1$m. The control signal is a random walk, where each step is sampled from a zero-mean Gaussian with 0.25 standard deviation. We used this control strategy instead of Gaussian or uniform sampling at each step. The latter approaches made identification of the parameters more difficult because the average control signal on each trajectory was zero.

After training a model $\hat{f}$ on complete observations, we evaluate it in the case where the pole length is hidden.We place a prior $p(H_L)$ and can sample from it to estimate the posterior distribution of the true hidden pole length.

On figure 4.4, we demonstrate how the model is used to detect that a change in context has occurred. We evaluate the negative log-likelihood of the model on each dataset assuming the pole length is 1. As a consequence, the model fit is very bad for the trajectories that were collected with different values. Facing with such a significant drop, we now show how to estimate what the actual value is.

On figure 4.5, we see the result of predicting a trajectory given several samples of the latent variable. Each of these samples will correspond to specific dynamics. Therefore, are looking to infer the posterior distribution over that variable that best fit to new trajectory data.

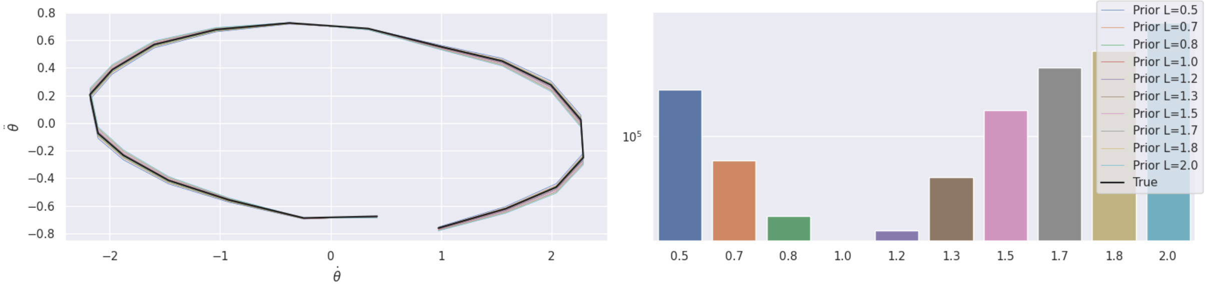


Figure 4.5: This shows the output of the model given different samples of pole length. The left plot shows a phase plane of the different models, with the black line representing true data, with $L = 1m$. The right plot displays Negative log-likelihood for each sample.

If we repeat this procedure on datasets conditioned by different pole length values, we can understand better how we may be able to estimate the ground truth. On figure 4.6, we collect 10 trajectories from 5 different pole length values. We then place a standard log-normal prior on H_L and compute the likelihood of the data under the model and each prior sample. We can see the regions highest likelihoods (lowest negative log-likelihood on the graph), are located close to where the true parameter is.

We use Monte Carlo simulation do estimate the posterior distribution of the latent variable because having trained the model already, the dimensionality of the inferred is h or in this case 1 because only the pole length is hidden. Specifically, we ran the No U-Turn Sampler (NUTS) [Homan and Gelman, 2014], a variant of Hamiltonian Monte Carlo method that requires fewer user-defined hyperparameters. We found that letter the algorithm sample 1600 draws (and as many tuning steps) over 4 chains was sufficient to obtain good posterior distributions of the variable.

On figure 4.7, we plot the posterior of the latent variable estimated with Monte Carlo sampling against the ground truth. Even with as little as 5 trajectories (100 samples), we are able to recover a good estimate of the true value of L. On figure 4.8, we plot the posterior distributions for different values L given increasing number of observed trajectories. Surprisingly, it shows

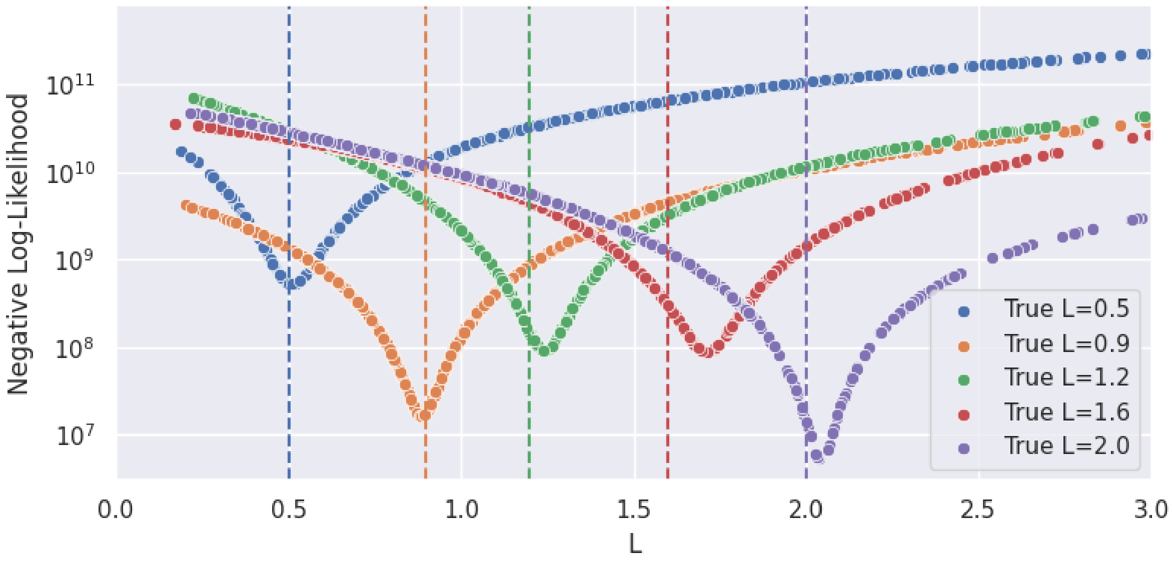


Figure 4.6: For each trajectory, we sample values from a uniform prior in $[0.1, 3]m$ interval and evaluate the likelihood of each sample under the model. Lower value means a better model fit.

the quality of the posterior does not depend on the number of observed trajectories. It seems instead, the initial position of the trajectories as well as the control sequence has an important impact on identification. As we mentioned before, uniform and Gaussian actions tend to cancel each other between successive steps, thus preventing the model to identify the hidden variable correctly.

As we can see, the model is able to quickly adapt even when the number of samples available is small. We repeated this approach with missing parameters g and M to validate the approach and showed the identification of the value was successful as well.

4.3.2 Latent Variable Model

In this section, we evaluate the model described in section 4.2.3. Contrary to the previous section, the context is not observed during training at all. As a consequence, we aim to learn both the model and latent variable during training. For testing a new environment, we will estimate a new distribution over H while retaining the rest of the model.

We train the following hierarchical model

$$f \sim \mathcal{GP}(\mathbf{0}, k_{RBF}(.,.)) \tag{4.15}$$

$$\log(Z) \sim \mathcal{N}(0, 1) \tag{4.16}$$

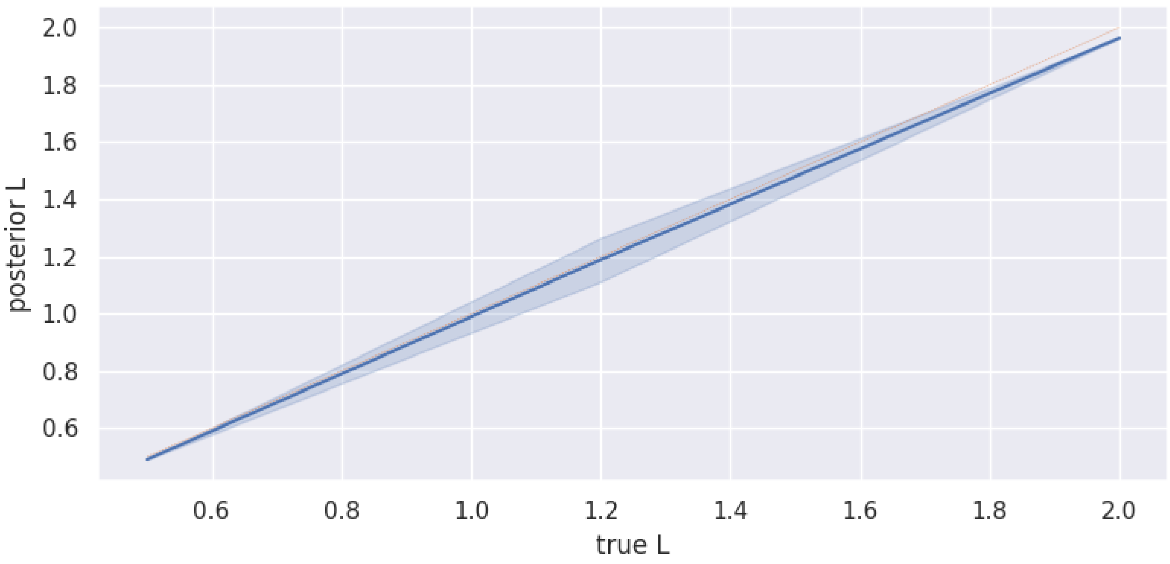


Figure 4.7: Plot of the posterior latent L against the true value which generated the data. Each test dataset consist of 5 trajectories. The orange dashed-line represents a perfect fit of the pole length value. The uncertainty associated with the prediction of the latent is low on the whole range of tested trajectories.

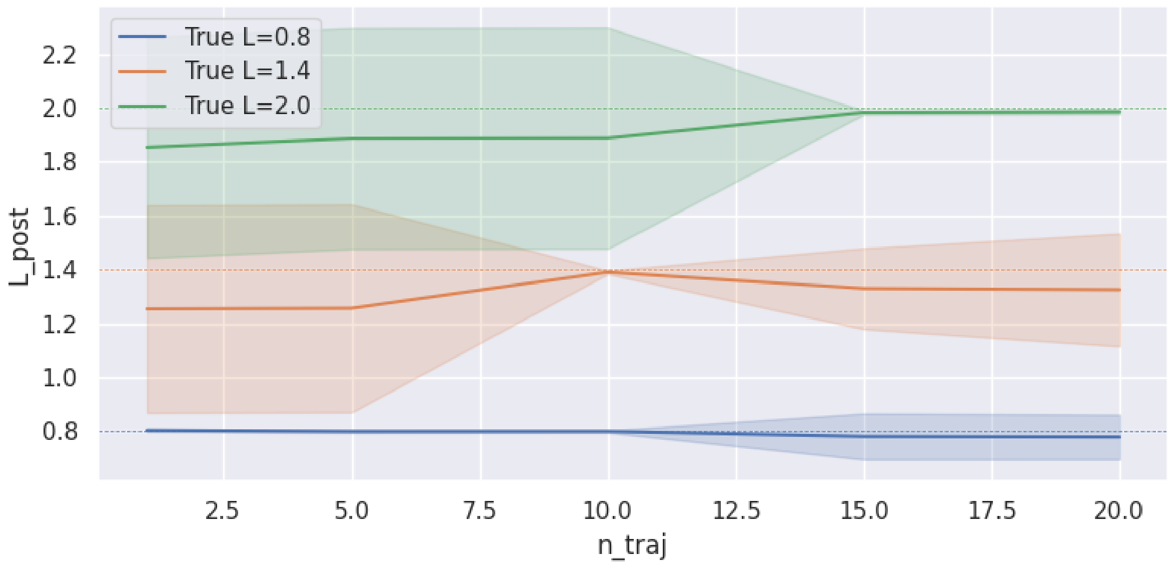


Figure 4.8: Posterior of the latent L against the ground truth (horizontal line) as a function of number of observed trajectories. The shaded area represents the standard deviation around the mean. We observe that the number of trajectories required for collapsing the uncertainty is dependent on the true value. The farther we are from the nominal of $L = 1m$, the more data is required.

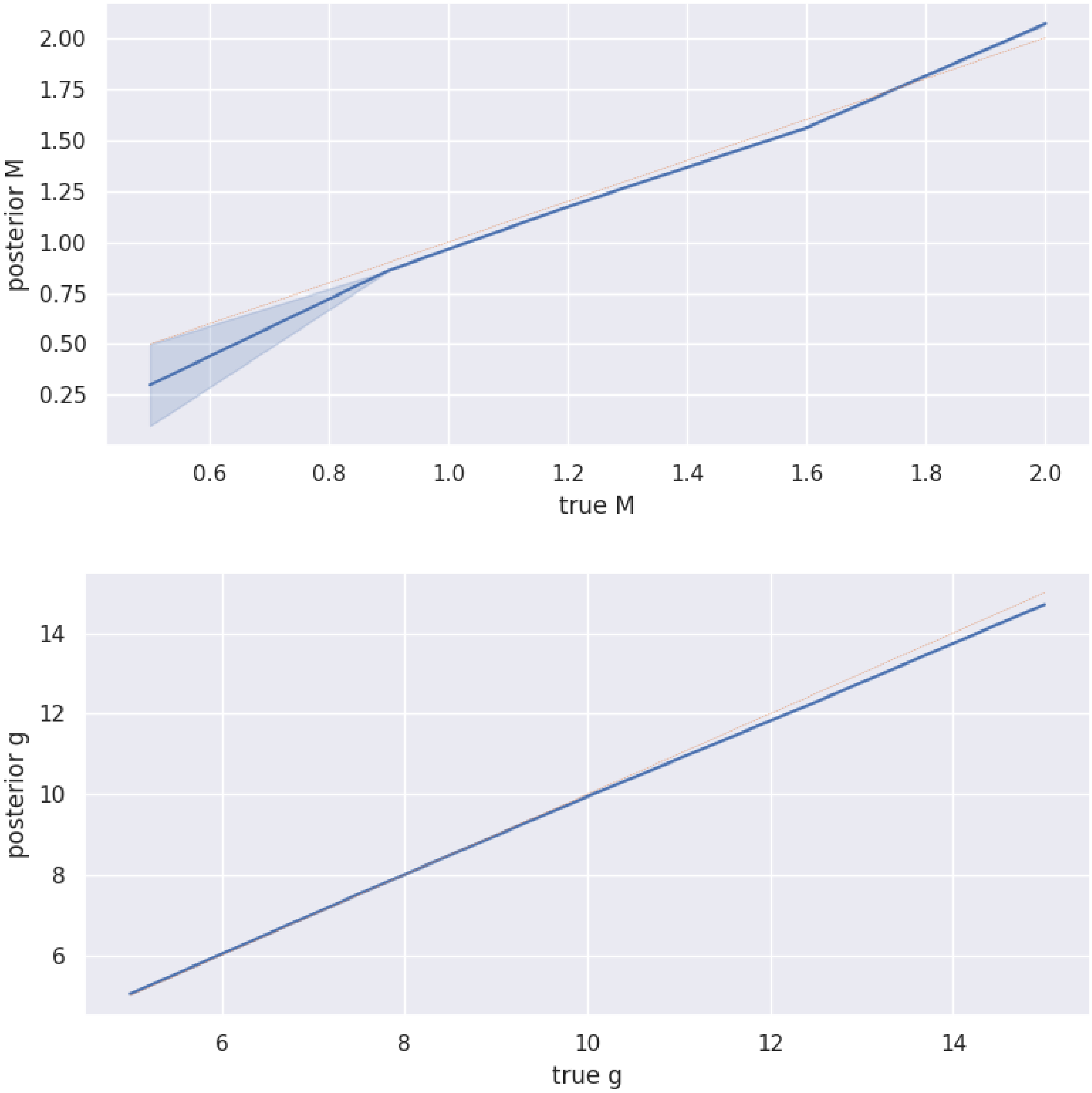


Figure 4.9: Posterior distribution of inferred M (top) and g (bottom). The blue line and shaded area represent the mean and standard deviation of the posterior. The red dashed line is the true value. We can see that predictions are over confident and the standard deviation of the posterior does not recover the truth in most cases.

with Variational Inference.

1D Latent Variable Model

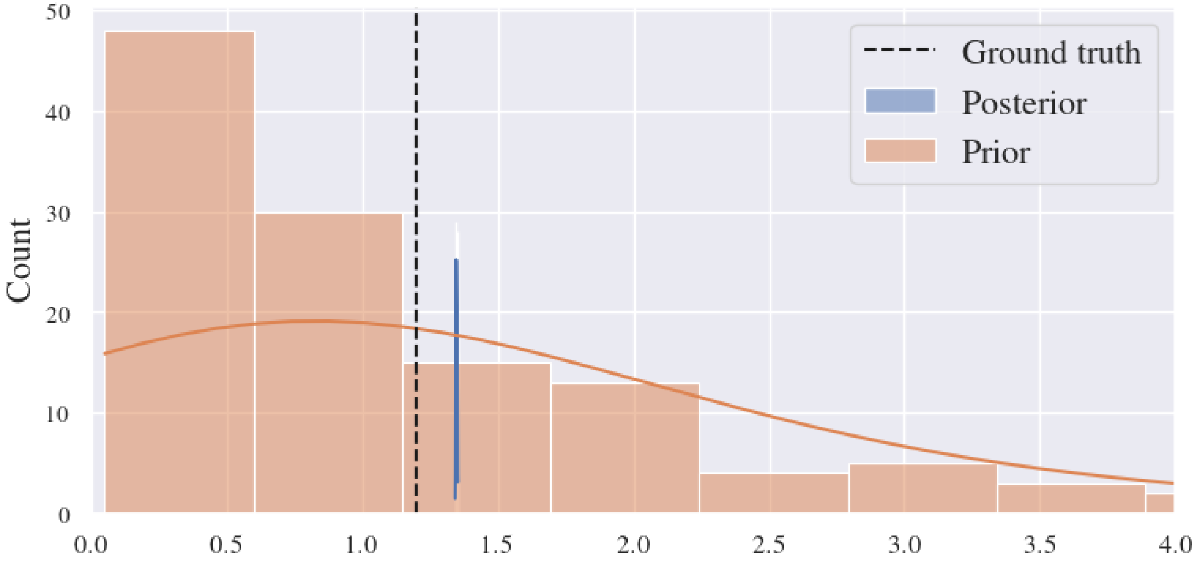


Figure 4.10: Prior and posterior distribution of the latent variable. The true value of the pole length is indicated by the black dashed line. We can see that the posterior is very narrow and does not coincide with the true parameter.

In this first section, we evaluate the setting where the pole length is hidden. The learned latent variable $h \in \mathbb{R}$ will here be used for transforming the variables $\Pi_{\dot{\theta}}$ and $\Pi_{\ddot{\theta}}$ according to equation 4.17.

$$\begin{aligned} \Pi_u &= \frac{u}{MgL} \\ \Pi_{\cos(\theta)} &= \cos(\theta) \\ \Pi_{\sin(\theta)} &= \sin(\theta) \\ \Pi_{\dot{\theta}} &= \dot{\theta}\sqrt{\frac{L}{g}} \\ \Pi_{\ddot{\theta}} &= \frac{L\ddot{\theta}}{g} \end{aligned}. \tag{4.17}$$

In figure 4.10, we display the prior and posterior distributions over the latent pole length as well as the actual value. We can see that the posterior is very peaked and that its support does not coincide with the true value. However, the model still fits well to its training data (figure 4.12, top-middle plot). In this specific case, the collapse of the uncertainty associated with the posterior does not prevent the model from making good predictions. It could however cause problems for identifying new parameters, especially in higher dimension spaces. The quality of the prediction however, decreases significantly when we move away from the training context. If we adapt the latent space whilst keeping the rest of the model fixed, we are able to make good predictions as

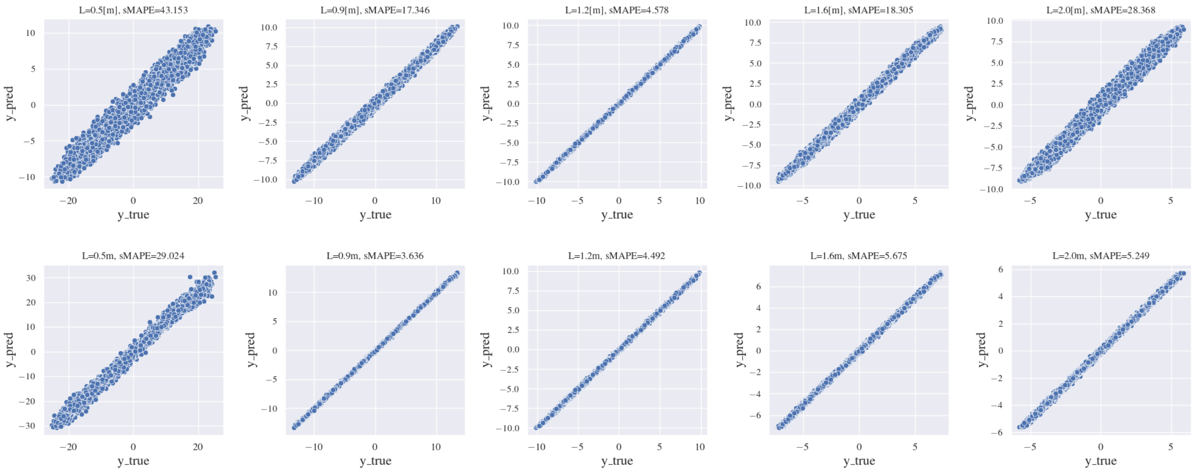


Figure 4.11: Prediction of the regression on different contexts. On the top plot, we show the predictions after training the model. On the bottom plot, we display the results after inferring the latent variables on new data. In the latter case, the model predictions are much closer to the true values.

can be seen on the bottom plot of figure 4.12.

We can verify the quality of predictions by comparing the mean predictions of the outputs y against their ground truth (figure 4.11). We can see that the predictions vary significantly when the context changes. However, when the latent variables are inferred on the new data, the model is able to make good predictions.

Now that we saw the model is able to fit the data wall in terms of prediction performance, we turn to the ability of the learned latent space to carry meaningful information about the hidden physical parameter it aims to model. To do so, we collect 50 datasets with different pole lengths equally spaced between 0.5 and 4 meters. We relearn the posterior over H for each of them, and compare samples from the posterior with the truth. We show the plot of the posterior against true value on figure 4.13. If the fit were perfect, we would see the points aligned at $y = x$ which is not the case: the inferred values are overestimated. However, that trend is monotonous and linear, as confirmed by the statistical tests shown on table 4.1. While that inferred value may not be used to estimate the true parameter, it could however be used to obtain some information about the new context. Because $\boldsymbol{c} = \kappa \boldsymbol{c}_0$, a decrease in the estimate of H_c would imply that the scaling factor κ is smaller than 1 provided the relationship between latent and true variable is monotonously increasing. In practice however, there is no way to verify that as long as the context is hidden.

This demonstrated that the latent space we have learned carries the dimensions of a distance, but it is not expressed in the unit of meters. Indeed, as we stated previously and mentioned in [Lee et al., 2021], two units of a physical dimension are equivalent up to a linear transformation.

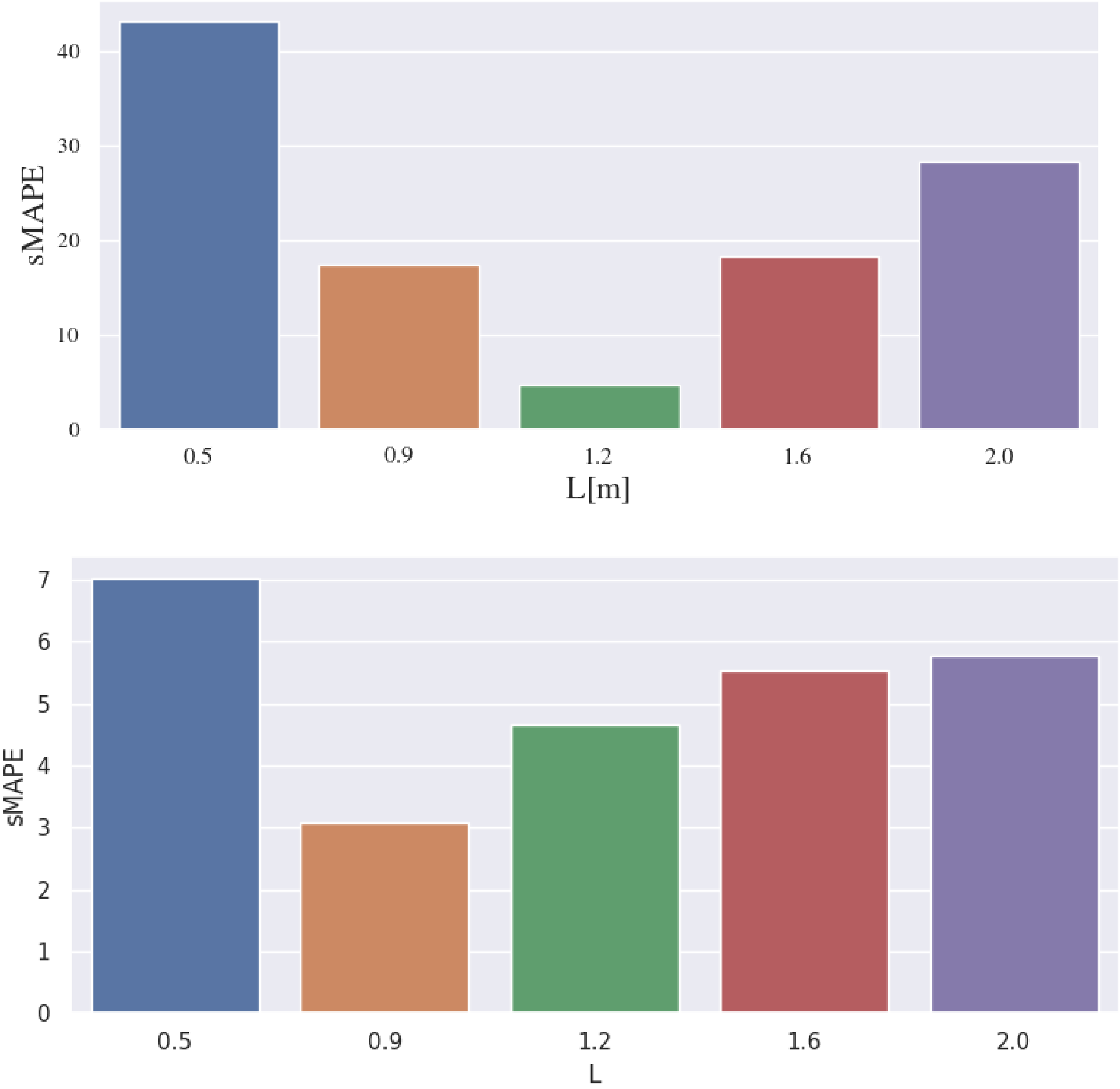


Figure 4.12: The top plot shows the test predictions of the model just after training. The errors on data close to the nominal of $L = 1m$ are lower than those further. On the bottom, we show the sMAPE score after inferring H_L on the new data, thus yielding better predictions.

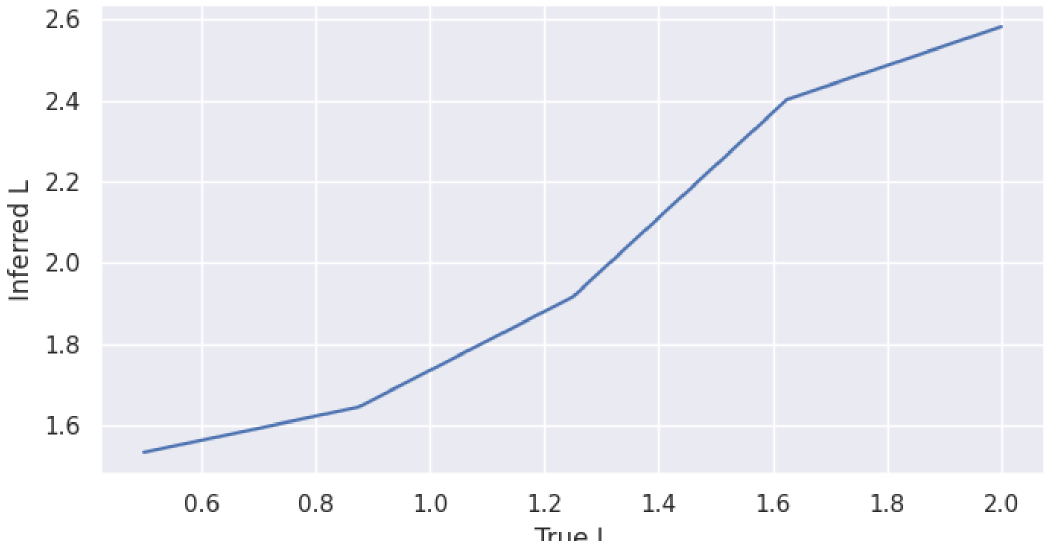


Figure 4.13: Regression Plot of inferred and true pole length in few-shot learning setting. As we can see, the values are not correctly estimated, but they increase monotonously with the true value.

Test	R	p-value
Pearson	0.9590	0.0
Spearman	0.9986	0.0

Table 4.1: Pearson and Spearman statistical test for correlation between inferred latent variable and true value of pole length. These tests suggest the estimation yields strongly correlated variables.

The benefit of this method is the ability to quickly adapt to a shift in distribution. After detecting a distribution shift, we can infer a new set of latent variables representing the local properties of the task, while retaining the model and the global properties.

2D Latent Variable Model

In this section, we take one step further and consider both pole length and mass are unobserved, as showed on figure 4.14. Following the methodology from the previous subsection, we infer a 2D latent variable, with each dimension representing the mass and length respectively, and use them to transform the data.

On figure 4.15, we show the prediction error of the model right after it has been trained on the nominal context, situated in the middle of the plot. We can see that as we draw context values farther from the training value, the predictions become increasingly worse. This can be fixed,

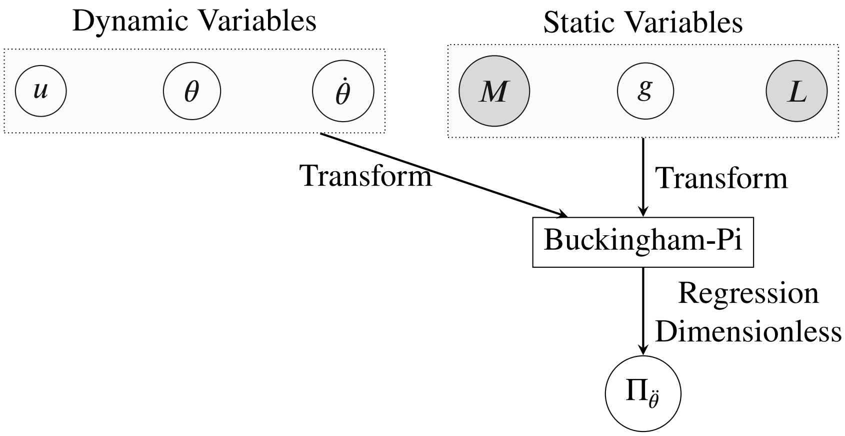


Figure 4.14: Regression task with hidden pole length and mass on angular velocity for the pendulum. In this case, both pole length L and mass M are hidden.

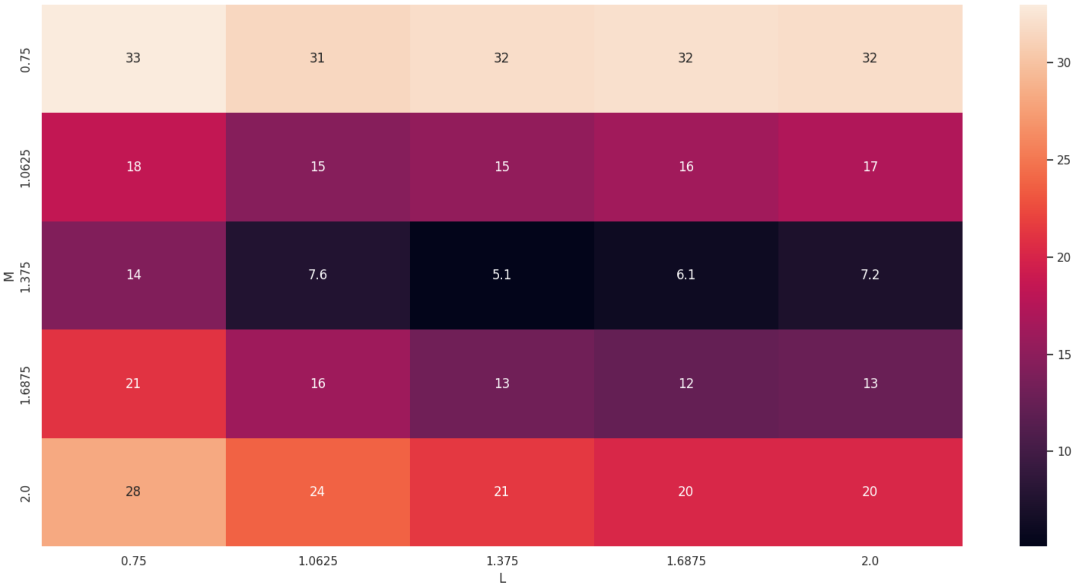


Figure 4.15: Influence of mass and length on model performance measured by sMAPE 3.16. The nominal training context is in the middle square with $M = 1.375kg$ and $L = 1.375m$. Dark purple color indicate better predictions. We can see that the model is more sensitive to a shift in mass than length and that the quality of predictions decrease significantly for large shifts.

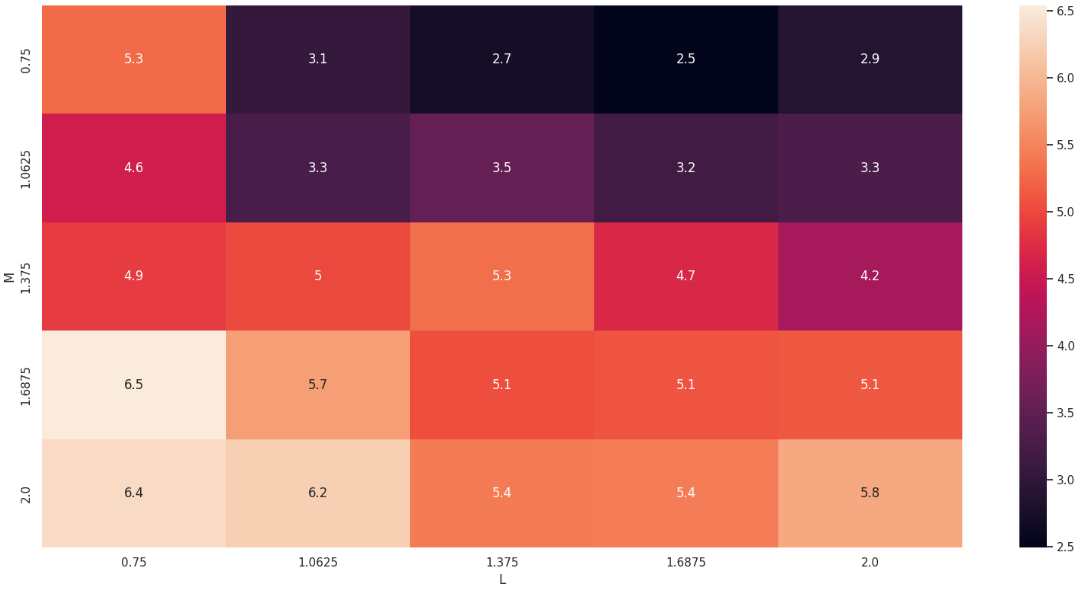


Figure 4.16: sMAPE score on the different context pairs after reinferring the latent variables. It shows that the model is able to adapt to the new context to yield good predictions.

Parameter	Test	R	p-value
L	Pearson	0.9798	0.0
L	Spearman	0.9876	0.0
M	Pearson	0.5825	0.0
M	Spearman	0.5596	0.0

Table 4.2: Pearson and Spearman statistical test for correlation between inferred latent variable and true value for pole length L and mass M.

however, by learning a new set of latent variables for each of these context pairs with the same predictive model.

For context adaptation, we collect ... trajectories on each new context and train the distribution of latent variables for 100 epochs. The prediction errors on figure 4.16 show the model has been able to adapt quickly. The leftmost values, corresponding to the lowest pole length are the ones with worse estimation, though still below 10% for all but the top-left one.

These results demonstrate a strong correlation between the inferred latent variables and their corresponding true values. We now aim to measure how the error is distributed, if we estimated the values of the variable given samples from the latent space. To do so, we first train a linear model

$$\hat{c} = \boldsymbol{w} Z_c + \beta, \tag{4.18}$$

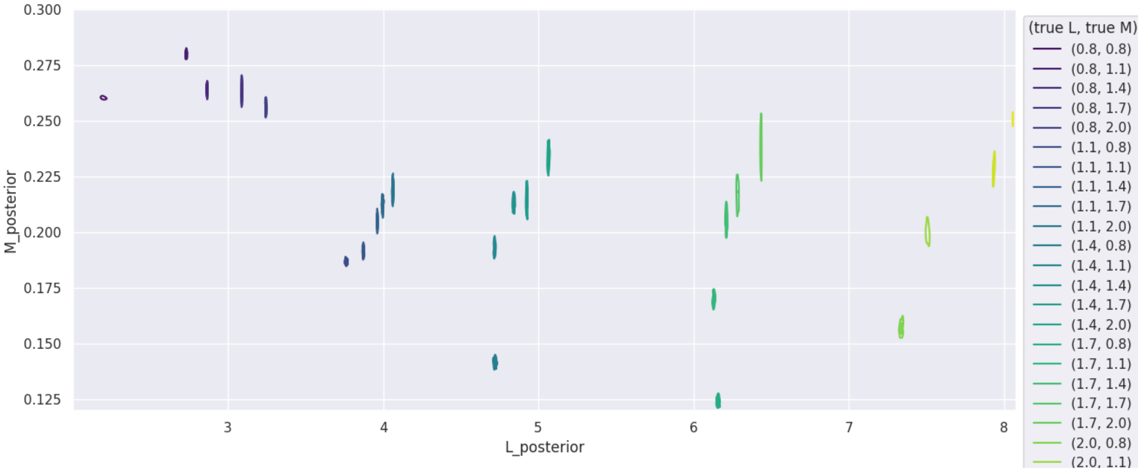


Figure 4.17: Here, we plot the inferred (Z_L, Z_M) pairs of variables for the different context values. The predictions are associated with higher uncertainty in the mass (vertical axis) than for the length (horizontal axis).

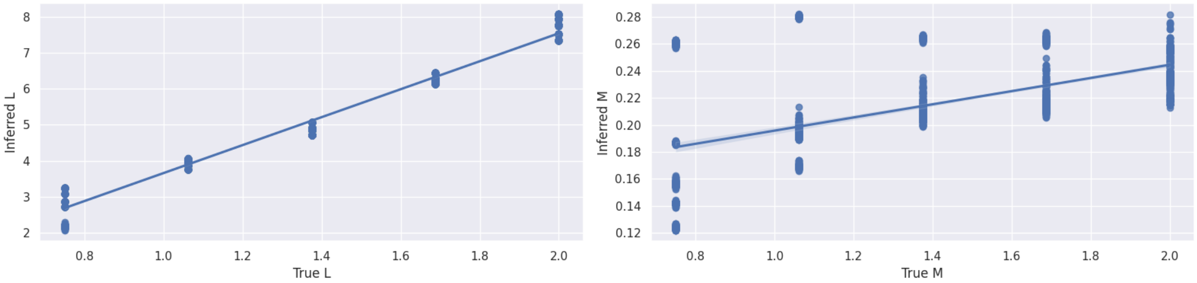


Figure 4.18: Comparison of latent inferred variable against their true value. Left plot is the pole length, right plot is the mass. The dots represent the means of the predictions and the line is an Ordinary Least Square estimator of the inferred against true context values.

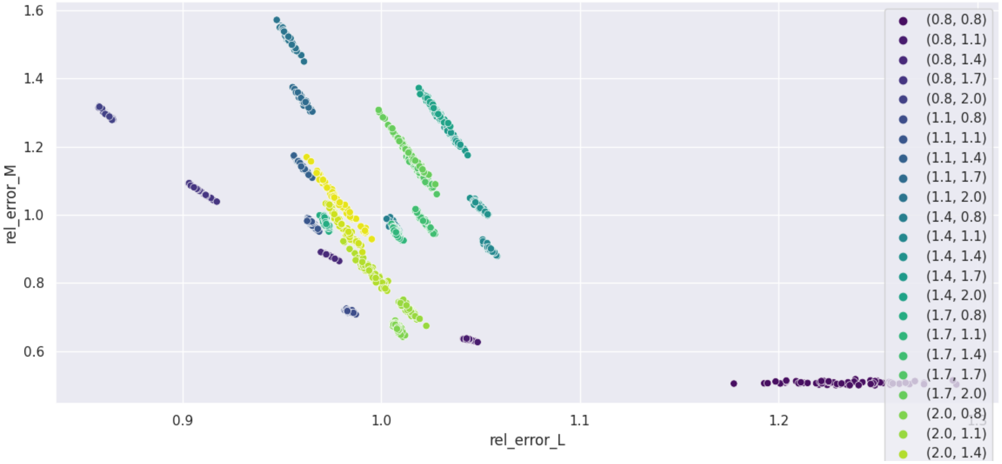


Figure 4.19: Plot of relative errors (equation 4.19) for different (L, M) values and corresponding posterior sample estimates. Perfect predictions would be located at the point with coordinates $(1, 1)$.

where $\hat{c}$ is an estimate of the ground truth. We can then plot the relative mismatch between the true values and their estimates using

$$\text{RelErr} = \frac{\hat{c}}{c}. \tag{4.19}$$

We plot this function for different values of M and L on figure 4.19.

We now investigate the quality of the estimation of the parameter compared to its true values. The question we ask is whether the learned latent variables estimate the ground truth correctly. We plot the marginals of each of them on figure 4.18 as well the linear regression line between true and inferred variable. The linear correlation between the latent and true parameters suggest the model is able to estimate variables with correct physical dimensions but not with the correct measurements units. The tests for correlation on table 4.2 confirm this hypothesis. It also suggests that knowing the correct context for 2 different environments would allow estimating a third one using the linear relation between the variables. The plot of relative error on figure 4.19 suggests the deviation is consistent across test environments. This means the error in estimation in M is consistently compensated by L.

Overall, we demonstrated on the pendulum how this latent variable model can be used to quickly transfer knowledge from a training environment to a perturbed version of it to make accurate predictions. If one is interested in estimating the value of the context, there is no way around measuring ground truth values for at least two of them.

4.4 Conclusion

In this chapter, we studied the feasibility of using a dimensionless feature space using the Buckingham-Pi theorem when the variables used for the transformation are unobserved. To do so, we augmented the probabilistic model with a set of latent variables that take values in a unit-typed space.

We made a first study under the hypothesis that all the variables are known during training, but then are hidden during deployment. We demonstrated empirically that a probabilistic model of the system yields the ability of detecting any change in the unobserved states (figure 4.4). This can be used to trigger the estimation of a latent variable that is assigned the physical dimensions of the hidden ones. We also showed that even in low-data regime, these latent variables represent a good estimate of the true hidden values (figures 4.7 and 4.8).

Second, we relaxed the assumption of full observation of the context variables during training. In that case, we need to learn the latent space concurrently as the global model itself. This model can then quickly adapt to a shift in the context space by re-learning the context-specific latent variables. We empirically showed such adaptation yields very good prediction performance on a wide range of test contexts (figures 4.12 and 4.16). Compared to the case where all states variables are available during training, the learned latent variables do not estimate accurately the ground truth. Nevertheless, statistical tests proved a strong correlation between them (tables 4.1 and 4.2). This result suggests that the latent space do represent correct physical dimensions albeit not in the correct units.

We showed in this chapter how the Buckingham-Pi theorem can be used for zero-shot transfer learning problems. The invariance induced by non-dimensionalization relaxes the need for domain randomization for learning a hierarchical model that disentangles the local from the global task properties. We demonstrated the effectiveness of the approach on an actuated pendulum with 1D and 2D hidden variables. The extension of our approach to higher-dimensional systems remains to be proven, and we leave it for future work.

Chapter 5.

Robust Model-Based Reinforcement Learning in Dimensionless State-Action Spaces

Controllers trained with Reinforcement Learning tend to be very specialized and are thus not able to generalize well outside their training environments. We propose a Model-Based approach where both the policy and the world model are trained in a dimensionless state-action space. To do so, we introduce the notion Π-MDP which is an extension of Contextual-MDPs where the state and action spaces are non-dimensionalized with the Buckingham-Π theorem. We then provide a generic model-based policy search algorithm in the Π-MDP and apply it with probabilistic state estimation using Gaussian Process models. This allows the controller to generalize well to new contexts in the zero-shot transfer setting, meaning no retraining is required. We demonstrate the usefulness of this approach on the actuated pendulum and cartpole environments.

5.1 Motivation for Robust Controllers

One of the main obstacles for deploying controllers trained with Reinforcement Learning in the real world is their lack of resilience to perturbations and noise that are absent during training. This problem of distribution shift, that we already described in chapter 3, has mostly been investigated in the supervised and unsupervised learning settings. Though the question can be phrased similarly in sequential decision-making, solving it remains difficult because of the dynamic nature of RL. Firstly, because errors and approximations accumulate during planning and rollout, secondly because the closed-loop nature of the learning process incurs a loss of identifiability [Ljung, 1989]. The issue is even more prevalent in Offline RL because of the lack of training data in some regions of the state-action space and the impossibility to collect more.

In this chapter, we focus our work on perturbations that affect the environment dynamics only, not the reward function which we assume to be known. The perturbations of the underlying transition kernel cause non-stationarity in the dynamics. These can be caused by hardware wear-and-tear, feedback loops or external perturbations and is admitted to be one of the main challenges

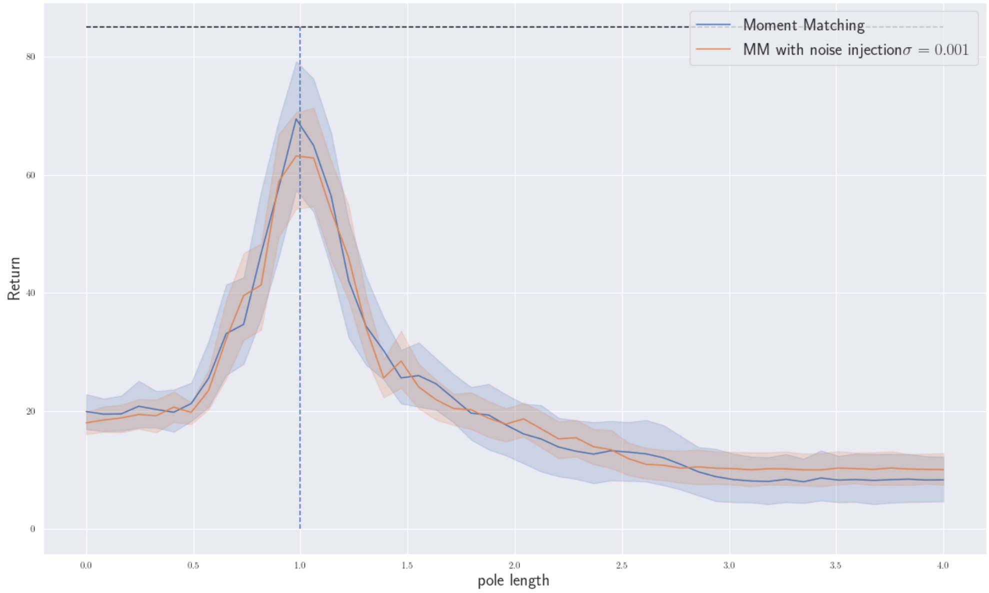


Figure 5.1: PILCO returns on the cartpole for varying pole length

to be solved for deploying RL agents in the real world [Dulac-Arnold et al., 2021a].

While a range of different approaches exist for approaching controller robustness (see section 2.1 for a review), many of them require several version of an environment during training. This does not pose a problem when a parameterized simulation is available as it allows the practitionner to randomize those parameters that will later change during testing. Our work instead focused on an approach that relaxes the domain randomization assumption required for policy transfer.

There are three main categories of approaches for improving controller robustness [Kirk et al., 2023]:

- Adapting the optimization objective with worst case estimators of the return
- Improving data collection
- Changing the model

The methods we propose fall into the latter category. We leverage prior knowledge given by the physics of the system such that by construction, the control policy will be agnostic to perturbations.

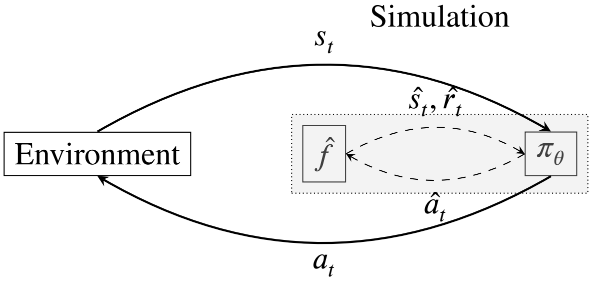


Figure 5.2: Simulation with world model $\hat{f}$. The parametric policy π_θ interacts with the actual environment but is trained with simulated transitions (grey box).

5.1.1 Model-Based RL

Model-Based Reinforcement Learning (MBRL) is a class of RL algorithms in which the policy is trained on data generated by a *world model*. For this reason, such algorithms are often called *indirect methods* as opposed to model-free approaches that optimize their decisions using data directly collected in the environment. We illustrate this concept on figure 5.2 where we can see the environment on the right and a simulation of it in the grey box. In principle, any model-free algorithm can be ported to a model-free counterpart simply by applying it within a model rather than actual interaction. In practice however, they require specific adaptation to counteract the modelling error.

The first requirement for such MBRL algorithms is the dynamics model. It is an estimator that mimics the behaviour of the MDP transition kernel,

$$\hat{f} : (\boldsymbol{s}_t, \boldsymbol{a}_t) \mapsto \hat{\boldsymbol{s}}_{t+1}. \tag{5.1}$$

This model is subsequently trained to predict one-step transitions using the batches of data collected so far. It is therefore a multidimensional regression problem where the inputs are the state-action vectors $\tilde{\boldsymbol{x}} = (\boldsymbol{s}, \boldsymbol{a}) \in \mathbb{R}^{d+f}$ and the targets are the successor states $\boldsymbol{y} = (\boldsymbol{s}_{t+1} - \boldsymbol{s}_t) \in \mathbb{R}^d$. Because the target $\boldsymbol{y}$ are vectors, MBRL methods are more sample-efficient than model-free methods since they learn from scalar reward signals instead. The procedure of inferring the dynamics of the environment is called *System Identification* The model can then be queried to generate one-step transitions or whole trajectories with a parametric policy π_θ. We write the closed-loop dynamics as

$$f_\theta : s \mapsto f(\boldsymbol{s}'|\boldsymbol{s}, \pi_\theta(s)). \tag{5.2}$$

Its estimate counterpart $\hat{f}_\theta$ is able to generate whole trajectories by functional composition in order to predict the future state of a system under the current policy. To do so, we start from an initial state $\boldsymbol{s}_0$ and iterate the predictions until desired time.

$$\hat{s}_t = \underbrace{\hat{f}_\theta \circ \cdots \circ \hat{f}_\theta(\boldsymbol{s}_0)}_{t \text{ times}}. \tag{5.3}$$

This ability to query the model to predict long-term states of the system is what makes this type of methods useful. It can generate trajectories $\tau = (\boldsymbol{s}_0, \cdots, \boldsymbol{s}_t)$ of arbitrary size. Assuming we know the reward function r, we can compute the simulated expected sum of rewards from the future state predictions. The policy search objective therefore writes down as,

$$\hat{R}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\hat{f}^\theta}\left[\sum_{t=0}^{T} r(\hat{\boldsymbol{s}}_t)|s_0\right]. \tag{5.4}$$

This quantity serves as a proxy for the return that would be obtained by rolling out in the environment. The objective (5.4) is very similar to (2.3) but with the expectation measured by the approximate dynamics. A controller is optimal for the model if it maximizes that quantity (5.4), however there is no guarantee that $\arg\max_\theta \hat{R} = \arg\max_\theta R$ because of *model bias*. Because during training the policy is only exposed to data generated by the model, any discrepency with the true dynamics will reflect on the quality of the policy. Moreover, because of compounding errors in 5.3 estimating the future states is a difficult task. One solution is to use the model on short rollouts only [Janner et al., 2019].

Alternatively, a probabilistic model is able to eliminate most of the bias associated with predictions. Given a state-action input, a probabilistic model will predict a distribution over plausible future states. Hence, rolling it out with 5.3 yields a distribution of trajectories $p(\tau)$. If the model is wrong, the trajectories will be associated with high levels of uncertainty that will propagate to the estimation of (5.4). On the other hand, a non-probabilistic model would not have that capacity and weigh equally all trajectories for gradient estimation however unlikely they are.

To optimize the parameters of the policy, different algorithms use different gradient of return estimation schemes like reparameterization trick [Kingma and Welling, 2014; Xu et al., 2019] or likelihood ratio [Williams, 1992] to backpropagate derivates through sampling the model [Mohamed et al., 2020]. Suppose $\hat{\nabla}_\theta \hat{R}(\boldsymbol{\theta})$ is a unbiased estimation of the gradient, we can optimize the policy with stochastic steps in the ascending direction

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \eta \hat{\nabla}_{\boldsymbol{\theta}} \hat{R}(\boldsymbol{\theta}), \tag{5.5}$$

with η the learning rate. Between each episode, the policy is optimized with the current dynamic model until the expected return stall. Then the policy collects a new episode of data which is fed into training the model. The model will improve using the new data, and so until some measure of convergence is reached.

For generalization, however, the objective is different as the transition kernel depends on a context that is different from the training one. Let us consider the distribution $p(\boldsymbol{c})$ from which context is sampled at the beginning of each episode at testing time. Then the expected return of a policy within that C-MDP is,

$$R(\boldsymbol{\theta}, \mathcal{M}_C) = \mathop{\mathbb{E}}_{c \sim p(c)} \left[R(\pi_{\boldsymbol{\theta}}, c) \right] . \tag{5.6}$$

While the equation (5.6) provides a good illustration for the generalization problem, we do not use that metric in practice. To evaluate the robustness of our policies, we instead sample the contexts $\boldsymbol{c}$ uniformly on a domain and compute the returns for each.

We extend the subclass of model-based policy gradient methods with Gaussian Process priors [Deisenroth and Rasmussen, 2011; Parmas et al., 2018; Amadio et al., 2022; Cowen-Rivers et al., 2022] because their ability to estimate uncertainty eliminates most of the bias. This ability to plan with uncertainty has allows model-based algorithms to compete with their model-free alternatives [Schrittwieser et al., 2020; Chua et al., 2018; Janner et al., 2019]. Instead of optimizing the controller in the natural state-space view, we do it in its dimensionless counterpart. This transformation essentially renders the controller invariant to small context changes and so is able to generalize outside its training support. We believe model-based RL constitutes a good method for the problem of generalization since their ability to simulate a system from empirical data allows counterfactual queries allows reasoning without environment interaction, which is a key component of cognition [Hamrick, 2019]. Moreover, inference based on model simulation will become an important aspect of applied sciences in the coming years [Lavin et al., 2021].

Link to Previous Work

The method we propose sheds a new light on a generalization method based on *Augmented World Models* [Ball et al., 2021]. In this work, the authors propose to increase the zero-shot generalization of a control policy learned offline from a single environment. To do so, they rescale the observations by a factor inferred from data. Our work proposes a similar transformation that is instead inferred from the physics of the system at hand. The Buckingham-Π theorem has also been applied to transfer learning problems for system identification [Therrien et al., 2024] and control [Girard, 2024] in robotics.

5.2 Equivariant Model-Based Reinforcement Learning

5.2.1 Control in Dimensionless Observation Spaces

We consider a Contextual Markov Decision Process (C-MDP) defined as

$$\mathcal{M}_c = \left(\mathcal{S}, \mathcal{A}, \mathcal{R}, f_c\right). \tag{5.7}$$

The only difference from an MDP is the transition kernel which depends on a context $\boldsymbol{c} \in \mathbb{R}^k$. In all the following, we assume the context always remains fixed in the duration of an episode. Robust RL maximizes the expected sum of rewards for large context set $\mathcal{C}$.

In order to reason about control policies within a dimensionless state-space, we introduce a new concept we call the Π-MDP. The Π-MDP is a generalization of the C-MDP, it is equipped with a dimensionless invertible feature mapping that transforms the state and actions spaces depending on the context vector $\boldsymbol{c}$.

Definition 5.2.1 (Π-MDP)
The dimensionless Markov Decision Process or Π*-MDP is a MDP in which the state and action spaces are dimensionless. They can be written*

$$\mathcal{M}_\Pi = \left(\mathcal{S}_\Pi, \mathcal{A}_\Pi, \mathcal{R}, f_\Pi\right), \tag{5.8}$$

where $\mathcal{R}$ is the reward function and f_Π the transition kernel that takes values in the dimensionless sate-action space. The dimensionless transition kernel is defined as,

$$f_\Pi = f \circ \Phi_\Pi \tag{5.9}$$

where $\circ$ denotes the functional composition and Φ_Π is the non-dimensionalization transformation.

The graph on figure 5.3, illustrates how an autonomous agent interacts within such a system. At each time step, the state observation is non-dimensionalized with the current value of the context and passed on to the input of the control policy. The policy then sends out a dimensionless control signal which is dimensionalized using the same context value to ensure homogeneity of the environment transition kernel.

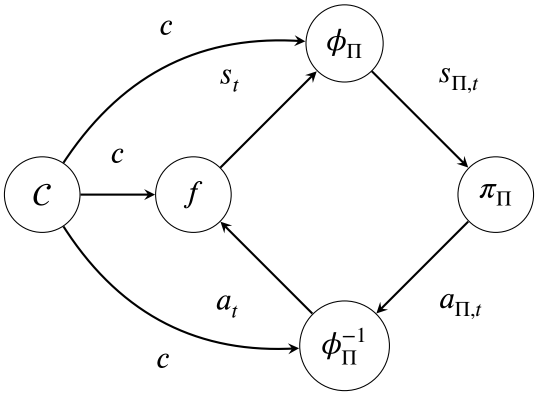


Figure 5.3: Interaction within a Π-MDP

Construction of a Π-MDP

Our goal is to design a model-based policy search algorithm within the dimensionless space so that the policy is robust to environmental perturbations.

$$\pi_\Pi(s, \boldsymbol{\theta}) = \pi_\Pi(\Phi_\Pi(s); \boldsymbol{\theta}) \tag{5.10}$$

We show the interaction in such a system on figure 5.3. It is important to notice how the context affects the dynamics f and transformation Φ_Π. It means if we have knowledge of it, then we can also use it directly to apply the non-dimensionalization operation ϕ_Π and its inverse.

5.2.2 Model-Based Reinforcement Learning in Π-MDP

Now that we framed the generalization problem within a dimensionless state-space, we describe how to integrate it within a Model-Based policy search algorithm. We introduce a new algorithm Π-PILCO: *Dimensionless Probabilistic Inference for Learning COntrol*, a variation of the data efficient PILCO algorithm that performs policy search within a dimensionless state space. Let us not that the methodology can in principle, be applied to any MBRL algorithm provided the state and action space can be non-dimensionalized with the Buckingham-Π theorem.

In essence, the algorithm is not very different from the one in natural space. The difference here is that both dynamics model and policy have dimensionless inputs and outputs. When the policy is interacting with the MDP, it non-dimensionalizes the observations, returns a dimensionless control, which is then projected back in natural space before being sent to environment. The procedure is described in extensive details in algorithm 2.

Algorithm 2 Interaction in a Π-MDP

```
Input policy $\pi_\Pi$, dimensionless feature map $\Phi$, initial state $s_0$
$s_t \leftarrow s_0$
for $t = 1, \dots, T$ do                          ▷ number of steps of an episode
    $s_{\Pi,t} = \Phi s_t$                        ▷ non-dimensionalize observation
    $a_{\Pi,t} = \pi_\Pi(s_{\Pi,t})$              ▷ choose dimensionless action
    $a_t = \Phi^{-1}(a_{\Pi,t})$                  ▷ dimensionalize action
    $s_t \leftarrow f(s_t, a_t)$                  ▷ 1-step Markov transition
    $r_t = R(s_t)$
return $\sum r_t$                                 ▷ Cumulative Rewards
```

The policy search resembles also closely to that in the natural spaces. The optimization objective is the same as previously described in equation 5.4, the difference lies in the way trajectories are computed. In equation 5.3, the closed-loop dynamics iterate one-step predictions on the natural state-action spaces. So at each time step, the policy selects a dimensionless action based on dimensionless observations and the model predicts the next state (or distribution thereof). Additionally, we compute the reward at each step by applying the inverse Buckingham transformation to the dimensionless state. We repeat the procedure until a horizon H is reached and the local rewards are summed to estimate the gradient of the return. For ease of exposition, algorithm 3 details the policy search methodology based on the Reparameterization Trick as in [Parmas et al., 2018].

Algorithm 3 Dimensionless Policy Search - Π-PILCO

```
Input policy $\pi_{\Pi,\theta}$, dimensionless feature map $\Phi$, dimensionless model $\hat{f}_\Pi$
for $i = 1, \dots, P$ do                          ▷ number of epochs
    $s_t \sim \rho_0$                             ▷ sample initial state
    $s_{\Pi,t} = \Phi s_t$
    $R = 0$
    for $t = 1, \dots, H$ do                      ▷ prediction horizon
        $a_{\Pi,t} = \pi_\Pi(s_{\Pi,t}; \theta)$
        $s_{\Pi,t} \leftarrow \hat{f}_\Pi(s_{\Pi,t}, a_{\Pi,t})$
        $R \leftarrow R + R_\Pi(s_t)$
    $\theta \leftarrow \theta + \nabla_\theta R$  ▷ gradient step
return $\pi_{\Pi,\theta}$
```

5.3 Experiments

Questions

- Can we learn an invariant controller from a single training environment
- Can we use model for few-shot adaptation?
- Are safety constraints respected?
- Can we infer the context during deployment?
- how to measure invariance of a decision? ie characterise the 'region of invariance'

Assumptions:

- the context variables are fully observables during training, meaning Φ is known
- the reward function is known a priori.

We will evaluate our algorithm on two second-order the systems, the first is the pendulum that we have already studied in depth in the previous chapters The cartpole, is a slightly more complicated one where a pendulum is attached to a cart on a horizontal axis that can move left and right. Initially, the pendulum is positioned downright and the control problem consists in learning a policy that can swing the pendulum up and stabilize it vertically at the middle of the cart. The nominal context values for each are summarized on table 5.1.

These two systems possess the appealing properties of having smooth dynamics and low dimensions. As such, they are well suited for studying dimensional analysis in RL. We used our own model-based RL code[1] for training the policies in the different environments. The control policy is parameterized as a single-layer Radial Basis Function network. We use Moment Matching [Girard et al., 2002] for trajecory predictions as in the original PILCO paper. For the cartpole, we used the benchmark for distribution shift from [Dulac-Arnold et al., 2021b] and adapted some of the code for our needs. For the pendulum, we used Gymnasium [Towers et al., 2023] on which context variables can be changed with no code modification.

Environment	$L[m]$	$M[kg]$	$g[m.s^{-2}]$
Pendulum	1	1	10
Cartpole	1	0.1	9.81

Table 5.1: Nominal context value for the cartpole and pendulum environments

The movement of the cartpole is described by the variables $(x, \cos(\theta), \sin(\theta), \dot{x}, \dot{\theta}), u$. Due to the similar structure between the two environments, the dimensionless groups for the cartpole

[1] https://git.dcs.gla.ac.uk/ValentinCharvet/pilco-torch

are very similar as those of the pendulum.

$$\begin{cases} \Pi_x = \frac{x}{L} \\ \Pi_{\cos\theta} = \cos(\theta) \\ \Pi_{\sin\theta} = \sin(\theta) \\ \Pi_{\dot{x}} = \frac{x}{\sqrt{Lg}} \\ \Pi_{\ddot{\theta}} = \ddot{\theta}\frac{g}{L} \\ \Pi_u = \frac{u}{Mg} \end{cases} \tag{5.11}$$

The derivations of the Π-groups in equation 5.11 are detailed in appendix A.1.2. As we did on the angular speed on the pendulum, we adapt the variable $\Pi_{\dot{x}}$ to avoid losing its sign.

5.3.1 Generalization Performance

We use two different metrics to evaluate the generalization capabilities of our algorithm. The return (2.3) is the most commonly used metric used in Markov Decision Processes. It measures the long-term performance of a controller given an initial state distribution and is computed by a discounted sum of rewards. The reward for our environments are inversely proportional to a distance from the current state and target as written in equation 5.12

$$r_t \propto -d(s_t, s^*), \tag{5.12}$$

where d is a distance function in $\mathcal{S}$. For our specific problems, we only consider finite-time MDPs and thus consider a discount rate $\gamma = 1$, which weighs identically the rewards from the beginning to end of each episode.

However, during the experiments we realized this metric was not sufficient to characterise the ability of the controller to stabilize the systems. The return translates the ability of the agent to stabilize a system at a target position as quickly as possible, which yields ignores two components of our tasks. The first is that is two controllers are able to solve the task but one requires more steps to do so, it will be penalized with a lower return since it spends less time in the optimal-rewards regions. The second inconvenience is that is the controller is able to push the system into a closed-loop equilibrium that deviates from the target, it will not receive an optimal return. In the next section, we will illustrate these two points for each of the environment we studied.

In order to alleviate the bias of the return metric, we had to find a metric that would translate

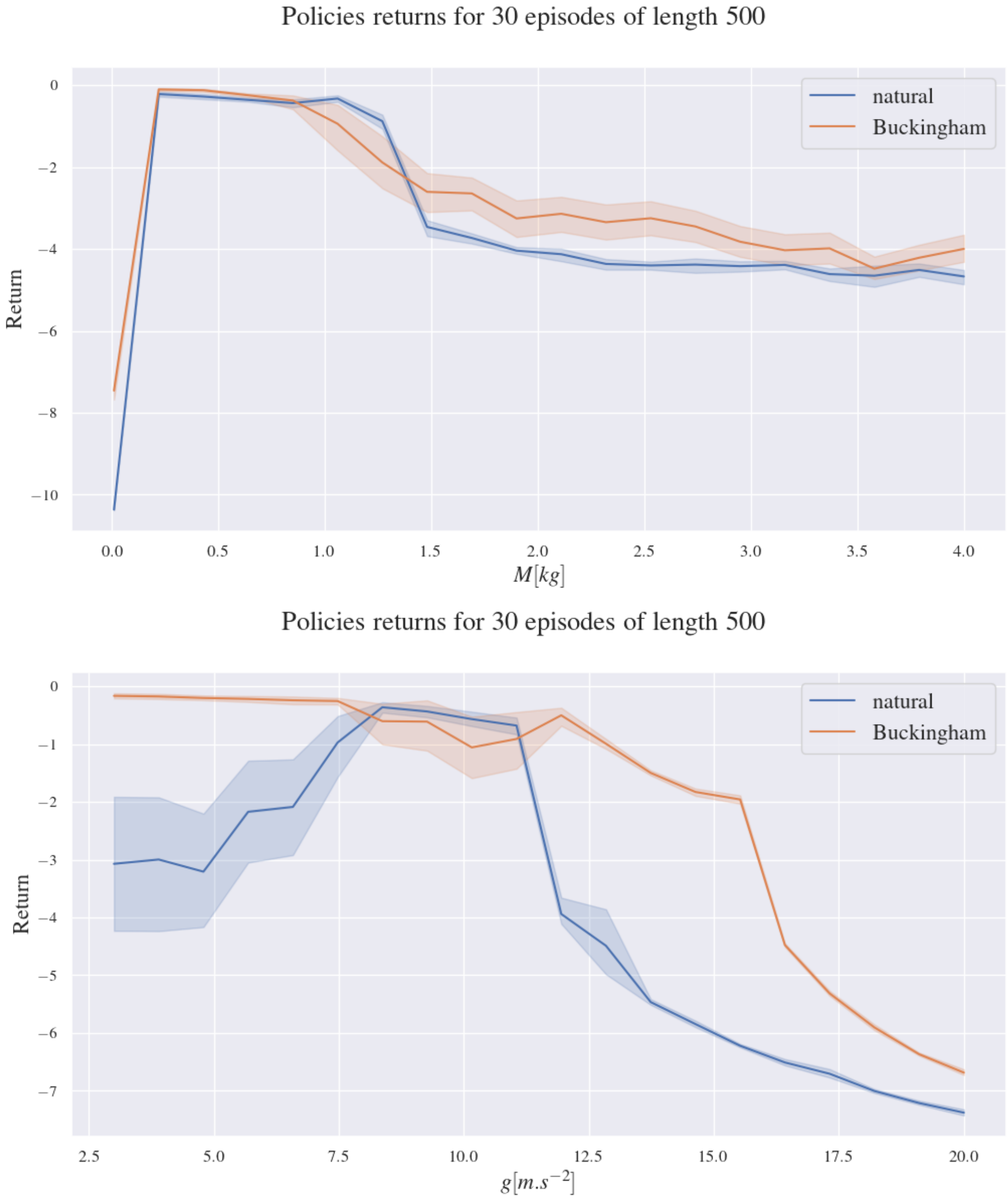


Figure 5.4: Return on the pendulum environment for different values of M (top) and g (bottom), with respective nominal values of $1m$ and $10m.s^{-2}$.

the ability of the controller to reach a closed-loop equilibrium. Therefore, we include a binary metric that measures whether in the last step of the episode, the velocity variables of the observations are equal to 0. We call such an episode *successful*, which allows us to measure the rate of successes for each controller across many different initializations. Our measure of success rate can be written as

$$\rho = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1} \left\{ \dot{s}_T \leq \epsilon \right\}, \tag{5.13}$$

where N is the number of evaluation episodes and ϵ a threshold. For our experiments, we used the values $N = 100$ and $\epsilon = 0.05$.

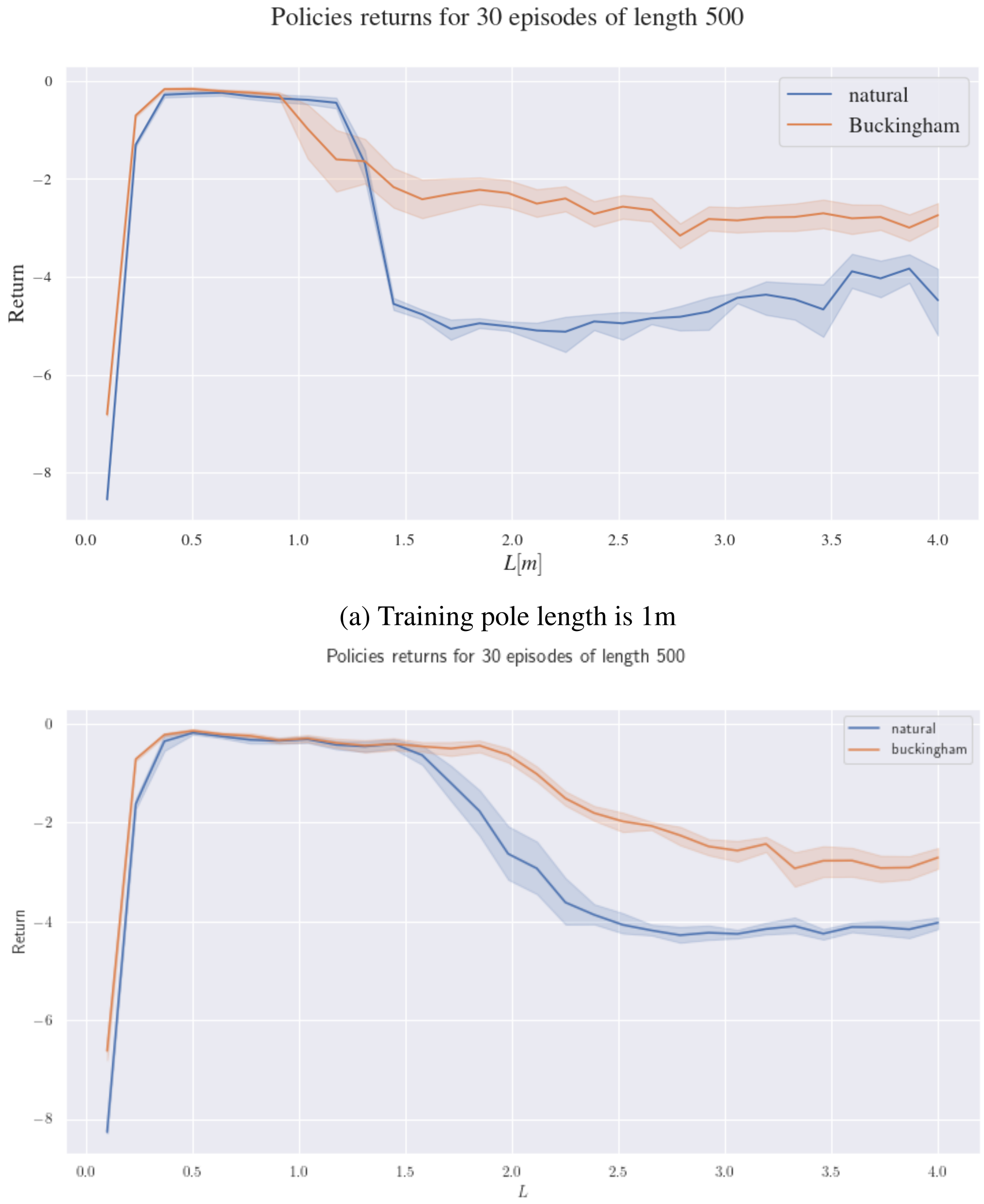
Return

We start with experiments to measure the return of the natural and dimensionless controllers in both pendulum and cartpole environment. The return is maximal when the pole is positioned vertically up (pendulum and cartpole) and the cart at the centre of the rail (cartpole).

The figure 5.4 shows the returns obtained on the pendulum when the pole mass M and gravity field g vary individually. The first observation is that the mass has less negative impact on the performance drop than the gravity. At 25% in increase of the nominal value, the performance of the natural controller has already significantly decreased for the latter. However, it is more obvious for the gravity case that the Buckingham controller obtains higher returns when the context drifts away from the nominal value, both in augmentation and reduction of the context.

A legitimate interrogation is if the Buckingham controller is maximally invariant with respect to the context. To verify this, we evaluate the returns of the controllers on the pendulum that are trained each on different pole length values. On figure 5.5 (left), we see the result for a training value of $L = 1$. We note that the performance degrades right after the nominal value in both case, but the drop is less significant on the Buckingham controller. However, when trained on a pole length $L = 1.5$, the Buckingham controller is able to maintain higher performance for longer than the natural one. This result suggests that the Buckingham controllers are more robust than their natural counterpart but they are not maximally equivariant either. If they were, the return would be the same whichever pole length they are trained on.

On figure 5.6, we repeat the same experiment on the cartpole environment. Again, we can see how the Buckingham controller is able to generalize and obtain high return when the context is scaled higher or lower than the nominal values. We can see on the right plot that the Buckingham returns tend to oscillate. This is caused by the fact that in this range, the controller is not able to maintain the pole up whilst the cart is at the centre of the rail.



(a) Training pole length is 1m

(b) Training pole length is 1.5m.

Figure 5.5: Return on the pendulum environment for different training values of L. We plot the mean and standard deviation for 50 episodes.

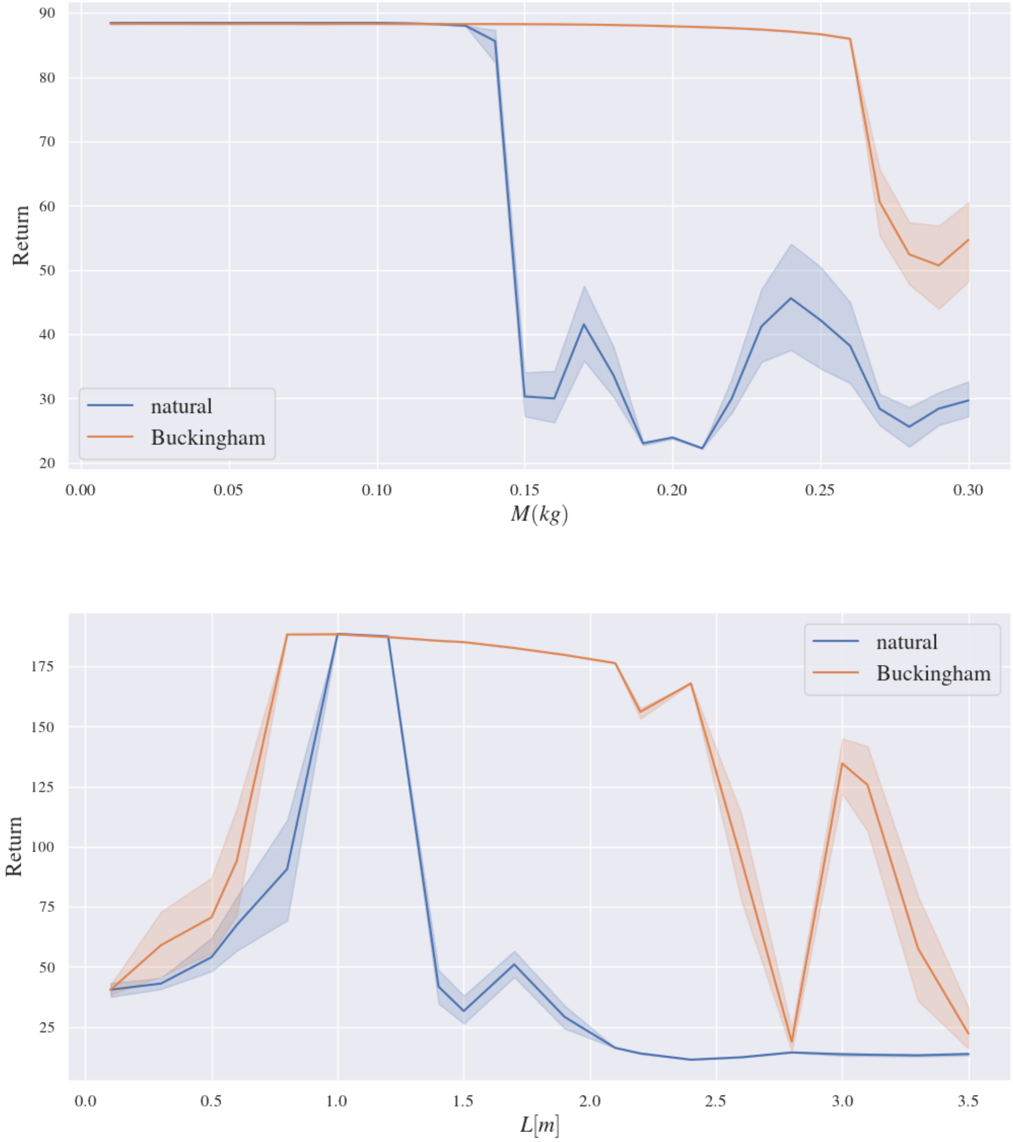


Figure 5.6: Return on the cartpole environment for different values of M (top) and L (bottom).

Success Rates

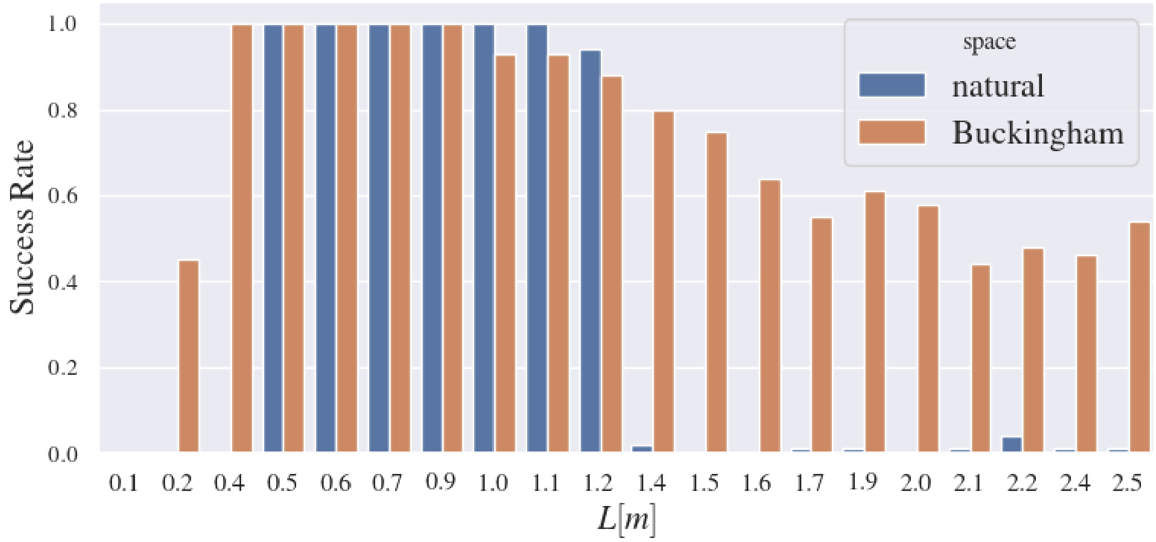


Figure 5.7: Success rates for different pole length values on the pendulum. We can see that beyond 1.2m, both controllers struggle to swing the mass up, even though the dimensionless controller succeeds in some cases.

We now look into the same experiments where we instead plot the success rates as given by equation 5.13. The figure 5.7 completes 5.5 as it shows the average lower cumulative rewards for high pole lengths is caused by the inability of the Buckingham controller to stabilize the system more than 20% of the episodes. This observation is confirmed by plotting specific reward trajectory given different initial states on figure 5.8. We can see on the left that the rewards associated with the Buckingham controller have much more variance when $L = 1.2m$ than for lower values.

We now turn to the cartpole problem to see if the return oscillation on high L is due to failure to solve the controller problem. We can see on figure 5.9 that in most for most of the evaluation context range, the episodes are counted as successful. This confirms our hypothesis that the controllers is still able to stabilize the system upright, albeit not at the target cart horizontal position. We can confirm this by plotting the reward trajectory of both controllers for $L = 2.1m$ on figure 5.10

Discrepancy between success rates and returns is due to reward relates to swinging up at centre of the cart whereas we count success as long as pole ends up vertically up. Additionally, for certain values of L, the controller is able to hack the environment and use the extra momentum given by hitting the wall at the extremity of the rail. We will see in more details in section 5.3.3 how the optimal performance of a controller are related to such constraints and the consequences of relaxing them. On figure 5.11, we show the cartpole environment at several time steps $t =$

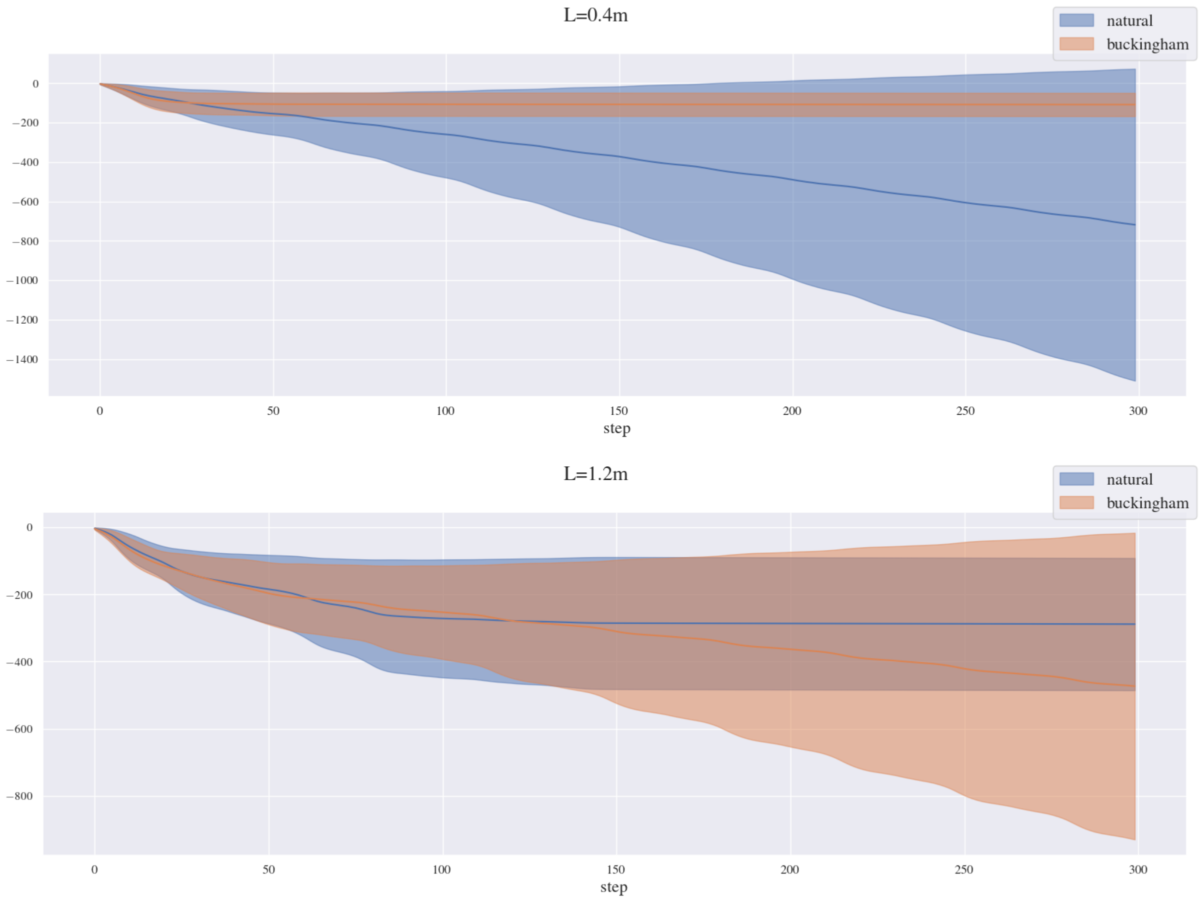


Figure 5.8: Cumulative rewards on the pendulum with $L = 0.4m$ (top) and $L = 1.2m$ (bottom). We observe that both mean and variance of cumulative rewards monotonously increase in the Buckingham case.

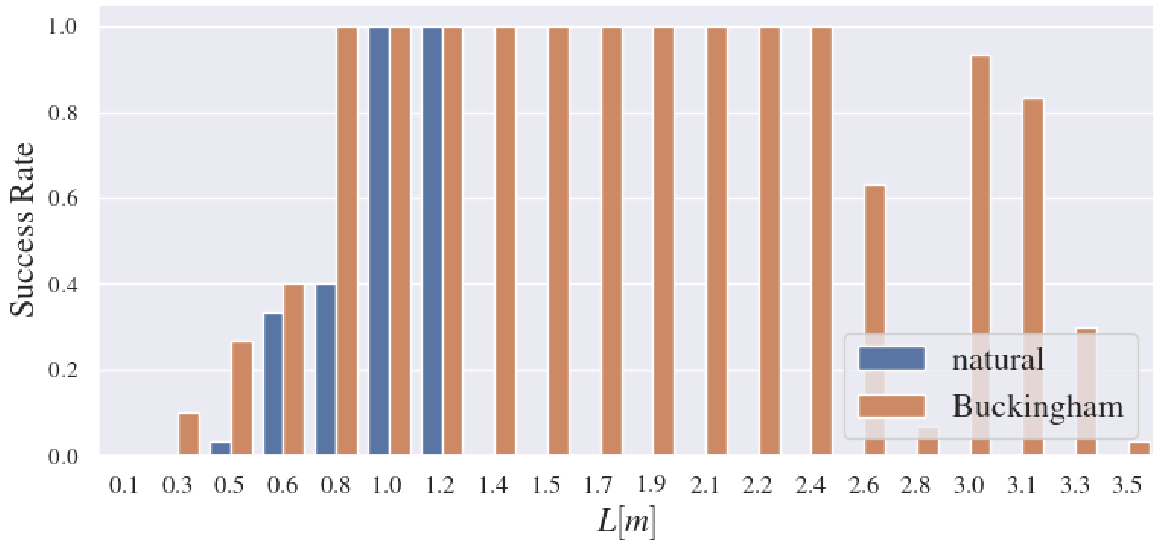


Figure 5.9: Success rates on the cartpole environment for different values of L. Below $L = 1.2m$, the natural controller is no longer able to swing the pendulum upright anymore.

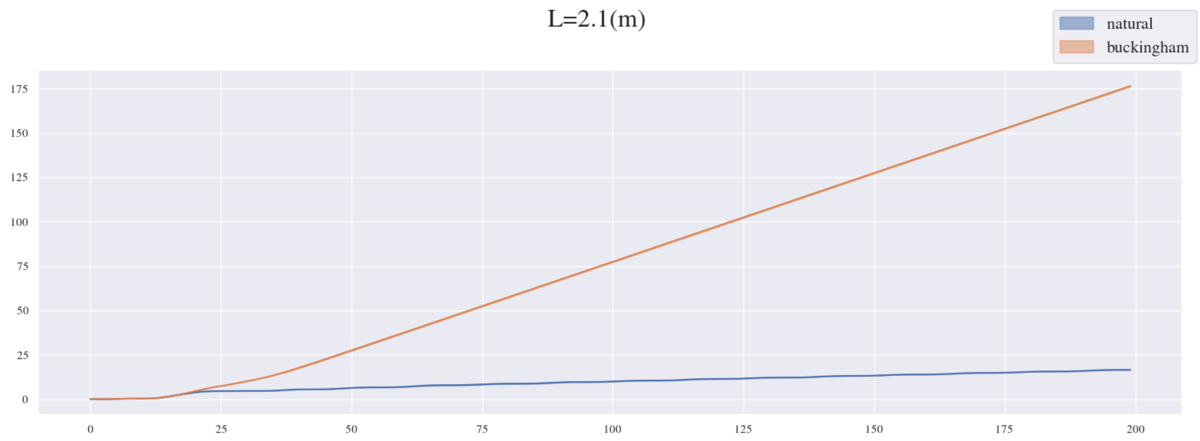


Figure 5.10: Cumulative reward on cartpole with $L = 2.1$.

$(0, 2, 4, 6, 8, 10)s$.

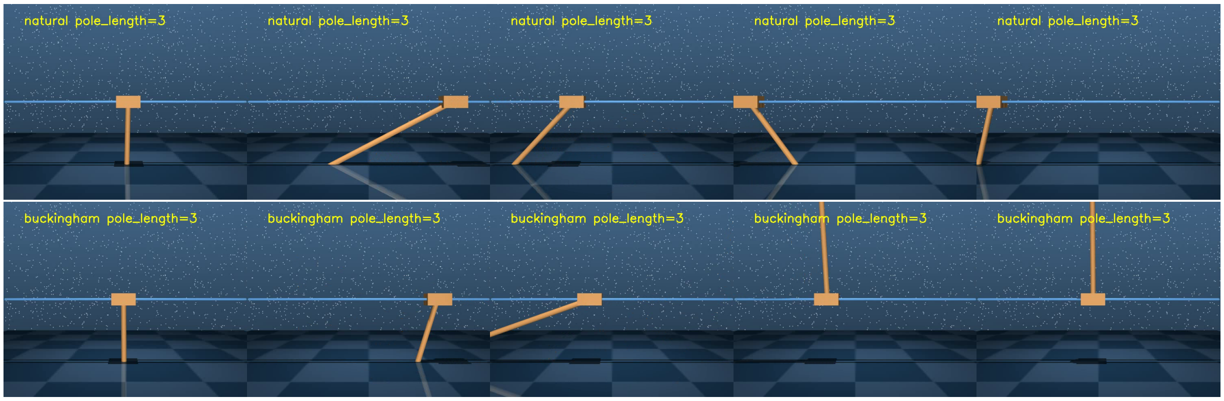


Figure 5.11: Evolution of the cartpole environment with a pole length $L = 1m$ in Mujoco. The leftmost frame is taken at $t = 0$ and the following frames are each spaced by 2 seconds. The pendulum starts with the pendulum initially vertically downright. The top row are the images for the natural controller and the bottom row for the dimensionless one.

Return 2D Perturbation

So far, we have only evaluated the generalization capabilities of the controllers with a shift in an atomic subset of the context. We now turn to the case where two parameters are perturbed at once. The context is therefore a 2-dimensional vector $\boldsymbol{c} \in \mathbb{R}^2$. For the pendulum on figure 5.12, both mass M and length L are perturbed around their nominal values. Bright colours indicate higher return meaning good generalization of the controller whereas dark ones indicate failure to stabilize the system. The same experiment for the cartpole is presented on figure 5.13.

The first thing we notice is that even in the natural case, the policy for the pendulum is already

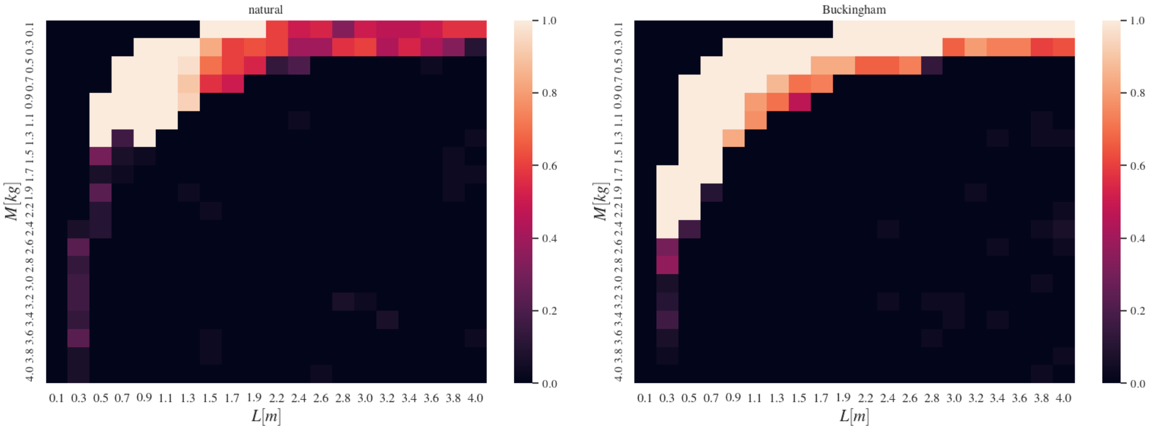


Figure 5.12: Pendulum success rates on the pole length when both M and L are varying for the natural (left) and dimensionless (right) controllers. Brighter values indicate higher rates.

able to generalize to a significant range of values. We hypothesize this is due to the probabilistic nature of the policy search which presents naturally robust capabilities [Charvet et al., 2021]. Nevertheless, the Buckingham transformation is able to enlarge this region, allowing large values of L when is small (below 0.5). It also allows larger M values, up to 2.4 when L is smaller than 0.9.

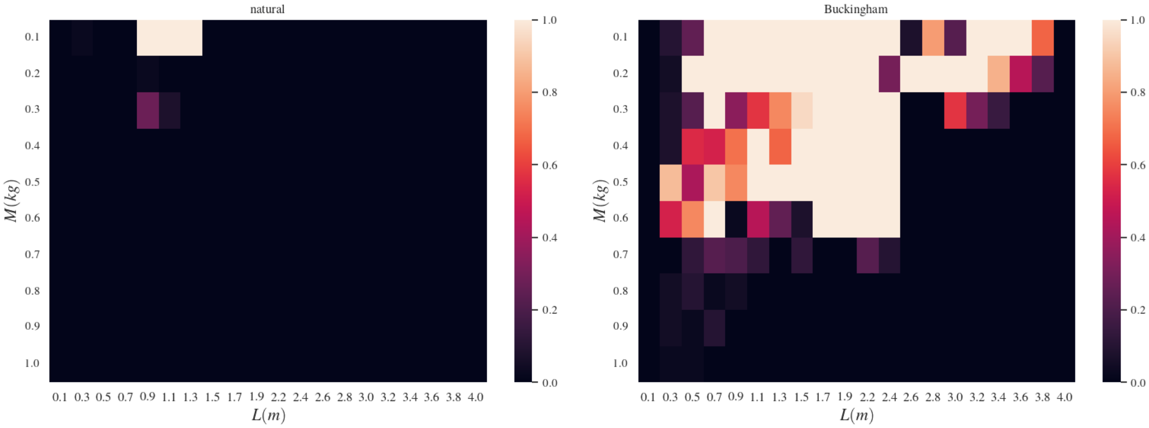
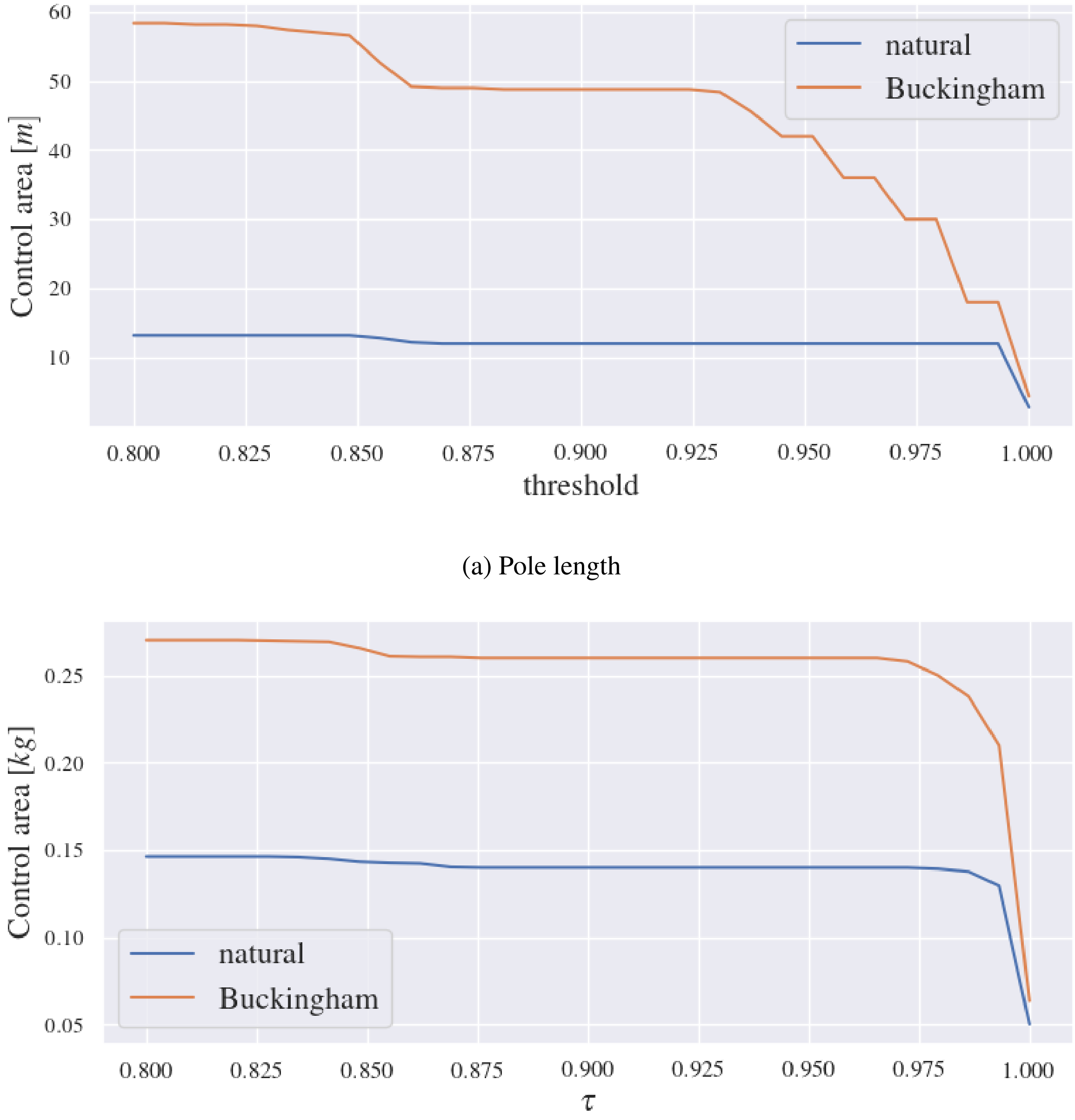


Figure 5.13: Cartpole success rates when both parameters L and M change simultaneously. We can see how the dimensionless controller (right) can solve the task on a much wider set of context pairs.

On the cart-pole however, the natural controller is very sensitive to small perturbation around the nominal value. As we see on the right plot of figure 5.13, only a very small region in the top left is demonstrating high returns. When using the Buckingham-Π features, this region is significantly widened. These results show that when two parameters are perturbed at once, the Buckingham transformation yields significant generalization performance.

5.3.2 Controllable Area



(a) Pole length

(b) Pole mass

Figure 5.14: Control area for pole length and mass. Higher values on the x-axis indicate better generalization.

Following the idea of complementing the return metric, we propose another metric that is specific to the problem of generalization and robustness. We call *controllable area* the surface in parameter space on which the performance of the controller drops by a given percentage $\tau \in [0, 1]$. The area can be mathematically described as follows,

$$\mathcal{S}_{control}(\tau) = \left\{ \boldsymbol{c} \in \mathcal{C},\ R_\pi(\boldsymbol{c}) \geq \tau R^*(\pi) \right\}. \tag{5.14}$$

This definition allows us to measure the region in context space in which the controller works close to its optimal regime. We can compute that value with means of an integral over the rate of episodes that have returns greater than the threshold for each infinitesimal context as,

$$\mathcal{S}_{control}(\tau) = \int_C \mathbb{1}[R_\pi(\boldsymbol{c}) \geq \tau R^*(\pi)] d\boldsymbol{c}. \tag{5.15}$$

We plot this area as a function of the performance dropoff τ on figure 5.14 for the pole mass and length. This figure confirms the findings from above as we can see the area of optimality of the controllers is much larger for the one in dimensionless space. This is confirmed by figure 5.15, on which both length and mass are perturbed. The resulting surface has the unit of $kg \times m$ (a mass times a length).

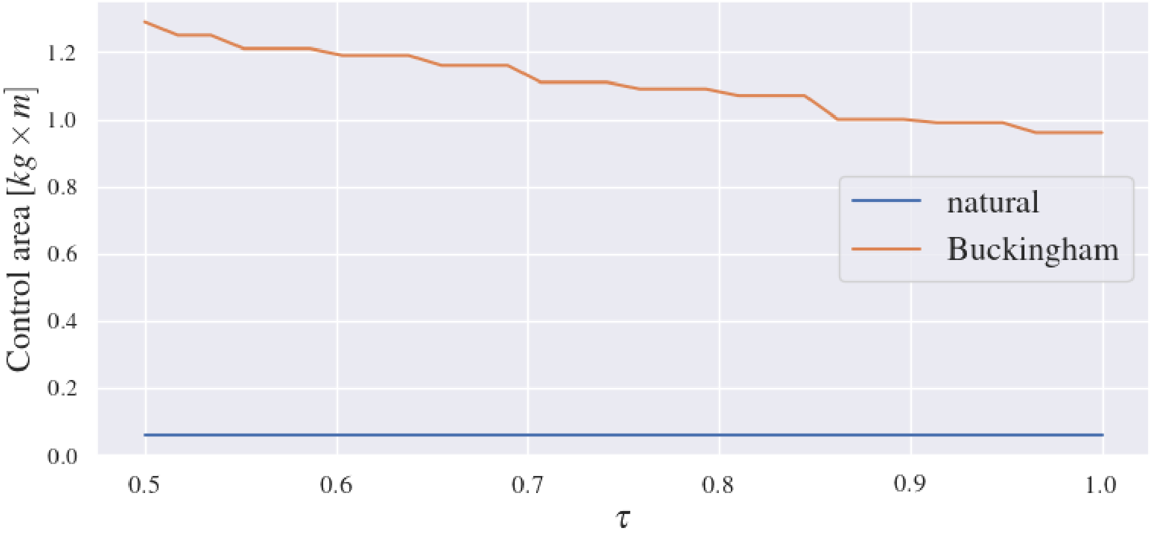


Figure 5.15: Area of control as a function of performance drop for 2D perturbations

Note that depending on the system at hand, the controllable region may not be compact set of the context space. It is a similar phenomenon that we observe on figure 5.13.

5.3.3 Optimal Performance and Constraints

It is worth noting at this point that what we call the optimal performance of a controller is conditioned by the constraints of the environment. In the specific case of the cartpole, these constraints take the form of a wall that the cart may hit at positions $x = \pm 2m$. The consequence is a discontinuity of the dynamics incurred by hitting the wall, which the model is not aware of when training on the nominal environment. That is because the permissible range of the cart allows naturally to swing the pendulum upright in that configuration. However, when the context takes extreme values the controller is physically denied the possibility to gather enough momentum to push the pendulum. This phenomenon is highlighted on figure 5.16. It shows that in the Buckingham case (left), the controller pushes the cartpole up to the wall when the context drifts too far from the nominal value. The problem is that upon contact, the dynamics change and cause an unexpected interaction between some variables.

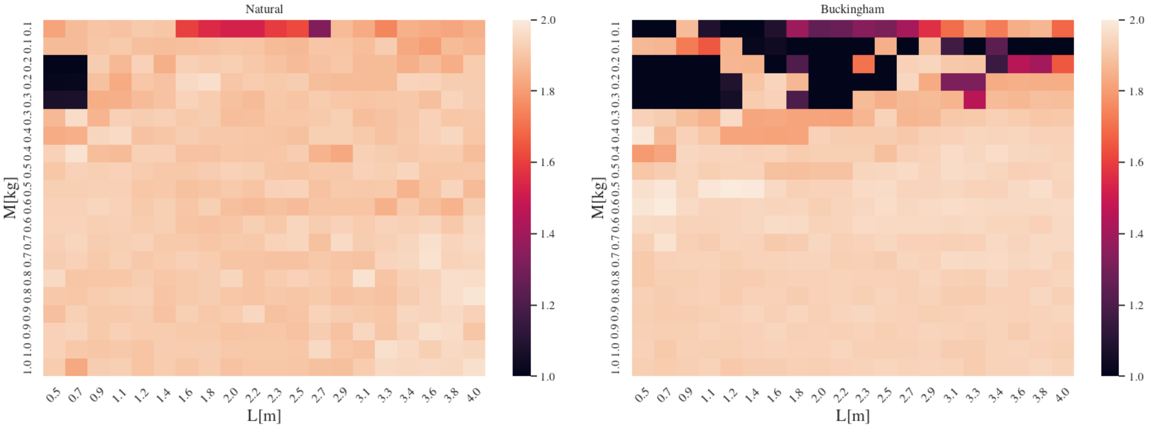


Figure 5.16: We plot the extreme horizontal $|x|$ positions on the cartpole along 10 episodes. The brightest values indicate that the wall is hit. We can see that the Buckingham controller (right) often reaches the constraint at $|x| = 2$ when the context is far from the nominal ($M = 0.1kg, L = 1m$).

On figure 5.17, we see what happens when those constraints are removed. We can see then that the dimensionless controller is able to solve the task more easily for large values of L.

We now illustrate the equivariance property of the dimensionless policy using data from the cartpole environment. The policy initially takes values in $\mathbb{R}$, it is then squashed into the domain $\mathcal{A}$ to allow the control signal to be passes into the environment.

$$\pi_\theta(s) = \max(\mathcal{A})\sigma\left(\pi\tilde{(s)}\right), \tag{5.16}$$

where in our case $\sigma(x) = \frac{1}{8}(9\sin(x) + \sin(3x))$. In this formulation of the actions, the initial policy $\tilde{\pi}$ outputs dimensionless vectors that then take the dimension given by the bound of the action set $\max\mathcal{A}$.

As a first experiment, we collect one trajectory with a natural and dimensionless controller on four different versions of the cartpole. Figure 5.18 highlights the influence of the context on the actions scaling. As the environment undergoes a scaling transformation of the context, the dimensionless control actions are also scaled accordingly. On the other hand, the natural controller is agnostic to the context change and thus not able to stabilize the cartpole and solve

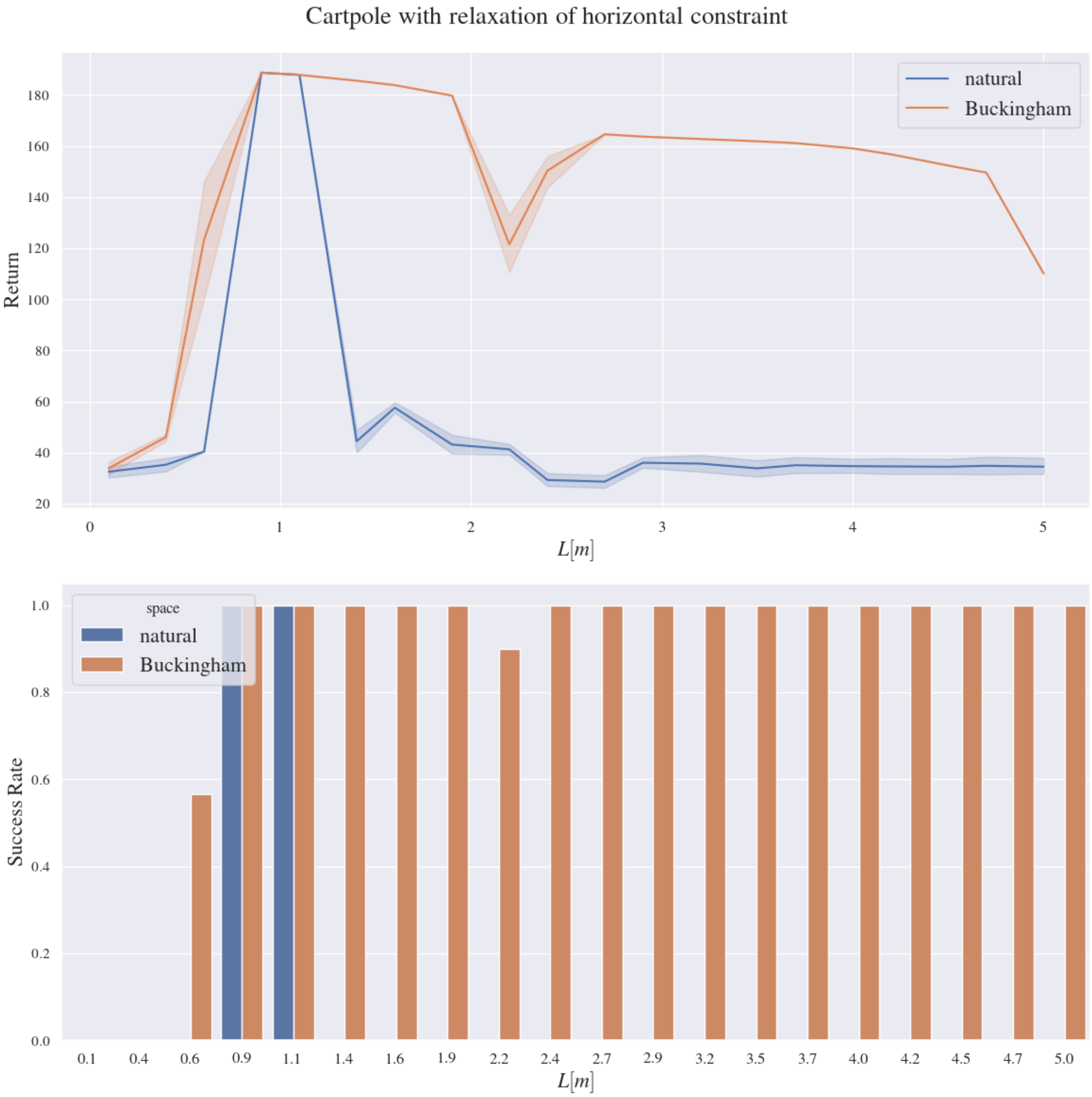


Figure 5.17: Plot of the return for the dimensionless controller when constraints are relaxed. The Buckingham controller is able to maintain good performance on most of the evalution context.

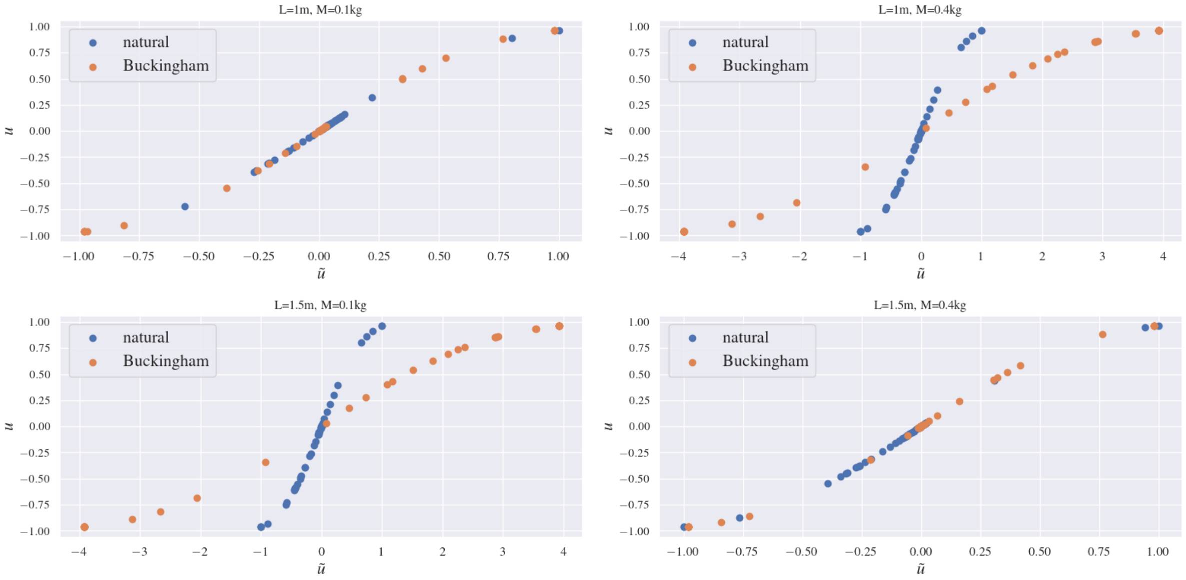


Figure 5.18: Comparing raw actions with their squashed equivalent for different context pairs (the nominal is on the top-left corner). The figure illustrates how the Buckingham transformation allows the policy to take extreme actions to adapt to the context-specific dynamics.

the task. This phenomenon is further highlighted on figure 5.19. Here we plot the natural against dimensionless controllers actions on the same sample of state data, but for different context. To do so, we sample a subset of 100 one-step transition from the data collected during training. We then plot the controller actions with the input state going through the Buckingham power-law transformation with appropriate context. As we can see, the Buckingham actions are rescaled to reflect the change in pole length. This ability to transform the controller input further explains how zero-shot generalization can be improved with no additional training data.

5.3.4 Parameter Identifiability

In the previous sections, we assumed the context was observed in order to transform the state-action space into a dimensionless one. This hypothesis might be constraining for many real-world problems, where accurate measures of some variables are not possible. We evaluate the possibility on inferring these variables in order to transform the input space. We note that this is not a trivial problem, since the data we record is recorded by a specific controller in closed-loop, which is known to prevent parameter identification [Ljung, 1989].

Similarly as in chapter 4, we roll out the trained controllers on a perturbed version of the environment to collect sample trajectories. We then sample parameters from a uniform prior and plot the likelihood of the hidden parameter under the Gaussian Process model. For the cartpole on figure 5.20, the log-likelihood surfaces for each observable $(x_\Pi, \dot{x}_\Pi, \cos(\theta), \sin(\theta), \dot{\theta}_\Pi)$

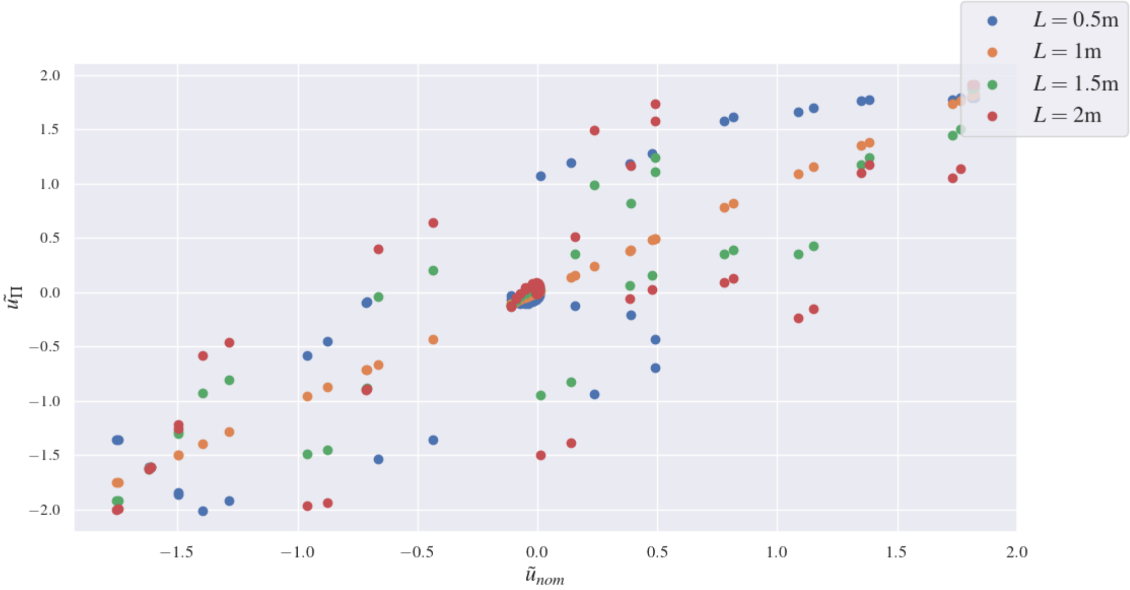


Figure 5.19: We plot the actions by the nominal controller (x-axis) against the perturbed ones (y-axis) for different pole lengths. All the controllers share the same natural input, but is transformed by the context-dependent Π-groups.

are not minimized where the true values are. One explanation is that whichever the context is, the optimal trajectories in phase planes are little impacted. As a consequence, the model is not able to accurately distinguish the correct parameters from the wrong ones.

5.4 Discussion

In this chapter, we investigated the problem of controller generalization when a dynamic system is subjected to environmental perturbations. We introduced the dimensionless Markov Decision Process in section 5.2.1, that allows an autonomous agent to take actions in a dimensionless observation space. The Π-MDP is a rescaling of a C-MDP state and actions spaces such that each variable becomes dimensionless. The resulting state-action space stems from additional assumptions about the units of the system and the observation of perturbing variables. The equivariance properties of the transformation allow *zero-shot transfer* from one context to the other.

From the Π-MDP formulation, we derived a generic framework for model-based policy search that we applied with a Gaussian Process dynamics model (algorithms 2 and 3). The new algorithm we proposed, is built on top of PILCO maintains its data-efficency and improves greatly its generalization capabilities with no further data collection.

We demonstrated empirically that this approach yields controllers that are invariant with

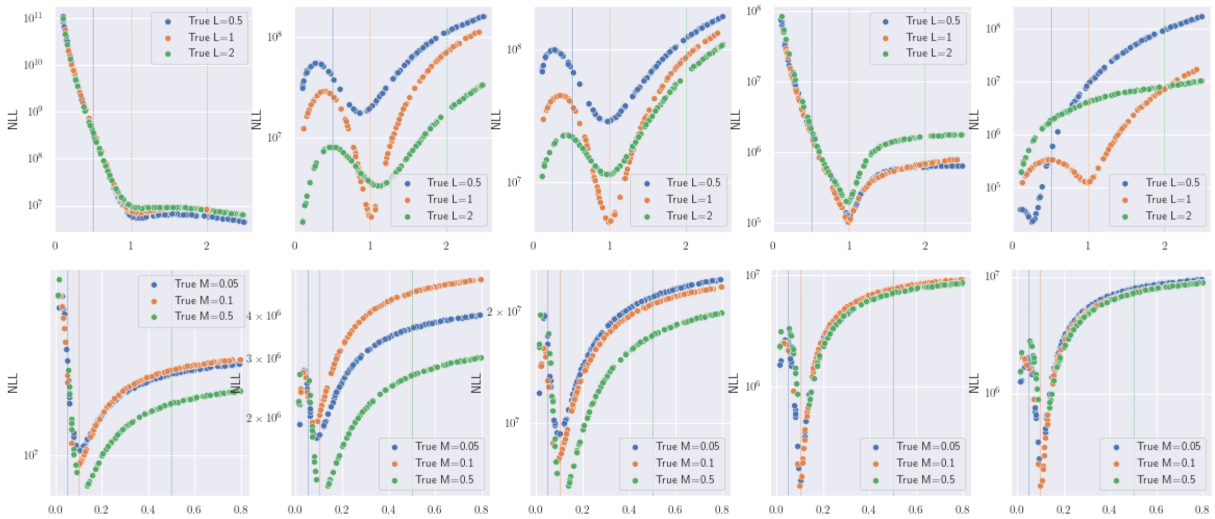


Figure 5.20: Negative Log-Likelihood for different samples of L (top) and M (bottom) on the cartpole. Each horizontal plot correspond to the NLL for each variable of the state space. As we can see, the model is not able to identify correctly the parameter that generated the trajectory.

respect to the context, provided it can be observed or measured. Our experiments focused on two different environments, an underactuated pendulum (figure 5.12) and a cartpole (figure 5.13). Our results show strong generalization properties of the controller when the physical properties of the system such as pole length and mass drift from their initial training value. While these are simple systems, because of their second-order dynamics and low dimension, the consistency of the results suggest the methodology could be successfully applied to more complex systems, which we leave to future work.

Conceptually, our approach comes within the scope of instilling physics prior in Machine Learning pipelines to increase model robustness [Botev et al., 2021]. The main weakness of this approach is the requirement for measuring what the perturbation variables are at any point in the deployment of the controller. Relaxing this assumption proved to be difficult because of the closed-loop nature of the problem, which is known to prevent identifiability of the parameters [Ljung, 1989], as we can see on figure 5.20. We believe identification of the parameters could be achieved with different control policies that aim to actively infer those values based on exploratory trajectories, and leave this direction for future work. The second limitation of this approach is the requirement for knowing the measurements dimensions which can be prohibitively expensive on high-dimensional systems. To alleviate this, one could either use physical priors to determine which transformation if most suited to type of perturbation that might be later encountered.

Chapter 6.

Conclusion

In this thesis, we examined the problem of distribution shift in dynamic systems through the lens of confounding variables acting on inputs and outputs. We call them *context* and it includes all the static variables that are present in the Ordinary Differential Equation that drives the system's temporal evolution. In the course of the deployment of an agent, this context may be subjected to modifications caused by external perturbations or hardware wear-and-tear. We considered the case where any of these perturbations are slow compared to the temporal evolution of the system such that within an episode, the system can be considered stationary. When the context is modified, the data generated in the environment will suffer a distribution shift. The problem of modelling accurately such systems in the presence of these perturbations is therefore one of *generalization*. It poses the question of making accurate predictions or taking optimal decisions outside the training domain. Conversely, a model is called *robust* when it is resilient to those perturbations. Data-driven approaches to solving this issue involve gathering additional data in the training phase to reflect the diversity of geometries incurred by the confounders. Data augmentation then allows either creating a general model that can interpolate between domains or a meta-learning model that can separate the local and global system properties. This type of approach has encountered great success, but comes with higher training and data storage cost as well. Furthermore, these methods lie on the ability to intervene on the context during training to gather data from several variants of the environment. This can only be achieved if a simulator is available because in general interventions on real systems can not be done at the risk of damaging the system. Throughout this thesis, we assumed that during training we only have access to a single version of the environment. Thus, any method based on the augmentation of training data is not permitted.

Alternatively, generalization may be improved by augmenting the set of assumptions on the training and testing distribution depending on the task. In the same way inductive biases are used to augment capacity without increase in complexity, domain-specific knowledge can be included in the modelling procedure to increase robustness. This thesis explores the ability of a variable transformation given by the Buckingham-Π theorem to create *equivariant* estimators. That is, estimators that can appropriately rescale their predictions or decisions on the basis of a transformation of its inputs. The theorem stems from the field of Dimensional Analysis and derives a dimensionality reduction by exploiting the symmetries incurred by the system of units.

This transformation requires the knowledge of the dimension of each measurement in order to make them invariant to a change in unit through a power law. By combining this approach with probabilistic Machine Learning models we are able to increase the robustness of estimators with respect to context perturbations.

In chapter 3, we adapt the Buckingham-Π theorem to second-order dynamic systems that are conditioned by a set of static physical variables. We showed that under the full-rank assumption of the context vector on the basis of the elementary physical dimensions we can project the dynamic variables into a dimensionless space. By construction, the models in that space are equivariant with respect to rescaling of the context. This transformation allows statistical estimators to make accurate predictions outside the training data support even when the context is poorly measured. The strength of the approach lies in the ability of models to generalize even when the training context is atomic (ie a single vector $\boldsymbol{c}_0$) and uncertain. Additionally, the transformation is oblivious to the estimator, so we have been able to apply it to a Neural Network as well as Gaussian Process models with Maximum A Posteriori and Variational Inference. While it requires the additional knowledge of the context, the transformation is beneficial since a natural model with the same access to all the variables is not able to generalize as well because of the form of the nominal context. However, the results of this chapter should be not be overstated since they have been obtained in simulation on a fairly simple environment, an actuated pendulum. We believe they should be confirmed on higher dimensional systems and other domains such as thermodynamics and electricity where mass, length and time do not constitute a sufficient basis of dimensions. Moreover, the Π-groups used for the transformation of the state space are not unique. The ones we used were found by trial-and-error and informed by physics intuition. We believe that understanding which Π-groups are optimal for a given system is an important step for extending this work to more complex and critical systems.

In chapter 4, we propose a solution for one of the main weaknesses of the Buckingham transformation, namely the requirement of observing the context variables to construct the appropriate transformation. Our contribution is a dimensional latent variable model, where the inferred variables are constrained to take values in a units-typed space. We approach that problem in two different ways. In the first, we build a predictive model on the Π-groups assuming the whole context is observed. In the second, the context is hidden, so the model is trained as we infer the hidden variables. Then, when facing a perturbed version of the system, we re-infer the latent variables whilst retaining the predictive model. In opposition with other latent variable models, we impose a dimension (in the sense of units) to each latent variable through the Buckingham transformation. This preserves the equivariant property of the predictive model and allows it to generalize to new context. At the same time, the learned latent dimensional variables can be used to estimate the true value of the hidden parameter. Doing so however requires at least one observation of the context during training. If it is not the case, we can however use the model to predict if the elements of new context are scaled-up or scaled-down versions of the nominal. Constraining latent variables to a specific dimension constitutes a promising research direction for making this type of model more explainable and transparent. For instance, some areas of research, such as medicine require model transparency [Winter and Carusi, 2023; Rubinger et al., 2023], may be reluctant to use architectures like Variational AutoEncoders due to the opacity of their latent space. Enforcing dimensional constraints with the Buckingham theorem could be a partial solution to that problem. We also believe an interesting question for future investigations is the number of training contexts required to infer the dimensional latent variables correctly. Because of the equivariance property of the dimensionless estimators, a single context should be enough in principles. In practice we have focused our work on atomic contexts and found it insufficient to learn latent space with more than two dimensions. Moreover, it will be important in the future to understand what causes the variance collapse in the latent variable posteriors. While it does not prevent the models to make accurate predictions, it might constitute an obstacle for few-shot transfer in higher dimensional spaces.

In chapter 5, we apply our previous findings for improving the robustness of model-based controllers to distribution shift. We introduce the concept of dimensionless Markov Decision Processes (Π-MDP), in which the state and action spaces are non-dimensionalized using appropriate Π-groups. A Π-MDP can be seen as the reparametrization of a Contextual-MDP through the power law given by the Buckingham theorem. We adapted the PILCO model-based Reinforcement Learning routine for iterating the policy search within that Π space. Our empirical results demonstrate that control policies trained in that way are able to generalize to a large range of testing contexts even when they are trained on a single nominal environment. The strength of our approach to Robust Reinforcement Learning is its conceptual simplicity. While we have only tested the Π-MDP with PILCO, we believe it can be extended to other model-based and model-free algorithms as long as the Π-groups exist. Demonstrating this extension on other algorithms should constitute a straightforward future research direction, and we made our code publicly available to help in that regard. Nevertheless, we have not been able to apply the dimensional latent variable models from chapter 4 to construct a Π-MDP when the context is hidden. We hypothesize it may be alleviated by training the model and controllers on several contexts instead of one. Alternatively, active exploration routines could be put in place with the specific objective to estimate the latent variables at test time [Memmel et al., 2024; Colas et al., 2019; ASMUTH, 2009; Liu et al., 2023]. Finally, we think that training Offline Reinforcement Learning algorithms in a Π-MDP could greatly increase their performances.

The overall directing contribution of this work is an approach to zero-shot transfer learning, that we achieved by means of building equivariant feature spaces for the estimator. As we could see, this methodology benefits from the conceptual simplicity to instil a physics-driven inductive bias that is agnostic to the choice of machine-learning model. As such, our approach could qualify as "grey-box" As we could see, this methodology benefits from the conceptual simplicity to instil a physics-driven inductive bias that is agnostic to the choice of machine-learning model. As such, our approach could be qualified of "gray-box" [Liu et al., 2021]. It means a mixing of data-driven and physics based models to benefit from flexibility and scalability on the one hand and generality on the other. While the need for measuring the context variables could be seen as a weakness compared to fully data-driven approaches, we believe it is a small price to pay with respect to the gains associated with model complexity and computing costs. The solution we propose in chapter 4 to relax this observability assumption constitutes a promising first result but needs to be evaluated within more complex environments. On the pendulum and cartpole, Dimensional Analysis brings enough constraints to make the modelling significantly easier. The scalability of this approach in terms of the number of dependant variables is an open question that we leave for future work. For more complex systems, additional issues may arise. The first comes from the non-uniqueness of the Π-groups: as the number of variables increases so does the number of homogeneously acceptable transformations. It follows that higher dimensional systems require additional expert knowledge and trial-and-error to find the optimal solution. The direct consequence is that the procedure may become more model-driven than data-driven and lose its benefits as a statistical learning method. Numerical approaches to non-dimensionalization such as [Bakarji et al., 2022] may constitute a natural way to maintain a data-centric approach. Second, the Buckingham theorem relies on the knowledge of all the dimensional variables that appear in the graphical model of the system. While we proposed a solution to deal with hidden variables in chapter 4, we have not studied the impact of the absence (or presence) of variables in the Π that should in fact be present (or absent). The validity of the selected variables is only given by first principles and expert knowledge. Therefore, the development of automated routines for selecting and validating the variables and their Π-groups will be an essential milestone for the deployment of such tools into more critical environments.

Bibliography

I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang. Solving Rubik's cube with a robot hand, 2019. URL https://arxiv.org/abs/1910.07113.

F. Amadio, A. Dalla Libera, R. Antonello, D. N. Nikovski, R. Carli, and D. Romeres. Model-based policy search using Monte Carlo gradient estimation with real systems application. *IEEE Transaction on Robotics*, 38(6):3879–3898, Dec. 2022. ISSN 1941-0468. doi: 10.1109/TRO.2022.3184837. URL https://www.merl.com/publications/TR2022-154.

M. Arjovsky. *Out of distribution generalization in machine learning*. PhD thesis, New York University, 2020.

K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath. Deep Reinforcement Learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):2638, Nov. 2017. ISSN 1053-5888. doi: 10.1109/msp.2017.2743240. URL http://dx.doi.org/10.1109/MSP.2017.2743240.

J. ASMUTH. A Bayesian sampling approach to exploration in Reinforcement Learning. *Proceedings of The 25th Conference on Uncertainty in Artificial Intelligence (UAI-09), June*, 2009.

J. Bakarji, J. Callaham, S. L. Brunton, and J. N. Kutz. Dimensionally consistent learning with Buckingham Pi. *Nature Computational Science*, 2(12):834–844, 2022.

P. J. Ball, C. Lu, J. Parker-Holder, and S. Roberts. Augmented world models facilitate zero-shot dynamics generalization from a single offline environment. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 619–629. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/ball21a.html.

W. Barnett. Dimensions and economics: some problems. *The quarterly journal of Austrian economics*, 7(1):95–104, 2004.

A. G. Barto. *Intrinsic Motivation and Reinforcement Learning*, pages 17–47. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-32375-1. doi: 10.1007/978-3-642-32375-1_2. URL https://doi.org/10.1007/978-3-642-32375-1_2.

R. Bellman. Adaptive control processes: A guided tour. *SIAM Review*, 4(2):163–163, 1962.

R. Bellman and R. E. Kalaba. *Dynamic programming and modern control theory*, volume 81. Academic Press New York, 1965.

E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. A. Szerlip, P. Horsfall, and N. D. Goodman. Pyro: Deep universal probabilistic programming. *J. Mach. Learn. Res.*, 20:28:1–28:6, 2019. URL http://jmlr.org/papers/v20/18-403.html.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017. ISSN 1537274X. doi: 10.1080/01621459.2017.1285773. URL https://doi.org/10.1080/01621459.2017.1285773.

J. K. Blitzstein. Sir David R. Cox: A Beautiful Mind With a Beautiful Heart. *Harvard Data Science Review*, 5(2), apr 27 2023. https://hdsr.mitpress.mit.edu/pub/hhcrzne1.

M. I. Board. Mars climate orbiter mishap investigation board phase i report november 10, 1999. Technical report, 1999.

A. Botev, A. Jaegle, P. Wirnsberger, D. Hennes, and I. Higgins. Which priors matter? benchmarking models for learning latent dynamics. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/f033ab37c30201f73f142449d037028d-Paper-round1.pdf.

E. Buckingham. On physically similar systems; illustrations of the use of dimensional equations. *Physical review*, 4(4):345, 1914.

E. Burnaev, M. Panov, and A. Zaytsev. Regression on the basis of nonstationary Gaussian processes with Bayesian regularization. *Journal of communications technology and electronics*, 61(6):661–671, 2016.

A. Chandra, J. Bakarji, and D. Tartakovsky. Role of physics in physics-informed machine learning. *Journal of Machine Learning for Modeling and Computing*. doi: 10.1615/jmachlearnmodelcomput.2024053170.

V. Charvet, B. S. Jensen, and R. Murray-Smith. Learning robust controllers via probabilistic model-based policy search. International Conference on Learning Representations - Robust ML Workshop, 2021. doi: 10.48550/ARXIV.2110.13576.

Z. Chen and B. Wang. How priors of initial hyperparameters affect Gaussian Process regression models. *Neurocomputing*, 275:1702–1710, 2018. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2017.10.028.

Z. Chen and Y. Yang. Assessing forecast accuracy measures. *Preprint Series*, 2010:2004–10, 2004.

K. Chua, R. Calandra, R. McAllister, and S. Levine. Deep Reinforcement Learning in a handful of trials using probabilistic dynamics models. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/3de568f8597b94bda53149c7d7f5958c-Paper.pdf.

C. Colas, P. Founder, O. Sigaud, M. Chetouani, and P. Y. Oudeyer. CURIOUS: Intrinsically motivated modular multi-goal Reinforcement Learning. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:2372–2387, 2019.

A. I. Cowen-Rivers, D. Palenicek, V. Moens, M. A. Abdullah, A. Sootla, J. Wang, and H. Bou-Ammar. Samba: Safe model-based & active reinforcement learning. *Machine Learning*, 111 (1):173–203, 2022.

L. Csató and M. Opper. Sparse On-Line Gaussian Processes. *Neural Computation*, 14(3):641–668, 03 2002. ISSN 0899-7667. doi: 10.1162/089976602317250933. URL https://doi.org/10.1162/089976602317250933.

J. Degrave, F. Felici, J. Buchli, M. Neunert, B. Tracey, F. Carpanese, T. Ewalds, R. Hafner, A. Abdolmaleki, D. de Las Casas, et al. Magnetic control of tokamak plasmas through deep Reinforcement Learning. *Nature*, 602(7897):414–419, 2022.

M. P. Deisenroth and C. E. Rasmussen. PILCO: A Model-Based and Data-Efficient Approach to Policy Search. In *International Conference on Machine Learning*. 467-472, 2011. doi: 10.1080/0034408960910404.

E. Derman, D. Mankowitz, T. Mann, and S. Mannor. A Bayesian approach to robust Reinforcement Learning. In R. P. Adams and V. Gogate, editors, *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pages 648–658. PMLR, 22–25 Jul 2020.

F. Doshi-Velez and G. Konidaris. Hidden parameter markov decision processes: A semiparametric regression approach for discovering latent task parametrizations. In *IJCAI: proceedings of the conference*, volume 2016, page 1432. NIH Public Access, 2016.

G. Dulac-Arnold, N. Levine, D. J. Mankowitz, J. Li, C. Paduraru, S. Gowal, and T. Hester. Challenges of real-world Reinforcement Learning: definitions, benchmarks and analysis. *Machine Learning*, 110(9):2419–2468, 2021a.

G. Dulac-Arnold, N. Levine, D. J. Mankowitz, J. Li, C. Paduraru, S. Gowal, and T. Hester. An empirical investigation of the challenges of real-world Reinforcement Learning, 2021b.

B. Eysenbach and S. Levine. Maximum Entropy RL (provably) solves some robust RL problems. In *International Conference on Learning Representations*, 2021.

M. Ghavamzadeh, S. Mannor, J. Pineau, A. Tamar, et al. Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 8(5-6):359–483, 2015.

M. Ghavamzadeh, Y. Engel, and M. Valko. Bayesian policy gradient and actor-critic algorithms. *Journal of Machine Learning Research*, 17(66):1–53, 2016. URL http://jmlr.org/papers/v17/10-245.html.

D. Ghosh, J. Rahme, A. Kumar, A. Zhang, R. P. Adams, and S. Levine. Why generalization in rl is difficult: Epistemic pomdps and implicit partial observability. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 25502–25515. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/d5ff135377d39f1de7372c95c74dd962-Paper.pdf.

A. Girard. Dimensionless policies based on the buckingham theorem: Is this a good way to generalize numerical results? *Mathematics*, 12(5), 2024. ISSN 2227-7390. doi: 10.3390/math12050709. URL https://www.mdpi.com/2227-7390/12/5/709.

A. Girard, C. Rasmussen, J. Q. n. Candela, and R. Murray-Smith. Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2002. URL https://proceedings.neurips.cc/paper_files/paper/2002/file/f3ac63c91272f19ce97c7397825cc15f-Paper.pdf.

A. Hallak, D. D. Castro, and S. Mannor. Contextual Markov Decision Processes, 2015.

J. B. Hamrick. Analogues of mental simulation and imagination in deep learning. *Current Opinion in Behavioral Sciences*, 29:8–16, 2019. ISSN 2352-1546. doi: https://doi.org/10.1016/j.cobeha.2018.12.011. Artificial Intelligence.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition n*, volume 77. 2009. doi: 10.1111/j.1751-5823.2009.00095_18.x. Publication Title: International Statistical Review Issue: 3 ISSN: 03067734.

J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian Processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI'13, page 282290, Arlington, Virginia, USA, 2013. AUAI Press.

J. Hensman, A. Matthews, and Z. Ghahramani. Scalable Variational Gaussian Process Classification. In G. Lebanon and S. V. N. Vishwanathan, editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 351–360, San Diego, California, USA, 09–12 May 2015. PMLR. URL https://proceedings.mlr.press/v38/hensman15.html.

T. Hishinuma and K. Senda. Weighted model estimation for offline model-based Reinforcement Learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 17789–17800. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/949694a5059302e7283073b502f094d7-Paper.pdf.

M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013. ISSN 15324435. arXiv: 1206.7051.

M. D. Homan and A. Gelman. The No-U-Turn Sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.*, 15(1):15931623, jan 2014. ISSN 1532-4435.

Z.-W. Hong, A. Kumar, S. Karnik, A. Bhandwaldar, A. Srivastava, J. Pajarinen, R. Laroche, A. Gupta, and P. Agrawal. Beyond uniform sampling: Offline Reinforcement Learning with imbalanced datasets. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 4985–5009. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/0ff3502bb29570b219967278db150a50-Paper-Conference.pdf.

P. Huang, X. Zhang, Z. Cao, S. Liu, M. Xu, W. Ding, J. Francis, B. Chen, and D. Zhao. What went wrong? closing the sim-to-real gap via differentiable causal discovery. In J. Tan, M. Toussaint, and K. Darvish, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 734–760. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/huang23c.html.

M. Igl, K. Ciosek, Y. Li, S. Tschiatschek, C. Zhang, S. Devlin, and K. Hofmann. Generalization in Reinforcement Learning with selective noise injection and information bottleneck. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, 2019.

M. Janner, J. Fu, M. Zhang, and S. Levine. When to trust your model: Model-based policy optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/5faf461eff3099671ad63c6f3f094f7f-Paper.pdf.

K. Kansky, T. Silver, D. A. Mély, M. Eldawy, M. Lázaro-Gredilla, X. Lou, N. Dorfman, S. Sidor, S. Phoenix, and D. George. Schema networks: Zero-shot transfer with a generative causal model of intuitive physics. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1809–1818. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/kansky17a.html.

R. Kidambi, A. Rajeswaran, P. Netrapalli, and T. Joachims. Morel: Model-based offline Reinforcement Learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21810–21823. Curran Associates, Inc., 2020.

B. Kim and M.-H. Oh. Model-based offline Reinforcement Learning with count-based conservatism. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 16728–16746. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kim23q.html.

D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representaions*, 2014.

R. Kirk, A. Zhang, E. Grefenstette, and T. Rocktäschel. A survey of zero-shot generalisation in deep Reinforcement Learning. *Journal of Artificial Intelligence Research*, 76:201–264, 2023.

N. Kumar, P. Rajagopalan, P. Pankajakshan, A. Bhattacharyya, S. Sanyal, J. Balachandran, and U. V. Waghmare. Machine learning constrained with dimensional analysis and scaling laws: simple, transferable, and interpretable models of materials from small datasets. *Chemistry of Materials*, 31(2):314–321, 2018.

A. Kupcsik, M. Deisenroth, J. Peters, and G. Neumann. Data-efficient generalization of robot skills with contextual policy search. In *Proceedings of the AAAI conference on artificial intelligence*, volume 27, pages 1401–1407, 2013.

J. Kwon, Y. Efroni, C. Caramanis, and S. Mannor. RL for latent MDPs: Regret guarantees and a lower bound. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 24523–24534. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/cd755a6c6b699f3262bcc2aa46ab507e-Paper.pdf.

V. Lalchand and C. E. Rasmussen. Approximate inference for fully Bayesian Gaussian Process regression. In C. Zhang, F. Ruiz, T. Bui, A. B. Dieng, and D. Liang, editors, *Proceedings of The 2nd Symposium on Advances in Approximate Bayesian Inference*, volume 118 of *Proceedings of Machine Learning Research*, pages 1–12. PMLR, 08 Dec 2020.

V. Lalchand, W. Bruinsma, D. Burt, and C. E. Rasmussen. Sparse Gaussian Process hyperparameters: Optimize or integrate? In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 16612–16623. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/69c49f75ca31620f1f0d38093d9f3d9b-Paper-Conference.pdf.

A. Lavin, H. Zenil, B. Paige, D. Krakauer, J. Gottschlich, T. Mattson, A. Anandkumar, S. Choudry, K. Rocki, A. G. Baydin, C. Prunkl, B. Paige, O. Isayev, E. Peterson, P. L. McMahon, J. Macke, K. Cranmer, J. Zhang, H. Wainwright, A. Hanuka, M. Veloso, S. Assefa, S. Zheng, and A. Pfeffer. Simulation intelligence: Towards a new generation of scientific methods, 2021.

N. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: The informative vector machine. *Advances in Neural Information Processing Systems*, pages 1–8, 2003. ISSN 10495258. ISBN: 0262025507.

T. Y. Lee, J. V. Zidek, and N. Heckman. Dimensional analysis in statistical modelling. *arXiv preprint arXiv:2002.11259*, 2021.

S. Levine, A. Kumar, G. Tucker, and J. Fu. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. May 2020.

H.-H. Liu, J. Zhang, F. Liang, C. Temizel, M. A. Basri, and R. Mesdour. Incorporation of Physics into Machine Learning for Production Prediction from Unconventional Reservoirs: A Brief Review of the Gray-Box Approach. *SPE Reservoir Evaluation & Engineering*, 24(04): 847–858, 11 2021. ISSN 1094-6470. doi: 10.2118/205520-PA. URL https://doi.org/10.2118/205520-PA.

Z. Liu, M. Lu, W. XIONG, H. Zhong, H. Hu, S. Zhang, S. Zheng, Z. Yang, and Z. Wang. Maximize to explore: One objective function fusing estimation, planning, and exploration. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 22151–22165. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/4640d5da5888238b9de7e0dbacd2c605-Paper-Conference.pdf.

L. Ljung. System identification-Theory for the user. 25(3):475–476, 1989. ISSN 00051098. doi: 10.1016/0005-1098(89)90019-8.

D. J. MacKay. *Bayesian Methods for Adaptive Models*. PhD thesis, California Institute of Technology, 1991.

D. Mankowitz, T. Mann, P.-L. Bacon, D. Precup, and S. Mannor. Learning robust options. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

D. J. Mankowitz, N. Levine, R. Jeong, A. Abdolmaleki, J. T. Springenberg, Y. Shi, J. Kay, T. Hester, T. Mann, and M. Riedmiller. Robust Reinforcement Learning for continuous control with model misspecification. In *International Conference on Learning Representations*, 2019.

M. Memmel, A. Wagenmaker, C. Zhu, D. Fox, and A. Gupta. ASID: Active exploration for system identification in robotic manipulation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jNR6s6OSBT.

T. M. Mitchell. The need for biases in learning generalizations. Technical report, Computer Science Department, Rutgers University, New Brunswick, MA, 1980.

J. Mitton. *Robustness, scalability and interpretability of equivariant neural networks across different low-dimensional geometries*. PhD thesis, University of Glasgow, 2023.

S. Mohamed, M. Rosca, M. Figurnov, and A. Mnih. Monte Carlo gradient estimation in machine learning. *Journal of Machine Learning Research*, 21(132):1–62, 2020. URL http://jmlr.org/papers/v21/19-346.html.

P. Molyneux. The dimensions of logarithmic quantities: Implications for the hidden concentration and pressure units in ph values, acidity constants, standard thermodynamic functions, and standard electrode potentials. *Journal of Chemical Education*, 68(6):467, 1991.

J. Morimoto and K. Doya. Robust Reinforcement Learning. *Neural computation*, 17(2):335–359, 2005.

D. Muglich, C. Schroeder de Witt, E. van der Pol, S. Whiteson, and J. Foerster. Equivariant networks for zero-shot coordination. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 6410–6423. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/29e4b51d45dc8f534260adc45b587363-Paper-Conference.pdf.

R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical report, Department of Computer Science, University of Toronto, 1993.

R. M. Neal. *Bayesian Learning for Neural Networks*. PhD thesis, University of Toronto, 1994.

M. W. Oppenheimer, D. B. Doman, and J. D. Merrick. Multi-scale physics-informed machine learning using the Buckingham pi theorem. *Journal of Computational Physics*, 474:111810, 2023. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2022.111810. URL https://www.sciencedirect.com/science/article/pii/S0021999122008737.

C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of Markov Decision Processes. *Mathematics of operations research*, 12(3):441–450, 1987.

P. Parmas, C. E. Rasmussen, J. Peters, and K. Doya. PIPPS: Flexible model-based policy search robust to the curse of chaos. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4065–4074. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/parmas18a.html.

L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta. Robust adversarial Reinforcement Learning. In *International Conference on Machine Learning*, pages 2817–2826, 2017.

J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 12 2008. ISBN 9780262255103. doi: 10.7551/mitpress/9780262170055.001.0001. URL https://doi.org/10.7551/mitpress/9780262170055.001.0001.

R. Ranganath, D. Tran, and D. Blei. Hierarchical variational models. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 324–333, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/ranganath16.html.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 11 2005. ISBN 9780262256834. doi: 10.7551/mitpress/3206.001.0001. URL https://doi.org/10.7551/mitpress/3206.001.0001.

A. G. B. Richard S. Sutton. *Reinforcement Learning: An Introduction*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2018. Publication Title: Encyclopedia of Neuroscience.

S. Rossi, M. Heinonen, E. Bonilla, Z. Shen, and M. Filippone. Sparse Gaussian Processes revisited: Bayesian approaches to inducing-variable approximations. In A. Banerjee and K. Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 1837–1845. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/rossi21a.html.

L. Rubinger, A. Gazendam, S. Ekhtiari, and M. Bhandari. Machine learning and artificial intelligence in research and healthcare. *Injury*, 54:S69–S73, 2023. ISSN 0020-1383. doi: https://doi.org/10.1016/j.injury.2022.01.046. URL https://www.sciencedirect.com/science/article/pii/S0020138322000766. AOTrauma Europe Supplement: Clinical Research: Lessons Learned-Looking Ahead.

S. Sæmundsson, K. Hofmann, and M. Deisenroth. Meta Reinforcement Learning with latent variable Gaussian processes. In *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, volume 34, pages 642–652. Association for Uncertainty in Artificial Intelligence (AUAI), 2018.

J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, et al. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison. Hidden technical debt in machine learning systems. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/86df7dcfd896fcaf2674f757a2463eba-Paper.pdf.

B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016. doi: 10.1109/JPROC.2015.2494218.

W. Shen. *Dimensional analysis in statistics: theories, methodologies and applications*. PhD thesis, The Pennsylvania State University, 2015.

W. Shen and D. K. J. Lin. A conjugate model for dimensional analysis. *Technometrics*, 60(1): 79–89, 2018. doi: 10.1080/00401706.2017.1291451. URL https://doi.org/10.1080/00401706.2017.1291451.

W. Shen and D. K. J. Lin. Statistical theories for dimensional analysis. *Statistica Sinica*, 29 (2):527–550, 2019. ISSN 10170405, 19968507. URL https://www.jstor.org/stable/26705477.

E. Snelson and Z. Ghahramani. Sparse Gaussian Processes using pseudo-inputs. *Advances in neural information processing systems*, 18, 2005.

A. A. Sonin. Dimensional analysis. Technical report, Department of Mechanical Engineering - MIT, 2001.

A. Tamar, S. Mannor, and H. Xu. Scaling up robust MDPs using function approximation. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 181–189, Bejing, China, 22–24 Jun 2014.

M. A. Texocotitla, M. D. Alvarez-Hernández, and S. E. Alvarez-Hernández. Dimensional analysis in economics: A study of the neoclassical economic growth model. *Journal of Interdisciplinary Economics*, 32(2):123–144, 2020. doi: 10.1177/0260107919845269. URL https://doi.org/10.1177/0260107919845269.

W. Therrien, O. Lecompte, and A. Girard. Using the Buckingham theorem for multi-system transfer learning: A case-study with 3 vehicles sharing a database. *Electronics*, 13(11), 2024. ISSN 2079-9292. doi: 10.3390/electronics13112041. URL https://www.mdpi.com/2079-9292/13/11/2041.

M. Titsias. Variational Learning of Inducing Variables in Sparse Gaussian Processes. In *AISTATS*, pages 567–574, Apr. 2009. URL http://proceedings.mlr.press/v5/titsias09a. ISSN: 1938-7228.

M. Towers, J. K. Terry, A. Kwiatkowski, J. U. Balis, G. d. Cola, T. Deleu, M. Goulão, A. Kallinteris, A. KG, M. Krimmel, R. Perez-Vicente, A. Pierré, S. Schulhoff, J. J. Tai, A. T. J. Shen, and O. G. Younis. Gymnasium, Mar. 2023. URL https://zenodo.org/record/8127025.

H. Unbehauen. Adaptive dual control systems: a survey. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, pages 171–180, 2000. doi: 10.1109/ASSPCC.2000.882466.

E. van der Pol, H. van Hoof, F. A. Oliehoek, and M. Welling. Multi-agent MDP homomorphic networks. In *International Conference on Learning Representations*, 2021.

S. Villar, D. W. Hogg, K. Storey-Fisher, W. Yao, and B. Blum-Smith. Scalars are universal: Equivariant machine learning, structured like classical physics. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 28848–28863. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/f1b0775946bc0329b35b823b86eeb5f5-Paper.pdf.

S. Villar, W. Yao, D. W. Hogg, B. Blum-Smith, and B. Dumitrascu. Dimensionless machine learning: Imposing exact units equivariance. *Journal of Machine Learning Research*, 24(109):1–32, 2023. URL http://jmlr.org/papers/v24/22-0680.html.

P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0

Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

W. Wiesemann, D. Kuhn, and B. Rustem. Robust Markov Decision Processes. *Mathematics of Operations Research*, 38(1):153–183, 2013. doi: 10.1287/moor.1120.0566. URL https://doi.org/10.1287/moor.1120.0566.

R. J. Williams. *Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning*, pages 5–32. Springer US, Boston, MA, 1992. ISBN 978-1-4615-3618-5. doi: 10.1007/978-1-4615-3618-5_2. URL https://doi.org/10.1007/978-1-4615-3618-5_2.

P. D. Winter and A. Carusi. (de)troubling transparency: artificial intelligence (ai) for clinical applications. *Medical Humanities*, 49(1):17–26, 2023. ISSN 1468-215X. doi: 10.1136/medhum-2021-012318. URL https://mh.bmj.com/content/49/1/17.

M. Xu, M. Quiroz, R. Kohn, and S. A. Sisson. Variance reduction properties of the reparameterization trick. In K. Chaudhuri and M. Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 2711–2720. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/xu19a.html.

S. Yang, Y. Ze, and H. Xu. Movie: Visual model-based policy adaptation for view generalization. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 21507–21523. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/43b77cef2a83a25aa27d3271d209e4fd-Paper-Conference.pdf.

H. Yu, T. Nghia, B. K. Hsiang Low, and P. Jaillet. Stochastic Variational Inference for Bayesian Sparse Gaussian Process Regression. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019. doi: 10.1109/IJCNN.2019.8852481.

T. Yu, G. Thomas, L. Yu, S. Ermon, J. Y. Zou, S. Levine, C. Finn, and T. Ma. Mopo: Model-based offline policy optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 14129–14142. Curran Associates, Inc., 2020.

Y. Yuan, C. S. Chen, Z. Liu, W. Neiswanger, and X. S. Liu. Importance-aware co-teaching for offline model-based optimization. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 55718–55733. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/ae8b0b5838ba510daff1198474e7b984-Paper-Conference.pdf.

W. Zhao, J. P. Queralta, and T. Westerlund. Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 737–744, 2020. doi: 10.1109/SSCI47803.2020.9308468.

C. Zhu, M. Simchowitz, S. Gadipudi, and A. Gupta. RePo: Resilient model-based Reinforcement Learning by regularizing posterior predictability. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 32445–32467. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6692e1b0e8a31e8de84bd90ad4d8d9e0-Paper-Conference.pdf.

Appendix A.

Supplementary Material

A.1 Buckingham-Pi Theorem and Application to Pendulum

A.1.1 Pendulum

Dynamic variables

$$\begin{cases} u: & \begin{bmatrix} 1 & 2 & -2 \end{bmatrix} \\ \theta: & \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \\ \dot{\theta}: & \begin{bmatrix} 0 & 0 & -1 \end{bmatrix} \\ \ddot{\theta}: & \begin{bmatrix} 0 & 0 & -2 \end{bmatrix} \end{cases} \tag{A.1}$$

Context

$$\begin{cases} M: & \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \\ g: & \begin{bmatrix} 0 & 1 & -2 \end{bmatrix} \\ L: & \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \end{cases} \tag{A.2}$$

The context matrix

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -2 \\ 0 & 1 & 0 \end{bmatrix} \tag{A.3}$$

is full rank and thus the variables (M, g, L) can be used for non-dimensionalizing the other ones.

$$\begin{cases} \left[u^{\alpha_u} \cdot M^{\beta_u} \cdot g^{\delta_u} \cdot L^{\gamma_u}\right] = 0 \\ \left[\theta^{\alpha_\theta} \cdot M^{\beta_\theta} \cdot g^{\delta_\theta} \cdot L^{\gamma_\theta}\right] = 0 \\ \left[\dot{\theta}^{\alpha_{\dot{\theta}}} \cdot M^{\beta_{\dot{\theta}}} \cdot g^{\delta_{\dot{\theta}}} \cdot L^{\gamma_{\dot{\theta}}}\right] = 0 \\ \left[\ddot{\theta}^{\alpha_{\ddot{\theta}}} \cdot M^{\beta_{\ddot{\theta}}} \cdot g^{\delta_{\ddot{\theta}}} \cdot L^{\gamma_{\ddot{\theta}}}\right] = 0 \end{cases} \tag{A.4}$$

Where the bracket signs $[x]$ represent the dimension of variable x and each power law within equation A.4 will be the Π-groups.

Because we know the dimension of the variables $u, \theta, \dot{\theta}, \ddot{\theta}$ and because $[x \times y] = [x] \times [y]$ the system can be rewritten as

$$\begin{cases} M^{\beta_u} \cdot L^{\alpha_u+\delta_u+\gamma_u} \cdot t^{-2\delta_u+\alpha_u} = 1 \\ M^{\beta_{\dot{\theta}}} \cdot L^{\delta_{\dot{\theta}}+\gamma_{\dot{\theta}}} \cdot t^{-2\delta_{\dot{\theta}}-1} = 1 \\ M^{\beta_{\ddot{\theta}}} \cdot L^{\delta_{\ddot{\theta}}+\gamma_{\ddot{\theta}}} \cdot t^{-2\delta_{\dot{\theta}}-2} = 1 \end{cases} \tag{A.5}$$

We removed the equation for θ because as an angle, this variable is naturally dimensionless. The coefficients are found by solving one system for each variable.

Torque u

$\Pi_u = u^\alpha \cdot M^\beta \cdot g^\delta \cdot L^\gamma$ Using the first term from A.5 and replacing the terms by their dimension we obtain,

$$M^{\alpha+\beta}.L^{\alpha+\delta+\gamma}t^{-2\alpha-2\delta} = 1. \tag{A.6}$$

All exponents must be 0 to ensure the homogeneity which yields

$$\begin{cases} \alpha+\beta & = 0 \\ \alpha+\delta+\gamma & = 0 \\ \alpha-2\delta & = 0 \end{cases} \tag{A.7}$$

The last equation implies $\alpha + \delta = 0$, which we substract to the first equation to obtain

$$\begin{cases} \beta = \delta \\ \alpha+\beta = 0 \\ \alpha+\delta+\gamma = 0 \end{cases} \tag{A.8}$$

and then using $\alpha + \delta = 0$

$$\begin{cases} \beta = \delta\gamma = 0 \\ \alpha + \beta = 0 \end{cases} \tag{A.9}$$

Because the solution is not unique, we choose $alpha = 1$ which gives the dimensionless torque

$$\Pi_u = \frac{u}{Mg} \tag{A.10}$$

Angular speed $\dot{\theta}$

$\Pi_{\dot{\theta}} = \dot{\theta}^\alpha \cdot M^\beta \cdot g^\delta \cdot L^\gamma$ We replace the variables with their dimensions to obtain

$$M^\beta.L^{\delta+\gamma}.t^{-\alpha-2\delta} = 1, \tag{A.11}$$

which we can solve with the systems

$$\begin{cases} \beta = 0 \\ \delta + \gamma = 0 \\ \alpha + 2\delta = 0 \end{cases} \tag{A.12}$$

By substracting twice the second equation to the third we obtain

$$\begin{cases} \beta = 0 \\ \alpha = 2\delta \\ \delta + \gamma = 0 \end{cases} \tag{A.13}$$

We choose $\delta = 1$ yielding

$$\Pi_{\dot{\theta}} = \dot{\theta}^2 \frac{g}{L} \tag{A.14}$$

Angular acceleration $\ddot{\theta}$

We the same process we obtain,

$$M^\beta.L^{\delta+\gamma}.t^{-\alpha-2\delta} = 1 \tag{A.15}$$

$\beta = 0$ so we have the systems

$$\begin{cases} \delta + \gamma = 0 \\ \alpha = \beta \end{cases} \tag{A.16}$$

This yields

$$\Pi_{\ddot{\theta}} = \ddot{\theta}\frac{g}{L} \tag{A.17}$$

A.1.2 Cartpole

The movement of the cartpole depends on the variables $(x, \cos(\theta), \sin(\theta), \dot{x}, \dot{\theta}), u$. A trivial Π-group for the cart position is $\Pi_x = \frac{x}{L}$, where L is the pole length. For the angular speed, we use the same transformation as the pendulum. Therefore, we need to compute the dimensionless variables for $\dot{x}$ and u

Cart speed $\dot{x}$

With $\Pi_{\dot{x}} = \dot{x}^{\alpha} \cdot M^{\beta} \cdot g^{\delta} \cdot L^{\gamma}$, we obtain with $[\dot{x}] = L.t^{-1}$,

$$M^{\beta}.L^{\alpha+\delta+\gamma}.t^{-\alpha-2\delta} \tag{A.18}$$

which yields $\beta = 0$. We then substract on equation with the other to obtain,

$$\begin{cases} \delta - \gamma = 0 \\ \alpha + 2\delta = 0 \end{cases} \tag{A.19}$$

which is solved with $\delta = \gamma = -1$.

Therefore the dimensionless variable for the cart is

$$\Pi_{\dot{x}} = \frac{\dot{x}^2}{Lg}. \tag{A.20}$$

Force u

$\Pi_u = u^\alpha \cdot M^\beta \cdot g^\delta \cdot L^\gamma$ The dimension of the control force is $[u] = M.L.t^{-2}$. Using that value yields the system

$$\begin{cases} \alpha + \beta = 0 \\ \alpha + \delta = 0 \\ \alpha + \delta + \gamma = 0 \end{cases} \tag{A.21}$$

and by substracting the first two equations we obtain

$$\begin{cases} \gamma = 0 \\ \beta = \delta \\ \alpha + \delta = 0 \end{cases} \tag{A.22}$$

With $\alpha = 1$, we obtain the resulting

$$\Pi_u = \frac{u}{MgL} \tag{A.23}$$