University of Glasgow | School of Computing Science

Variational Inference for Inverse Reinforcement Learning with Gaussian Processes

Paulius Dilkas (2146879)

16th April 2019

ABSTRACT

The inverse reinforcement learning (IRL) problem asks us to find a reward function of a Markov decision process that explains observed behaviour. Many approaches are only able to express reward functions as linear combinations of state features. Out of those that can handle nonlinearity, none can provide a full posterior distribution of rewards. Providing variance estimates for rewards would allow one to judge how well the model has learned its policy and discover any weak spots the model may have. We show how to perform variational inference (VI) on a Gaussian process-based IRL model in order to approximate the posterior distribution of rewards. We prove the correctness of the approach and demonstrate the model's behaviour in practice. Being able to provide full posterior probability distributions in IRL unlocks many new research frontiers ranging from integrating recent developments in VI to make the models more efficient and flexible, to developing complex reinforcement learning agents that can explicitly search for opportunities to fix their weaknesses.

1. INTRODUCTION

Imagine using a machine learning (ML) algorithm to teach a robot how to move around people so that it learns to predict where people are going and adjust its path accordingly. The ML algorithm would learn from data about proper actions in a range of different situations. But do we have enough data to ensure reasonably optimal behaviour? Perhaps the robot behaves well in most situations, but fails in less common scenarios. Can the ML model itself describe its weaknesses so that we could ensure it is exposed to sufficiently many uncommon or difficult situations?

This learning problem [12] as well as many others have benefited from an approach called *inverse reinforcement learning* (IRL), also known as apprenticeship learning and inverse optimal control. IRL proposes a way to learn behaviour from demonstrations that typically come from human actions. More formally, the IRL problem asks us to find a reward function for a Markov decision process (MDP), where demonstrations are encoded as sets of state-action pairs, showing observed actions in a number of states.

IRL is an important problem because adjusting the reward function by hand is often unwieldy, since human behaviour often depends on many factors in complicated ways [2]. Moreover, learning the reward function rather than the policy itself makes the model more transferable to new environments: a minor change in the environment can reorganise the whole policy but only have a local effect in the reward structure [11, 15]. IRL has been used to teach helicopters how to perform tricks [1], predict taxi destinations [34], and make driving safer and more efficient by predicting pedestrian movement [35] and the driver's intentions [28].

Most IRL models in the literature make a convenient yet unjustified assumption that the reward function can be expressed as a linear combination of features [2, 18, 33]. This assumption severely restricts what reward structures can be modelled. Out of the non-linear models proposed to date, none can answer the questions posed in the first paragraph. More specifically, without knowing the ground truth for the learning problem, we have no reliable way to know whether the learned behaviour is optimal and which parts of the model could benefit from more data. Quite often, the models assume that rewards have no variance [15, 11].

In this paper, we show how that assumption can be lifted by switching from maximum likelihood estimation to *variational inference* (VI), i.e., we approximate the posterior distribution of the model by optimising the parameters of a simpler distribution to make it similar to the posterior. This approach can prove useful in three major ways:

1. Variance estimates can be used to guide what data should be collected next, i.e., if the rewards of some states have abnormally high variance, we might want to expose the model to more data visiting those and surrounding states.

2. Variances estimates can also be used to judge whether we can trust the predictions of the model or, perhaps, the model could benefit from some adjustments or more data.

3. By adopting a more Bayesian approach we can set prior distributions on random variables. This can help prevent overfitting as well as encode additional information about the reward function [11].

No fully Bayesian nonlinear IRL model has been proposed yet. We will uncover one reason for this when we set up our IRL model for VI in Section 4. Applying VI to this model in a theoretically sound way (and without introducing unjustified simplifying assumptions) requires us to use Lebesgue's dominated convergence theorem in order to derive the gradient of the expectation of the MDP value function—we provide a complete proof for this (and some other results) in Section 5. Finally, in Section 6 we demonstrate the model's correctness and properties experimentally using several example scenarios.

Notation and Conventions.

For any matrix $\mathbf{A}$, we write either $A_{i,j}$ or $[\mathbf{A}]_{i,j}$ to denote the element of $\mathbf{A}$ in row i and column j. We use $\operatorname{tr}(\mathbf{A})$

to denote its *trace* and adj($\mathbf{A}$) for its *adjugate* (or *classical adjoint*). For any vector $\mathbf{x}$, we write $\mathbb{R}_d[\mathbf{x}]$ to denote a vector space of polynomials with degree at most d, where variables are elements of $\mathbf{x}$, and coefficients are in $\mathbb{R}$.

Throughout the paper, all integrals should be interpreted as definite integrals over the entire sample space. When referencing measurability, we assume Lebesgue measure in a Euclidean space with a suitable number of dimensions. Similarly, whenever we consider the existence of an integral, we use the Lebesgue definition of integration.

2. BACKGROUND

The IRL problem itself was originally proposed by Russell in 1998 [25]. Most of the early approaches had the aforementioned reward linearity assumption. One of the first papers on the subject by Ng and Russell [18] introduced several linear programming algorithms and identified an important issue: there are typically many reward functions that can explain the data equally well. This problem was solved by Ziebart et al. [33] with the introduction of IRL based on the principle of maximum causal entropy in a linearly-solvable MDP.

Levine et al. [15] were the first to lift the linearity assumption without imposing additional restrictions on the problem by modelling rewards with a Gaussian process (GP). They do, however, assume that rewards have no variance—our work removes this restriction without any compromises.

An alternative to GPs for modelling nonlinear functions is, of course, neural networks. Wulfmeier et al. [30] have shown how they can be used in the IRL setting. While this approach benefits from constant-time inference and the ability to learn complex features from data, neural networks often need significantly more data for the weights across all layers to stabilise.

Recently, Jin et al. [11] have adapted the model proposed by Levine et al. [15] to use deep GPs, harnessing the power of deep learning to make the model less dependent on what features are provided. Although they use VI, their approximating distribution for rewards at inducing points is simply the Dirac δ function, which is essentially equivalent to the assumption of no variance.

We argue that developing an IRL model with a variational approximation of the full posterior distribution has eluded previous research in the area for two reasons. First, it is not directly relevant to the goal of producing accurate point estimates of rewards. However, we described several key benefits of variance prediction in the previous section, and we postulate how this work could affect the wider fields of reinforcement learning and artificial intelligence in Section 7. Second, as it will become apparent in Sections 4 and 5, optimising such a VI model involves estimating gradients, and establishing correctness of these estimates requires a significant amount of theoretical groundwork.

2.1 Variational Inference

VI has come a long way from the initial *mean field* approach where each variable is approximated by a univariate Gaussian, independent of all other variables. Recent years have brought improvements in both computational complexity and flexibility of the variational approximation [4]. In this section, we will mention some of the most important and relevant developments in the field, while referring the interested reader to recent survey articles [4, 32] for a broader overview of the area.

A major development in approximating complex distributions came in the form of *normalizing flows*, i.e., a collection of invertible functions—parametrised by additional variational parameters—that are applied to latent variables [23]. A sequence of such functions can transform a simple initial distribution (e.g., uniform or Gaussian) into an arbitrarily complex one.

Another approach to flexibility in modelling can be achieved by considering different GP kernels, i.e., functions that model similarity between pairs of points. For instance, Wilson and Adams [29] show how all *stationary* (i.e., invariant to translations) kernels can be generated (or at least approximated) from a mixture of Gaussians in their spectral representation using Bochner's theorem [5]. It looks promising to combine these kernels with the variational Fourier features approach by Hensman et al. [9] that leverages the same spectral representations for efficient VI. This algorithm, however, is currently restricted to Matérn kernels, which could be restrictive, considering how IRL models tend to use kernels that can adjust the importance of each feature [11, 15].

While these and many other advancements in VI promise great improvements, there are hidden challenges when it comes to applying them to IRL. Many suggested improvements implicitly impose restrictions on the model that are unlikely to be satisfied in the context of IRL, e.g., the evidence of the model being a Gaussian. Nevertheless, the work mentioned here seems adaptable to the IRL setting and can make variational approximations both more accurate and faster.

3. PROBLEM DESCRIPTION

In this section, we describe the problem in more detail, introducing the notation and definitions that will be used throughout the paper.

Definition 3.1 (MDP). *A* Markov decision process *is a set $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{T}, \gamma, \mathbf{r}\}$, where $\mathcal{S}$ and $\mathcal{A}$ are sets of states and actions, respectively; $\mathcal{T}\colon \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is a function defined so that $\mathcal{T}(s, a, s')$ is the probability of moving to state s' after taking action a in state s; $\gamma \in [0, 1)$ is the discount factor; and $\mathbf{r} \in \mathbb{R}^{|\mathcal{S}|}$ is the reward vector.*

While it is more common to represent rewards as a function $r\colon \mathcal{S} \to \mathbb{R}$, the vector notation is generally more appropriate in our situation. However, we switch between the two notations as needed.

Definition 3.2 (IRL). *Given an MDP without rewards $\mathcal{M} \setminus \{\mathbf{r}\}$, an $|\mathcal{S}| \times d$ feature matrix $\mathbf{X}$ (where d is the number of features), and a set of* demonstrations *$\mathcal{D} = \{\zeta_i\}_{i=1}^{N}$, where each demonstration $\zeta_i = \{(s_{i,t}, a_{i,t})\}_{t=1}^{T}$ is a multiset of state-action pairs that represents optimal (or near-optimal) actions, find a reward function that maximises the probability of observing the demonstrations, i.e.,*

$$\arg\max_{\mathbf{r}} p(\mathcal{D} \mid \mathbf{r}).$$

Example 3.3. *Figure 1 shows a possible setup for IRL.*

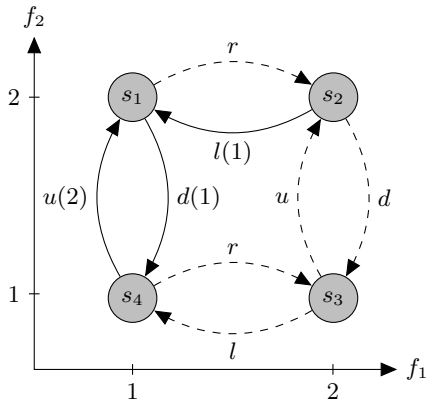

Figure 1: An example MDP with deterministic actions. States are represented by grey circles, actions by directed labelled edges between them. The numbers in parentheses next to some actions' names denote how many times that state-action pair appears in $\mathcal{D}$. If that number is zero, the edge is dashed. Finally, the axes are used to assign two features to each state.

In this case, $\mathcal{S} = \{s_1, s_2, s_3, s_4\}$, $\mathcal{A} = \{l, r, u, d\}$,

$$\mathcal{T}(s,a,s') = \begin{cases} 1 & \text{if } (s,a,s') = (s_1,r,s_2) \\ 1 & \text{if } (s,a,s') = (s_1,d,s_4) \\ 1 & \text{if } (s,a,s') = (s_2,l,s_1) \\ 1 & \text{if } (s,a,s') = (s_2,d,s_3) \\ 1 & \text{if } (s,a,s') = (s_3,l,s_4) \\ 1 & \text{if } (s,a,s') = (s_3,u,s_2) \\ 1 & \text{if } (s,a,s') = (s_4,r,s_3) \\ 1 & \text{if } (s,a,s') = (s_4,u,s_1) \\ 0 & \text{otherwise,} \end{cases}$$

$$\mathbf{X} = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \end{array} \begin{array}{c} \begin{array}{cc} f_1 & f_2 \end{array} \\ \begin{bmatrix} 1 & 2 \\ 2 & 2 \\ 2 & 1 \\ 1 & 1 \end{bmatrix} \end{array},$$

and $\gamma \in [0, 1)$ *can be assigned any value. The distribution of example state-action pairs into separate demonstrations is irrelevant to our model (and presumably all other models) and is only a relic of the practical motivation behind the problem's initial definition. Hence, we can keep each pair in its own set as follows:*

$$\mathcal{D} = \{\{(s_1, d)\}, \{(s_2, l)\}, \{(s_4, u)\}, \{(s_4, u)\}\}.$$

A solution to this problem is then an assignment of rewards to states, e.g.,

$$\mathbf{r} = \begin{bmatrix} 2 \\ 0 \\ -1 \\ 1 \end{bmatrix} \quad \text{or} \quad r(s) = \begin{cases} 2 & \text{if } s = s_1 \\ 0 & \text{if } s = s_2 \\ -1 & \text{if } s = s_3 \\ 1 & \text{if } s = s_4 \end{cases}.$$

The optimal (deterministic) *policy* $\pi \colon \mathcal{S} \to \mathcal{A}$ (i.e., a choice of actions for each state that maximises reward over time) is usually constructed by defining a *value (utility) function* $V_{\mathbf{r}} \colon \mathcal{S} \to \mathbb{R}$ that measures how good a state is based on the reward $\mathbf{r}$ as well as the structure of the MDP. One can then find $V_{\mathbf{r}}$ by applying the Bellman backup operator until convergence to all states $s \in \mathcal{S}$ (this algorithm is known as *value iteration*) [26]:

$$V_{\mathbf{r}}(s) \coloneqq r(s) + \gamma \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} \mathcal{T}(s,a,s') V_{\mathbf{r}}(s').$$

However, we follow previous work on IRL with GPs [11, 15] and use a *linearly solvable* (or *maximum causal entropy*) MDP with a stochastic policy that defines probability distributions over actions instead of suggesting a single action for each state [33]. This type of MDP can be solved by applying the 'soft' version of the operator [15, 16]:

$$V_{\mathbf{r}}(s) \coloneqq \log \sum_{a \in \mathcal{A}} \exp\left(r(s) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{T}(s,a,s') V_{\mathbf{r}}(s') \right). \quad (1)$$

With this model, we can express the likelihood as [11, 15]

$$\begin{aligned} p(\mathcal{D} \mid \mathbf{r}) &= \prod_{i=1}^{N} \prod_{t=1}^{T} p(a_{i,t} \mid s_{i,t}) \\ &= \exp\left(\sum_{i=1}^{N} \sum_{t=1}^{T} Q_{\mathbf{r}}(s_{i,t}, a_{i,t}) - V_{\mathbf{r}}(s_{i,t}) \right), \end{aligned}$$

where

$$Q_{\mathbf{r}}(s,a) = r(s) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{T}(s,a,s') V_{\mathbf{r}}(s').$$

A significant part of learning is about being able to generalise to previously unseen input states (otherwise the problem becomes knowledge representation instead of learning). For this reason, we represent each state by a d-dimensional feature vector. This way, similar states are represented by similar features, and the ML model can predict rewards for new states based on their similarity to training data.

In this paper, we model rewards as a function from feature space to $\mathbb{R}$ using a *Gaussian process*. A GP represents an infinite collection of random variables from a joint (infinitely-dimensional) probability distribution, any finite number of which has a Gaussian distribution [22]. A GP is defined by its mean function (which is always $\mathbf{x} \mapsto 0$ in our case) and covariance function, which we denote by k. *Covariance functions* (also known as *kernels*) take two state feature vectors as input and quantify how similar the two states are, in a sense that we would expect high covariance scores to be associated with similar rewards.

A common way to scale GPs to larger data sets is by selecting m points in the feature space–called *inducing points*—and focus most of the training effort on them [17]. Let $\mathbf{X}_{\mathbf{u}}$ be the $m \times d$ matrix of features at inducing points, and let $\mathbf{u}$ be the rewards at those states. Then the full joint probability distribution can be factorised as

$$p(\mathcal{D}, \mathbf{u}, \mathbf{r}) = p(\mathbf{u}) \times p(\mathbf{r} \mid \mathbf{u}) \times p(\mathcal{D} \mid \mathbf{r}), \quad (2)$$

where $p(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{0}, \mathbf{K}_{\mathbf{u},\mathbf{u}})$ is the GP prior [22]. The GP posterior is then a multivariate Gaussian [15] defined as

$$p(\mathbf{r} \mid \mathbf{u}) = \mathcal{N}(\mathbf{r}; \mathbf{K}_{\mathbf{r},\mathbf{u}}^{\intercal} \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{u}, \mathbf{K}_{\mathbf{r},\mathbf{r}} - \mathbf{K}_{\mathbf{r},\mathbf{u}}^{\intercal} \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{K}_{\mathbf{r},\mathbf{u}}). \quad (3)$$

The matrices such as $\mathbf{K}_{\mathbf{r},\mathbf{u}}$ are called *covariance matrices* (also *kernel matrices* and *Gram matrices*) and are defined as

$[\mathbf{K}_{\mathbf{r},\mathbf{u}}]_{i,j} = k(\mathbf{x}_{\mathbf{r},i}, \mathbf{x}_{\mathbf{u},j})$, where $\mathbf{x}_{\mathbf{r},i}$ and $\mathbf{x}_{\mathbf{u},j}$ denote feature vectors for the ith state in $\mathcal{S}$ and the jth inducing point, respectively [11].

We can then use VI in order to approximate $p(\mathbf{u}, \mathbf{r} \mid \mathcal{D})$. This is done by minimising the *Kullback-Leibler* (KL) divergence between the original probability distribution and our approximation. KL divergence (asymmetrically) measures the difference between two probability distributions. Let $q(\mathbf{u}, \mathbf{r})$ denote our approximation. The KL divergence is then defined as [4]

$$\begin{aligned} D_{\mathrm{KL}}(q(\mathbf{u},\mathbf{r}) \parallel p(\mathbf{u},\mathbf{r} \mid \mathcal{D})) &= \mathbb{E}_{q(\mathbf{u},\mathbf{r})}[\log q(\mathbf{u},\mathbf{r}) - \log p(\mathbf{u},\mathbf{r} \mid \mathcal{D})] \\ &= \mathbb{E}_{q(\mathbf{u},\mathbf{r})}[\log q(\mathbf{u},\mathbf{r}) - \log p(\mathcal{D},\mathbf{u},\mathbf{r})] \\ &\quad + \mathbb{E}_{q(\mathbf{u},\mathbf{r})}[\log p(\mathcal{D})]. \end{aligned}$$

The last term is both hard to compute and constant with respect to $q(\mathbf{u}, \mathbf{r})$, so we can remove it from our optimisation objective. The negation of what remains is known as the *evidence lower bound* (ELBO) and is defined as [3, 4]

$$\begin{aligned} \mathcal{L} &= \mathbb{E}_{q(\mathbf{u},\mathbf{r})}\left[\log \frac{p(\mathcal{D},\mathbf{u},\mathbf{r})}{q(\mathbf{u},\mathbf{r})}\right] \\ &= \iint \log \frac{p(\mathcal{D},\mathbf{u},\mathbf{r})}{q(\mathbf{u},\mathbf{r})} q(\mathbf{u},\mathbf{r})\,\mathrm{d}\mathbf{r}\,\mathrm{d}\mathbf{u}. \end{aligned} \tag{4}$$

Thus, instead of minimising KL divergence, we focus on maximising $\mathcal{L}$ by optimising the values of parameters that we define in Section 4. Note that hereafter we drop the subscript notation, as all expected values will be with respect to $q(\mathbf{u}, \mathbf{r})$.

4. THE MODEL

In order to have a fully functional model, we still need to make several key design decisions. In this section, we describe the covariance function used to define the GP, consider what parameters should be used to optimise $\mathcal{L}$ and how they should be initialised, and in Section 4.1 we derive our final expression for $\mathcal{L}$.

We stick with the same covariance function as in the work by Levine et al. [15], which is a version of the *automatic relevance detection kernel*:

$$\begin{aligned} k(\mathbf{x}_i, \mathbf{x}_j) = \lambda_0 \exp\Bigg(&-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^\intercal \mathbf{\Lambda} (\mathbf{x}_i - \mathbf{x}_j) \\ &- \mathbb{1}[i \neq j]\sigma^2 \operatorname{tr}(\mathbf{\Lambda}) \Bigg). \end{aligned}$$

Here, λ_0 is the overall 'scale' factor, $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_d)$ is a diagonal matrix that determines the importance of each feature, $\mathbb{1}$ is defined as

$$\mathbb{1}[b] = \begin{cases} 1 & \text{if } b \text{ is true} \\ 0 & \text{otherwise,} \end{cases}$$

and σ^2 is fixed as $10^{-2}/2$, since this value has little influence on the behaviour of the algorithm and is here as a noise factor used to avoid singular covariance matrices [15]. We will write $\boldsymbol{\lambda} = (\lambda_0, \ldots, \lambda_d)^\intercal$ to refer to both λ_0 and $\mathbf{\Lambda}$ at the same time.

Ideally, we would like to model $\boldsymbol{\lambda}$ with an approximating distribution. However, due to how the prior probability density function (PDF) for $\mathbf{u}$ involves $\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}$, and $\mathcal{L}$ has an expected value that cannot be eliminated (see Section 4.1), we are unable to show that such an $\mathcal{L}$ is well-defined. More generally, we pose the following problem, which, to the best of our knowledge, is currently open:

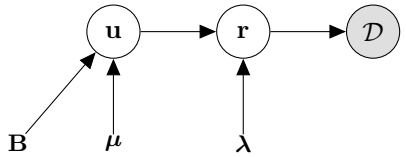

Figure 2: Our VI problem expressed as a (simplified) Bayesian network. The only observed variable (representing the demonstrations) is in a grey circle, modelled latent variables are in white circles, and the variational parameters are at the bottom.

Open Problem 4.1. *Let* $\mathbf{A}$ *be a* $n \times n$ *matrix of coefficients,* X *be a random variable, and* $\mathbf{M}$ *be an* $n \times n$ *matrix such that* $M_{i,j} = f(X, A_{i,j})$*, where* f *is an arbitrary function. Under what circumstances does* $\mathbb{E}[\mathbf{M}^{-1}]$ *exist?*

While there are some obvious examples where the expected value exists (e.g., $f(X, A_{i,j}) = A_{i,j}X$ for an invertible $\mathbf{A}$ and many distributions of X), it would be particularly interesting to know whether the answer is 'always'. A proof of such a result would allow us to model $\boldsymbol{\lambda}$ instead of treating it as a variational parameter, which would make the model fully Bayesian. For now, $\boldsymbol{\lambda}$ will have to be treated as a variational parameter.

It remains to decide on the approximating distribution for $\mathbf{u}$ and $\mathbf{r}$. As is commonly done when applying VI to GPs [6], we set

$$q(\mathbf{u}, \mathbf{r}) = q(\mathbf{u}) q(\mathbf{r} \mid \mathbf{u}), \tag{5}$$

where $q(\mathbf{r} \mid \mathbf{u}) = p(\mathbf{r} \mid \mathbf{u})$ and $q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \boldsymbol{\mu}, \mathbf{\Sigma})$.

Ong et al. [19] have recently suggested that, in order to make variational approximation of a multivariate Gaussian more scalable, the covariance matrix should be decomposed as $\mathbf{\Sigma} = \mathbf{B}\mathbf{B}^\intercal + \mathbf{D}^2$, where $\mathbf{B}$ is a lower triangular $m \times p$ matrix with positive diagonal entries, and $\mathbf{D}$ is a diagonal matrix. Typically, we would set p so that $p \ll m$ to get an efficient approximation, but in this case we will simply set $p = m$ and $\mathbf{D} = \mathbf{O}_m$ in order to retain full covariance structure.

The resulting model is summarised in Figure 2. We rely on $p(\mathcal{D} \mid \mathbf{r})$ as the only link between data and our model. Since the expression for $q(\mathbf{r} \mid \mathbf{u})$ has both $\mathbf{u}$ and covariance matrices in it, $\mathbf{r}$ depends on both $\mathbf{u}$ and the parameters of the kernel, $\boldsymbol{\lambda}$. The two remaining dependencies stem from the fact that the approximating distribution for $\mathbf{u}$ is $\mathcal{N}(\boldsymbol{\mu}, \mathbf{B}\mathbf{B}^\intercal)$.

As we want to restrict some parameters (namely, $\boldsymbol{\lambda}$ and the diagonal of $\mathbf{B}$) to positive values, we express them as exponentials and later adjust their derivatives accordingly. Specifically, we can set $\lambda_i = e^{\lambda_i'}$ and optimise λ_i' using the chain rule:

$$\frac{\partial \mathcal{L}}{\partial \lambda_i'} = e^{\lambda_i'} \frac{\partial \mathcal{L}}{\partial \lambda_i}. \tag{6}$$

This way, we restrict λ_i to positive values while allowing λ_i' to range over $\mathbb{R}$.

Finally, the parameters can be initialised as follows[1]:

$$\mu_i \sim \mathcal{U}(0,1) \quad \text{for } i = 1, \ldots, m, \tag{7}$$
$$\lambda_0 \sim \chi^2_5, \tag{8}$$
$$\lambda_i \sim \chi^2_1 \quad \text{for } i = 1, \ldots, d, \tag{9}$$
$$\operatorname{diag}(\mathbf{B}) \sim \chi^2_4, \tag{10}$$
$$\text{the rest of } \mathbf{B} \sim \mathcal{N}(0,1). \tag{11}$$

The initialisation of $\boldsymbol{\mu}$ mirrors the initialisation of $\mathbf{r}$ in previous work [15]. In contrast, while they have constant initial values for $\boldsymbol{\lambda}$, we sample from χ^2 distributions centred around those values (5 for λ_0 and 1 for any other λ_i). The distributions for initial values of $\mathbf{B}$ are simply set to provide a reasonable spread of positive values for the diagonal, and both positive and negative values for all other entries in the matrix.

4.1 Evidence Lower Bound

It remains to express $\mathcal{L}$ for our (now fully specified) model. Note that in order to keep the derivation simple, we drop all constant terms in the expression of $\mathcal{L}$, i.e., equality is taken to mean 'equality up to an additive constant'.

Firstly, let us return to Equation (4) and write

$$\mathcal{L} = \mathbb{E}[\log p(\mathcal{D}, \mathbf{u}, \mathbf{r})] - \mathbb{E}[\log q(\mathbf{u}, \mathbf{r})].$$

By substituting in Equations (2) and (5), we get

$$\begin{aligned}\mathcal{L} = \mathbb{E}[&\log p(\mathbf{u}) + \log p(\mathbf{r} \mid \mathbf{u}) + \log p(\mathcal{D} \mid \mathbf{r})] \\ &- \mathbb{E}[\log q(\mathbf{u}) + \log q(\mathbf{r} \mid \mathbf{u})].\end{aligned}$$

Since $q(\mathbf{r} \mid \mathbf{u}) = p(\mathbf{r} \mid \mathbf{u})$, they cancel each other out. Also notice that

$$\begin{aligned}\mathbb{E}[\log p(\mathbf{u}) - \log q(\mathbf{u})] &= -D_{\mathrm{KL}}(q(\mathbf{u}) \parallel p(\mathbf{u})) \\ &= -\frac{1}{2}(\operatorname{tr}(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\boldsymbol{\Sigma}) + \boldsymbol{\mu}^\intercal \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\boldsymbol{\mu} - m \\ &\quad + \log|\mathbf{K}_{\mathbf{u},\mathbf{u}}| - \log|\boldsymbol{\Sigma}|),\end{aligned}$$

by the definition of KL divergence between two multivariate Gaussians [7]. Hence,

$$\begin{aligned}\mathcal{L} = \mathbb{E}&\left[\sum_{i=1}^{N}\sum_{t=1}^{T} Q_{\mathbf{r}}(s_{i,t}, a_{i,t}) - V_{\mathbf{r}}(s_{i,t})\right] \\ &- \frac{1}{2}\left(\operatorname{tr}\left(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\boldsymbol{\Sigma}\right) + \boldsymbol{\mu}^\intercal \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\boldsymbol{\mu} + \log|\mathbf{K}_{\mathbf{u},\mathbf{u}}| - \log|\boldsymbol{\Sigma}|\right).\end{aligned}$$

Using the expressions for $Q_{\mathbf{r}}$ we get

$$\begin{aligned}\mathcal{L} = \mathbb{E}&\left[\sum_{i=1}^{N}\sum_{t=1}^{T} r(s_{i,t}) - V_{\mathbf{r}}(s_{i,t}) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{T}(s_{i,t}, a_{i,t}, s') V_{\mathbf{r}}(s')\right] \\ &- \frac{1}{2}\left(\operatorname{tr}\left(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\boldsymbol{\Sigma}\right) + \boldsymbol{\mu}^\intercal \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\boldsymbol{\mu} + \log|\mathbf{K}_{\mathbf{u},\mathbf{u}}| - \log|\boldsymbol{\Sigma}|\right).\end{aligned}$$

We can simplify $\sum_{i=1}^{N}\sum_{t=1}^{T} r(s_{i,t})$ by defining a new vector $\mathbf{t} = (t_1, \ldots, t_{|\mathcal{S}|})^\intercal$, where t_i is the number of times the state associated with reward r_i has been visited across all demonstrations. Then,

$$\begin{aligned}\mathbb{E}\left[\sum_{i=1}^{N}\sum_{t=1}^{T} r(s_{i,t})\right] &= \mathbb{E}[\mathbf{t}^\intercal \mathbf{r}] = \mathbf{t}^\intercal \mathbb{E}[\mathbf{r}] \\ &= \mathbf{t}^\intercal \mathbb{E}\left[\mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}\right] = \mathbf{t}^\intercal \mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\boldsymbol{\mu}.\end{aligned}$$

[1]In practice, it is often easier to start with a fixed $\mathbf{B} = \mathbf{I}_m$, as random values of $\mathbf{B}$ can often result in a near-singular $\boldsymbol{\Sigma}$.

This allows us to simplify $\mathcal{L}$ to

$$\begin{aligned}\mathcal{L} = &\mathbf{t}^\intercal \mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\boldsymbol{\mu} - \mathbb{E}[v] \\ &- \frac{1}{2}\left(\operatorname{tr}\left(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\boldsymbol{\Sigma}\right) + \boldsymbol{\mu}^\intercal \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\boldsymbol{\mu} + \log|\mathbf{K}_{\mathbf{u},\mathbf{u}}| - \log|\boldsymbol{\Sigma}|\right),\end{aligned}$$

where

$$v = \sum_{i=1}^{N}\sum_{t=1}^{T} V_{\mathbf{r}}(s_{i,t}) - \gamma \sum_{s' \in \mathcal{S}} \mathcal{T}(s_{i,t}, a_{i,t}, s') V_{\mathbf{r}}(s').$$

We will need to optimise this expression in order to train our model.

5. THEORETICAL JUSTIFICATION

The typical way to optimise a quantity (the ELBO, in this case) involves computing its gradient. Unfortunately, the term $\mathbb{E}[v]$ in $\mathcal{L}$ complicates the situation. Indeed, in order to take the derivative of an expected value, certain conditions must be satisfied. Without proving the validity of our algorithm, we risk building it on faulty assumptions and having it misbehave—perhaps all the time, perhaps only in certain situations. The goal of this section is to show how Lebesgue's dominated convergence theorem can be applied to our model in order to derive $\nabla\mathbb{E}[v]$, i.e., the derivative of $\mathbb{E}[v]$ with respect to our variational parameters $\boldsymbol{\mu}$, $\mathbf{B}$, and $\boldsymbol{\lambda}$. This technique is inspired by black box VI [21], but takes a more detailed look at the problem and requires significantly more work to prove correctness. After showing that the theorem applies to our situation, we can then estimate $\nabla\mathbb{E}[v]$ with

$$\begin{aligned}\nabla\mathbb{E}[v] &= \nabla \iint v q(\mathbf{r} \mid \mathbf{u}) q(\mathbf{u}) \,\mathrm{d}\mathbf{r}\,\mathrm{d}\mathbf{u} \\ &= \iint \nabla[v q(\mathbf{r} \mid \mathbf{u}) q(\mathbf{u})] \,\mathrm{d}\mathbf{r}\,\mathrm{d}\mathbf{u} \\ &= \iint \frac{\nabla[v q(\mathbf{r} \mid \mathbf{u}) q(\mathbf{u})]}{q(\mathbf{r} \mid \mathbf{u}) q(\mathbf{u})} q(\mathbf{r} \mid \mathbf{u}) q(\mathbf{u}) \,\mathrm{d}\mathbf{r}\,\mathrm{d}\mathbf{u} \\ &\approx \frac{1}{S}\sum_{s=1}^{S} \frac{\nabla[v q(\mathbf{r}_s \mid \mathbf{u}_s) q(\mathbf{u}_s)]}{q(\mathbf{r}_s \mid \mathbf{u}_s) q(\mathbf{u}_s)},\end{aligned}$$

which can be computed by drawing S samples $\{(\mathbf{u}_s, \mathbf{r}_s)\}_{s=1}^{S}$ from $q(\mathbf{u}, \mathbf{r})$.

Our main goal is Theorem 5.13, which allows us to move differentiation inside the integral. In order to prove it, we use a number of intermediate results. We begin by introducing a few important tools and techniques in Section 5.1. In Section 5.2, we state a few derivatives of PDFs and covariance matrices and bound their values with some easy-to-work-with polynomials. In Section 5.3, we provide a sketch proof of the measurability of the MDP value function as well as new upper and lower bounds on its values. After another quick lemma, the main proof of the paper is in Section 5.4. See Figure 3 for an overview of how these results fit together.

5.1 Mathematical Preliminaries

We introduce a few definitions and results from linear algebra, numerical analysis, and measure theory that will be used later in the paper. Namely, we will use several different vector and matrix norms, consider how an inverse of a matrix changes with a small perturbation, and introduce Lebesgue's dominated convergence theorem which plays a major role in Section 5.4.

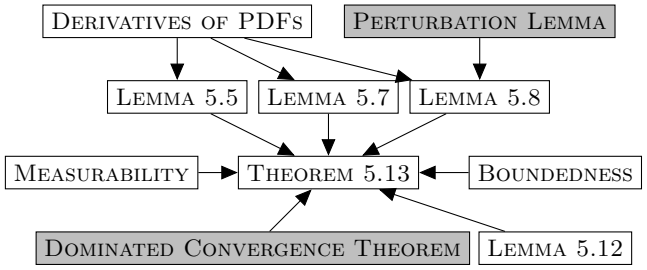

Figure 3: A graphical representation of dependencies between our theoretical results. An arrow from A to B means that A was used to prove B. Results from the literature are in grey.

Definition 5.1 (Norms). *For any finite-dimensional vector* $\mathbf{x} = (x_1, \dots, x_n)^\intercal$*, its* maximum norm *(*$\ell_\infty$-norm*) is*

$$\|\mathbf{x}\|_\infty = \max_i |x_i|$$

whereas its ℓ_1-norm *is*

$$\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|.$$

Let $\mathbf{A}$ *be a matrix. For any vector norm* $\|\cdot\|_p$*, we can also define its* induced norm *for matrices as*

$$\|\mathbf{A}\|_p = \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{A}\mathbf{x}\|_p}{\|\mathbf{x}\|_p}.$$

In particular, for $p = \infty$*, we have*

$$\|\mathbf{A}\|_\infty = \max_i \sum_j |A_{i,j}|.$$

Lemma 5.2 (Perturbation Lemma [14]). *Let* $\|\cdot\|$ *be any matrix norm, and let* $\mathbf{A}$ *and* $\mathbf{E}$ *be matrices such that* $\mathbf{A}$ *is invertible and* $\|\mathbf{A}^{-1}\|\|\mathbf{E}\| < 1$*. Then* $\mathbf{A} + \mathbf{E}$ *is invertible, and*

$$\|(\mathbf{A}+\mathbf{E})^{-1}\| \leq \frac{\|\mathbf{A}^{-1}\|}{1 - \|\mathbf{A}^{-1}\|\|\mathbf{E}\|}.$$

Theorem 5.3 (Dominated Convergence Theorem [24]). *Let* $(X, \mathcal{M}, \mu)$ *be a measure space and* $\{f_n\}$ *a sequence of measurable functions on* X *for which* $\{f_n\} \to f$ *pointwise almost everywhere on* X*, and the function* f *is measurable. Assume there is a non-negative function* g *that is integrable over* X *and dominates the sequence* $\{f_n\}$ *on* X *in the sense that*

$$|f_n| \leq g \text{ almost everywhere on } X \text{ for all } n.$$

Then f *is integrable over* X *and*

$$\lim_{n\to\infty} \int_X f_n \,\mathrm{d}\mu = \int_X f \,\mathrm{d}\mu.$$

5.2 Derivatives of PDFs and Their Bounds

The goal of this section is to find derivatives of $q(\mathbf{u})$ and $q(\mathbf{r} \mid \mathbf{u})$ with respect to variational parameters and bound their values. Note that throughout this section the word 'constant' means 'constant with respect to $\mathbf{u}$ and $\mathbf{r}$'. We begin by introducing a few extra variables in order to simplify expressions of derivatives:

$$\mathbf{U} = (\mathbf{u}-\boldsymbol{\mu})(\mathbf{u}-\boldsymbol{\mu})^\intercal,$$
$$\mathbf{S} = \mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1},$$
$$\boldsymbol{\Gamma} = \mathbf{K}_{\mathbf{r},\mathbf{r}} - \mathbf{S}\mathbf{K}_{\mathbf{r},\mathbf{u}},$$
$$\mathbf{R} = \mathbf{S}\frac{\partial \mathbf{K}_{\mathbf{r},\mathbf{u}}}{\partial \lambda_i} - \frac{\partial \mathbf{K}_{\mathbf{r},\mathbf{r}}}{\partial \lambda_i} + \left(\frac{\partial \mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal}{\partial \lambda_i} - \mathbf{S}\frac{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}}{\partial \lambda_i}\right)\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{r},\mathbf{u}}.$$

Next, we state a few key derivatives, derivations of which can be found in Appendix A.

Lemma 5.4 (Derivatives of PDFs).

1. $\frac{\partial q(\mathbf{u})}{\partial \boldsymbol{\mu}} = \frac{1}{2} q(\mathbf{u})(\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}^{-\intercal})(\mathbf{u}-\boldsymbol{\mu})$.
2. (a) $\frac{\partial q(\mathbf{u})}{\partial \boldsymbol{\Sigma}} = \frac{1}{2} q(\mathbf{u})(\boldsymbol{\Sigma}^{-1}\mathbf{U}\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1})$.
   (b) $\frac{\partial q(\mathbf{u})}{\partial \mathbf{B}} = q(\mathbf{u})(\boldsymbol{\Sigma}^{-1}\mathbf{U}\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1})\mathbf{B}$.
3. *For* $i = 0, \dots, d$*,*
   (a)
   $$\frac{\partial q(\mathbf{r} \mid \mathbf{u})}{\partial \lambda_i} = \frac{1}{2} q(\mathbf{r} \mid \mathbf{u})(|\boldsymbol{\Gamma}|^{-1} \operatorname{tr}(\mathbf{R}\operatorname{adj}(\boldsymbol{\Gamma})) - (\mathbf{r}-\mathbf{S}\mathbf{u})^\intercal \boldsymbol{\Gamma}^{-1}\mathbf{R}\boldsymbol{\Gamma}^{-1}(\mathbf{r}-\mathbf{S}\mathbf{u})).$$
   (b) *For any covariance matrix* $\mathbf{K}$*,*
   $$\frac{\partial \mathbf{K}}{\partial \lambda_i} = \begin{cases} \frac{1}{\lambda_i}\mathbf{K} & \text{if } i = 0, \\ \mathbf{L} & \text{otherwise,} \end{cases}$$
   where
   $$L_{j,k} = k(\mathbf{x}_j, \mathbf{x}_k)\left(-\frac{1}{2}(x_{j,i} - x_{k,i})^2 - \mathbb{1}[j \neq k]\sigma^2\right).$$

Lemma 5.5. *Let* $i \in \{0, \dots, d\}$ *be arbitrary, and let* $c\colon \mathbb{R}^{|\mathcal{S}|} \times \mathbb{R}^m \to (\lambda_i - \epsilon, \lambda_i + \epsilon) \subset \mathbb{R}$ *be a function with a codomain arbitrarily close to* λ_i*. Then*

$$\left.\frac{\partial q(\mathbf{r} \mid \mathbf{u})}{\partial \lambda_i}\right|_{\lambda_i = c(\mathbf{r},\mathbf{u})}$$

has upper and lower bounds of the form $q(\mathbf{r} \mid \mathbf{u})d(\mathbf{u})$*, where* $d(\mathbf{u}) \in \mathbb{R}_2[\mathbf{u}]$*.*

Proof. Let $\mathbf{K}$ be any covariance matrix and set

$$\mathbf{A} = \frac{1}{\lambda_0}\mathbf{K}.$$

First, we can easily deduce that[2]

$$\lim_{\epsilon\to 0} \mathbf{K}|_{\lambda_i = c(\mathbf{r},\mathbf{u})} = \mathbf{K}, \tag{12}$$

and

$$\lim_{\epsilon\to 0} \left.\frac{\partial \mathbf{K}}{\partial \lambda_i}\right|_{\lambda_i = c(\mathbf{r},\mathbf{u})} = \frac{\partial \mathbf{K}}{\partial \lambda_i}. \tag{13}$$

For Equation (12), the result is obvious if $i = 0$. Otherwise, it follows from the continuity of the exponential function. For Equation (13), observe that if $i = 0$, then

$$\left.\frac{\partial \mathbf{K}}{\partial \lambda_0}\right|_{\lambda_0 = c(\mathbf{r},\mathbf{u})}$$

[2]Note that we often switch between proving limits and bounds. We are fundamentally interested in proving constant bounds, but sometimes it is more convenient to show convergence instead. In this situation, convergence as $\epsilon \to 0$ implies constant upper and lower bounds.

as an expression contains no λ_0, so

$$\left.\frac{\partial \mathbf{K}}{\partial \lambda_0}\right|_{\lambda_0 = c(\mathbf{r},\mathbf{u})} = \frac{\partial \mathbf{K}}{\partial \lambda_0}.$$

Finally, if $i > 0$, then each element of

$$\left.\frac{\partial \mathbf{K}}{\partial \lambda_i}\right|_{\lambda_i = c(\mathbf{r},\mathbf{u})}$$

is a constant multiple of the corresponding element of

$$\mathbf{K}|_{\lambda_i = c(\mathbf{r},\mathbf{u})},$$

so the same reasoning applies as in case of Equation (12).

Next, we will show that $\mathbf{K}^{-1}|_{\lambda_i = c(\mathbf{r},\mathbf{u})}$ exists and

$$\lim_{\epsilon \to 0} \mathbf{K}^{-1}|_{\lambda_i = c(\mathbf{r},\mathbf{u})} = \mathbf{K}^{-1}. \tag{14}$$

If $i = 0$, then $\mathbf{K}|_{\lambda_i = c(\mathbf{r},\mathbf{u})} = c(\mathbf{r},\mathbf{u})\mathbf{A}$. Therefore[3],

$$\begin{aligned}\mathbf{K}^{-1}|_{\lambda_i = c(\mathbf{r},\mathbf{u})} &= \frac{1}{c(\mathbf{r},\mathbf{u})}\mathbf{A}^{-1} \\ &\to \frac{1}{\lambda_0}\mathbf{A}^{-1} = \frac{1}{\lambda_0}\left(\frac{1}{\lambda_0}\mathbf{K}\right)^{-1} = \mathbf{K}^{-1}\end{aligned}$$

as $\epsilon \to 0$. For $i > 0$, by Equation (12) and continuity of $\mathbf{A} \mapsto \mathbf{A}^{-1}$ we immediate get Equation (14).

This is enough to prove constant upper and lower bounds on $\mathbf{S}$, $\mathbf{\Gamma}$, and $\mathbf{R}$ (all with λ_i replaced with $c(\mathbf{r},\mathbf{u})$), which means that $(\mathbf{r} - \mathbf{S}\mathbf{u})^\intercal \mathbf{\Gamma}^{-1}\mathbf{R}\mathbf{\Gamma}^{-1}(\mathbf{r} - \mathbf{S}\mathbf{u})|_{\lambda_i = c(\mathbf{r},\mathbf{u})}$ has upper and lower bounds in $\mathbb{R}_2[\mathbf{u}]$. Furthermore, having convergence results for arbitrary covariance matrices and their inverses means that

$$\lim_{\epsilon \to 0} \mathbf{\Gamma}|_{\lambda_i = c(\mathbf{r},\mathbf{u})} = \mathbf{\Gamma} \tag{15}$$

and, by continuity of the determinant function,

$$\lim_{\epsilon \to 0} \det(\mathbf{\Gamma})|_{\lambda_i = c(\mathbf{r},\mathbf{u})} = \det(\mathbf{\Gamma}). \tag{16}$$

Assuming that $\mathbf{\Gamma}$ is invertible so that $q(\mathbf{r} \mid \mathbf{u})$ exists,

$$\det(\mathbf{\Gamma})|_{\lambda_i = c(\mathbf{r},\mathbf{u})} \neq 0$$

for small enough ϵ, and, thus, $\det(\mathbf{\Gamma})^{-1}|_{\lambda_i = c(\mathbf{r},\mathbf{u})}$ exists and is bounded.

From Equation (15) and continuity of $\mathbf{A} \mapsto \mathbf{A}^{-1}$ we can immediately deduce that

$$\lim_{\epsilon \to 0} \mathbf{\Gamma}^{-1}|_{\lambda_i = c(\mathbf{r},\mathbf{u})} = \mathbf{\Gamma}^{-1}. \tag{17}$$

Similarly, from Equations (16) and (17) we get that

$$\operatorname{adj}(\mathbf{\Gamma})|_{\lambda_i = c(\mathbf{r},\mathbf{u})} = \det(\mathbf{\Gamma})\mathbf{\Gamma}^{-1}|_{\lambda_i = c(\mathbf{r},\mathbf{u})} \to \det(\mathbf{\Gamma})\mathbf{\Gamma}^{-1}$$

as $\epsilon \to 0$. This gives us constant bounds on

$$|\mathbf{\Gamma}|^{-1} \operatorname{tr}(\mathbf{R} \operatorname{adj}(\mathbf{\Gamma}))|_{\lambda_i = c(\mathbf{r},\mathbf{u})},$$

which completes the proof that

$$\left.\frac{\partial q(\mathbf{r} \mid \mathbf{u})}{\partial \lambda_i}\right|_{\lambda_i = c(\mathbf{r},\mathbf{u})}$$

has the required quadratic bounds. □

[3]Note that since $\lambda_0 > 0$, $c(\mathbf{r},\mathbf{u}) \neq 0$ for small enough ϵ.

REMARK 5.6. *In order to find a derivative such as $\frac{\partial q(\mathbf{u})}{\partial \mu_i}$, we can find $\frac{\partial q(\mathbf{u})}{\partial \boldsymbol{\mu}}$ and simply take the ith element. A similar line of reasoning applies to matrices as well. Thus, we only need to consider derivatives with respect to $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$.*

LEMMA 5.7. *Let $c\colon \mathbb{R}^{|\mathcal{S}|} \times \mathbb{R}^m \to (a,b) \subset \mathbb{R}$ be an arbitrary bounded function. Then, for $i = 1, \ldots, m$, every element of*

$$\left.\frac{\partial q(\mathbf{u})}{\partial \boldsymbol{\mu}}\right|_{\mu_i = c(\mathbf{r},\mathbf{u})}$$

has upper and lower bounds of the form $q(\mathbf{u})d(\mathbf{u})$, where $d(\mathbf{u}) \in \mathbb{R}_1[\mathbf{u}]$.

PROOF. Using Lemma 5.4,

$$\left.\frac{\partial q(\mathbf{u})}{\partial \boldsymbol{\mu}}\right|_{\mu_i = c(\mathbf{r},\mathbf{u})} = \frac{1}{2}q(\mathbf{u})(\mathbf{\Sigma}^{-1} + \mathbf{\Sigma}^{-\intercal})(\mathbf{u} - \mathbf{c}(\mathbf{r},\mathbf{u})),$$

where $\mathbf{c}(\mathbf{r},\mathbf{u}) = (\mu_1, \ldots, \mu_{i-1}, c(\mathbf{r},\mathbf{u}), \mu_{i+1} \ldots, \mu_m)^\intercal$. Since $c(\mathbf{r},\mathbf{u})$ is bounded and $\mathbf{\Sigma}^{-1} + \mathbf{\Sigma}^{-\intercal}$ is a constant matrix, we can use the bounds on $c(\mathbf{r},\mathbf{u})$ to manufacture both upper and lower bounds on

$$\left.\frac{\partial q(\mathbf{u})}{\partial \boldsymbol{\mu}}\right|_{\mu_i = c(\mathbf{r},\mathbf{u})}$$

of the required form. □

LEMMA 5.8. *Let $i, j = 1, \ldots, m$, and let*

$$c\colon \mathbb{R}^{|\mathcal{S}|} \times \mathbb{R}^m \to (\Sigma_{i,j} - \epsilon, \Sigma_{i,j} + \epsilon) \subset \mathbb{R}$$

be a function with a codomain arbitrarily close to $\Sigma_{i,j}$. Then every element of

$$\left.\frac{\partial q(\mathbf{u})}{\partial \mathbf{\Sigma}}\right|_{\Sigma_{i,j} = c(\mathbf{r},\mathbf{u})}$$

has upper and lower bounds of the form $q(\mathbf{u})d(\mathbf{u})$, where $d(\mathbf{u}) \in \mathbb{R}_2[\mathbf{u}]$.

PROOF. Using Lemma 5.4,

$$\left.\frac{\partial q(\mathbf{u})}{\partial \mathbf{\Sigma}}\right|_{\Sigma_{i,j} = c(\mathbf{r},\mathbf{u})} = \frac{1}{2}q(\mathbf{u})(\mathbf{C}(\mathbf{r},\mathbf{u})^{-1}\mathbf{U}\mathbf{C}(\mathbf{r},\mathbf{u})^{-1} - \mathbf{C}(\mathbf{r},\mathbf{u})^{-1}),$$

where

$$[\mathbf{C}(\mathbf{r},\mathbf{u})]_{k,l} = \begin{cases} c(\mathbf{r},\mathbf{u}) & \text{if } (k,l) = (i,j), \\ \Sigma_{k,l} & \text{otherwise.} \end{cases}$$

We can also express $\mathbf{C}(\mathbf{r},\mathbf{u})$ as $\mathbf{C}(\mathbf{r},\mathbf{u}) = \mathbf{\Sigma} + \mathbf{E}(\mathbf{r},\mathbf{u})$, where

$$[\mathbf{E}(\mathbf{r},\mathbf{u})]_{k,l} = \begin{cases} c(\mathbf{r},\mathbf{u}) - \Sigma_{i,j} & \text{if } (k,l) = (i,j), \\ 0 & \text{otherwise.} \end{cases}$$

We will show how we can establish bounds on $\mathbf{C}(\mathbf{r},\mathbf{u})^{-1}$ without using the continuity of matrix inversion. For this, we use the maximum norm $\|\cdot\|_\infty$ on both vectors and matrices. We can apply Lemma 5.2 to $\mathbf{\Sigma}$ and $\mathbf{E}(\mathbf{r},\mathbf{u})$ since

$$\|\mathbf{E}(\mathbf{r},\mathbf{u})\|_\infty = \max_k \sum_l |[\mathbf{E}(\mathbf{r},\mathbf{u})]_{k,l}| = |c(\mathbf{r},\mathbf{u}) - \Sigma_{i,j}| < \epsilon \to 0,$$

ensuring that $\|\mathbf{\Sigma}^{-1}\|_\infty \|\mathbf{E}(\mathbf{r},\mathbf{u})\|_\infty < 1$. Then $\mathbf{C}(\mathbf{r},\mathbf{u})$ is invertible, and

$$\|\mathbf{C}(\mathbf{r},\mathbf{u})^{-1}\|_\infty \leq \frac{\|\mathbf{\Sigma}^{-1}\|_\infty}{1 - \|\mathbf{\Sigma}^{-1}\|_\infty \|\mathbf{E}(\mathbf{r},\mathbf{u})\|_\infty} < \frac{\|\mathbf{\Sigma}^{-1}\|_\infty}{1 - \epsilon\|\mathbf{\Sigma}^{-1}\|_\infty},$$

which means that

$$\max_k \sum_l \left|[\mathbf{C}(\mathbf{r},\mathbf{u})^{-1}]_{k,l}\right| < \frac{\|\boldsymbol{\Sigma}^{-1}\|_\infty}{1-\epsilon\|\boldsymbol{\Sigma}^{-1}\|_\infty},$$

i.e., for any row k and column l,

$$\left|[\mathbf{C}(\mathbf{r},\mathbf{u})^{-1}]_{k,l}\right| < \frac{\|\boldsymbol{\Sigma}^{-1}\|_\infty}{1-\epsilon\|\boldsymbol{\Sigma}^{-1}\|_\infty},$$

which bounds all elements of $\mathbf{C}(\mathbf{r},\mathbf{u})^{-1}$ as required. Since every element of $\mathbf{U} = (\mathbf{u}-\boldsymbol{\mu})(\mathbf{u}-\boldsymbol{\mu})^\intercal$ is in $\mathbb{R}_2[\mathbf{u}]$, and the elements of $\mathbf{C}(\mathbf{r},\mathbf{u})^{-1}$ are bounded, the desired result follows. □

5.3 The MDP Value Function

In this section, we present two results that are necessary for the main proof in Section 5.4 but also interesting in their own right. Firstly, we sketch a proof for the measurability of the MDP value function. Afterwards, we establish upper and lower bounds on said function.

REMARK 5.9. *MDP values are characterised by both a state and a reward function/vector. In this section, we will think of the value function as* $V\colon \mathcal{S} \to \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}$*, i.e.,* V *takes a state* $s \in \mathcal{S}$ *and returns a function* $V(s)\colon \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}$ *that takes a reward vector* $\mathbf{r} \in \mathbb{R}^{|\mathcal{S}|}$ *and returns a value of the state* s*,* $V_\mathbf{r}(s) \in \mathbb{R}$*. Given a reward vector, the function* $V(s)$ *computes the values of all states and returns the value of state* s*.*

PROPOSITION 5.10 (MEASURABILITY). *MDP value functions* $V(s)\colon \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}$ *(for* $s \in \mathcal{S}$*) are Lebesgue measurable.*

PROOF. For any reward vector $\mathbf{r} \in \mathbb{R}^{|\mathcal{S}|}$, the set of converged value functions $\{V_\mathbf{r}(s) \mid s \in \mathcal{S}\}$ satisfy

$$V_\mathbf{r}(s) = \log \sum_{a\in\mathcal{A}} \exp\left(r(s) + \gamma \sum_{s'\in\mathcal{S}} \mathcal{T}(s,a,s') V_\mathbf{r}(s')\right) \quad (18)$$

for all $s \in \mathcal{S}$. Let $s_0 \in \mathcal{S}$ be an arbitrary state. In order to prove that $V(s_0)$ is measurable, it is enough to show that for any $\alpha \in \mathbb{R}$, the set

$$\left\{ \mathbf{r} \in \mathbb{R}^{|\mathcal{S}|} \;\middle|\; \begin{array}{l} V_\mathbf{r}(s_0) \in (-\infty, \alpha); \\ V_\mathbf{r}(s) \in \mathbb{R} \text{ for all } s \in \mathcal{S}\setminus\{s_0\}; \\ \text{Equation (18) is satisfied by all } s \in \mathcal{S} \end{array} \right\}$$

is measurable. Since this set can be constructed in Zermelo-Fraenkel set theory *without* the axiom of choice, it is measurable [10], which proves that $V(s)$ is a measurable function for any $s \in \mathcal{S}$. □

PROPOSITION 5.11 (BOUNDEDNESS). *If the initial values of the MDP value function satisfy the following bound, then the bound remains satisfied throughout value iteration:*

$$|V_\mathbf{r}(s)| \le \frac{\|\mathbf{r}\|_\infty + \log|\mathcal{A}|}{1-\gamma}.$$

PROOF. We begin by considering the desired inequality without taking the absolute value of $V_\mathbf{r}(s)$, i.e.,

$$V_\mathbf{r}(s) \le \frac{\|\mathbf{r}\|_\infty + \log|\mathcal{A}|}{1-\gamma}, \quad (19)$$

and assuming that the initial values of $\{V_\mathbf{r}(s) \mid s \in \mathcal{S}\}$ already satisfy Inequality (19). We will show that it remains satisfied using an inductive argument. Recall that for each $s \in \mathcal{S}$, the value of $V_\mathbf{r}(s)$ is updated by applying Equation (1). Note that both the natural logarithm and the exponential function are increasing, $\gamma > 0$, and the $\mathcal{T}$ function gives a probability (a non-negative number). Thus,

$$\begin{aligned} V_\mathbf{r}(s) &\le \log \sum_{a\in\mathcal{A}} \exp\left(r(s) + \gamma \sum_{s'\in\mathcal{S}} \mathcal{T}(s,a,s') \frac{\|\mathbf{r}\|_\infty + \log|\mathcal{A}|}{1-\gamma}\right) \\ &= \log \sum_{a\in\mathcal{A}} \exp\left(r(s) + \frac{\gamma(\|\mathbf{r}\|_\infty + \log|\mathcal{A}|)}{1-\gamma} \sum_{s'\in\mathcal{S}} \mathcal{T}(s,a,s')\right) \\ &= \log \sum_{a\in\mathcal{A}} \exp\left(r(s) + \frac{\gamma(\|\mathbf{r}\|_\infty + \log|\mathcal{A}|)}{1-\gamma}\right) \end{aligned}$$

by the definition of $\mathcal{T}$. Then

$$\begin{aligned} V_\mathbf{r}(s) &\le \log\left(|\mathcal{A}| \exp\left(r(s) + \frac{\gamma(\|\mathbf{r}\|_\infty + \log|\mathcal{A}|)}{1-\gamma}\right)\right) \\ &= \log\left(\exp\left(\log|\mathcal{A}| + r(s) + \frac{\gamma(\|\mathbf{r}\|_\infty + \log|\mathcal{A}|)}{1-\gamma}\right)\right) \\ &= \log|\mathcal{A}| + r(s) + \frac{\gamma(\|\mathbf{r}\|_\infty + \log|\mathcal{A}|)}{1-\gamma} \\ &= \frac{\gamma(\|\mathbf{r}\|_\infty + \log|\mathcal{A}|) + (1-\gamma)(\log|\mathcal{A}| + r(s))}{1-\gamma} \\ &\le \frac{\gamma(\|\mathbf{r}\|_\infty + \log|\mathcal{A}|) + (1-\gamma)(\log|\mathcal{A}| + \|\mathbf{r}\|_\infty)}{1-\gamma} \\ &= \frac{\|\mathbf{r}\|_\infty + \log|\mathcal{A}|}{1-\gamma} \end{aligned}$$

by the definition of $\|\mathbf{r}\|_\infty$.

The proof for

$$V_\mathbf{r}(s) \ge \frac{\|\mathbf{r}\|_\infty + \log|\mathcal{A}|}{\gamma-1} \quad (20)$$

follows the same argument until we get to

$$\begin{aligned} V_\mathbf{r}(s) &\ge \frac{\gamma(\|\mathbf{r}\|_\infty + \log|\mathcal{A}|) + (\gamma-1)(\log|\mathcal{A}| + r(s))}{\gamma-1} \\ &\ge \frac{\gamma(\|\mathbf{r}\|_\infty + \log|\mathcal{A}|) + (\gamma-1)(-\log|\mathcal{A}| - \|\mathbf{r}\|_\infty)}{\gamma-1} \\ &= \frac{\|\mathbf{r}\|_\infty + \log|\mathcal{A}|}{\gamma-1}, \end{aligned}$$

where we use the fact that $r(s) \ge -\|\mathbf{r}\|_\infty - 2\log|\mathcal{A}|$. Combining Inequalities (19) and (20) gives the desired result. □

LEMMA 5.12.

$$\int \|\mathbf{r}\|_\infty q(\mathbf{r} \mid \mathbf{u})\,\mathrm{d}\mathbf{r} \le a + \|\mathbf{S}\mathbf{u}\|_1,$$

where a *is a constant independent of* $\mathbf{u}$*.*

PROOF. Since $\|\mathbf{r}\|_\infty \le \|\mathbf{r}\|_1$,

$$\int \|\mathbf{r}\|_\infty q(\mathbf{r} \mid \mathbf{u})\,\mathrm{d}\mathbf{r} \le \int \|\mathbf{r}\|_1 q(\mathbf{r} \mid \mathbf{u})\,\mathrm{d}\mathbf{r} = \sum_{i=1}^{|\mathcal{S}|} \mathbb{E}[|r_i|].$$

As each $\mathbb{E}[|r_i|]$ is a mean of a folded Gaussian distribution,

$$\mathbb{E}[|r_i|] = \sigma_i\sqrt{\frac{2}{\pi}} \exp\left(-\frac{\xi_i^2}{2\sigma_i^2}\right) + \xi_i\left(1 - 2\Phi\left(-\frac{\xi_1}{\sigma_1}\right)\right),$$

where $\xi_i = [\mathbf{Su}]_i$, $\sigma_i = \sqrt{\Gamma_{i,i}}$[4], and Φ is the cumulative distribution function of the standard Gaussian. Furthermore,

$$\mathbb{E}[|r_i|] \leq \sigma_i \sqrt{\frac{2}{\pi}} + |\xi_i|,$$

as σ_i is non-negative, and $\Phi(x) \in [0, 1]$ for all x. Since

$$\sum_{i=1}^{|\mathcal{S}|} |\xi_i| = \|\mathbf{Su}\|_1,$$

we can set

$$a = \sum_{i=1}^{|\mathcal{S}|} \sigma_i \sqrt{\frac{2}{\pi}}$$

to get the desired result. □

5.4 Differentiability

Our main theorem is a specialised version of an integral differentiation result by Timoney [27].

THEOREM 5.13. *Whenever the derivative exists,*

$$\frac{\partial}{\partial t} \iint V_{\mathbf{r}}(s) q(\mathbf{r} \mid \mathbf{u}) q(\mathbf{u}) \, \mathrm{d}\mathbf{r} \, \mathrm{d}\mathbf{u} = \iint \frac{\partial}{\partial t} [V_{\mathbf{r}}(s) q(\mathbf{r} \mid \mathbf{u}) q(\mathbf{u})] \, \mathrm{d}\mathbf{r} \, \mathrm{d}\mathbf{u},$$

where t is any scalar part of $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$, or $\boldsymbol{\lambda}$.

PROOF. Let

$$f(\mathbf{r}, \mathbf{u}, t) = V_{\mathbf{r}}(s) q(\mathbf{r} \mid \mathbf{u}) q(\mathbf{u}),$$
$$F(t) = \iint f(\mathbf{r}, \mathbf{u}, t) \, \mathrm{d}\mathbf{r} \, \mathrm{d}\mathbf{u},$$

and fix the value of t. Let $(t_n)_{n=1}^{\infty}$ be any sequence such that $\lim_{n\to\infty} t_n = t$, but $t_n \neq t$ for all n. We want to show that

$$F'(t) = \lim_{n\to\infty} \frac{F(t_n) - F(t)}{t_n - t} = \iint \left. \frac{\partial f}{\partial t} \right|_{(\mathbf{r},\mathbf{u},t)} \mathrm{d}\mathbf{r} \, \mathrm{d}\mathbf{u}. \quad (21)$$

We have

$$\frac{F(t_n) - F(t)}{t_n - t} = \iint \frac{f(\mathbf{r}, \mathbf{u}, t_n) - f(\mathbf{r}, \mathbf{u}, t)}{t_n - t} \, \mathrm{d}\mathbf{r} \, \mathrm{d}\mathbf{u}$$
$$= \iint f_n(\mathbf{r}, \mathbf{u}) \, \mathrm{d}\mathbf{r} \, \mathrm{d}\mathbf{u},$$

where

$$f_n(\mathbf{r}, \mathbf{u}) = \frac{f(\mathbf{r}, \mathbf{u}, t_n) - f(\mathbf{r}, \mathbf{u}, t)}{t_n - t}.$$

Since

$$\lim_{n\to\infty} f_n(\mathbf{r}, \mathbf{u}) = \left. \frac{\partial f}{\partial t} \right|_{(\mathbf{r},\mathbf{u},t)},$$

Equation (21) follows from Theorem 5.3 as soon as we show that both f and f_n are measurable and find a non-negative integrable function g such that for all n, $\mathbf{r}$, $\mathbf{u}$,

$$|f_n(\mathbf{r}, \mathbf{u})| \leq g(\mathbf{r}, \mathbf{u}).$$

The MDP value function is measurable by Proposition 5.10. The result of multiplying or adding measurable functions (e.g., PDFs) to a measurable function is still measurable. Thus, both f and f_n are measurable.

[4]The diagonal of $\boldsymbol{\Gamma}$ is non-negative because $\boldsymbol{\Gamma}$ is a covariance matrix of a Gaussian distribution, hence also positive semi-definite.

It remains to find g. For notational simplicity and without loss of generality, we will temporarily assume that t is a parameter of $q(\mathbf{r} \mid \mathbf{u})$. Then

$$|f_n(\mathbf{r}, \mathbf{u})| = |V_{\mathbf{r}}(s)| \left| \frac{q(\mathbf{r} \mid \mathbf{u})|_{t=t_n} - q(\mathbf{r} \mid \mathbf{u})}{t_n - t} \right| q(\mathbf{u}),$$

since PDFs are non-negative. An upper bound for $|V_{\mathbf{r}}(s)|$ is given by Proposition 5.11, while

$$\frac{q(\mathbf{r} \mid \mathbf{u})|_{t=t_n} - q(\mathbf{r} \mid \mathbf{u})}{t_n - t} = \left. \frac{\partial q(\mathbf{r} \mid \mathbf{u})}{\partial t} \right|_{t=c(\mathbf{r},\mathbf{u})}$$

for some function $c\colon \mathbb{R}^{|\mathcal{S}|} \times \mathbb{R}^m \to (\min\{t, t_n\}, \max\{t, t_n\})$ due to the mean value theorem (since q is a continuous and differentiable function of t, regardless of the specific choices of q and t).

We then have that

$$|f_n(\mathbf{r}, \mathbf{u})| \leq \frac{\|\mathbf{r}\|_\infty + \log|\mathcal{A}|}{1 - \gamma} \left| \left. \frac{\partial q(\mathbf{r} \mid \mathbf{u})}{\partial t} \right|_{t=c(\mathbf{r},\mathbf{u})} \right| q(\mathbf{u}).$$

The bound is clearly non-negative and measurable. It remains to show that it is also integrable. Depending on what t represents, we can use one of the Lemmas 5.5, 5.7, and 5.8, which gives us two polynomials $p_1(\mathbf{u}), p_2(\mathbf{u}) \in \mathbb{R}_2[\mathbf{u}]$ such that

$$p_1(\mathbf{u}) q(\mathbf{r} \mid \mathbf{u}) < \left. \frac{\partial q(\mathbf{r} \mid \mathbf{u})}{\partial t} \right|_{t=c(\mathbf{r},\mathbf{u})} < p_2(\mathbf{u}) q(\mathbf{r} \mid \mathbf{u}).$$

Then

$$\left| \left. \frac{\partial q(\mathbf{r} \mid \mathbf{u})}{\partial t} \right|_{t=c(\mathbf{r},\mathbf{u})} \right| < q(\mathbf{r} \mid \mathbf{u}) \max\{|p_1(\mathbf{u})|, |p_2(\mathbf{u})|\}.$$

We can now apply Lemma 5.12, which allows us to integrate out $\mathbf{r}$, and we are left with showing the existence of

$$\int (a + \|\mathbf{Su}\|_1) \max\{|p_1(\mathbf{u})|, |p_2(\mathbf{u})|\} q(\mathbf{u}) \, \mathrm{d}\mathbf{u}, \quad (22)$$

where a is a constant. The integral

$$\int \max \left\{ \begin{matrix} |p_1(\mathbf{u})|, \\ |p_2(\mathbf{u})| \end{matrix} \right\} q(\mathbf{u}) \, \mathrm{d}\mathbf{u} = \int \max \left\{ \begin{matrix} |p_1(\mathbf{u}) q(\mathbf{u})|, \\ |p_2(\mathbf{u}) q(\mathbf{u})| \end{matrix} \right\} \mathrm{d}\mathbf{u}$$

exists because $p_1(\mathbf{u})q(\mathbf{u})$ and $p_2(\mathbf{u})q(\mathbf{u})$ are both integrable, hence their absolute values are integrable, and the maximum of two integrable functions is also integrable. Since $\|\mathbf{Su}\|_1 \in \mathbb{R}_1[\mathbf{u}]$, a similar argument can be applied to the rest of Integral (22) as well. □

Having proven that differentiation can be moved inside the integral allows us to use $\nabla\mathcal{L}$ for optimisation, knowing that our approach is sound. This allows us to implement a simple optimisation algorithm, the description of which is in the next section.

6. EXPERIMENTS

The main goal of this section is to use experimental evidence to show that the implementation agrees with the theory and to investigate its properties such as convergence and how the learned model changes with varying amounts and types of data. We begin with an outline of the implementation in Algorithm 1. The algorithm was implemented in

MATLAB to be compatible with the toolkit[5] for evaluating IRL algorithms developed by Levine et al. [15]. We decided to use a constant step size because the stochastic nature of the gradient makes dynamic step size algorithms such as AdaGrad [8] and AdaDelta [31] ineffective, as spikes in the gradient estimates drive the step size down to zero. Similarly, although we check for convergence using the ℓ_∞-norm, the algorithm usually only stops when the maximum number of iterations is exhausted. Throughout the experiments, we discard runs that result in a near-singular $\mathbf{\Sigma}$ or $\mathbf{\Gamma}$, repeatedly testing each combination of parameters as many times as necessary until a non-failing run is achieved.

Data: number of samples S, constant step size stepSize, convergence criteria tolerance and numIterations
Result: optimised parameters
initialise parameters as in Equations (7)–(11);
for $i \leftarrow 1$ **to** numIterations **do**
  $\{\mathbf{u}_s\}_{s=1}^{S} \sim \mathcal{N}(\text{parameters.}\boldsymbol{\mu}, \text{parameters.}\mathbf{\Sigma})$;
  $\nabla\mathcal{L} \leftarrow \texttt{exactPart(parameters)} + \frac{1}{S}\sum_{s=1}^{S} \texttt{estimatedPart(parameters, } \mathbf{u}_s\texttt{)}$;
  $\mathbf{v} \leftarrow \texttt{pack(parameters)}$;
  $\mathbf{v}' \leftarrow \mathbf{v} + \text{stepSize} \times \texttt{transformGradient}(\nabla\mathcal{L})$;
  parameters $\leftarrow \texttt{unpack}(\mathbf{v}')$;
  if $\|\mathbf{v}' - \mathbf{v}\|_\infty <$ tolerance **then** break;
end
Function estimatedPart(parameters, $\mathbf{u}$)
  $\mathbf{r} \sim \mathcal{N}(\text{parameters.}\mathbf{Su}, \text{parameters.}\mathbf{\Gamma})$;
  $V_\mathbf{r} \leftarrow \texttt{valueIteration(r)}$;
  return *the estimated part of* $\nabla\mathcal{L}$ *according to* $\mathbf{u}$, $\mathbf{r}$, $V_\mathbf{r}$, *and* parameters;
end

Algorithm 1: The optimisation procedure. All parameters are kept in a single variable parameters. Converting it to/from vector form and taking logarithms, exponentiating, and using Equation (6) to handle the optimisation of parameters restricted to positive values is handled by functions `pack`, `unpack`, and `transformGradient`. Each derivative from Appendix B is split into parts outside and inside $\mathbb{E}[\cdot]$ that are handled by functions `exactPart` and `estimatedPart`, respectively. The function `valueIteration` simply applies Equation (1) until convergence.

We will first focus on a simple three-state MDP where the agent can deterministically move from any state to any other state. More formally, we set $\mathcal{S} = \{s_1, s_2, s_3\}$, $\mathcal{A} = \{a_1, a_2\}$,

$$\begin{aligned} \mathcal{T}(s_1, a_1, s_2) &= 1, \quad \mathcal{T}(s_1, a_2, s_3) = 1, \\ \mathcal{T}(s_2, a_1, s_1) &= 1, \quad \mathcal{T}(s_1, a_2, s_3) = 1, \\ \mathcal{T}(s_3, a_1, s_1) &= 1, \quad \mathcal{T}(s_1, a_2, s_2) = 1, \end{aligned}$$

all other values of $\mathcal{T}$ to zero, and $\gamma = 0.9$. We also set the inducing points to be equal to the three states in $\mathcal{S}$, add a single feature $f\colon \mathcal{S} \to \mathbb{R}$ such that

$$f(s_1) = 1, \quad f(s_2) = 2, \quad f(s_3) = 3,$$

and create two demonstrations $\zeta_1 = \{(s_1, a_1)\}$ and $\zeta_2 = \{(s_3, a_2)\}$ that correspond to moving from s_1 and s_3 to s_2.

[5] `https://graphics.stanford.edu/projects/gpirl/`

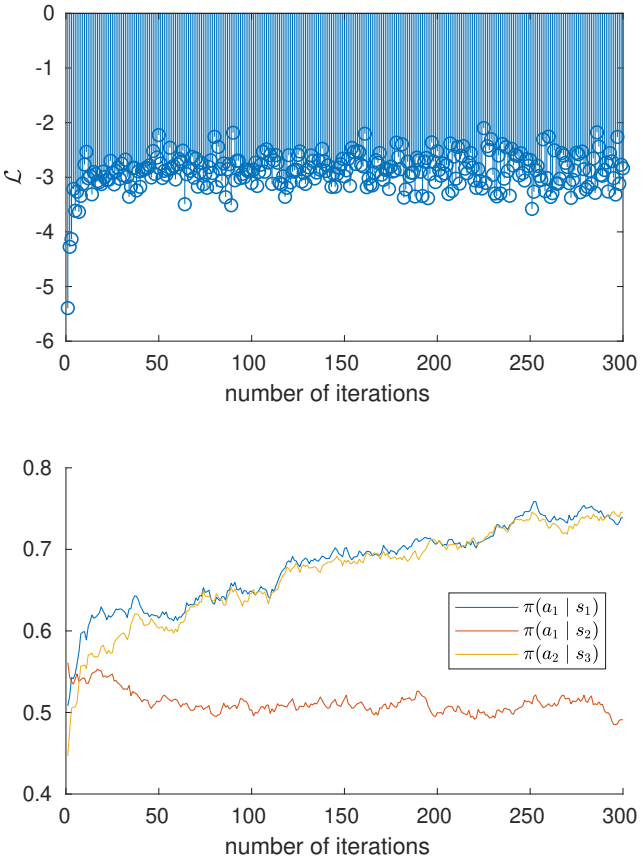

Figure 4: Convergence of $\mathcal{L}$ (at the top) as well as several example policies (at the bottom)

With this setup, we would expect the reward of s_2 to be higher than the other two rewards.

Convergence.

We plot how $\mathcal{L}$ as well as policies $\pi(a_1 \mid s_1)$, $\pi(a_2 \mid s_3)$, and $\pi(a_1 \mid s_2)$ converge over a number of iterations in Figure 4. The first two policies correspond to actions taken in the set of demonstrations $\mathcal{D}$, so we would expect to see their probabilities converge to values near 1. The third policy, however, has no data in $\mathcal{D}$ to guide its value, so the maximum causal entropy framework would put the probability at around 0.5. The first thing to note is that while $\mathcal{L}$ converges in just a few iterations, policies continue to improve for much longer. In addition, all policies behave as expected: $\pi(a_1 \mid s_2)$ stays around 0.5, while both $\pi(a_1 \mid s_1)$ and $\pi(a_2 \mid s_3)$ keep increasing. Although they stay at around 0.75 (and not 1), this may be optimal for the maximum causal entropy model (and this question will be addressed in more detail with a different experiment).

We also plot how the parameters of the model converge in Figure 5. While some parameters could benefit from a higher number of iterations[6], none look alarmingly wrong. Note that μ_2, a variable closely related to $r(s_2)$, stabilises at a positive value whereas both μ_1 and μ_2 move to negative

[6] The limit of 300 iterations was chosen so that the initial increase in $\mathcal{L}$ would not be overshadowed by a long straight line afterwards.

values as expected.

Adding More Data.

We would expect policies to converge closer to extreme values (i.e., 0 and 1) as we add more data to the model. We test this hypothesis with a small-scale experiment in Figure 6. The policy $\pi(a_1 \mid s_2)$, in particular, confirms our prediction. As this policy denotes the probability of going from s_2 to s_1, we expect it to stay around 0.5 as long as we have equal amounts of ζ_1 and ζ_2—this corresponds to plots at coordinates $(1,1)$, $(2,2)$, and $(3,3)$. Moreover, as we increase the number of times state s_1 appears in our demonstrations (going to the right across any row of plots), the model recognises the increasing value of state s_1, and the probability increases. In contrast, going up across the plots reduces the probability, as state s_3 becomes more valuable, and the model prefers it over state s_1. Interestingly, the other two policies seem to increase regardless of what data is added. This is likely due to the fact that a higher number of demonstrations results in higher gradient values, and thus leads to better-converged policies.

We would also like to see similar changes in reward covariances. Firstly, we will show how to derive the covariance matrix for $\mathbf{r}$ from the posterior distributions of $\mathbf{u}$ and $\mathbf{r} \mid \mathbf{u}$. Remember that $\mathbf{u} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\mathbf{r} \mid \mathbf{u} \sim \mathcal{N}(\mathbf{S}\mathbf{u}, \boldsymbol{\Gamma})$, and let $\mathbf{r} \sim \mathcal{N}(\boldsymbol{\mu}', \boldsymbol{\Sigma}')$, where our goal is to find $\boldsymbol{\mu}'$ and $\boldsymbol{\Sigma}'$. Then,

$$\boldsymbol{\mu}' = \mathbb{E}[\mathbf{r}] = \mathbb{E}[\mathbf{S}\mathbf{u}] = \mathbf{S}\boldsymbol{\mu}.$$

Furthermore,

$$\begin{aligned}\mathbb{E}_{\mathbf{r}}[\mathbf{r}\mathbf{r}^\intercal] &= \boldsymbol{\Gamma} + \mathbb{E}_{\mathbf{u}}[\mathbf{S}\mathbf{u}\mathbf{u}^\intercal\mathbf{S}^\intercal] = \boldsymbol{\Gamma} + \mathbf{S}\mathbb{E}_{\mathbf{u}}[\mathbf{u}\mathbf{u}^\intercal]\mathbf{S}^\intercal \\ &= \boldsymbol{\Gamma} + \mathbf{S}(\boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^\intercal)\mathbf{S}^\intercal\end{aligned}$$

by an identity for $\mathbb{E}[\mathbf{x}\mathbf{x}^\intercal]$ by Petersen and Pedersen [20]. Now we can find the covariance matrix as follows:

$$\begin{aligned}\boldsymbol{\Sigma}' &= \mathbb{E}[(\mathbf{r} - \mathbf{S}\boldsymbol{\mu})(\mathbf{r} - \mathbf{S}\boldsymbol{\mu})^\intercal] \\ &= \mathbb{E}[\mathbf{r}\mathbf{r}^\intercal - \mathbf{r}\boldsymbol{\mu}^\intercal\mathbf{S}^\intercal - \mathbf{S}\boldsymbol{\mu}\mathbf{r}^\intercal + \mathbf{S}\boldsymbol{\mu}\boldsymbol{\mu}^\intercal\mathbf{S}^\intercal] \\ &= \boldsymbol{\Gamma} + \mathbf{S}\boldsymbol{\Sigma}\mathbf{S}^\intercal.\end{aligned}$$

Therefore, $\mathbf{r} \sim \mathcal{N}(\mathbf{S}\boldsymbol{\mu}, \boldsymbol{\Gamma} + \mathbf{S}\boldsymbol{\Sigma}\mathbf{S}^\intercal)$.

Secondly, we normalise the elements of this new covariance matrix into the interval $[-1, 1]$ for visualisation purposes. Diagonal (state) and non-diagonal (edge) covariances are normalised separately in order to represent the full range of values more clearly. All covariances are normalised according to this rule:

$$x \mapsto \begin{cases} x/M & \text{if } x \geq 0 \\ -x/m & \text{otherwise,} \end{cases}$$

where M is the maximum (diagonal or non-diagonal) covariance across all data, and m is the minimum. This way, all covariances are scaled to $[-1, 1]$, with both extreme points guaranteed to be reached unless all values are on one side of zero on the real number axis.

However, the resulting plot in Figure 7 shows no clear pattern, suggesting that adding more of the same demonstrations does not result in lower covariances. The only other observation from this plot is that there are only two instances of negative covariance, and both of them are between states s_2 and s_3.

Structure versus Randomness.

If having more copies of the same demonstrations has no effect on covariances, we can test if covariances tend to be lower in situations that exhibit more structure, i.e., where demonstrations provide a clearer picture of which states are more valuable. For this, we consider a completely deterministic MDP with ten states and nine actions that allow the agent to move from any state to any other state—the *clique MDP*. We denote the states by $\mathcal{S} = \{s_i\}_{i=0}^{9}$, and, similarly to the previous example, set up a single feature f such that $f(s_i) = i$ for $i = 0, \ldots, 9$. Finally, $\gamma = 0.9$ as before. We consider two scenarios, with 100 randomly generated demonstrations in each. The scenarios are visualised in Figure 8, although we only show a small subset of demonstrations. In the first scenario, we draw both the starting state and the action uniformly at random. In the second scenario, we draw the starting state uniformly from $\mathcal{S} \setminus \{s_0\}$, and the action always points to s_0. We would expect at least variances of state rewards to be lower in the second scenario in order to reflect the more structured, certain behavioural pattern expressed by the demonstrations. Indeed, Figure 9 shows exactly that. The first two box plots show that in roughly 3/4 of the runs reward variances were higher in the random scenario[7]. Interestingly, the same applies to covariances.

7. CONCLUSIONS AND FURTHER WORK

In this paper, we showed how VI can be used to solve the IRL problem. We proved the validity of our approach, establishing subsidiary results such as new bounds for the linearly solvable MDP model along the way. The implementation of the model was well-behaved and showed that covariance can indeed be used to recognise a well-learned model from a model which is less sure about what policies are optimal.

Building on that, an interesting question for MDP (or, perhaps, dynamical systems) research would be: how much does a reward have to change in order to affect the deterministic policy? A simple answer to this question would allow us to use variance estimates in order to quantify the model's confidence regarding optimal behaviour.

It is also worth noting that the approach presented in this paper requires solving S MDPs for every iteration of optimisation (where S is the number of samples drawn from $q(\mathbf{u}, \mathbf{r})$). Thus, any improvements in MDP solving speed are highly desirable. Two simple improvements can come from requiring less precision when running value iteration and from reusing the solutions of previous MDPs to initialise the new (unsolved) MDP.

Finally, an interesting extension to our work would be to consider IRL in the context of a reinforcement learning (RL) agent. Suppose we have an agent whose purpose is to learn optimal behaviour from observing other agents using IRL. It could then take reward variance estimates into account when choosing what states to visit next. It would have to handle the balance between exploration and exploitation similarly to many RL agents, but the information about rewards would come from observing (presumably near-optimal) behaviour exhibited by other agents rather than directly from the environment.

8. REFERENCES

[7]Note that outliers were removed from the plot. However, all outliers were positive.

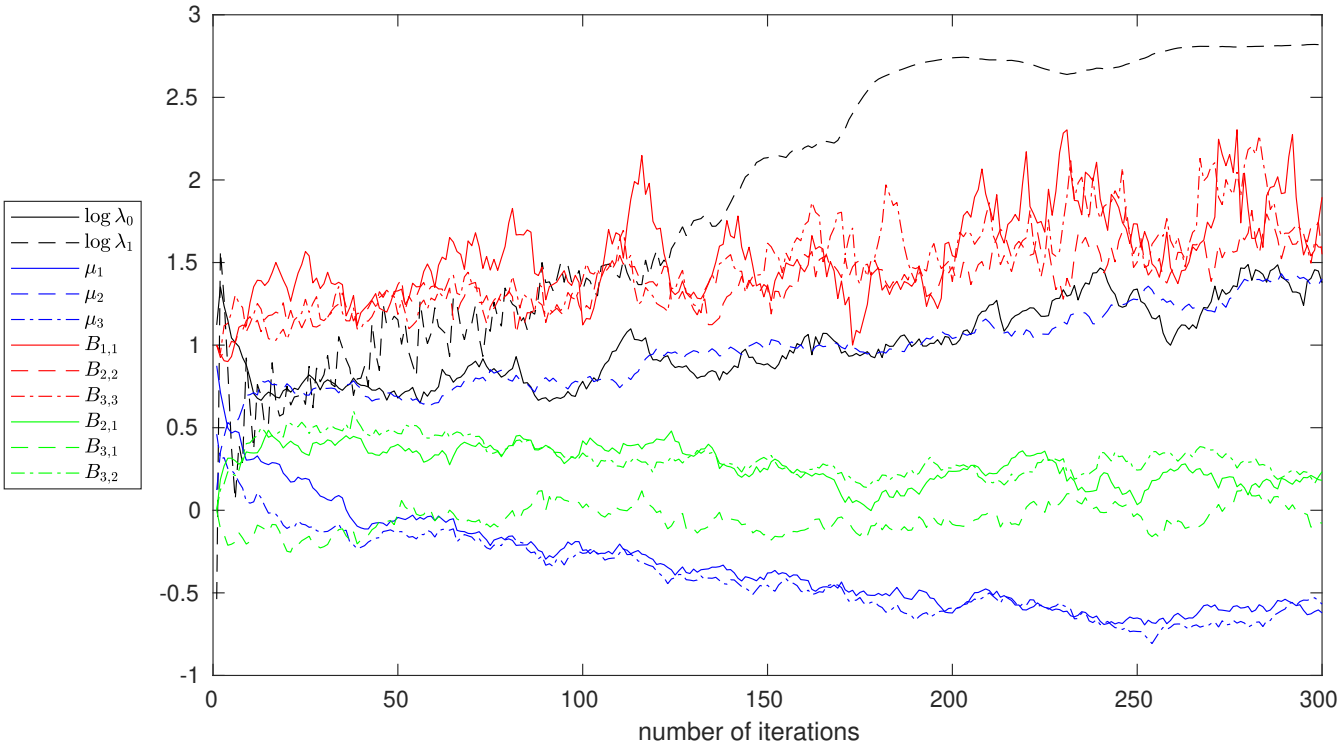

Figure 5: Convergence of variational parameters. In order to represent different variables on the same scale, some variables have been log-transformed. Colours denote which vector or matrix each scalar comes from: black for $\boldsymbol{\lambda}$, blue for $\boldsymbol{\mu}$, red for diagonal elements of B, and green for its non-diagonal elements.

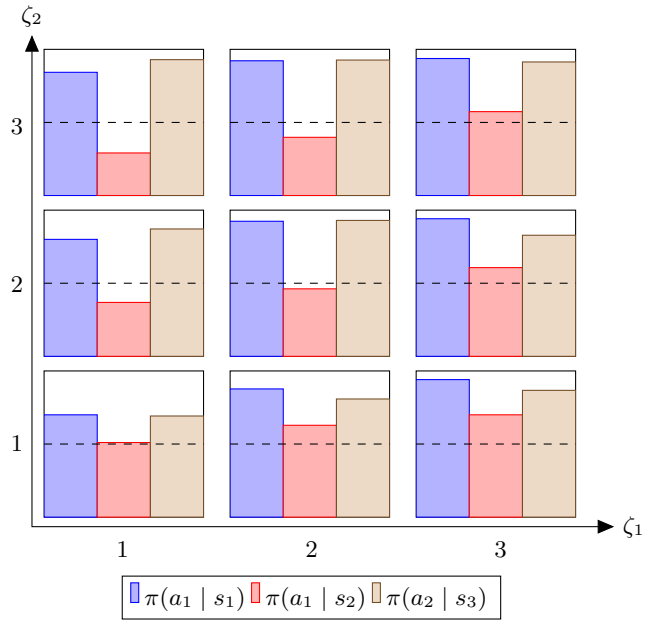

Figure 6: Changes in policies after adding more demonstrations. The x-axis (y-axis) shows how many copies of ζ_1 (ζ_2) are in $\mathcal{D}$. Each plot shows the values of three policies after 300 iterations (averaged out over ten runs). The bottom (top) of each bar plot corresponds to probability 0 (1), and dashed lines mark probability 0.5.

[1] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng. An application of reinforcement learning to aerobatic helicopter flight. In B. Schölkopf, J. C. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pages 1–8. MIT Press, 2006.

[2] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In C. E. Brodley, editor, *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004*, volume 69 of *ACM International Conference Proceeding Series*. ACM, 2004.

[3] C. M. Bishop. *Pattern recognition and machine learning, 5th Edition*. Information science and statistics. Springer, 2007.

[4] 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.

[5] S. Bochner. *Lectures on Fourier integrals*. Princeton University Press, 1959.

[6] C. Cheng and B. Boots. Variational inference for Gaussian process models with linear complexity. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural*

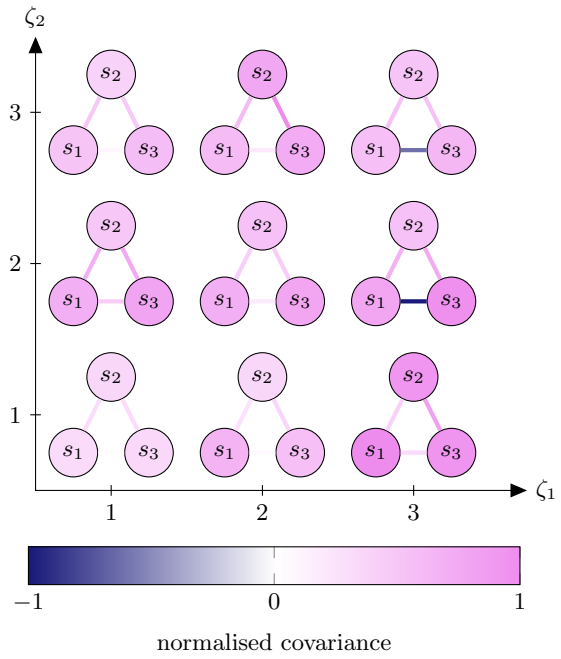

Figure 7: Changes in reward covariances after adding more demonstrations. The meaning of external axes is the same as in Figure 6. Colours denote reward covariance values, e.g., the colour of node s_1 denotes the variance of $r(s_1)$, and the colour of the edge between nodes s_1 and s_2 denotes the covariance between $r(s_1)$ and $r(s_2)$. The colours represent median covariances across ten runs, normalised to the interval $[-1, 1]$ while preserving their positivity/negativity.

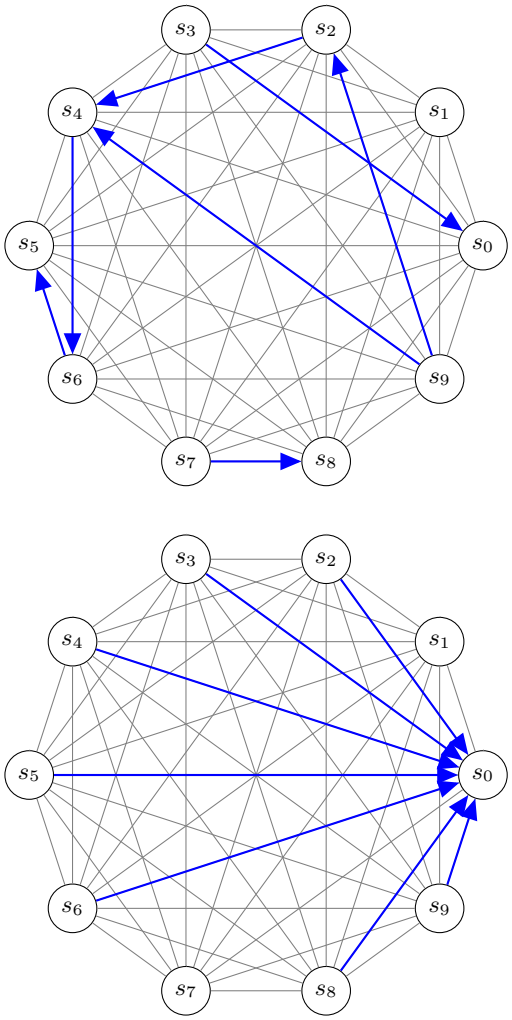

Figure 8: The clique MDP with two sets of demonstrations denoted by blue arrows

Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, pages 5190–5200, 2017.

[7] J. Duchi. Derivations for linear algebra and optimization. Stanford University.

[8] J. C. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[9] J. Hensman, N. Durrande, and A. Solin. Variational Fourier features for Gaussian processes. *Journal of Machine Learning Research*, 18:151:1–151:52, 2017.

[10] H. Herrlich. *Axiom of choice*. Springer, 2006.

[11] M. Jin, A. C. Damianou, P. Abbeel, and C. J. Spanos. Inverse reinforcement learning via deep Gaussian process. In G. Elidan, K. Kersting, and A. T. Ihler, editors, *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press, 2017.

[12] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard. Socially compliant mobile robot navigation via inverse reinforcement learning. *I. J. Robotics Res.*, 35(11):1289–1307, 2016.

[13] S. Laue, M. Mitterreiter, and J. Giesen. Computing higher order derivatives of matrix and tensor expressions. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 2755–2764, 2018.

[14] W. Layton and M. Sussman. *Numerical linear algebra*. Lulu.com, 2014.

[15] S. Levine, Z. Popovic, and V. Koltun. Nonlinear inverse reinforcement learning with Gaussian processes. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.*, pages 19–27, 2011.

[16] S. Levine, Z. Popovic, and V. Koltun. Supplementary material: Nonlinear inverse reinforcement learning with Gaussian processes. `http://graphics.stanford.edu/projects/gpirl/gpirl_supplement.pdf`, December 2011.

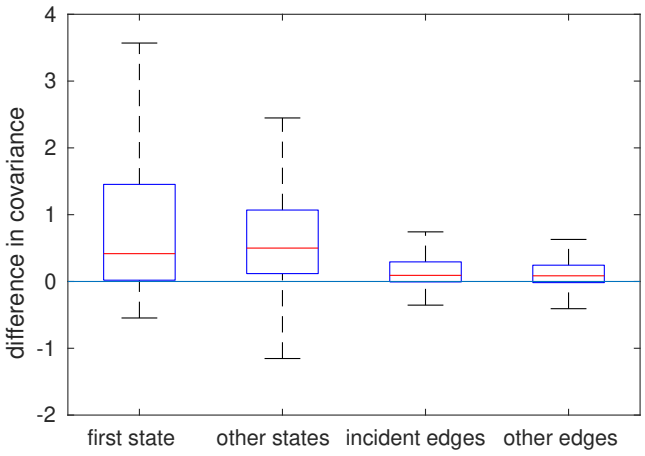

Figure 9: Box plots of the difference between (absolute values of) reward covariances in random and semi-structured scenarios from Figure 8, averaged out over 100 runs and grouped into four key categories. 'First state' refers to the variance of $r(s_0)$, the reward of the state targeted by all demonstrations in the semi-structured case. 'Other states' are variances of other states of the MDP, i.e., diagonal entries of the covariance matrix. 'Incident edges' refers to covariances between $r(s_0)$ and all other states, i.e., the first row or column of the covariance matrix. Finally, 'other edges' is the category for the remaining covariances between pairs of rewards.

[17] H. Liu, Y. Ong, X. Shen, and J. Cai. When Gaussian process meets big data: A review of scalable GPs. *CoRR*, abs/1807.01065, 2018.

[18] A. Y. Ng and S. J. Russell. Algorithms for inverse reinforcement learning. In P. Langley, editor, *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pages 663–670. Morgan Kaufmann, 2000.

[19] V. M.-H. Ong, D. J. Nott, and M. S. Smith. Gaussian variational approximation with a factor covariance structure. *Journal of Computational and Graphical Statistics*, 27(3):465–478, 2018.

[20] K. B. Petersen, M. S. Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008.

[21] R. Ranganath, S. Gerrish, and D. M. Blei. Black box variational inference. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, AISTATS 2014, Reykjavik, Iceland, April 22-25, 2014*, volume 33 of *JMLR Workshop and Conference Proceedings*, pages 814–822. JMLR.org, 2014.

[22] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006.

[23] D. J. Rezende and S. Mohamed. Variational inference with normalizing flows. In F. R. Bach and D. M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1530–1538. JMLR.org, 2015.

[24] H. Royden and P. Fitzpatrick. *Real Analysis*. Prentice Hall, 2010.

[25] S. J. Russell. Learning agents for uncertain environments (extended abstract). In P. L. Bartlett and Y. Mansour, editors, *Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT 1998, Madison, Wisconsin, USA, July 24-26, 1998.*, pages 101–103. ACM, 1998.

[26] S. J. Russell and P. Norvig. *Artificial Intelligence - A Modern Approach (3. internat. ed.)*. Pearson Education, 2010.

[27] R. M. Timoney. The dominated convergence theorem and applications. Trinity College Dublin, `https://www.maths.tcd.ie/~richardt/MA2224/MA2224-ch4.pdf`, March 2018.

[28] A. Vogel, D. Ramachandran, R. Gupta, and A. Raux. Improving hybrid vehicle fuel efficiency using inverse reinforcement learning. In J. Hoffmann and B. Selman, editors, *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada*. AAAI Press, 2012.

[29] A. Wilson and R. Adams. Gaussian process kernels for pattern discovery and extrapolation. In S. Dasgupta and D. McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1067–1075, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[30] M. Wulfmeier, P. Ondruska, and I. Posner. Maximum entropy deep inverse reinforcement learning. *arXiv preprint arXiv:1507.04888*, 2015.

[31] M. D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.

[32] C. Zhang, J. Bütepage, H. Kjellström, and S. Mandt. Advances in variational inference. *CoRR*, abs/1711.05597, 2017.

[33] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

[34] B. D. Ziebart, A. L. Maas, A. K. Dey, and J. A. Bagnell. Navigate like a cabbie: probabilistic reasoning from observed context-aware behavior. In H. Y. Youn and W. Cho, editors, *UbiComp 2008: Ubiquitous Computing, 10th International Conference, UbiComp 2008, Seoul, Korea, September 21-24, 2008, Proceedings*, volume 344 of *ACM International Conference Proceeding Series*, pages 322–331. ACM, 2008.

[35] B. D. Ziebart, N. D. Ratliff, G. Gallagher, C. Mertz, K. M. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. S. Srinivasa. Planning-based prediction for pedestrians. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 11-15, 2009, St. Louis, MO, USA*, pages 3931–3936. IEEE, 2009.

APPENDIX

A. PROOFS

LEMMA 5.4 (DERIVATIVES OF PDFS).

1. $\frac{\partial q(\mathbf{u})}{\partial \boldsymbol{\mu}} = \frac{1}{2} q(\mathbf{u})(\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}^{-\intercal})(\mathbf{u} - \boldsymbol{\mu})$.
2. (a) $\frac{\partial q(\mathbf{u})}{\partial \boldsymbol{\Sigma}} = \frac{1}{2} q(\mathbf{u})(\boldsymbol{\Sigma}^{-1}\mathbf{U}\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1})$.
   (b) $\frac{\partial q(\mathbf{u})}{\partial \mathbf{B}} = q(\mathbf{u})(\boldsymbol{\Sigma}^{-1}\mathbf{U}\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1})\mathbf{B}$.
3. *For* $i = 0, \dots, d$,
   (a)
$$\frac{\partial q(\mathbf{r} \mid \mathbf{u})}{\partial \lambda_i} = \frac{1}{2} q(\mathbf{r} \mid \mathbf{u})(|\boldsymbol{\Gamma}|^{-1} \operatorname{tr}(\mathbf{R} \operatorname{adj}(\boldsymbol{\Gamma})) - (\mathbf{r} - \mathbf{S}\mathbf{u})^\intercal \boldsymbol{\Gamma}^{-1} \mathbf{R} \boldsymbol{\Gamma}^{-1} (\mathbf{r} - \mathbf{S}\mathbf{u})).$$
   (b) *For any covariance matrix* $\mathbf{K}$,
$$\frac{\partial \mathbf{K}}{\partial \lambda_i} = \begin{cases} \frac{1}{\lambda_i}\mathbf{K} & \text{if } i = 0, \\ \mathbf{L} & \text{otherwise,} \end{cases}$$
   where
$$L_{j,k} = k(\mathbf{x}_j, \mathbf{x}_k) \left(-\frac{1}{2}(x_{j,i} - x_{k,i})^2 - \mathbb{1}[j \neq k]\sigma^2 \right).$$

PROOF.

1.
$$\begin{aligned} \frac{\partial q(\mathbf{u})}{\partial m} &= q(\mathbf{u}) \frac{\partial}{\partial \boldsymbol{\mu}} \left[-\frac{(\mathbf{u} - \boldsymbol{\mu})^\intercal \boldsymbol{\Sigma}^{-1} (\mathbf{u} - \boldsymbol{\mu})}{2} \right] \\ &= -\frac{1}{2} q(\mathbf{u})(\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}^{-\intercal})(\mathbf{u} - \boldsymbol{\mu}) \frac{\partial}{\partial \boldsymbol{\mu}} [\mathbf{u} - \boldsymbol{\mu}] \\ &= \frac{1}{2} q(\mathbf{u})(\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}^{-\intercal})(\mathbf{u} - \boldsymbol{\mu}). \end{aligned}$$

2. An online tool by Laue et al.[8] [13] can be used to find both derivatives.

3. (a) Since
$$\begin{aligned} q(\mathbf{r} \mid \mathbf{u}) &= \mathcal{N}(\mathbf{r}; \mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{u}, \mathbf{K}_{\mathbf{r},\mathbf{r}} - \mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{K}_{\mathbf{r},\mathbf{u}}) \\ &= \mathcal{N}(\mathbf{r}; \mathbf{S}\mathbf{u}, \boldsymbol{\Gamma}), \end{aligned}$$
   we have
$$\frac{\partial q(\mathbf{r} \mid \mathbf{u})}{\partial \lambda_i} = -\frac{1}{2} q(\mathbf{r} \mid \mathbf{u}) \frac{\partial}{\partial \lambda_i} [(\mathbf{r} - \mathbf{S}\mathbf{u})^\intercal \boldsymbol{\Gamma}^{-1} (\mathbf{r} - \mathbf{S}\mathbf{u}) + \log |\boldsymbol{\Gamma}|].$$
   The same online tool can be used to show that
$$\frac{\partial}{\partial \lambda_i} \log |\boldsymbol{\Gamma}| = -|\boldsymbol{\Gamma}|^{-1} \operatorname{tr}(\mathbf{R} \operatorname{adj}(\boldsymbol{\Gamma})),$$
   and
$$\frac{\partial}{\partial \lambda_i} \boldsymbol{\Gamma}^{-1} = \boldsymbol{\Gamma}^{-1} \mathbf{R} \boldsymbol{\Gamma}^{-1}.$$
   (b) If $i = 0$, then
$$\frac{\partial \mathbf{K}}{\partial \lambda_i} = \frac{1}{\lambda_i} \mathbf{K}$$
   by the structure of each element of $\mathbf{K}$. If $i \neq 0$, then each element of $\frac{\partial \mathbf{K}}{\partial \lambda_i}$ is
$$\begin{aligned} L_{j,k} &= \frac{\partial k(\mathbf{x}_j, \mathbf{x}_k)}{\partial \lambda_i} \\ &= k(\mathbf{x}_j, \mathbf{x}_k) \frac{\partial}{\partial \lambda_i} \left[-\frac{1}{2}(\mathbf{x}_j - \mathbf{x}_k)^\intercal \boldsymbol{\Lambda} (\mathbf{x}_j - \mathbf{x}_k) - \mathbb{1}[j \neq k]\sigma^2 \operatorname{tr}(\boldsymbol{\Lambda}) \right] \\ &= k(\mathbf{x}_j, \mathbf{x}_k) \frac{\partial}{\partial \lambda_i} \left[-\frac{1}{2} \sum_{l=1}^{d} \lambda_l (x_{j,l} - x_{k,l})^2 - \mathbb{1}[j \neq k]\sigma^2 \sum_{l=1}^{d} \lambda_l \right] \\ &= k(\mathbf{x}_j, \mathbf{x}_k) \left(-\frac{1}{2}(x_{j,i} - x_{k,i})^2 - \mathbb{1}[j \neq k]\sigma^2 \right). \end{aligned}$$

□

[8] http://www.matrixcalculus.org/

B. DERIVATIVES OF THE ELBO

B.1 $\partial / \partial \boldsymbol{\mu}$

We begin by removing terms independent of $\boldsymbol{\mu}$:
$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}} = \frac{\partial}{\partial \boldsymbol{\mu}} [\mathbf{t}^\intercal \mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} \boldsymbol{\mu}] - \frac{1}{2} \frac{\partial}{\partial \boldsymbol{\mu}} \left[\boldsymbol{\mu}^\intercal \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} \boldsymbol{\mu} \right] - \frac{\partial}{\partial \boldsymbol{\mu}} \mathbb{E}[v].$$
Here
$$\frac{\partial}{\partial \boldsymbol{\mu}} \left[\boldsymbol{\mu}^\intercal \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} \boldsymbol{\mu} \right] = (\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} + \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-\intercal}) \boldsymbol{\mu}$$
by Petersen and Pedersen [20], and
$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{\mu}} \mathbb{E}[V_{\mathbf{r}}(s)] &= \frac{\partial}{\partial \boldsymbol{\mu}} \iint V_{\mathbf{r}}(s) q(\mathbf{r} \mid \mathbf{u}) q(\mathbf{u}) \, \mathrm{d}\mathbf{r} \, \mathrm{d}\mathbf{u} \\ &= \iint V_{\mathbf{r}}(s) q(\mathbf{r} \mid \mathbf{u}) \frac{\partial q(\mathbf{u})}{\partial \boldsymbol{\mu}} \, \mathrm{d}\mathbf{r} \, \mathrm{d}\mathbf{u} \\ &= \frac{1}{2} \mathbb{E}[V_{\mathbf{r}}(s)(\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}^{-\intercal})(\mathbf{u} - \boldsymbol{\mu})] \end{aligned}$$
by Theorem 5.13 and Lemma 5.4. Hence,
$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}} &= \mathbf{t}^\intercal \mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} - \frac{1}{2}(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} + \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-\intercal}) \boldsymbol{\mu} \\ &\quad - \frac{1}{2} \mathbb{E}\left[(\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}^{-\intercal})(\mathbf{u} - \boldsymbol{\mu}) v \right]. \end{aligned}$$

B.2 $\partial / \partial \mathbf{B}$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{B}} = \frac{1}{2} \left(\frac{\partial}{\partial \mathbf{B}} \log |\boldsymbol{\Sigma}| - \frac{\partial}{\partial \mathbf{B}} \operatorname{tr}\left(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} \boldsymbol{\Sigma} \right) \right) - \frac{\partial}{\partial \mathbf{B}} \mathbb{E}[v].$$
By Theorem 5.13,
$$\frac{\partial}{\partial \mathbf{B}} \mathbb{E}[V_{\mathbf{r}}(s)] = \iint V_{\mathbf{r}}(s) q(\mathbf{r} \mid \mathbf{u}) \frac{\partial q(\mathbf{u})}{\partial \mathbf{B}} \, \mathrm{d}\mathbf{r} \, \mathrm{d}\mathbf{u}.$$
Then, using the aforementioned tool by Laue et al. [13], we get
$$\frac{\partial}{\partial \mathbf{B}} \log |\boldsymbol{\Sigma}| = 2\boldsymbol{\Sigma}^{-1}\mathbf{B}, \quad \frac{\partial}{\partial \mathbf{B}} \operatorname{tr}\left(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} \boldsymbol{\Sigma} \right) = 2\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{B},$$
and Lemma 5.4 gives
$$\frac{\partial q(\mathbf{u})}{\partial \mathbf{B}} = q(\mathbf{u})(\boldsymbol{\Sigma}^{-1}\mathbf{U}\boldsymbol{\Sigma}^{-1} - |\boldsymbol{\Sigma}|^{-1} \operatorname{adj}(\boldsymbol{\Sigma}))\mathbf{B}.$$

Therefore,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{B}} = \left(\boldsymbol{\Sigma}^{-1} - \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\right)\mathbf{B} - \mathbb{E}[(\boldsymbol{\Sigma}^{-1}\mathbf{U}\boldsymbol{\Sigma}^{-1} - |\boldsymbol{\Sigma}|^{-1}\operatorname{adj}(\boldsymbol{\Sigma}))\mathbf{B}v].$$

B.3 $\partial/\partial\lambda_j$

For $j = 0, \ldots, d$,

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \lambda_j} &= \mathbf{t}^\intercal \frac{\partial}{\partial \lambda_j}\left[\mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\right]\boldsymbol{\mu} - \frac{\partial}{\partial \lambda_j}\mathbb{E}[v] \\ &\quad - \frac{1}{2}\left(\frac{\partial}{\partial \lambda_j}\operatorname{tr}\left(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\boldsymbol{\Sigma}\right) + \boldsymbol{\mu}^\intercal \frac{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}}{\partial \lambda_j}\boldsymbol{\mu} + \frac{\partial}{\partial \lambda_j}\log|\mathbf{K}_{\mathbf{u},\mathbf{u}}|\right),\end{aligned}$$

where

$$\begin{aligned}\frac{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}}{\partial \lambda_j} &= -\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\frac{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}}{\partial \lambda_j}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}, \\ \frac{\partial}{\partial \lambda_j}\left[\mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\right] &= \frac{\partial \mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal}{\partial \lambda_j}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} + \mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal \frac{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}}{\partial \lambda_j} \\ &= \left(\frac{\partial \mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal}{\partial \lambda_j} - \mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\frac{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}}{\partial \lambda_j}\right)\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}, \\ \frac{\partial}{\partial \lambda_j}\operatorname{tr}(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\boldsymbol{\Sigma}) &= \operatorname{tr}\left(\frac{\partial}{\partial \lambda_j}\left[\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\boldsymbol{\Sigma}\right]\right) = \operatorname{tr}\left(\frac{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}}{\partial \lambda_j}\boldsymbol{\Sigma}\right) \\ &= -\operatorname{tr}\left(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\frac{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}}{\partial \lambda_j}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\boldsymbol{\Sigma}\right), \\ \frac{\partial}{\partial \lambda_j}\log|\mathbf{K}_{\mathbf{u},\mathbf{u}}| &= \operatorname{tr}\left(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\frac{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}}{\partial \lambda_j}\right)\end{aligned}$$

by Petersen and Pedersen [20], and

$$\begin{aligned}\frac{\partial}{\partial \lambda_j}\mathbb{E}[V_{\mathbf{r}}(s)] &= \iint V_{\mathbf{r}}(s)\frac{\partial q(\mathbf{r} \mid \mathbf{u})}{\partial \lambda_j}q(\mathbf{u})\,\mathrm{d}\mathbf{r}\,\mathrm{d}\mathbf{u} \\ &= \frac{1}{2}\mathbb{E}[V_{\mathbf{r}}(s)(|\boldsymbol{\Gamma}|^{-1}\operatorname{tr}(\mathbf{R}\operatorname{adj}(\boldsymbol{\Gamma})) \\ &\qquad - (\mathbf{r} - \mathbf{S}\mathbf{u})^\intercal \boldsymbol{\Gamma}^{-1}\mathbf{R}\boldsymbol{\Gamma}^{-1}(\mathbf{r} - \mathbf{S}\mathbf{u}))]\end{aligned}$$

by Theorem 5.13 and Lemma 5.4. Thus,

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \lambda_j} &= \mathbf{t}^\intercal \left(\frac{\partial \mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal}{\partial \lambda_j} - \mathbf{K}_{\mathbf{r},\mathbf{u}}^\intercal \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\frac{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}}{\partial \lambda_j}\right)\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\boldsymbol{\mu} \\ &\quad + \frac{1}{2}\left[\operatorname{tr}\left(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\frac{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}}{\partial \lambda_j}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\boldsymbol{\Sigma}\right) + \boldsymbol{\mu}^\intercal \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\frac{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}}{\partial \lambda_j}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\boldsymbol{\mu}\right. \\ &\qquad \left. - \operatorname{tr}\left(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\frac{\partial \mathbf{K}_{\mathbf{u},\mathbf{u}}}{\partial \lambda_j}\right)\right] \\ &\quad - \frac{1}{2}\mathbb{E}[(|\boldsymbol{\Gamma}|^{-1}\operatorname{tr}(\mathbf{R}\operatorname{adj}(\boldsymbol{\Gamma})) \\ &\qquad - (\mathbf{r} - \mathbf{S}\mathbf{u})^\intercal \boldsymbol{\Gamma}^{-1}\mathbf{R}\boldsymbol{\Gamma}^{-1}(\mathbf{r} - \mathbf{S}\mathbf{u}))v],\end{aligned}$$

where the remaining derivatives can be found in Lemma 5.4.