Variational Methods for Probabilistic Inference and Decision-Making

Tim Georg Johann Rudner

New College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2024

To my parents.

Acknowledgements

I am deeply grateful to my advisors, Yee Whye Teh and Yarin Gal, for their guidance and support, and for giving me the freedom to explore. I feel incredibly fortunate to have had the privilege of learning from them. I also thank my thesis examiners, Chris Holmes and Francisco J. R. Ruiz. Their comments were invaluable, and I greatly enjoyed discussing my research with them.

It was a rare piece of luck to spend several years in a place as magical and vibrant as Oxford. So many people made it special, and I will forever cherish the memories and friendships I made. I am grateful to the Rhodes community for piquing my interest in artificial intelligence, and to my colleagues in the Oxford Applied and Theoretical Machine Learning (OATML) group, the Oxford Computational Statistics and Machine Learning group, and the AIMS Centre for Doctoral Training for many conversations that have broadened my horizons. I would especially like to thank my labmates Aidan Gomez, Andreas Kirsch, Andrew Jesson, Angelos Filos, Clare Lyle, Gunshi Gupta, Jan Brauner, Jannik Kossen, Joost van Amersfoort, Lewis Smith, Lisa Schut, Milad Alizadeh, Mo Razzak, Pascal Notin, Sebastian Farquhar, and Sören Mindermann—I have learned a great deal from them—and my collaborators in and outside of Oxford: Helen Toner, Cong Lu, Neil Band, Leo Klarner, Qixuan Feng, Oscar Key, Vitchyr Pong, and Zonghao Chen. It has been a privilege to work with them, and they have made me a better researcher.

I am thankful to Steve Roberts, who encouraged me to switch fields and pursue a DPhil in computer science; to Michael Osborne, whose kind and uplifting comments on paper drafts brightened my day; to Dino Sejdinović, who supervised my first machine learning research project; and to Sergey Levine at UC Berkeley, Sekhar Tatikonda at Yale, and Julia Kempe at NYU for their research advice, guidance, and mentorship. I am especially thankful to Wendy Poole, who was a continual source of support and encouragement and who made the impossible possible, over and over again, and to Benjamin Garfinkel for being a wonderful friend.

I am thankful for my friends, my family, my mentors, and everyone who has contributed to my path.

I am deeply grateful to my mother, Ursula Rudner, for her endless love and support. She taught me how to be persistent, to dare, and to believe in myself. Her kindness inspires me. She is my role model.

Finally, I am thankful to my wife, Elizabeth Zhang, for her love, for being my life partner, and for making me a better person every day.

Abstract

Machine learning models have achieved significant success across a wide range of tasks, but their reliability in safety-critical applications remains a concern, particularly when faced with unfamiliar data. This thesis focuses on improving the reliability and uncertainty quantification of neural networks using probabilistic inference. The first contribution of this thesis addresses shortcomings in parameter-space variational inference, which fails to provide reliable uncertainty estimates. We propose a function-space variational inference method that approximates distributions over functions instead of parameters and improves uncertainty quantification under distribution shifts and significantly reduces catastrophic forgetting in continual learning. The second contribution tackles the challenge of specifying meaningful priors for neural networks. We introduce a method for constructing data-driven priors that incorporate domain knowledge, significantly improving uncertainty quantification in computer vision and language modeling tasks and increasing group robustness under subpopulation shifts. The third contribution is a variational formulation of reinforcement learning in infinite-horizon Markov decision processes. This framework allows learning reward functions and dynamic discount factors directly from environmental interactions, offers a probabilistic justification for KL-regularized reinforcement learning, and improves the performance of RL agents in goal-conditioned sequential decision-making tasks. Overall, this thesis presents probabilistically principled, practical methods for making neural networks more reliable and more suitable for deployment in high-stakes, safety-critical environments.

Contents

List of Figures

1 Introduction

1.1 Probabilistic Inference for Reliable Decision-Making

Machine learning models succeed at an increasingly wide range of tasks (Krizhevsky et al., 2012; Mnih et al., 2013; Silver et al., 2016; Radford et al., 2019; Jumper et al., 2021) but may fail without warning when applied to inputs that are meaningfully different from the training data (Amodei et al., 2016; Hendrycks et al., 2021; Rudner et al., 2021b; Rudner et al., 2021c). To deploy machine learning models in safety-critical environments where failures are costly or may cause severe harm, machine learning methods must be reliable and have the ability to 'fail gracefully.'

In this thesis, we approach the goal of ensuring reliability in neural networks from a probabilistic perspective. My work has two main themes. The first theme is that using probabilistic methods for training neural networks allows us to meaningfully represent uncertainty about the modeling process, which can help prevent overly confident yet incorrect predictions. The second theme is that probabilistically principled methods for incorporating prior beliefs about a given application domain can help make models more reliable and enable them to better adapt to previously unseen settings.

Bayesian inference provides a probabilistically well-founded approach to updating beliefs based on new information while also taking into account prior beliefs. Bayesian methods have long promised to make modern neural networks more reliable by improving their uncertainty estimates and improving predictive accuracy via Bayesian model averaging. However, they have not consistently delivered on this promise. We revisit the reasons for these shortcomings and introduce three probabilistic methods to address them.

First, we examine why variational inference has struggled to provide high-quality uncertainty quantification in neural networks. Instead of attributing the issue to the restrictive nature of Gaussian mean-field variational inference, we investigate whether the real issue lies in parameter-space variational inference with uninformative priors, which fails to find an approximation to the posterior that induces a distribution over functions close to the distribution over functions induced by the (exact) posterior. To address this, we derive a function-space variational objective and show that explicitly approximating the distribution over functions significantly improves uncertainty quantification and helps prevent catastrophic forgetting in continual learning—two domains where Bayesian methods have previously fallen short.

Second, we tackle the challenge of specifying meaningful priors for neural networks. While simple priors have been studied, incorporating useful domain knowledge into the training process has remained largely unattainable. Given the importance of priors in Bayesian inference, this gap has limited the appeal of probabilistic methods. Building on my work in function-space variational inference, we propose a general method for defining meaningful, data-driven priors over network parameters. We present two examples of such priors: one for improving uncertainty quantification, and another for ensuring robustness to subpopulation shifts. Incorporating these priors into existing variational inference methods, such as Gaussian mean-field variational inference, leads to significant performance improvements in tasks like semantic shift detection, selective prediction, and robustness under subpopulation shifts.

Third, we reformulate reinforcement learning in infinite-horizon Markov decision processes as a variational inference problem and use this framework to derive temporal-difference algorithms for both behavior-driven and outcome-driven reinforcement learning. This probabilistic approach not only justifies KL-regularized reinforcement learning but also extends it by learning the reward function from interactions with the environment and introducing a time-dependent, variationally learned discount factor. Empirical results show that an actor-critic method based on this framework performs well in practice, eliminating the need to specify reward functions in settings where the desired outcome is clear.

1.2 Summary

This thesis addresses the problem of how to use probabilistic methods to make machine learning models more trustworthy and reliable. Specifically, it proposes a tractable approach for performing function-space variational inference in stochastic neural networks, develops a general method for specifying interpretable, data-driven priors for neural networks and demonstrates how to train neural networks using such priors, and formulates a probabilistically-motivated framework for solving infinite-horizon Markov decision processes without access to a pre-specified reward function.

In Chapter 2, we introduce Bayesian inference and explain how to approximate a posterior distribution using variational methods. We define stochastic neural networks as parametric models with stochastic parameters and describe Gaussian mean-field variational inference. Finally, we introduce several uncertainty-aware evaluation metrics relevant to safety-critical real-world applications.

In Chapter 3, we propose a novel approach to approximate inference in Bayesian neural networks by framing posterior inference as a problem of finding a posterior over induced stochastic functions. We consider a function-space variational objective, show for which classes of prior distributions over functions this optimization objective is well-defined, and devise a simple and computationally scalable estimator for computing the Kullback-Leibler divergence between a variational induced distribution over functions and a prior induced distribution over functions. We demonstrate that function-space variational inference facilitates the incorporation of prior beliefs into neural network training and enhances the reliability and uncertainty quantification of neural networks.

Based on:

1. N. Band*, **T. G. J. Rudner***, Q. Feng, A. Filos, Z. Nado, M. Dusenberry, G. Jerfel, D. Tran, Y. Gal. *Benchmarking Bayesian Deep Learning on Diabetic Retinopathy Detection Tasks*. Conference on Neural Information Processing Systems (**NeurIPS**), 2021.
2. **T. G. J. Rudner**, F. Bickford Smith, Q. Feng, Y. W. Teh, Y. Gal. *Continual Learning via Sequential Function-Space Variational Inference*. International Conference on Machine Learning (**ICML**), 2022.
3. **T. G. J. Rudner**, Z. Chen, Y. W. Teh, Y. Gal. *Tractable Function-Space Variational Inference in Bayesian Neural Networks*. Conference on Neural Information Processing Systems (**NeurIPS**), 2022.

In Chapter 4, we tackle the longstanding challenge of specifying meaningful and informative prior distributions over neural network parameters. Starting with the question of whether it is possible to specify prior distributions over neural networks conditioned on useful domain-specific information, we derive a family of data-driven

priors and show how to perform tractable variational inference with such priors. Building on this approach, we demonstrate how data-driven priors improve the ability of neural networks to quantify predictive uncertainty and ensure fairness in predictions.

Based on:

1. **T. G. J. Rudner**, S. Kapoor, S. Qiu, A. G. Wilson. *Function-Space Regularization in Neural Networks: A Probabilistic Perspective*. International Conference on Machine Learning (**ICML**), 2023.
2. **T. G. J. Rudner**, Y. Zhang†, A. G. Wilson, J. Kempe. *Mind the GAP: Improving Robustness to Subpopulation Shifts with Group-Aware Priors*. International Conference on Artificial Intelligence and Statistics (**AISTATS**), 2024.
3. **T. G. J. Rudner**, X. Pan†, Y. L. Li, R. Shwartz-Ziv, A. G. Wilson. *Uncertainty-Aware Priors for Fine-Tuning Pre-trained Vision and Language Models*. International Conference on Artificial Intelligence and Statistics (**AISTATS**), 2025.

In Chapter 5, we approach sequential decision-making from a probabilistic perspective and show that state-of-the-art reinforcement learning methods can be derived from, and are special cases of, performing variational Bayesian inference in a transdimensional probabilistic model over state–action trajectories. We prove corresponding policy iteration theorems for behavior- and outcome-driven reinforcement learning problems and demonstrate empirically that framing reinforcement learning as Bayesian variational inference enables improved performance across challenging goal-conditioned sequential decision-making tasks.

Based on:

1. **T. G. J. Rudner***, V. H. Pong*, R. McAllister, Y. Gal, S. Levine. *Outcome-Driven Reinforcement Learning via Variational Inference*. Conference on Neural Information Processing Systems (**NeurIPS**), 2021.
2. **T. G. J. Rudner**. *A Variational Formulation of Reinforcement Learning in Infinite-Horizon Markov Decision Processes*. Workshop on Foundations of Reinforcement Learning & Control. International Conference on Machine Learning (**ICML Workshop**), 2024.

2

Preliminaries

In this chapter, we introduce relevant concepts in Bayesian inference, describe variational inference as an approach to approximate inference in settings where exact Bayesian inference is analytically or computationally intractable, and explain how variational inference can be used to learn an approximate posterior distribution over the parameters of a stochastic neural network. We also introduce uncertainty-aware model evaluation metrics that will be used in later chapters to assess the reliability of neural network uncertainty estimates for computer vision and language classification tasks.

2.1 Notation

In this thesis, we will consider supervised learning tasks on data $\mathcal{D} \stackrel{\text{def}}{=} \{(x_n, y_n)\}_{n=1}^N$ with inputs $x_n \in \mathcal{X} \subseteq \mathbb{R}^D$ and targets $y_n \in \mathcal{Y}$, where $\mathcal{Y} \subseteq \mathbb{R}^Q$ for regression and $\mathcal{Y} \subseteq \{0,1\}^Q$ for classification tasks. To denote the sets of training inputs and targets, we will use the shorthands $x_\mathcal{D}$ and $y_\mathcal{D}$, respectively. Let $f(x;\theta)$ be a mapping parameterized by stochastic parameters $\Theta \in \mathbb{R}^P$. We define a conditional distribution of targets given function values $f(x;\theta)$: $p_{Y|X,\Theta}(y \,|\, x, \theta; f)$. For regression, the conditional distribution $p_{Y|X,\Theta}(y \,|\, x, \theta; f)$ is typically chosen to be a Gaussian distribution with mean $f(x;\theta)$ and a learned or fixed variance, and for classification tasks, $p_{Y|X,\Theta}(y \,|\, x, \theta; f)$ is typically chosen to be a categorical distribution defined via a softmax model (Bridle, 1990; Murphy, 2013).

2.2 Bayesian Inference

Bayesian inference provides a probabilistic framework for updating beliefs about a stochastic quantity of interest, given some prior beliefs, an observation model, and some observed data (Jaynes, 2003).

More specifically, for the supervised learning settings specified above, given observed data $\mathcal{D}$, observation model $p_{Y|X,\Theta}(y \mid x, \theta; f)$, and a prior distribution over parameters, $p_\Theta(\theta)$, Bayes' Theorem provides a mathematical formalism for expressing the conditional distribution over stochastic model parameters Θ given the observed data. This conditional distribution is called the posterior distribution over Θ and is given by

$$p_{\Theta|\mathcal{D}}(\theta \mid \mathcal{D}; f) = \frac{p_{Y|X,\Theta}(y_\mathcal{D} \mid x_\mathcal{D}, \theta; f) p_\Theta(\theta)}{p_{Y|X}(y_\mathcal{D} \mid x_\mathcal{D}; f)}, \tag{2.1}$$

where $p_{Y|X}(y_\mathcal{D} \mid x_\mathcal{D}; f)$ is called the marginal distribution (of the observed targets given the inputs),

$$p_{Y|X}(y_\mathcal{D} \mid x_\mathcal{D}; f) = \int_{\mathbb{R}^P} p(y_\mathcal{D} \mid x_\mathcal{D}, \theta; f) p_\Theta(\theta) \, \mathrm{d}\theta. \tag{2.2}$$

2.2.1 Prior and Posterior Predictive Distributions

Given an observation model, a prior distribution, and a posterior distribution, we can express the prior predictive and posterior predictive distributions—the distributions over targets Y given an input x induced by the prior and posterior distributions, respectively—as

$$p_{Y|X}(y \mid x; f) = \int_{\mathbb{R}^P} p_{Y|X,\Theta}(y \mid x, \theta; f) p_\Theta(\theta) \, \mathrm{d}\theta \tag{2.3}$$

and

$$p_{Y|X}(y \mid x, \mathcal{D}; f) = \int_{\mathbb{R}^P} p_{Y|X,\Theta}(y \mid x, \theta; f) p_{\Theta|\mathcal{D}}(\theta \mid \mathcal{D}) \, \mathrm{d}\theta. \tag{2.4}$$

The prior predictive and posterior predictive distributions can be viewed as incorporating all possible parameter configurations that have non-zero probability density under the prior and posterior distributions over model parameters, respectively. Going forward, we will drop the subscripts from probability density functions unless they are needed for clarity.

2.2.2 Variational Bayes

For some observation models, performing exact inference to find the posterior distribution over the network parameters is analytically intractable. Variational inference is a variational approach for finding an approximate posterior distribution over a given stochastic quantity of interest, such as the model parameters Θ (Wainwright et al., 2008; Blei et al., 2017).

In variational Bayes, we express the posterior inference problem of computing $p_{\Theta|\mathcal{D}}(\theta \mid \mathcal{D}; f)$ variationally as

$$\min_{q_\Theta \in \mathcal{Q}_\Theta} \mathbb{D}_{\mathrm{KL}}(q_\Theta \parallel p_{\Theta|\mathcal{D}}), \tag{2.5}$$

where q_Θ is a variational distribution in a variational family $\mathcal{Q}_\Theta$. $\mathcal{Q}_\Theta$ is typically chosen to be a family of distributions that results in a tractable variational optimization procedure.

To solve the variational problem in Equation (2.5), we can derive a mathematically equivalent optimization problem. To do so, we note that we can write

$$\begin{aligned}
&\mathbb{D}_{\mathrm{KL}}(q_\Theta \parallel p_{\Theta|\mathcal{D}}) \\
&= \int_{\mathbb{R}^P} q(\theta) \log \frac{q(\theta)}{p(\theta \mid \mathcal{D})} \,\mathrm{d}\theta && (2.6)\\
&= \int_{\mathbb{R}^P} q(\theta) \log \frac{q(\theta) p(y_\mathcal{D} \mid x_\mathcal{D}; f)}{p(y_\mathcal{D} \mid x_\mathcal{D}, \theta; f) p(\theta)} \,\mathrm{d}\theta && (2.7)\\
&= \int_{\mathbb{R}^P} q(\theta) \log \frac{q(\theta)}{p(y_\mathcal{D} \mid x_\mathcal{D}, \theta; f) p(\theta)} \,\mathrm{d}\theta + \int_{\mathbb{R}^P} q(\theta) \log p(y_\mathcal{D} \mid x_\mathcal{D}; f) \,\mathrm{d}\theta && (2.8)\\
&= \int_{\mathbb{R}^P} q(\theta) \log \frac{q(\theta)}{p(\theta)} \,\mathrm{d}\theta - \int_{\mathbb{R}^P} q(\theta) \log p(y_\mathcal{D} \mid x_\mathcal{D}, \theta; f) \,\mathrm{d}\theta + \log p(y_\mathcal{D} \mid x_\mathcal{D}; f) && (2.9)\\
&= \mathbb{D}_{\mathrm{KL}}(q_\Theta \parallel p_\Theta) - \mathbb{E}_{q_\Theta}\left[\log p(y_\mathcal{D} \mid x_\mathcal{D}, \Theta; f)\right] + \log p(y_\mathcal{D} \mid x_\mathcal{D}; f), && (2.10)
\end{aligned}$$

where we have used Bayes' Theorem as expressed in Equation (2.1) and the fact that the marginal likelihood $p(y_\mathcal{D} \mid x_\mathcal{D}; f)$ is independent of θ, so that

$$\int_{\mathbb{R}^P} q(\theta) \log p(y_\mathcal{D} \mid x_\mathcal{D}; f) \,\mathrm{d}\theta = \underbrace{\left(\int_{\mathbb{R}} q(\theta) \,\mathrm{d}\theta\right)}_{=1} \log p(y_\mathcal{D} \mid x_\mathcal{D}; f) = \log p(y_\mathcal{D} \mid x_\mathcal{D}; f). \tag{2.11}$$

Rearranging and using the fact that the KL divergence between two distributions is guaranteed to be non-negative, we get

$$\begin{aligned}
&\log p(y_\mathcal{D} \mid x_\mathcal{D}; f) \\
&= \mathbb{E}_{q_\Theta}\left[\log p(y_\mathcal{D} \mid x_\mathcal{D}, \Theta; f)\right] - \mathbb{D}_{\mathrm{KL}}(q_\Theta \parallel p_\Theta) + \underbrace{\mathbb{D}_{\mathrm{KL}}(q_\Theta \parallel p_{\Theta|\mathcal{D}})}_{\geq 0} && (2.12)\\
&\geq \mathbb{E}_{q_\Theta}\left[\log p(y_\mathcal{D} \mid x_\mathcal{D}, \Theta; f)\right] - \mathbb{D}_{\mathrm{KL}}(q_\Theta \parallel p_\Theta) && (2.13)\\
&\stackrel{\text{def}}{=} \mathcal{F}(q_\Theta) && (2.14)
\end{aligned}$$

which implies that

$$\arg\max_{q_\Theta \in \mathcal{Q}_\Theta} \mathcal{F}(q_\Theta) = \arg\min_{q_\Theta \in \mathcal{Q}_\Theta} \mathbb{D}_{\mathrm{KL}}(q_\Theta \parallel p_{\Theta|\mathcal{D}}). \tag{2.15}$$

That is, minimizing $\mathbb{D}_{\mathrm{KL}}(q_\Theta \parallel p_{\Theta|\mathcal{D}})$ with respect to q_Θ is mathematically equivalent to maximizing the variational lower bound $\mathcal{F}(q_\Theta)$ with respect to q_Θ. The variational objective in Equation (2.14) is commonly referred to as the evidence lower bound (ELBO).

Unlike the variational objective in Equation (2.5), the variational lower bound in Equation (2.14) is defined in terms of known quantities, that is, the prior, the observation model, and the observed data, and it does not require access to the posterior distribution. However, Equation (2.14) may still not be analytically or computationally tractable in practice. To enable tractability, a common choice for the variational family $\mathcal{Q}_\Theta$ is the family of mean-field distributions. Under a mean-field assumption, the joint distribution over the stochastic quantity of interest is assumed to fully factorize, that is

$$q(\theta) = \prod_{i=1}^{P} q(\theta_i). \tag{2.16}$$

Under a suitable choice of prior, a mean-field assumption can make the expected log-likelihood, $\mathbb{E}_{q_\Theta}\left[\log p(y_\mathcal{D} \mid x_\mathcal{D}, \Theta; f)\right]$ and the KL divergence, $\mathbb{D}_{\mathrm{KL}}(q_\Theta \parallel p_\Theta)$, easier to estimate or even make them analytically tractable.

2.2.3 Variational Inference in Stochastic Neural Networks

We can apply the Bayesian inference framework to neural networks by defining the mapping $f(\cdot; \Theta)$ to be a neural network parameterized by stochastic parameters Θ. Since neural networks are non-linear in their parameters, exact inference is analytically intractable (MacKay, 1992; Neal, 1996). However, we can use variational inference for finding an approximate posterior distribution over the network parameters (Hinton et al., 1993; Graves, 2011; Wainwright et al., 2008). A common choice for $\mathcal{Q}_\Theta$ is the variational family of Gaussian mean-field distributions (Blundell et al., 2015), that is,

$$q(\theta) = \prod_{i=1}^{P} \mathcal{N}(\theta_i; \mu_i, \Sigma_i) \tag{2.17}$$

for variational mean parameters $\{\mu_i\}_{i=1}^P$ and variational variance parameters $\{\Sigma_i\}_{i=1}^P$. Under an isotropic Gaussian prior,

$$p(\theta) = \mathcal{N}(\theta \mid \mathbf{0}, \mathbf{I}), \tag{2.18}$$

this approximation results in a tractable and scalable variational objective,

$$\mathcal{F}(q_\Theta) \stackrel{\text{def}}{=} \mathbb{E}_{q_\Theta}[\log p(y_\mathcal{D} \mid \Theta, x_\mathcal{D}; f)] - \mathbb{D}_{\text{KL}}(q_\Theta \,\|\, p_\Theta), \tag{2.19}$$

where $\mathbb{D}_{\text{KL}}(q_\Theta \,\|\, p_\Theta)$ can be computed analytically, the expected log-likelihood can be estimated using simple Monte Carlo estimation, and the objective can be used to scale variational inference to large datasets using stochastic variational inference (Hoffman et al., 2013).

2.3 Uncertainty-Aware Model Evaluation

In this section, we introduce three approaches to assessing the reliability of uncertainty estimates generated by predictive models.

2.3.1 Selective Prediction

Selective prediction modifies the standard prediction pipeline by introducing a rejection class (El-Yaniv et al., 2010). In selective prediction pipelines, input points for which a model abstains from making a prediction can be deferred to a human expert for review, enabling collaborative human-machine decision-making. This way, selective prediction can provide a responsible approach to leveraging machine learning systems in safety-critical domains while maintaining human oversight. Given a model f whose output $[f(x)]_k$ is the predicted probability of class k, the classifier predicts $\hat{y}(x) = \arg\max_{k \in \mathcal{Y}} [f(x)]_k$, the selective prediction model introduces a selection function s which determines whether a prediction should be made. This selection function can be based on the outputs of f, such as $s(x) = \max_{k \in \mathcal{Y}} f(x|k)$ The selective prediction model f_s is then defined as

$$f_s(x; \tau) = \begin{cases} f(x) & \text{if } s(x) \geq \tau \\ \perp & \text{otherwise} \end{cases} \tag{2.20}$$

where τ represents the rejection threshold and $\perp$ denotes that the model abstains from making a prediction.

Predictive uncertainty is a natural choice for a selection function: If a model's predictive uncertainty is above a certain threshold, the selection function will decline to make a prediction. Using predictive uncertainty as a selection function, selective prediction allows us to assess both a model's predictive accuracy and the quality of its uncertainty estimates. If a model is successful at rejecting data points for which it would have made an incorrect prediction based on its level of predictive uncertainty, the accuracy of the remaining, non-rejected points will increase as more and more points are rejected. To evaluate the selective prediction model, we compute its predictive accuracy over a range of thresholds τ and compute the area under the selective prediction accuracy curve. Successful selective prediction models are able to obtain high accuracy across many thresholds.

2.3.2 Calibration

Calibration expresses how closely the confidence of a model's predictions is aligned with its accuracy, and well-calibrated models allow users to reliably estimate the accuracy of a model's prediction from its confidence. This is especially important in safety-critical applications, where it is crucial to identify inaccurate predictions. Well-calibrated models are also more trustworthy, as they provide users with a clearer understanding of when to rely on the model.

One notion of miscalibration is the Expected Calibration Error (ECE; Naeini et al., 2015), which computes the gap between model accuracy and confidence. The ECE estimator is defined as

$$\text{ECE} = \sum_{m=1}^{M'} \frac{|B_m|}{n} |\text{Accuracy}(B_m) - \text{Confidence}(B_m)|, \tag{2.21}$$

where n is the number of samples, M' is the number of bins, $\text{Accuracy}(B_m)$ is the accuracy of samples within bin B_m, and $\text{Confidence}(B_m)$ is the average maximum probability outputs of the classifier of all samples within the bin. A perfectly calibrated model has an ECE of zero since its accuracy is perfectly aligned with its confidence, and a more miscalibrated model has a higher ECE.

2.3.3 Semantic Shift Detection

In real-world prediction tasks, models may be tested on data that is meaningfully different from the training data. *Semantic shift* is a type of distribution shift where the test data contains labels that are semantically different from any labels seen during training, that is, $p_{\text{train}}(x, y) \neq p_{\text{test}}(x, y)$. This poses a significant challenge to model performance, as a model will, by definition, not be able to output the correct prediction for semantically different inputs. It is, therefore, important to detect instances of semantic shift so that users can trust a model's prediction.

Uncertainty estimates can be used to identify semantic shifts in the data, and reliable models should have high uncertainty for input points whose true labels are semantically different from the training labels. To evaluate the model's performance on detecting semantic shift, we design test sets that consist of a mixture of in-distribution and semantically shifted test samples. We then construct a binary classifier that only uses the model's predictive entropy $\mathcal{H}(p(y \mid x))$—which serves as a proxy of the model's uncertainty—to predict whether any given point is semantically different from the training set. Finally, we compute the area under the ROC curve (AUROC) of this classifier to assess whether the model's predictive uncertainty can be used to identify data points that are semantically different from the training data. Models that are able to successfully detect semantic shift will achieve higher AUROC scores.

3

Function-Space Variational Inference In Neural Network Models

3.1 Introduction

In this chapter, we develop a function-space variational inference method for obtaining reliable uncertainty estimates in Bayesian neural networks (BNNs, Neal (1996)). While BNNs have promised to combine the advantages of deep learning and Bayesian inference, existing approaches for approximate inference in BNNs fall short of this promise and have been demonstrated to result in approximate posterior predictive distributions that underperform 'non-Bayesian' methods both in terms of predictive accuracy and uncertainty quantification—making them of limited use in practice (Ovadia et al., 2019; Foong et al., 2019; Farquhar et al., 2020a; Band et al., 2021). A potential reason for this shortcoming is that commonly used parameter-space inference methods make it difficult to meaningfully incorporate information about the data-generating process into inference.

To avoid this limitation, we consider function-space variational inference (FSVI), which involves maximizing a variational objective defined explicitly in terms of distributions over functions induced by distributions over parameters. In contrast to prior works that rely on approximation techniques that prevent such function-space variational objectives from being used with high-dimensional inputs and modern neural networks, we propose a simple estimator of the Kullback-Leibler divergence between distributions over functions that enables us to perform stochastic variational inference. The proposed estimation procedure allows defining priors that explicitly encourage high predictive uncertainty away from the training data as well as priors

that reflect relevant information about the task at hand. Building on this approach, we extend FSVI to the continual learning setting and propose a sequential function-space variational inference (S-FSVI) optimization objective.

We present an extensive empirical evaluation in which we demonstrate that function-space variational inference leads to posterior approximations that exhibit significantly improved predictive uncertainty estimates compared to a wide array of state-of-the-art Bayesian and non-Bayesian methods. Furthermore, we demonstrate that sequential function-space variational inference outperforms existing objective-based continual learning methods—in some cases by a significant margin—on a wide range of task sequences, including single-head split MNIST, multi-head split CIFAR, and multi-head sequential Omniglot and show that sequential function-space variational inference is less reliant on careful selection of datapoints that summarize past tasks than other methods.

3.2 Related Work

There is a growing body of work on function-space approaches to inference in BNNs and continual learning (Benjamin et al., 2019; Sun et al., 2019; Titsias et al., 2020; Burt et al., 2021; Pan et al., 2020; Ma et al., 2021; Rudner et al., 2022b). Below, we review related work in these areas.

3.2.1 Function-Space Inference in Bayesian Neural Networks

Existing methods for FSVI in BNNs are based on approximate gradient estimators and either replace the intractable supremum in a function-space variational objective with an expectation (Sun et al., 2019) or do not define an explicit variational objective at all (Wang et al., 2019). Sun et al. (2019) and Carvalho et al. (2020) used Gaussian process priors over functions for which the function-space variational inference problem is not well-defined (see Section 3.3.2 and Burt et al. (2021)). More recent work has attempted to circumvent the intractability of the variational objective derived by Sun et al. (2019) by proposing alternative objectives for function-space inference in BNNs (Ma et al., 2019; Ober et al., 2020; Ma et al., 2021). Immer et al. (2020) and Khan et al. (2019) showed that an approximate BNN posterior distribution obtained via the Laplace and Generalized Gauss-Newton approximations corresponds to the exact posterior under linearization of a different model. Unlike in our approach, they used a Laplace approximation and did not perform variational inference or optimize the variance parameters. Furthermore, Immer et al. (2020) and Khan et al. (2019) used a neural network model to obtain a parameter maximum a posteriori estimate, but then used a linearization of the neural network model to compute a posterior predictive distribution.

Burt et al. (2021) considered the function-space variational objective considered by Sun et al. (2019) and showed that the KL divergence between BNNs with

different network architectures is not well-defined. A parallel line of research showed that posterior predictive distributions of shallow BNNs with mean-field variational distributions have a limited ability to represent complex covariance structures in function space (Foong et al., 2019; Foong et al., 2020) but that deep BNNs do not suffer from this limitation (Farquhar et al., 2020b).

3.2.2 Continual Learning and Function-Space Inference Approaches

There are three main (partially overlapping) categories of methods for continual learning in a deep neural network. Objective-based approaches modify the objective function used to train the neural network. Replay-based approaches summarize past tasks using either stored data or freshly generated synthetic data. Architecture-based approaches change the neural network's structure from one task to another. For extensive reviews, see De Lange et al. (2021) and Parisi et al. (2019). We focus on objective-based approaches in this review.

For a neural network to retain abilities it has previously learned, its predictions on data associated with past tasks must not change significantly from one task to another. One way of achieving this is to include in the training objective a form of function-space regularization to discourage important changes in the network's predictions or internal representations. "Learning without forgetting" (Li et al., 2018b) uses a modified cross-entropy loss that penalizes the difference between the predictions of the current network on the current task data and the predictions of the previous network on the current task data. "Less-forgetful learning" (Jung et al., 2018) employs the same method but uses squared Euclidean distance rather than the modified cross-entropy loss and applies it to the penultimate-layer representations rather than the network's predictions. "Keep and learn" (Kim et al., 2018) also uses internal representations as a basis for regularization. The method subsequently proposed by Benjamin et al. (2019) involves comparing the current network with all previous versions of the network and on data from all past tasks instead of with only the most recent network on data from the current task. Each pair of networks is compared by computing the Euclidean distance between the networks' predictions. "Dark experience replay" (Buzzega et al., 2020) extends this method to work in a setting where task boundaries are not clearly defined.

While the approaches described above mitigate forgetting, they do not explicitly account for predictive uncertainty, which is an issue if the neural network is a poor fit to the data. This deficiency is addressed by probabilistic approaches to function-space regularization, which encourage a network's predictions to agree with a prior distribution over functions rather than with a single function. "Functional regularization for continual learning" (FRCL; Titsias et al., 2020) considers a network whose final layer is a Bayesian linear model. Based on the duality between parameter space and function space, the FRCL objective includes the KL divergence between predictive distributions at a selection of input points. This encourages similarity

between the network's current predictive distribution and the distributions from past tasks. FRCL is theoretically appealing, building on a well-understood method for stochastic variational inference using inducing points, but is only applicable to Bayesian linear models. In contrast, "functional regularization of the memorable past" (FROMP; Pan et al., 2020) maintains a posterior distribution over all the parameters of a neural network. While FROMP achieves state-of-the-art performance on several continual-learning task sequences, it relies on a change in the underlying probabilistic model and uses a surrogate objective for optimization, which divorces it from function-space variational objectives. As we show, this results in suboptimal performance compared to sequential function-space variational inference, which maintains a stronger link to the underlying Bayesian approximation.

Although our focus is on methods for training deep neural networks, for completeness, we also note methods based on Gaussian processes (GPs). Incremental variational sparse GP regression (Cheng et al., 2016), streaming sparse GPs (Bui et al., 2017) and online sparse multi-output GP regression (Yang et al., 2019) build on the work of Csató et al. (2002) and Csató (2002), and are effective approaches to continual learning for regression tasks. Continual multi-task GPs (Moreno-Muñoz et al., 2019) extend to multi-output settings with non-Gaussian likelihoods. Variational autoregressive GPs (VAR-GP; Kapoor et al., 2021) perform well on task sequences with image inputs but scale poorly with the number of tasks: the time complexity for inference is cubic in the number of context points and hence in the number of tasks, which may limit their applicability to task sequences like sequential Omniglot.

3.3 Bayesian Inference Over Stochastic Functions

We consider supervised learning tasks on data $\mathcal{D} \stackrel{\text{def}}{=} \{(x_n, y_n)\}_{n=1}^N = (x_\mathcal{D}, y_\mathcal{D})$ with inputs $x_n \in \mathcal{X} \subseteq \mathbb{R}^D$ and labels $y_n \in \mathcal{Y}$, where $\mathcal{Y} \subseteq \mathbb{R}^Q$ for regression and $\mathcal{Y} \subseteq \{0, 1\}^Q$ for classification tasks. Bayesian neural networks (BNNs) are stochastic neural networks trained using (approximate) Bayesian inference. Denoting the parameters of such a stochastic neural network by the multivariate random variable $\Theta \in \mathbb{R}^P$ and letting the function mapping defined by a neural network architecture be given by $f : \mathcal{X} \times \mathbb{R}^P \to \mathbb{R}^Q$, we obtain a random function $f(\cdot\,; \Theta)$. For a parameter realization θ, we obtain a corresponding function realization, $f(\cdot\,; \theta)$. When evaluated at a finite collection of points $x = \{x_i\}_{i=1}^m$, $f(x; \Theta)$ is a multivariate random variable and $f(x; \theta)$ is a vector.

Letting $p_{y|f(x;\Theta)}$ be a likelihood function and $p_{y|f(x;\Theta)}(y_\mathcal{D} \mid f(x_\mathcal{D}; \theta))$ be the likelihood of observing the labels $y_\mathcal{D}$ under the stochastic function $f(\cdot\,; \Theta)$ evaluated at inputs $x_\mathcal{D}$ and letting p_Θ be a prior distribution over the stochastic network parameters Θ, we can use approximate inference methods to find an approximation to the posterior distribution, $p_{\Theta|\mathcal{D}}$.

Below, instead of seeking to infer an approximate posterior distribution over the parameters, we frame variational inference in stochastic neural networks as inferring an approximation to the posterior distribution over *functions* $p_{f(\cdot\,;\Theta)|\mathcal{D}}$ induced by the posterior distribution over parameters $p_{\Theta|\mathcal{D}}$. Informally, we can write this induced distribution over functions as

$$p_{f(\cdot\,;\Theta)|\mathcal{D}}(f(\cdot\,;\theta)\,|\,\mathcal{D}) = \int_{\mathbb{R}^P} p_{\Theta|\mathcal{D}}(\theta'\,|\,\mathcal{D})\,\delta(f(\cdot\,;\theta) - f(\cdot\,;\theta'))\,\mathrm{d}\theta', \tag{3.1}$$

where $\delta(\cdot)$ is the Dirac delta function (Wolpert, 1993). Considering the prior distribution over functions $p_{f(\cdot\,;\Theta)}$ induced by a prior distribution over parameters p_Θ,

$$p_{f(\cdot\,;\Theta)}(f(\cdot\,;\theta)) = \int_{\mathbb{R}^P} p_\Theta(\theta')\,\delta(f(\cdot\,;\theta) - f(\cdot\,;\theta'))\,\mathrm{d}\theta', \tag{3.2}$$

and the variational distribution over functions $q_{f(\cdot\,;\Theta)}$ induced by a variational distribution over parameters q_Θ,

$$q_{f(\cdot\,;\Theta)}(f(\cdot\,;\theta)) = \int_{\mathbb{R}^P} q_\Theta(\theta')\,\delta(f(\cdot\,;\theta) - f(\cdot\,;\theta'))\,\mathrm{d}\theta', \tag{3.3}$$

we can express the problem of finding a posterior distribution over functions variationally as

$$\min_{q_\Theta \in \mathcal{Q}_{q_\Theta}} \mathbb{D}_{\mathrm{KL}}(q_{f(\cdot\,;\Theta)} \,\|\, p_{f(\cdot\,;\Theta)|\mathcal{D}}), \tag{3.4}$$

which allows us to effectively incorporate meaningful prior information about the underlying data-generating process into training.

3.3.1 A Function-Space Variational Objective

To derive a function-space variational objective, we follow Matthews et al. (2016). Consider measures $\hat{P}$ and P both of which define distributions over some function f, indexed by an infinite index set x. Let $\mathcal{D}$ be a dataset and let $x_\mathcal{D}$ denote a set of inputs and $y_\mathcal{D}$ a set of labels. Consider the measure-theoretic version of Bayes' Theorem (Schervish, 1995):

$$\frac{d\hat{P}}{dP}(f) = \frac{p_x(y\,|\,f)}{p_x(y)}, \tag{3.5}$$

where $p_x(y\,|\,f)$ is the likelihood and $p_x(y) = \int_{\mathbb{R}^x} p_x(y\,|\,f) dP(f)$ is the marginal likelihood. We assume that the likelihood function is evaluated at a finite subset of the index set x. Denote by $\pi_z : \mathbb{R}^x \to \mathbb{R}^z$ a projection function that takes a function and returns the same function, evaluated at a finite set of points z, so for a dataset $\mathcal{D}$ consisting of $x_\mathcal{D}$ and $y_\mathcal{D}$, we can write

$$\frac{d\hat{P}}{dP}(f) = \frac{d\hat{P}_{x_\mathcal{D}}}{dP_{x_\mathcal{D}}}(\pi_{x_\mathcal{D}}(f)) = \frac{p(y_\mathcal{D}\,|\,\pi_{x_\mathcal{D}}(f))}{p_{x_\mathcal{D}}(y_\mathcal{D})}, \tag{3.6}$$

and the marginal likelihood becomes

$$p_{x_\mathcal{D}}(y_\mathcal{D}) = \int p_{y|f_x}(y_\mathcal{D} \mid \pi_{x_\mathcal{D}}(f))) \, \mathrm{d}P_{x_\mathcal{D}}(\pi_{x_\mathcal{D}}(f)). \tag{3.7}$$

Now, considering the measure-theoretic version of the KL divergence between an approximating stochastic process Q and a posterior stochastic process $\hat{P}$, we can write

$$\mathbb{D}_{\mathrm{KL}}(Q \,\|\, \hat{P}) = \int \log \frac{dQ}{dP}(f) \, \mathrm{d}Q(f) - \int \log \frac{d\hat{P}}{dP}(f) \, \mathrm{d}Q(f), \tag{3.8}$$

where P is some prior stochastic process. Now, we can apply the measure-theoretic Bayes' Theorem to obtain

$$\begin{aligned}
&\mathbb{D}_{\mathrm{KL}}(Q \,\|\, \hat{P}) \\
&\quad = \int \log \frac{dQ}{dP}(f) \, \mathrm{d}Q(f) - \int \log \frac{d\hat{P}}{dP}(f) \, \mathrm{d}Q(f) && (3.9) \\
&\quad = \int \log \frac{dQ^\pi}{dP^\pi}(f) \, \mathrm{d}Q^\pi(f) - \int \log \frac{d\hat{P}_{x_\mathcal{D}}}{dP_{x_\mathcal{D}}}(\pi_{x_\mathcal{D}}(f)) \, \mathrm{d}Q_{x_\mathcal{D}}(\pi_{x_\mathcal{D}}(f)) && (3.10) \\
&\quad = \int \log \frac{dQ^\pi}{dP^\pi}(f) \, \mathrm{d}Q^\pi(f) - \mathbb{E}_{Q_{x_\mathcal{D}}(\pi_{x_\mathcal{D}}(f))} \left[\log p(y_\mathcal{D} \mid \pi_{x_\mathcal{D}}(f)) \right] + \log p_{x_\mathcal{D}}(y_\mathcal{D}), && (3.11)
\end{aligned}$$

where $\frac{dQ^\pi}{dP^\pi}(f)$ is marginally consistent given the projection π. Defining $f_{x_\mathcal{D}} \overset{\text{def}}{=} \pi_{x_\mathcal{D}}(f))$, using the more conventional notation for the log likelihood, $\log p(y_\mathcal{D} \mid x_\mathcal{D})$, and rearranging, we get

$$\begin{aligned}
&\log p(y_\mathcal{D} \mid x_\mathcal{D}) \\
&\quad = \mathbb{E}_{Q_{x_\mathcal{D}}(f_{x_\mathcal{D}})} \left[\log p(y_\mathcal{D} \mid f_{x_\mathcal{D}}) \right] - \int \log \frac{dQ^\pi}{dP^\pi}(f) \, \mathrm{d}Q^\pi(f) + \mathbb{D}_{\mathrm{KL}}(Q^\pi \,\|\, \hat{P}) && (3.12) \\
&\quad \geq \mathbb{E}_{Q_{x_\mathcal{D}}(f_{x_\mathcal{D}})} \left[\log p(y_\mathcal{D} \mid f_{x_\mathcal{D}}) \right] - \int \log \frac{dQ^\pi}{dP^\pi}(f) \, \mathrm{d}Q^\pi(f) && (3.13) \\
&\quad = \mathbb{E}_{Q_{x_\mathcal{D}}(f_{x_\mathcal{D}})} \left[\log p(y_\mathcal{D} \mid f_{x_\mathcal{D}}) \right] - \mathbb{D}_{\mathrm{KL}}(Q^\pi \,\|\, P^\pi), && (3.14)
\end{aligned}$$

by the non-negativity property of the KL divergence. This lower bound can equivalently be expressed as

$$\log p(y_\mathcal{D} \mid x_\mathcal{D}) \geq \mathbb{E}_{Q_{x_\mathcal{D}}(f_{x_\mathcal{D}})} \left[\log p(y_\mathcal{D} \mid f_{x_\mathcal{D}}) \right] - \mathbb{D}_{\mathrm{KL}}(Q_{x_\mathcal{D}, x_{\setminus\mathcal{D}}} \,\|\, P_{x_\mathcal{D}, x_{\setminus\mathcal{D}}}), \tag{3.15}$$

where $x_{\setminus\mathcal{D}}$ is an infinite index set excluding the finite index set $x_\mathcal{D}$, that is, $x_{\setminus\mathcal{D}} \cap x_\mathcal{D} = \varnothing$, or by Theorem 1 in Sun et al. (2019), we can write

$$\log p(y_\mathcal{D} \mid x_\mathcal{D}) \geq \mathbb{E}_{Q_{x_\mathcal{D}}(f_{x_\mathcal{D}})} \left[\log p(y_\mathcal{D} \mid f_{x_\mathcal{D}}) \right] - \sup_{x \in \mathcal{X}_\mathbb{N}} \mathbb{D}_{\mathrm{KL}}(Q_x \,\|\, P_x), \tag{3.16}$$

where $\mathcal{X}_\mathbb{N} \overset{\text{def}}{=} \bigcup_{n \in \mathbb{N}} \{x \in \mathcal{X}_n \mid \mathcal{X}_n \subseteq \mathbb{R}^{n \times D}\}$ is the collection of all finite sets of evaluation points.

Finally, due to marginal consistency, we can equivalently express the variational lower bound as

$$\log p(y_{\mathcal{D}} \mid x_{\mathcal{D}}) \geq \mathbb{E}_{Q_\Theta}\left[\log p(y_{\mathcal{D}} \mid f(x_{\mathcal{D}}; \Theta))\right] - \sup_{x \in \mathcal{X}_{\mathbb{N}}} \mathbb{D}_{\mathrm{KL}}(Q_x \,\|\, P_x), \tag{3.17}$$

where Q_Θ is a measure on the parameter space $\mathbb{R}^P$ that induces the measure Q.

In the remainder of this chapter, we will denote the distribution over functions induced by a variational distribution over parameters, q_Θ, and a prior distribution over parameters, p_Θ, by $q_{f(\cdot\,;\Theta)}$ and $p_{f(\cdot\,;\Theta)}$, respectively.

3.3.2 Validity of the Function-Space Variational Objective

As discussed by Burt et al. (2021), this variational objective is guaranteed to be well-defined for suitably chosen prior distributions over functions. Specifically, the KL divergence between two distributions over functions generated from different distributions over parameters applied to the same mapping (e.g., the same neural network architecture) is well-defined (i.e., finite) if the KL divergence between the distributions over parameters is finite, since, by the strong data processing inequality (Polyanskiy et al., 2017),

$$\mathbb{D}_{\mathrm{KL}}(q_{f(\cdot\,;\Theta)} \,\|\, p_{f(\cdot;\Theta)}) \leq \mathbb{D}_{\mathrm{KL}}(q_\Theta \,\|\, p_\Theta). \tag{3.18}$$

As a result, if $\mathbb{D}_{\mathrm{KL}}(q_\Theta \,\|\, p_\Theta) < \infty$, which is the case for finite-dimensional parameter vectors Θ and q_Θ absolutely continuous with respect to p_Θ, then the function-space KL divergence is finite and thus well-defined as a variational objective.

Hence, for a likelihood function defined on a finite set of training labels $y_{\mathcal{D}}$ and a suitably defined prior distribution over functions, we can express the variational problem above equivalently as the well-defined maximization problem $\max_{q_\Theta \in \mathcal{Q}_\Theta} \mathcal{F}(q_\Theta)$ with

$$\mathcal{F}(q_\Theta) \stackrel{\text{def}}{=} \mathbb{E}_{q_\Theta}[\log p(y_{\mathcal{D}} \mid f(x_{\mathcal{D}}; \theta))] - \mathbb{D}_{\mathrm{KL}}(q_{f(\cdot;\Theta)} \,\|\, p_{f(\cdot;\Theta)}), \tag{3.19}$$

where $\mathbb{D}_{\mathrm{KL}}(q_{f(\cdot;\Theta)} \,\|\, p_{f(\cdot;\Theta)})$ is also a KL divergence between distributions over functions.

In the next section, we will describe an approximation and estimation procedure that allows scaling function-space variational inference to large neural networks and high-dimensional input data.

3.4 Tractable Function-Space Variational Inference

The primary obstacle to computing the objective in Equation (3.19) is the KL divergence from $q_{f(\cdot;\Theta)}$ to $p_{f(\cdot;\Theta)}$. There are two reasons why the KL divergence in Equation (3.17) is intractable: First, for BNNs or other non-linear models, we do not have access to the probability density functions of the multivariate distributions $q_{f(x;\Theta)}$ and $p_{f(x;\Theta)}$; second, for all but extremely simple input spaces, we are unable to compute the supremum over all possible finite sets of evaluation points. In the remainder of this section, we outline an approach for obtaining an estimator of a locally accurate approximation to the KL divergence that allows for scalable gradient-based optimization of Equation (3.17).

We first approach the problem of computing the KL divergence between two BNNs evaluated at a finite set of points. To do so, we first derive tractable approximations to the distributions over functions $q_{f(x;\Theta)}$ and $p_{f(x;\Theta)}$. Next, we show that under these approximations, we are able to obtain a closed-form approximation to the KL divergence and describe a simple Monte Carlo estimator of the supremum in the function-space KL divergence.

3.4.1 Approximating Distributions Over Functions Via Local Linearization

To obtain an approximation to the probability distributions of $q_{f(x;\Theta)}$ and $p_{f(x;\Theta)}$, we use a first-order Taylor expansion of the *mapping* f about the mean parameters of q_Θ and p_Θ, respectively, and derive the induced distributions under the linearized mapping.

For a stochastic function $f(\cdot\,;\Theta)$ defined in terms of stochastic parameters Θ distributed according to distribution g_Θ with $m \stackrel{\text{def}}{=} \mathbb{E}_{g_\Theta}[\Theta]$ and $S \stackrel{\text{def}}{=} \mathrm{Cov}_{g_\Theta}[\Theta]$, we denote the linearization of the stochastic function $f(\cdot\,;\Theta)$ about m by

$$f(\cdot\,;\Theta) \approx \tilde{f}(\cdot\,;m,\Theta) \stackrel{\text{def}}{=} f(\cdot\,;m) + \mathcal{J}(\cdot\,;m)(\Theta - m), \tag{3.20}$$

where $\mathcal{J}(\cdot\,;m) \stackrel{\text{def}}{=} (\partial f(\cdot\,;\Theta)/\partial\Theta)|_{\Theta=m}$ is the Jacobian of $f(\cdot\,;\Theta)$ evaluated at $\Theta = m$, and the mean and covariance of the distribution over the linearized mapping $\tilde{f}$ at $x, x' \in \mathcal{X}$ are given by

$$\mathbb{E}[\tilde{f}(x;\Theta)] = f(x;m) \tag{3.21}$$

$$\mathrm{Cov}[\tilde{f}(x;\Theta), \tilde{f}(x';\Theta)] = \mathcal{J}(x;m) S \mathcal{J}(x';m)^\top. \tag{3.22}$$

For a derivation of this result, see Appendix A.1. Since Gaussianity is preserved under affine transformations, if g_Θ is a multivariate Gaussian distribution with mean m and diagonal covariance S, then the distribution $\tilde{g}$ over $\tilde{f}(x\,;\Theta)$ is given by

$$\tilde{g}_{\tilde{f}(x;m,\Theta)} = \mathcal{N}(f(x;m), \mathcal{J}(x;m) S \mathcal{J}(x;m)^\top). \tag{3.23}$$

For stochastic functions parameterized by many millions of parameters, obtaining the covariance of $\tilde{g}_{\tilde{f}(x;\Theta)}$—which requires computing an inner product of two Jacobian matrices—can be computationally expensive. Instead of computing the distribution over the linearized mapping exactly, we can construct a suitable Monte Carlo estimator. To do so, we consider a partition of the set of parameters into sets α and β (with $|\beta| \ll |\alpha|$) and note that the linearized mapping can then be expressed as

$$\tilde{f}(\cdot\,; m, \Theta) = f(\cdot\,; m) + \tilde{f}_\alpha(\cdot\,; m, \Theta_\alpha) + \mathcal{J}_\beta(\cdot\,; m)(\Theta_\beta - m_\beta), \tag{3.24}$$

with

$$\tilde{f}_\alpha(\cdot\,; m, \Theta_\alpha) \stackrel{\text{def}}{=} \mathcal{J}_\alpha(\cdot\,; m)(\Theta_\alpha - m_\alpha), \tag{3.25}$$

where $\mathcal{J}_\alpha(\cdot\,; m)$ and $\mathcal{J}_\beta(\cdot\,; m)$ are the columns of the Jacobian matrix corresponding to the sets of parameters α and β, respectively, and Θ_α and Θ_β are the corresponding random parameter vectors. Noting that Equation (3.24) expresses $\tilde{f}$ as a sum of (affine transformations of) random variables, we can use the fact that for independent Gaussian random variables X and Y, the distribution h_Z of $Z = X + Y$ is equal to the convolution of the distributions h_X and h_Y to obtain an approximation to $\tilde{f}$. In particular, we can show that if g_Θ is a multivariate Gaussian distribution with $\Theta_\alpha \perp \Theta_\beta$, the distribution $\tilde{g}_{\tilde{f}(x;\Theta)}$ can be approximated by the Monte Carlo estimator

$$\hat{\tilde{g}}_{\tilde{f}(x;m,\Theta)} = \frac{1}{R}\sum_{j=1}^{R} \mathcal{N}\Big(f(x; m) + \tilde{f}_\alpha(x; m, \Theta_\alpha)^{(j)}, \mathcal{J}_\beta(x; m) S_\beta \mathcal{J}_\beta(x; m)^\top\Big), \tag{3.26}$$

where $g_{\Theta_\beta} = \mathcal{N}(m_\beta, S_\beta)$ and samples $\tilde{f}_\alpha(x; m, \Theta_\alpha)^{(j)}$ are obtained by sampling parameters from the distribution $g_{\Theta_\alpha} = \mathcal{N}(m_\alpha, S_\alpha)$. For a derivation of this result, see Appendix A.1. This estimator is biased for finite R but converges to $\tilde{g}_{\tilde{f}(x;m,\Theta)}$ as $R \to \infty$. Similarly, for finite R, the smaller $[S_\alpha]_{ii}$, the more accurate and less biased the estimator will be. In our empirical evaluation, we use a single Monte Carlo sample, $R = 1$, to preserve Gaussianity and choose α to be the set of parameters in neural network layers $1 : L - 1$ and β to be the set of parameters in the final neural network layer to ensure fast gradient computation.

3.4.2 Approximating the Function-Space Kullback–Leibler Divergence

From Section 3.4.1, we know that if q_Θ and p_Θ are both Gaussian distributions, then the induced distributions under the linearized mapping $\tilde{f}$ evaluated at a finite set of evaluation points will be Gaussian as well. This means that for Gaussian variational and prior distributions over Θ, we can obtain locally accurate approximations to the induced distributions $q_{f(\cdot;\Theta)}$ and $p_{f(\cdot;\Theta)}$ and use them to approximate the KL divergence in the variational objective by $\mathbb{D}_{\mathrm{KL}}(\tilde{q}_{\tilde{f}(x;\Theta)} \,\|\, \tilde{p}_{\tilde{f}(x;\Theta)})$. Moreover, for

an isotropic Gaussian prior and a mean-field Gaussian variational distribution, $\mathbb{D}_{\mathrm{KL}}(\tilde{q}_{\tilde{f}(x;\Theta)} \,\|\, \tilde{p}_{\tilde{f}(x;\Theta)})$ is a KL divergence between two multivariate Gaussians and can be obtained analytically.

Using this approximation, we obtain an estimator of the variational objective given by

$$\tilde{\mathcal{F}}(q_\Theta) \stackrel{\text{def}}{=} \mathbb{E}_{q_{f(x_\mathcal{D};\Theta)}}[\log p_{y|f(x;\Theta)}(y_\mathcal{D} \mid f(x_\mathcal{D};\theta))] - \sup_{x \in \mathcal{X}_\mathbb{N}} \mathbb{D}_{\mathrm{KL}}(\tilde{q}_{\tilde{f}(x;\Theta)} \,\|\, \tilde{p}_{\tilde{f}(x;\Theta)}), \quad (3.27)$$

where the arguments of the KL divergence have been replaced by the (locally accurate) approximations to the variational and prior distributions over functions evaluated at x, respectively. Since the stochastic functions $\tilde{f}(\cdot\,;\Theta)$ induced by q_Θ and p_Θ under the linearized mapping will be closer to the stochastic function f the smaller the variance of q_Θ and p_Θ, respectively, the approximation to the KL divergence will be more accurate the smaller the variance of q_Θ and p_Θ.

Next, we turn to computing the supremum. Unlike Sun et al. (2019), who consider the supremum as a separate optimization problem, we do not seek to compute the supremum by searching over points $x \in \mathcal{X}_\mathbb{N}$ but instead propose to estimate the supremum at every gradient step via a simple finite-sample estimator. Specifically, letting $I(x) \stackrel{\text{def}}{=} \mathbb{D}_{\mathrm{KL}}(\tilde{q}_{\tilde{f}(x;\Theta)} \,\|\, \tilde{p}_{\tilde{f}(x;\Theta)})$, we estimate $G = \sup_{x \in \mathcal{X}_\mathbb{N}} I(x)$ using the Monte Carlo estimator

$$\hat{G}(\mathcal{X}_c) = \max_{x \in \mathcal{X}_c} I(x), \quad (3.28)$$

where $\mathcal{X}_c \stackrel{\text{def}}{=} \{x_c^{(i)}\}_{i=1}^K$ is a collection of K sets of *context points* $x_c^{(i)} \stackrel{\text{def}}{=} \{x^{(j)}\}_{j=1}^M$ jointly sampled from a context distribution p_{X_c}. Each context set $x_c^{(i)}$ can be viewed as a single Monte Carlo sample from the input space so that the estimator $\hat{G}(\mathcal{X}_c)$ provides a K-sample Monte Carlo estimate of the supremum. While this estimator is crude and only provides a rough approximation to the true supremum, it encourages the variational distribution over functions to match the prior distribution over functions on the sets of context points. The choice of the context distribution p_{X_c} can be informed by knowledge about the prediction task and should be viewed as a problem-specific modeling choice. Similarly, the numbers of samples K and M are hyperparameters to be optimized with a validation set.

3.4.3 Estimation of the Approximate Function-Space Variational Objective

Let q_Θ be a Gaussian mean-field variational distribution, let p_Θ be an isotropic Gaussian prior, let $(x_\mathcal{B}, y_\mathcal{B})$ be a mini-batch of the training data, and reparameterize Θ as $\breve{\Theta}(\mu, \Sigma, \boldsymbol{\epsilon}^{(j)}) \stackrel{\text{def}}{=} \mu + \Sigma \odot \boldsymbol{\epsilon}^{(j)}$. Using the estimator $\hat{G}(\mathcal{X}_c)$ defined above and

estimating the expected log-likelihood via Monte Carlo sampling, we obtain a Monte Carlo estimator for the function-space variational objective:

$$\bar{\mathcal{F}}(\mu, \Sigma) = \frac{1}{S} \sum_{i=1}^{S} \log p_{y|f(x;\Theta)}(y_{\mathcal{B}} \mid f(x_{\mathcal{B}}; \check{\Theta}(\mu, \Sigma, \boldsymbol{\epsilon}^{(i)}))) - \max_{x \in \mathcal{X}_c} \mathbb{D}_{\mathrm{KL}}(\tilde{q}_{\tilde{f}(x;\Theta)} \,\|\, \tilde{p}_{\tilde{f}(x;\Theta)}) \tag{3.29}$$

with $\boldsymbol{\epsilon}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_P)$ and $\mathcal{X}_c$ as defined above. This Monte Carlo estimator is biased due to the linearization and context-set approximations but allows for scalable gradient-based stochastic optimization.

3.5 Sequential Function-Space Variational Inference

Continual learning promises to enable applications of machine learning to settings with resource constraints, privacy concerns, or non-stationary data distributions. However, continual learning in deep neural networks remains a challenge. While progress has been made to mitigate "forgetting" of previously learned abilities, existing objective-based approaches to continual learning still fall short.

A popular family of objectives penalizes changes in parameters from one task to another (Ahn et al., 2019; Aljundi et al., 2018; Chaudhry et al., 2018; Kirkpatrick et al., 2017a; Lee et al., 2017; Liu et al., 2018; Loo et al., 2020; Nguyen et al., 2018; Park et al., 2019; Ritter et al., 2018; Schwarz et al., 2018; Swaroop et al., 2019; Yin et al., 2020a; Yin et al., 2020b; Zenke et al., 2017). However, explicitly regularizing parameters in this way may be ineffective, since parameters are only a proxy for a neural network's predictive function. For example, predictive functions defined by overparameterized neural networks may be obtained with several different parameter configurations, and small changes in a network's parameters may cause large changes in its predictions.

An alternative approach that addresses this shortcoming is to regularize the *predictive function* directly (Benjamin et al., 2019; Bui et al., 2017; Buzzega et al., 2020; Jung et al., 2018; Kapoor et al., 2021; Kim et al., 2018; Li et al., 2018b; Moreno-Muñoz et al., 2019; Pan et al., 2020; Titsias et al., 2020). Following this approach, we frame continual learning as sequential function-space variational inference (S-FSVI) and adapt the variational objective proposed in Section 3.4 to the continual-learning setting. The resulting variational optimization objective has a key advantage over existing alternatives: It is expressed purely in terms of distributions over predictive functions, which allows greater flexibility than with parameter-space regularization methods (Figure 3.1).

3.5.1 Continual Learning as Bayesian Inference over Stochastic Functions

We consider a sequence of tasks indexed by $t \in \{1, \ldots, T\}$. Each task involves making predictions on a supervised dataset $\mathcal{D}_t = \{(x_{tn}, y_{tn})\}_{n=1}^{N_t}$. To denote the sets of training inputs and targets for task t, we will use the shorthands x_t and y_t, respectively. Continual learning is the problem of inferring a distribution over predictive functions that fits the whole collection of datasets $\{\mathcal{D}_1, \ldots, \mathcal{D}_T\}$ as well as possible given access to only a single full dataset at a time.

Sequential Bayesian inference over predictive functions f provides a natural framework for this. To improve readability, we slightly change the notation for distributions over functions in this section: Assuming we have a prior $p(f)$, the posterior distribution over f at task 1 is

$$p(f \mid \mathcal{D}_1) = p(\mathcal{D}_1 \mid f)p(f)/p(\mathcal{D}_1). \tag{3.30}$$

For subsequent tasks t, the posterior can be expressed as

$$p(f \mid \mathcal{D}_1, \ldots, \mathcal{D}_t) \propto p(\mathcal{D}_t \mid f)p(f \mid \mathcal{D}_1, \ldots, \mathcal{D}_{t-1}), \tag{3.31}$$

where the posterior after task $t-1$ is treated as the prior for task t. Given the intractability of computing this posterior exactly, we need to use approximate inference.

3.5.2 A Sequential Function-Space Variational Objective

To approximate the posterior in Equation (3.31) at task t, we would like to find a variational distribution $q_t(f) \in \mathcal{Q}_f$ that minimizes

$$\mathbb{D}_{\mathrm{KL}}(q_t(f) \,\|\, p_t(f|\mathcal{D}_1, ..., \mathcal{D}_t)), \tag{3.32}$$

which can equivalently be expressed as maximizing

$$\mathbb{E}_{q_t(f)}[\log p(y_t \mid f(x_t))] - \mathbb{D}_{\mathrm{KL}}(q_t(f) \,\|\, p_t(f|\mathcal{D}_1, ..., \mathcal{D}_{t-1})). \tag{3.33}$$

Since we do not have access to $p_t(f|\mathcal{D}_1, ..., \mathcal{D}_{t-1})$, we simplify the inference problem to maximizing the variational objective

$$\mathbb{E}_{q_t(f)}[\log p(y_t \mid f(x_t))] - \mathbb{D}_{\mathrm{KL}}(q_t(f) \,\|\, p_t(f)), \tag{3.34}$$

where for $t = 1$ we assume some prior $p_1(f)$ and for $t > 1$ the prior is given by the variational posterior distribution over functions inferred on the previous task. That is,

$$p_t(f) \stackrel{\text{def}}{=} q_{t-1}(f).$$

While this objective is in general intractable for distributions over functions induced by neural networks with stochastic parameters, in Section 3.4, we proposed an

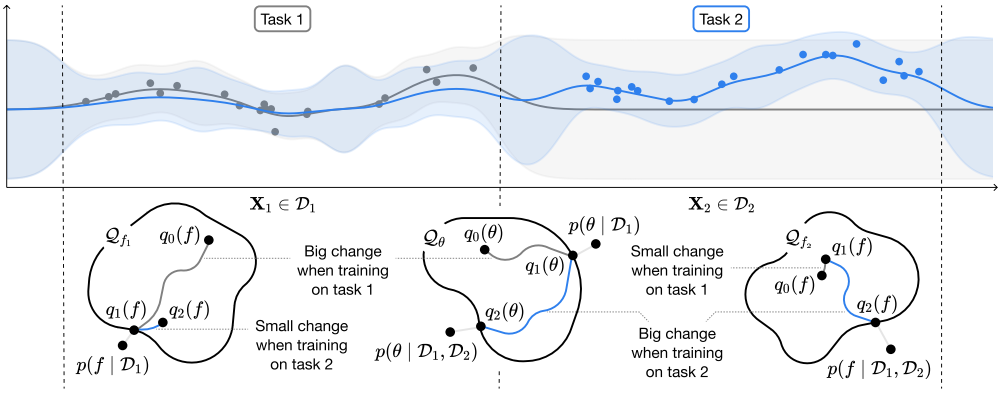


Figure 3.1: Schematic of how sequential function-space variational inference (S-FSVI) allows a Bayesian neural network to learn new tasks while maintaining previously learned abilities. (Top: predictive distributions.) On task 1, the model fits dataset $\mathcal{D}_1$ by updating an initial distribution over parameters $q_0(\theta)$ to a variational posterior $q_1(\theta)$, which in turn induces a distribution over functions $q_1(f)$. On task 2, the variational objective encourages the posterior distribution over functions to match $q_1(f)$ on a small set of data points from task 1 while also fitting dataset $\mathcal{D}_2$. The mean and two standard deviations of the distributions over functions learned on task 1 and task 2 are shown in gray and blue, respectively. (Bottom: learning trajectories.) On task 1, the distribution over functions changes by a large amount for inputs x_1 (left) but by a small amount for inputs x_2 (right). On task 2, the reverse is true. On both tasks, the change in the distribution over parameters (center) is only constrained by the desired distribution over functions (left, right).

approximation that makes this objective amenable to gradient-based optimization and scalable to large neural networks. To perform sequential function-space variational inference, we adapt the estimation procedure in Section 3.4 to the continual-learning setting. To do so, we let D_t be the number of model output dimensions for t tasks, let $f : \mathcal{X} \times \mathbb{R}^P \to \mathbb{R}^{D_t}$ be a mapping defined by a neural network architecture, let $\Theta \in \mathbb{R}^P$ be a multivariate random vector of network parameters, and let

$$q_t(\theta) \stackrel{\text{def}}{=} \mathcal{N}(\mu_t, \Sigma_t) \tag{3.35}$$

be a variational distribution over Θ and

$$q_{t-1}(\theta) \stackrel{\text{def}}{=} \mathcal{N}(\mu_{t-1}, \Sigma_{t-1}) \tag{3.36}$$

be a prior distribution over Θ. Finally, we define $\mathcal{X}_c^t \stackrel{\text{def}}{=} \{x_c^{(i)}\}_{i=1}^K$ as a collection of K sets of *context points* $x_c^{(i)} \stackrel{\text{def}}{=} \{x^{(j)}\}_{j=1}^M$ jointly sampled from a context distribution

$$p_{X_c^t} = \begin{cases} \text{Unif}(x_t \cup x_c^{:t-1}) & \text{if } t > 1 \\ \text{Unif}(x_t) & \text{if } t = 1 \end{cases}, \tag{3.37}$$

where $x_c^{:t-1}$ is a coreset constructed from $\{x_{t'}\}_{t'=1}^{t-1}$ for $t > 1$. Letting $\tilde{f}$ be a linearized mapping, as defined before, we get the variational objective

$$\begin{aligned}&\mathcal{F}(q_t, q_{t-1}, \mathcal{X}_c^t, x_t, y_t)\\&\stackrel{\text{def}}{=} \mathbb{E}_{q_t(\theta)}[\log p(y_t \mid f(x_t;\theta))] - \max_{x\in\mathcal{X}_c^t} \mathbb{D}_{\mathrm{KL}}(\tilde{q}_t(\tilde{f}(x;\Theta)) \,\|\, \tilde{q}_{t-1}(\tilde{f}(x;\Theta))).\end{aligned} \tag{3.38}$$

Under a diagonal approximation of the prior and variational posterior covariance functions across output dimensions, the objective can be expressed as

$$\begin{aligned}&\mathcal{F}(q_t, q_{t-1}, \mathcal{X}_c^t, x_t, y_t)\\&\stackrel{\text{def}}{=} \mathbb{E}_{q_t(\theta)}[\log p(y_t \mid f(x_t;\theta))] - \max_{x\in\mathcal{X}_c^t} \sum_{k=1}^{D_t} \frac{1}{2}\Big(\log \frac{|[\mathbf{K}^{p_t}]_k|}{|[\mathbf{K}^{q_t}]_k|} - |x| + \operatorname{Tr}([\mathbf{K}^{p_t}]_k^{-1}[\mathbf{K}^{q_t}]_k)\\&\qquad + \Delta(x;\mu_t,\mu_{t-1})^\top [\mathbf{K}^{p_t}]_k^{-1} \Delta(x;\mu_t,\mu_{t-1})\Big),\end{aligned} \tag{3.39}$$

where

$$\Delta_k(x;\mu_t,\mu_{t-1}) \stackrel{\text{def}}{=} [f(x;\mu_t)]_k - [f(x;\mu_{t-1})]_k \tag{3.40}$$

and

$$\mathbf{K}^{p_t} \stackrel{\text{def}}{=} \mathcal{J}(x;\mu_{t-1})\Sigma_{t-1}\mathcal{J}(x;\mu_{t-1})^\top \tag{3.41}$$

$$\mathbf{K}^{q_t} \stackrel{\text{def}}{=} \mathcal{J}(x,\mu_t)\Sigma_t\mathcal{J}(x,\mu_t)^\top, \tag{3.42}$$

are covariance matrix estimates constructed from Jacobians $\mathcal{J}(\cdot,\mathbf{m}) \stackrel{\text{def}}{=} \frac{\partial f(\cdot;\Theta)}{\partial\Theta}|_{\Theta=\mathbf{m}}$ with $\mathbf{m} = \{\mu_t, \mu_{t-1}\}$.

For ease of computation and to ensure scalability to large neural networks, we consider mean-field distributions $q_t^{\mathrm{MF}}(\theta)$ for all tasks, diagonalize the covariance matrix estimates $\mathbf{K}^{p_t}$ and $\mathbf{K}^{q_t}$ across input points in x, and let $(x_\mathcal{B}, y_\mathcal{B}) \subset \mathcal{D}_t$ be a mini-batch from the current dataset. This way, we obtain the simplified variational objective

$$\begin{aligned}&\tilde{\mathcal{F}}(q_t^{\mathrm{MF}}, q_{t-1}^{\mathrm{MF}}, \mathcal{X}_c^t, x_\mathcal{B}, y_\mathcal{B})\\&= \frac{1}{S}\sum_{i=1}^{S} \log p(y_\mathcal{B} \mid f(x_\mathcal{B}; \breve{\Theta}(\mu_t, \Sigma_t, \boldsymbol{\epsilon}^{(i)})))\\&\quad - \max_{x\in\mathcal{X}_c^t} \sum_{j=1}^{|x|}\sum_{k=1}^{D_t} \frac{1}{2}\Big(\log \frac{[\mathbf{K}^{p_t}]_{j,k}}{[\mathbf{K}^{q_t}]_{j,k}} + \frac{[\mathbf{K}^{q_t}]_{j,k}}{[\mathbf{K}^{p_t}]_{j,k}} - 1 + \frac{([f(x;\mu_t)]_{j,k} - [f(x;\mu_{t-1})]_{j,k})^2}{[\mathbf{K}^{p_t}]_{j,k}}\Big),\end{aligned} \tag{3.43}$$

where $\breve{\Theta}(\mu_t, \Sigma_t, \boldsymbol{\epsilon}^{(i)}) \stackrel{\text{def}}{=} \mu_t + \Sigma_t \odot \boldsymbol{\epsilon}^{(i)}$ is a reparameterization of $\Theta \in \mathbb{R}^P$ with $\boldsymbol{\epsilon}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_P)$, S is the number of Monte Carlo samples, D_t is as defined before, and

$$\mathbf{K}^{p_t} \stackrel{\text{def}}{=} \operatorname{diag}\left(\mathcal{J}(x, \mu_{t-1})\Sigma_{t-1}\mathcal{J}(x, \mu_{t-1})^\top\right) \tag{3.44}$$

$$\mathbf{K}^{q_t} \stackrel{\text{def}}{=} \operatorname{diag}\left(\mathcal{J}(x, \mu_t)\Sigma_t\mathcal{J}(x, \mu_t)^\top\right). \tag{3.45}$$

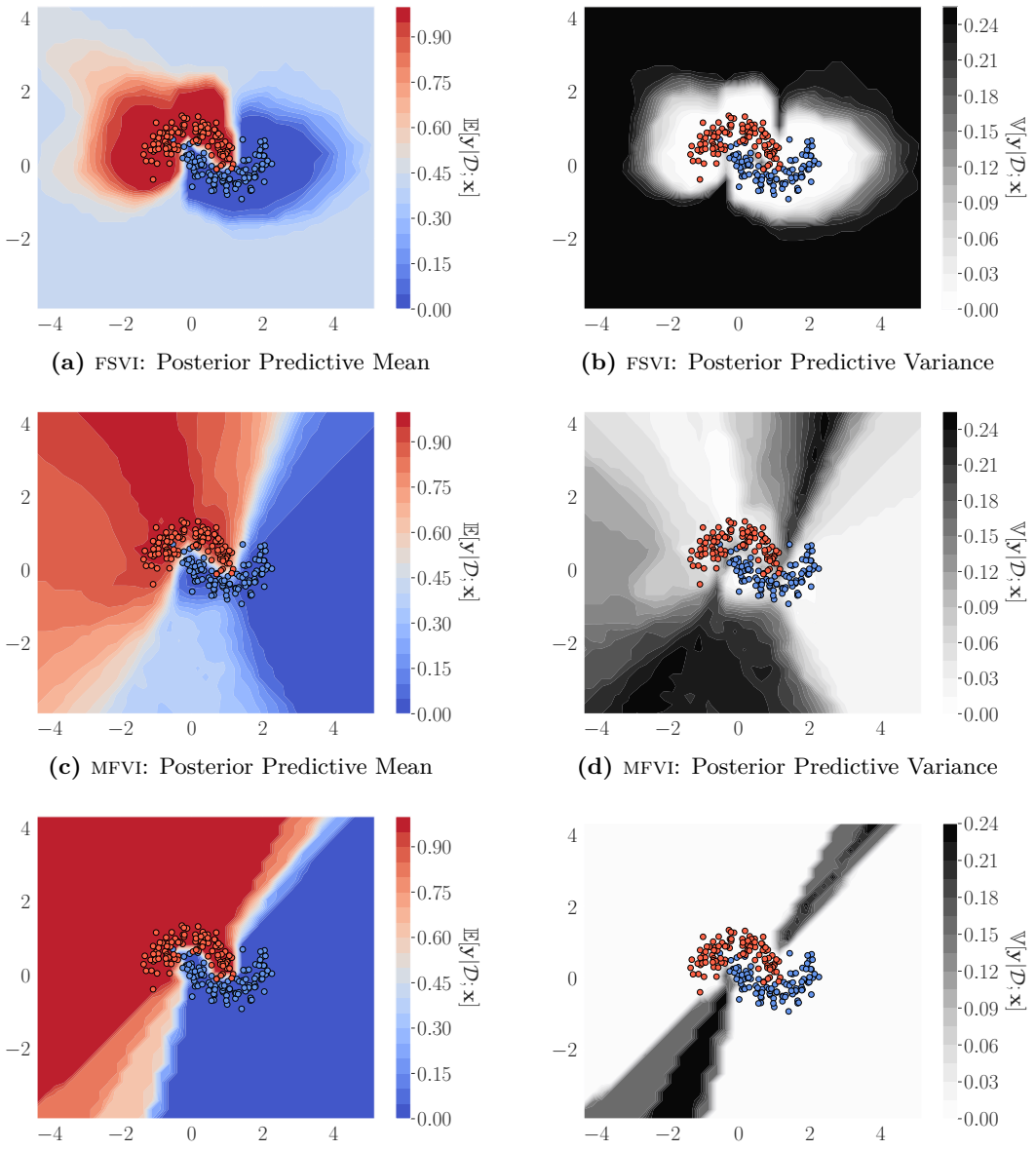
This simplified objective does not require matrix inversion, and the time and space complexity for gradient estimation and prediction scale linearly in the number of context points x and network parameters. The context set $x_c^{:t-1}$ can be constructed from coresets containing representative points from previous tasks.

3.6 Empirical Evaluation: Robustness to Distribution Shifts

3.6.1 Experiment Setup

For all experiments that involve uncertainty quantification, we chose a prior distribution over parameters that induces a prior distribution over functions $p_{f(\cdot;\Theta)}$ and a prior predictive distribution that exhibits a high degree of predictive uncertainty at evaluation points from regions in input space where p_{X_c} has non-zero support and, under smoothness constraints, on evaluation points in nearby regions. For settings where prior information is encoded in data—for example, in the form of expert demonstrations of robotic manipulation tasks (Rudner et al., 2021a) or in the form of pre-trained networks in continual or transfer learning (Rudner et al., 2022b)—an empirical prior that reflects this information can be specified. For further details, see Appendix A.2.

The distribution p_{X_c} allows us to incorporate information about the data-generating process into training and encourage the variational distribution to match the prior over functions in relevant parts of the input space. By taking advantage of the abundance of data available in real-world settings, context distributions can be constructed from large datasets like ImageNet (Krizhevsky et al., 2012), from small but diverse datasets like CIFAR-100, or by using any set of task-related unlabeled data. In our experiments, we choose two types of context distributions. One of the context distributions is constructed from the training data and only contains randomly sampled monochrome images, and one is constructed from a real-world dataset generated from a data distribution related to that of the training data. For example, when training on FashionMNIST, we use KMNIST as the context distribution, and when training on CIFAR-10, we use CIFAR-100 as the context distribution. For further details, see Appendix A.2.



(a) FSVI: Posterior Predictive Mean

(b) FSVI: Posterior Predictive Variance

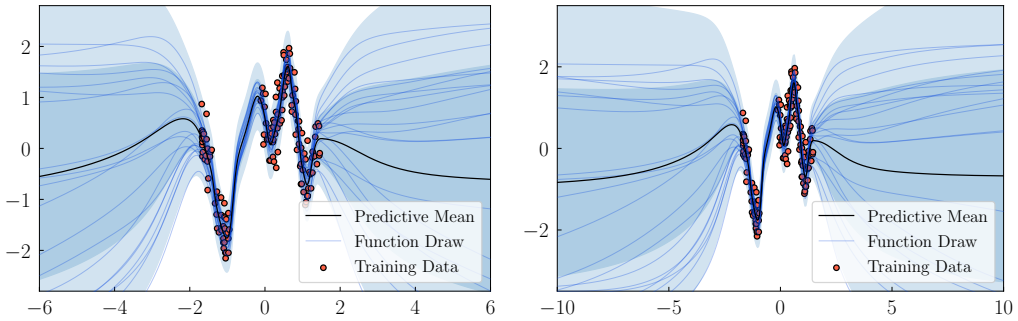
(c) MFVI: Posterior Predictive Mean

(d) MFVI: Posterior Predictive Variance

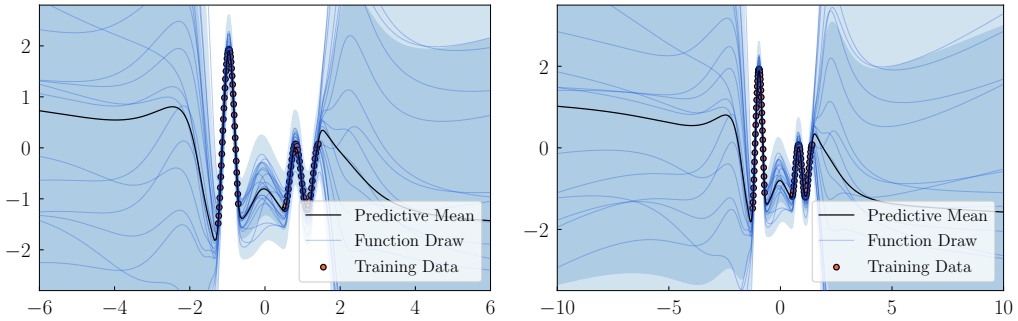
(e) MAP Ensemble: Predictive Mean

(f) MAP Ensemble: Predictive Variance

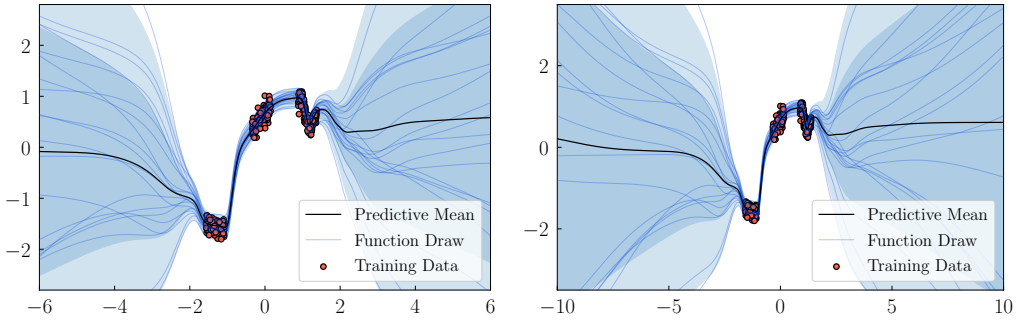
Figure 3.2: Binary classification on the *Two Moons* dataset. The plots show the posterior predictive mean and variance of a BNN trained via FSVI (Figure 3.2a and Figure 3.2b), of a BNN trained via MFVI (Figure 3.2c and Figure 3.2d), and an ensemble of MAP models (Figure 3.2e and Figure 3.2f). The predictive means represent the expected class probabilities and the predictive variance the model's epistemic uncertainty over the class probabilities. With FSVI, the predictive distribution is able to faithfully capture the geometry of the data manifold and exhibits high uncertainty over the class probabilities in areas of the data space about which the data is not informative. In contrast, neither MFVI nor MAP ensembles are able to accurately capture the geometry of the data manifold, and both only exhibit high uncertainty around the decision boundary.



(a) "Snelson" Dataset (Snelson et al., 2006)



(b) "OAT-1D" Dataset (Amersfoort et al., 2021)



(c) "Subspace Inference" Dataset (Izmailov et al., 2020)

Figure 3.3: 1D Regression with FSVI on a selection of datasets used to demonstrate desirable predictive uncertainty estimates in prior works. The left column is zoomed in.

Table 3.1: Comparison of in- and out-of-distribution performance metrics on FashionMNIST (mean ± standard error over ten random seeds). The last two columns show the AUROC for binary in- vs. out-of-distribution detection on MNIST (M) and NotMNIST (NM). MNIST and NotMNIST are used as out-of-distribution datasets. Best overall results for single and ensemble models are printed in boldface with gray shading. Results within a 95% confidence interval of the best overall result are printed in boldface only. All methods use the same four-layer CNN architecture. For further details about model architectures and training and evaluation protocols, see Appendix A.2.

Method	**Accuracy**↑	**ECE**↓	**AUROC** M↑	**AUROC** NM↑
MAP	91.73±0.08	0.037±0.001	87.00±0.30	74.85±1.31
MFVI (Blundell et al., 2015)	91.03±0.04	0.038±0.001	93.10±0.34	88.88±0.74
MFVI (tempered)	91.38±0.05	0.058±0.001	86.30±0.29	80.78±0.68
MFVI (radial) (Farquhar et al., 2020a)	90.31±0.11	0.035±0.001	84.40±0.68	82.11±1.15
MC DROPOUT (Gal et al., 2016)	90.55±0.04	0.012±0.001	88.46±0.57	80.02±1.04
SWAG (Maddox et al., 2019)	92.56±0.05	0.043±0.001	85.18±0.35	80.31±0.30
DUQ (Amersfoort et al., 2020)	92.40±0.20	–	95.50±0.70	94.60±1.80
BNN-LAPLACE (Immer et al., 2020)	92.25±0.10	0.012±0.003	95.55±0.60	–
SPG (Ma et al., 2021)	91.60±0.14	–	95.60±6.00	–
FSVI (p_{X_c} = synthetic)	93.13±0.13	0.012±0.002	96.23±0.46	95.02±0.69
FSVI (p_{X_c} = KMNIST)	**93.48**±0.12	**0.010**±0.001	**99.80**±0.20	**97.26**±0.23
Deep Ensemble	92.49±0.01	**0.019**±0.000	89.22±0.09	83.17±0.91
FSVI Ensemble (p_{X_c} = synthetic)	**94.44**±0.07	0.020±0.001	**97.85**±0.15	**96.95**±0.20

Table 3.2: Comparison of in- and out-of-distribution performance metrics on CIFAR-10 (mean ± standard error over ten random seeds). SVHN and corrupted CIFAR-10 (C-CIFAR) are used as out-of-distribution datasets. The penultimate column shows the AUROC for binary in- vs. out-of-distribution detection on SVHN. Best overall results for single and ensemble models are printed in boldface with gray shading. Results within a 95% confidence interval of the best overall result are printed in boldface only. All methods use a ResNet-18 architecture. For further details about model architectures and training and evaluation protocols, see Appendix A.2.

Method	**Accuracy**↑	**ECE**↓	**OOD-AUROC**↑	**C-CIFAR Acc**↑
MAP	93.19±0.11	0.043±0.001	94.65±0.27	78.87±1.39
MFVI (Blundell et al., 2015)	89.98±0.09	0.040±0.001	92.14±0.34	79.36±1.35
MFVI (tempered)	90.87±0.11	0.048±0.001	91.82±0.90	79.86±1.32
MC DROPOUT (Gal et al., 2016)	93.55±0.07	0.040±0.001	92.44±0.57	80.13±1.37
SWAG (Maddox et al., 2019)	93.13±0.14	0.067±0.002	89.79±0.50	76.12±0.51
VOGN (Osawa et al., 2019)	84.27±0.20	0.040±0.002	87.60±0.20	–
DUQ (Amersfoort et al., 2020)	**94.10**±0.20	–	92.70±1.30	–
SPG (Ma et al., 2021)	77.69±0.64	–	88.30±4.00	–
FSVI (p_{X_c} = synthetic)	93.35±0.04	0.034±0.001	94.76±0.24	**80.81**±0.43
FSVI (p_{X_c} = CIFAR-100)	93.57±0.04	**0.026**±0.001	**98.07**±0.10	**81.20**±0.42
Deep Ensemble	95.13±0.06	0.019±0.001	**98.04**±0.07	**81.22**±0.37
FSVI Ensemble (p_{X_c} = synthetic)	**95.19**±0.03	**0.013**±0.001	**99.19**±0.41	**81.35**±0.48

3.6.2 Illustrative Examples

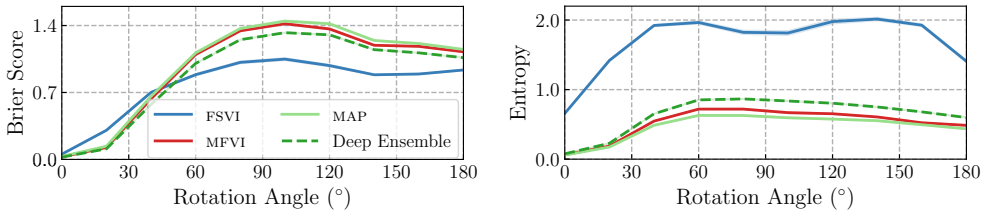
To illustrate how the predictive uncertainty of BNNs trained with FSVI differs from alternative uncertainty quantification methods, we use three different 1D regression datasets previously used in the literature and the *Two Moons* 2D classification dataset. As can be seen in Figure 3.2 and Figure 3.3, the predictive uncertainty produced by BNNs trained via FSVI has several desirable properties. First, it fits the data well in regions of the input space where many training points are available. Second, the uncertainty gradually increases further away from the training data. Third, the predictive distribution faithfully represents the properties of the data-generating process. That is, in Figure 3.2, the predictive distribution accurately captures the data geometry, and in Figure 3.3, the individual function draws accurately reflect the level of smoothness suggested by the training data and maintain patterns such as signs of periodicity. As can be seen in the comparison in Figure 3.2, the predictive distribution obtained with FSVI better reflects the underlying data geometry than alternative methods, such as deep ensembles and parameter-space variational inference.

3.6.3 Predictive Performance and Calibration on In-Domain Computer Vision Classification Tasks

To assess in-distribution predictive performance and calibration, we report the test accuracy and expected calibration error (ECE) for models trained on FashionMNIST and CIFAR-10 in Tables 3.1 and 3.2. On both FashionMNIST and CIFAR-10, FSVI achieves the best or second-best predictive accuracy and the best ECE across all methods. Notably, FSVI significantly outperforms SPG (Ma et al., 2021), an alternative function-space variational inference method.

3.6.4 Predictive Uncertainty under Distribution Shift in Computer Vision Classification Tasks

In Tables 3.1 and 3.2, we report evaluation metrics that elucidate the reliability of different methods' predictive uncertainty under distribution shift. FSVI exhibits reliable predictive uncertainty estimates that allow distinguishing between in- and out-of-distribution inputs with high accuracy. As would be expected, we observe that using context distributions that reflect our knowledge about the data-generating process can significantly improve uncertainty quantification under FSVI. For the FashionMNIST experiment, we used the KMNIST dataset, which contains grayscale images of Kuzushiji letters, and for the CIFAR-10 experiment, we used the CIFAR-100 dataset, which contains RGB images of 100 classes. Both KMNIST and CIFAR-100 differ from the OOD datasets (MNIST, NotMNIST, and SVHN, respectively) used to compute OOD-AUROC metrics in Tables 3.1 and 3.2, but using them as context distributions significantly increased the ability of BNNs trained via FSVI to identify distributionally shifted samples. The variational objective encourages matching the prior (which we chose to have high variance) on samples from the context distribution



(a) Brier score **(b)** Predictive entropy

Figure 3.4: Predictive uncertainty and accuracy/Brier score on rotated MNIST. Models with reliable uncertainty estimates should exhibit higher predictive uncertainty the more the digits are rotated. Ideally, such models would maintain high predictive accuracy (i.e., low Brier score). FSVI exhibits an increase in Brier scores as the predictive entropy increases, as desired. The fact that FSVI has a lower Brier score on highly rotated images than alternative methods demonstrates that it leads to improved predictive accuracy and uncertainty quantification under distribution shifts.

can improve uncertainty estimation in regions of the input space far from the training data.

3.6.5 Generalization and Reliability of Predictive Uncertainty under Distribution Shift: Rotated MNIST Classification

To assess the reliability of predictive models in deep learning, Ovadia et al. (2019) propose the following desiderata: In order for a model to be considered reliable, it ought to (i) exhibit low predictive uncertainty on training data and high predictive uncertainty on out-of-distribution inputs, (ii) generate predictive uncertainty estimates that allow distinguishing in- from out-of-distribution inputs, and (iii) if possible, maintain high predictive accuracy even under distribution shift. Models that satisfy these desiderata are less likely to make poor, high-confidence predictions and better suited for use in safety-critical downstream tasks.

To illustrate these desiderata, we follow Ovadia et al. (2019) and consider the rotated MNIST task, where a model is trained on MNIST and evaluated on rotated MNIST digits. The goal is to maintain a high level of predictive accuracy (measured in terms of Brier scores) while exhibiting an increasing level of predictive uncertainty on distribution shifts of increasing magnitude. Figure 3.4 shows Brier scores (lower is better) and predictive entropy estimates (higher means more uncertain) of four different models. As rotating the MNIST digits gradually shifts the data distributions, we would expect Brier scores to increase (corresponding to worse predictive accuracy) as the rotation angle increases. A model with reliable predictive entropy estimates would only experience a small increase under distribution shift while exhibiting a large increase in predictive uncertainty. As can be seen in the plot, the Brier scores of FSVI increase the least, while FSVI's uncertainty is significantly higher than that of other models. To assess the reliability of different uncertainty quantification methods

on a more challenging distribution-shift task, we consider corrupted CIFAR-10 inputs under the second-mildest corruption level used in Ovadia et al. (2019) and report our results in Table 3.2. Consistent with the rotated MNIST results, FSVI achieves the highest accuracy on the corrupted data.

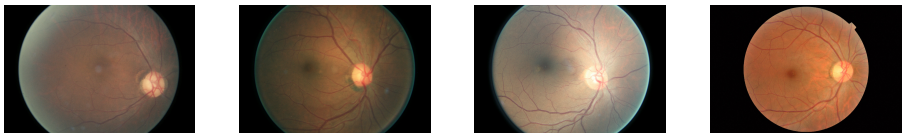
3.6.6 Safety-Critical Uncertainty-Aware Selective Prediction: Diabetic Retinopathy Diagnosis

To evaluate the reliability of the predictive uncertainty of FSVI in a real-world safety-critical setting, we consider the task of diagnosing diabetic retinopathy (DR)—a medical condition that can lead to impaired vision—from retina scans (Leibig et al., 2017; Filos et al., 2019; Band et al., 2021). We use two publicly available datasets, EyePACS (2015) and APTOS (2019), each containing RGB images of a human retina graded by a medical expert on the following scale: 0 (no DR), 1 (mild DR), 2 (moderate DR), 3 (severe DR), and 4 (proliferative DR). The EyePACS dataset was collected from patients in the United States, while the APTOS dataset was collected from patients in India using cheaper but more modern scanning devices. For sample retina scans, see Figure 3.5. We follow Leibig et al. (2017), Filos et al. (2019), and Band et al. (2021) and binarize all examples from both the EyePACS and APTOS datasets by dividing the classes up into sight-threatening diabetic retinopathy—defined as moderate diabetic retinopathy or worse (classes $\{2, 3, 4\}$)—and non-sight-threatening diabetic retinopathy—defined as no or mild diabetic retinopathy (classes $\{0, 1\}$). This results in a binary prediction task.

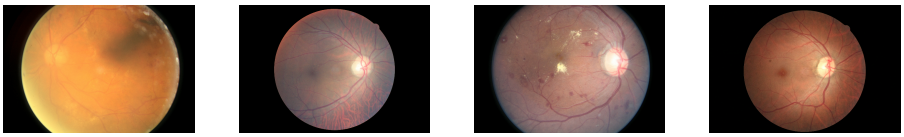
To assess the reliability of predictive models when medical training and test data are obtained from different patient populations or collected with different medical equipment, we follow Band et al. (2021) and use the Kaggle dataset for training and the distributionally shifted APTOS dataset for evaluation. The results are shown in Figure 3.6, which plots the ROC curves for the binary prediction problems as well as the area under the ROC curve for an uncertainty-aware selective prediction task. For further details about the uncertainty-aware selective prediction evaluation protocol, see Appendix A.2.4. Figure 3.6 shows that FSVI performs well on all four tasks and is only outperformed by MC DROPOUT. For full tabular results, see Appendix A.3.1.

3.7 Empirical Evaluation: Continual Learning

After visualizing how S-FSVI works in practice (Section 3.7.1), we compare S-FSVI's performance with that of existing objective-based methods for continual learning (Sections 3.7.2 to 3.7.4). For a comprehensive comparison, we evaluate S-FSVI on a range of task sequences used in related work. Aiming to use the strongest possible baselines, we report results taken directly from the literature in most cases (and mention when we do not). Reporting baselines in this way leaves gaps in our

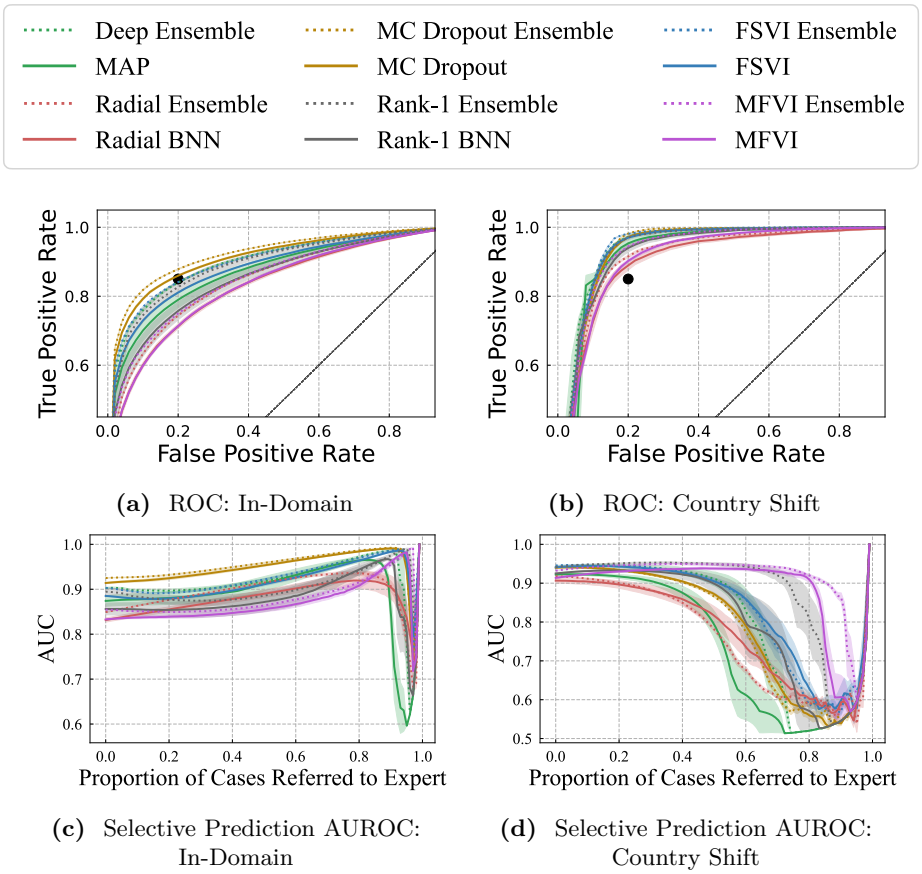


(a) Retina images exhibiting non-sight-threatening diabetic retinopathy ($y = 0$).



(b) Retina images exhibiting sight-threatening diabetic retinopathy ($y = 1$).

Figure 3.5: Samples of retina scans from the EyePACS dataset showing varying degrees of diabetic retinopathy.



(a) ROC: In-Domain

(b) ROC: Country Shift

(c) Selective Prediction AUROC: In-Domain

(d) Selective Prediction AUROC: Country Shift

Figure 3.6: Joint assessment of model predictive performance and uncertainty quantification on in-domain and distributionally shifted data. **Top:** ROC curves for in-population and country shift. **Bottom:** Selective prediction AUROC for different settings. The black dot denotes the NHS-recommended sensitivity and specificity ratios (Widdowson, 2016).

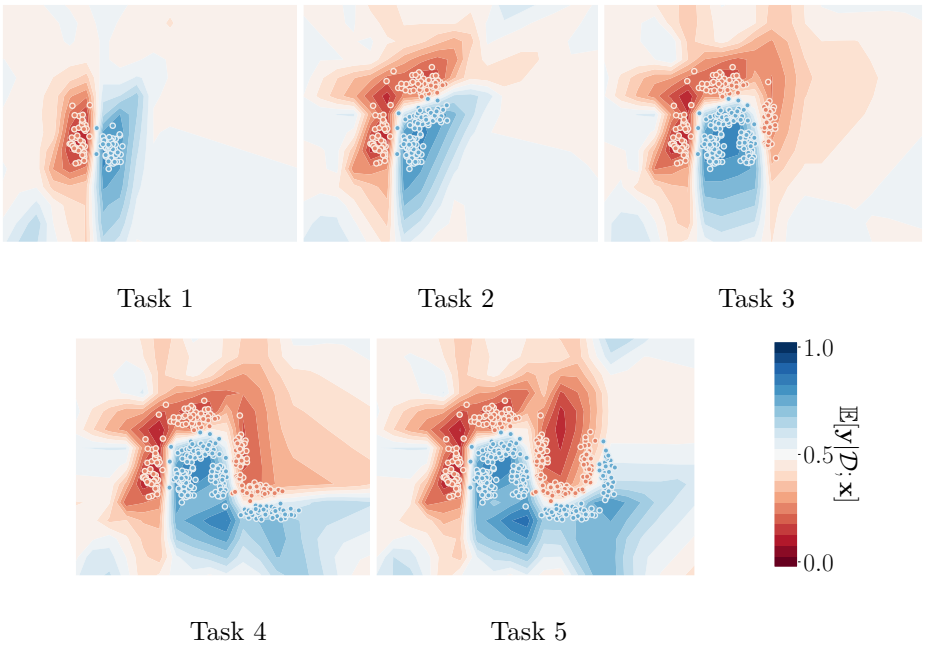


Figure 3.7: A practical demonstration of sequential function-space variational inference (S-FSVI) on a sequence of five binary-classification tasks with 2D inputs. The neural network infers a decision boundary between the two classes while maintaining high predictive uncertainty away from the data. The experimental setup is described in detail in Appendix A.4.

comparison: for each existing technique, results are available for only a subset of the task sequences we consider here (e.g., Pan et al. (2020) report results for split CIFAR but not sequential Omniglot, while Titsias et al. (2020) do the reverse).

Our evaluation pays attention to two factors important in the assessment of continual-learning methods: the use of task identifiers when making predictions, and the use of coresets of data points to summarize past tasks (Farquhar et al., 2018). To provide some commentary on the first of these factors, we run an experiment that compares the performance of a single-head neural network (which does not use task identifiers) to that of a multi-head neural network (which uses task identifiers). Regarding the second factor, we explore how performance changes when the coreset size changes or a context set unrelated to previous tasks is used.

Details about the experimental setups (e.g., optimization routines and hyperparameter searches) can be found in Appendix A.4.

3.7.1 Illustrative Example

To provide intuition for how S-FSVI allows learning on new tasks while maintaining previously acquired abilities, we apply it to a task sequence based on easy-to-visualize

Table 3.3: Predictive accuracies of a selection of objective-based methods for continual learning. Results are reported for three task sequences: split MNIST (S-MNIST), split Fashion MNIST (S-FMNIST) and permuted MNIST (P-MNIST). In some cases, a multi-head setup (MH) is used; in others, a single-head setup (SH). Best results for identical network architectures are printed in boldface (exception: VAR-GP uses a non-parametric model). Best overall results are highlighted in gray. Each numerical entry denotes the mean accuracy across tasks at the end of training. Where possible, this accuracy is based on experiments repeated with different random seeds (10 repeats for S-FSVI), with both the mean value and standard error reported. All methods use the same architecture and coreset size unless indicated otherwise. See Appendix A.4 for more experimental details. [1]Accuracies computed using the best coreset-selection method (either random or k-center). [2]Uses random coreset selection. [3]Requires a multi-head setup with task identifiers, including for permuted MNIST. This requirement explains the missing FRCL result for S-MNIST (SH). [4]Uses a larger MLP architecture (see Table A.3 in the appendix).[5]Evaluates the KL divergence at points sampled from the empirical data distribution of the current task. [6]Uses one sample per class as a coreset.

Method	S-MNIST (MH)	S-FMNIST (MH)	P-MNIST (SH)	S-MNIST (SH)
EWC (Kirkpatrick et al., 2017a)	63.10%	—	84.00%	—
SI (Zenke et al., 2017)	98.90%	—	86.00%	—
VCL (Nguyen et al., 2018)[1]	98.40%	98.60%±0.04	93.00%	32.11%±1.16
VCL (no coreset)	97.00%	89.60%±1.75	87.50%±0.61	17.74%±1.20
FRCL (Titsias et al., 2020)[3]	97.80%±0.22	97.28%±0.17	94.30%±0.06	—
FROMP (Pan et al., 2020)	99.00%±0.04	99.00%±0.03	94.90%±0.04	35.29%±0.52
VAR-GP (Kapoor et al., 2021)	—	—	**97.20%**±0.08	90.57%±1.06
S-FSVI (**ours**)[2]	**99.54%**±0.04	**99.19%**±0.02	95.76%±0.02	**92.87%**±0.14
S-FSVI Ablation Study:				
S-FSVI (larger networks)[4]	**99.76%**±0.00	99.16%±0.03	**97.50%**±0.01	**93.38%**±0.10
S-FSVI (no coreset)[5]	99.62%±0.02	**99.54%**±0.01	84.06%±0.46	20.15%±0.52
S-FSVI (minimal coreset)[6]	—	—	89.59%±0.30	51.44%±1.22

synthetic 2D data, originally proposed by Pan et al. (2020). In this task sequence, each data point belongs to one of two classes, and more data points are revealed as the task sequence progresses. The data-generating process is assumed to reveal data from mostly non-overlapping subsets of the input space. The continual-learning problem is then to infer the decision boundary around data points revealed up to and including the current task without forgetting the decision boundary inferred on previous tasks. We use a single-head neural network.

In Figure 3.7, we plot the model's posterior predictive distribution after training on each of five tasks. After training on task 1, the model has low predictive uncertainty close to the data points and high uncertainty (class probabilities around 0.5) everywhere else. On task 2, S-FSVI seeks to match the distribution over functions inferred on the previous task while fitting the new set of data points. S-FSVI achieves this and expands the area in input space where the model is confident in its predictions.

As more tasks and data are revealed, S-FSVI allows the model to continually explore the data space and infer the decision boundary while maintaining accurate, high-confidence predictions on data points in parts of the input space where it was previously trained on observed data. Finally, after training on five tasks, the model has inferred the decision boundary between the two classes, while maintaining high predictive uncertainty in parts of the input space where no data points have been observed yet. The model maintains high predictive uncertainty away from the data, which makes it easier to learn on new tasks. This is unlike deterministic neural networks, which tend to make highly confident predictions in parts of the input space where no data has been observed, or on data points that lie outside of the distribution of the training data.

3.7.2 Split (Fashion) MNIST & Permuted MNIST

Having established some intuition for how S-FSVI works, we demonstrate how this translates to high predictive accuracy on three task sequences commonly used to evaluate continual-learning methods. First is split MNIST (S-MNIST), in which each task consists of binary classification on a pair of MNIST classes (0 vs. 1, 2 vs. 3, and so on). Second is split Fashion MNIST (S-FMNIST), which has the same structure but uses data from Fashion MNIST, posing a harder problem. Third is permuted MNIST (P-MNIST), in which each task consists of ten-way classification on MNIST images whose pixels have been randomly reordered. A multi-head setup (MH) with task identifiers provided at prediction time is the default for S-MNIST and S-FMNIST, while a single-head setup (SH) without task identifiers is standard for P-MNIST. In addition to running the default setup for all three task sequences, we run a single-head setup for S-MNIST.

With a standard configuration, S-FSVI outperforms all existing methods based on deep neural networks by a statistically significant margin on all task sequences (Table 3.3). VAR-GP performs better than our standard configuration of S-FSVI on permuted MNIST, but this advantage disappears once a larger neural network is used with S-FSVI. Moreover, VAR-GP is unlikely to scale well to more challenging task sequences, such as those in Sections 3.7.3 and 3.7.4.

3.7.3 Sequential Omniglot

Sequential Omniglot (Lake et al., 2015; Schwarz et al., 2018) provides a more challenging task sequence than those considered in Section 3.7.2. It consists of 50 classification tasks, where the number of classes varies between the tasks (details in Appendix A.4). We find that S-FSVI produces better predictive accuracy than all available baselines, including FRCL, by a statistically significant margin (Table 3.4). To illustrate the stability of S-FSVI across long task sequences, we plot its mean accuracy over 50 tasks in Figure 3.8.

Table 3.4: Predictive accuracies of S-FSVI and related methods on sequential Omniglot. For S-FSVI and FRCL, the coreset consists of two data points per class. All baseline results are from Titsias et al. (2020). For all methods, the mean and standard deviation over five random task permutations are reported. [1]Li et al. (2018b). [2]Kirkpatrick et al. (2017b). [3]Schwarz et al. (2018). [4]Coreset selected using FRCL's "trace" method. [5]Details in Appendix A.4.

Method	Test Accuracy
Learning Without Forgetting[1]	62.06%±2.0
EWC	67.32%±4.7
Online EWC[2]	69.99%±3.2
Progress & Compress[3]	70.32%±3.3
FRCL[4]	81.47%±1.6
S-FSVI (**ours**)[5]	**83.29%**±1.2

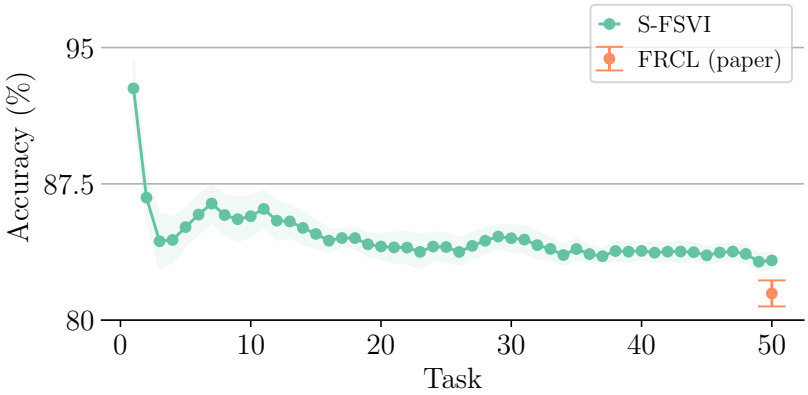
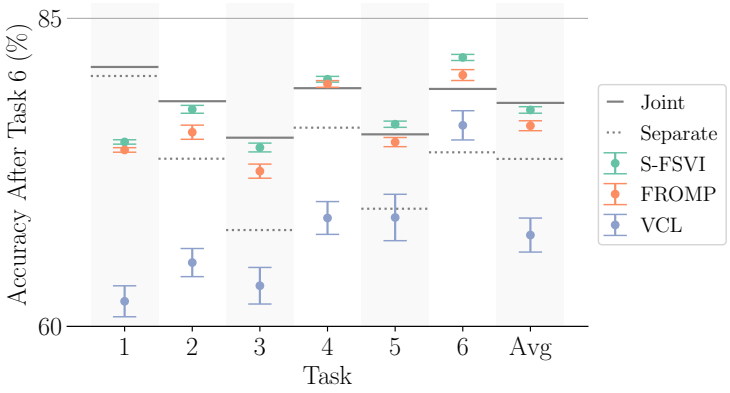


Figure 3.8: Predictive accuracies of S-FSVI and FRCL on sequential Omniglot. For S-FSVI, the accuracy shown at task t is the mean accuracy across all tasks up to that point (mean ± one standard error as computed across five permutations of the task order). We were unable to reproduce the result reported in Titsias et al. (2020) using the authors' code. However, we compare against the result from the paper (only the accuracy at task 50 is reported) here to provide a strong baseline.

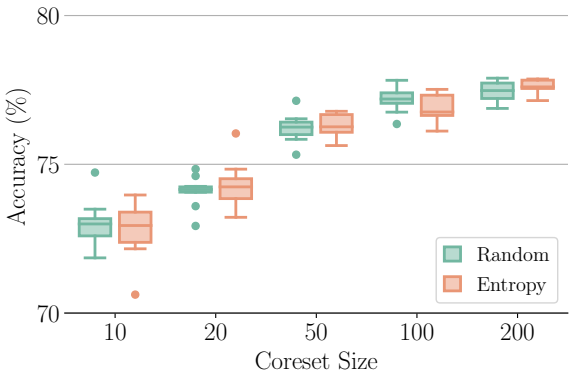
3.7.4 Split CIFAR

Moving beyond classification tasks on grayscale images, we evaluate S-FSVI on split CIFAR (Pan et al., 2020; Zenke et al., 2017). This uses the full CIFAR-10 dataset for the first task, followed by five ten-way classification tasks drawn from CIFAR-100. Our results show S-FSVI achieving higher accuracy on all tasks than FROMP and VCL after learning all six tasks (Figure 3.9a). Notably, on each task except the first, S-FSVI performs close to or better than two baselines: a model trained only on that task, and a model trained on all tasks jointly. The latter is a particularly strong baseline, because all data is available during training.

We compute the forward transfer and backward transfer for S-FSVI on the split CIFAR dataset. Backward transfer indicates the performance gain on past tasks when new tasks are learned, while forward transfer quantifies how much knowledge from past tasks helps the learning of new tasks. More specifically, for T tasks, let



(a) Split CIFAR Accuracies After Training on All Tasks

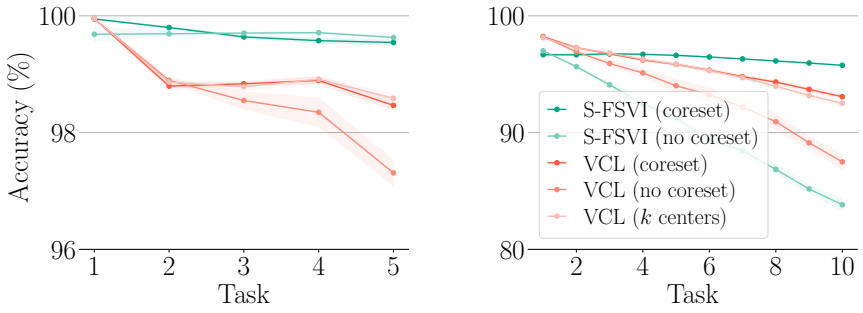


(b) S-FSVI Accuracy as a Function of Coreset Size

Figure 3.9: Predictive accuracies of S-FSVI and related methods on split CIFAR. (a) Per-task and average accuracy after training on six tasks. The result of the "joint" baseline is obtained using a model trained on data from all tasks at the same time. The accuracy at task t for the "separate" baseline is the accuracy of an independent model trained only on task t. We use the best-performing method for each baseline: FROMP for "joint", S-FSVI for "separate". (b) Average accuracy after training on six tasks with different coreset sizes. "Random" coreset selection denotes uniform sampling from the training set. "Entropy" coreset selection denotes sampling from the training set with probability proportional to the entropy of the model's posterior predictive distribution.

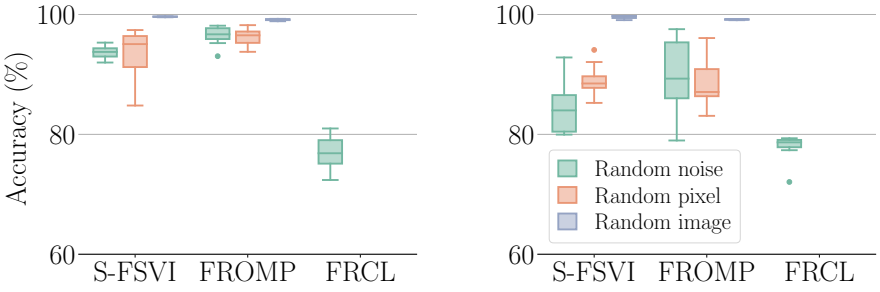
$R_{j,i}$ be the accuracy of a model on task t_i after training on task t_j, and let R_i^{ind} be the accuracy of an independent model trained only on task t_i. We can then define backward transfer as

$$\text{Backward Transfer} = \frac{1}{T-1} \sum_{i=1}^{T-1} (R_{T,i} - R_{i,i}) \tag{3.46}$$



(a) S-MNIST (MH) **(b)** P-MNIST (SH)

Figure 3.10: Predictive accuracies of S-FSVI and parameter-space variational inference (VCL) on split MNIST and permuted MNIST. The accuracy shown at task t is the mean accuracy across all tasks up to that point (mean $\pm$ one standard error as computed across ten repetitions of the experiment). With a coreset, S-FSVI outperforms VCL on both task sequences. Without a coreset, S-FSVI performs poorly on permuted MNIST.



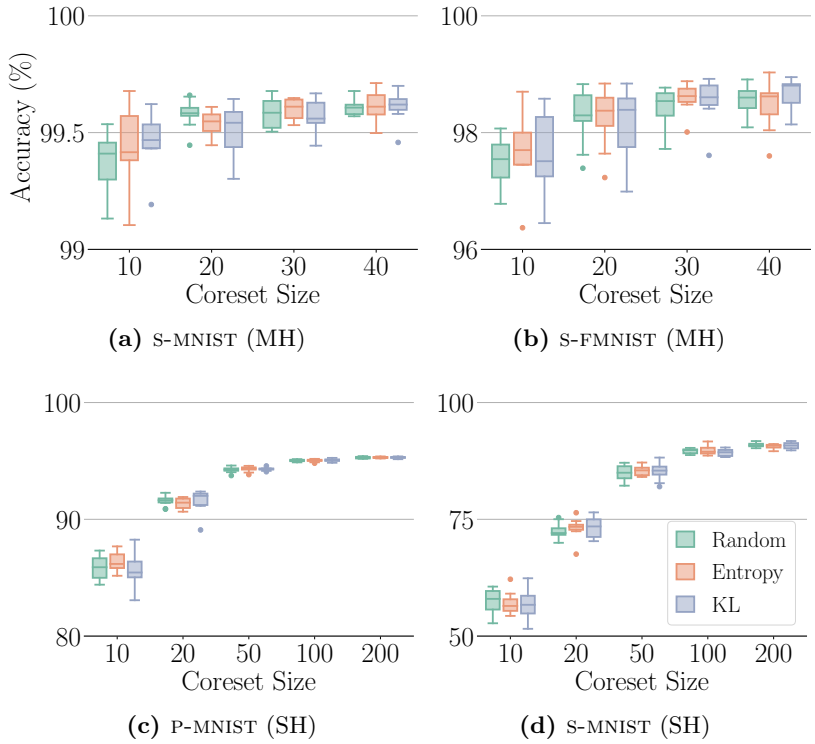
(a) S-MNIST (MH) **(b)** S-FMNIST (MH)

Figure 3.11: Predictive accuracies of S-FSVI, FROMP, and FRCL on multi-head split (Fashion) MNIST without using coresets. Inducing inputs for evaluating the KL divergence are sampled according to three different sampling schemes derived from the current task's empirical data distribution (see Appendix A.4 for details). Using S-FSVI with images sampled from the current task's training set significantly outperforms all other methods.

and forward transfer as

$$\text{Forward Transfer} = \frac{1}{T-1}\sum_{i=2}^{T}\left(R_{i,i} - R_i^{\text{ind}}\right). \tag{3.47}$$

As can be seen in Table 3.5, in addition to having the best overall accuracy, S-FSVI significantly outperforms all baselines in terms of forward transfer and has a backward transfer comparable to EWC and FROMP.



(a) S-MNIST (MH) **(b)** S-FMNIST (MH)

(c) P-MNIST (SH) **(d)** S-MNIST (SH)

Figure 3.12: Effect of the coreset size and coreset-selection method on the predictive accuracy of S-FSVI. Three coreset-selection methods are presented: sampling data points with uniform probability; sampling with probability proportional to the model's predictive entropy; and sampling with probability proportional to the KL divergence between the posterior predictive distribution and the prior predictive distribution. Ten inducing points are used in each case. No coreset-selection method consistently yields higher accuracy.

3.7.5 Function- vs. Parameter-Space Inference

To demonstrate the importance of performing inference in function space, we compare how the accuracies of S-FSVI and VCL evolve from one task to another on split MNIST and permuted MNIST (Figure 3.10). We find that S-FSVI consistently outperforms VCL, whose predictive performance steadily degrades, suggesting that function-space inference may be more effective than parameter-space inference at transferring prior knowledge from one task to another, and that this may offset the information loss in the KL divergence between distributions over functions compared to the KL divergence between distributions over parameters.

3.7.6 Coreset Size and Selection

Similar to existing methods such as FROMP and FRCL, S-FSVI includes in the training objective a function-space regularization term that encourages matching the prior distribution over functions at a set of context points. Typically, this requires keeping

Table 3.5: Forward transfer and backward transfer of S-FSVI and related methods on split CIFAR. All baseline results are from Pan et al. (2020). For all methods, the mean and standard error over five repeated experiments are reported.

Method	Test Accuracy	Forward Transfer	Backward Transfer
EWC	71.6%±0.4	0.2±0.4	**-2.3**±0.6
VCL	67.4%±0.6	1.8±1.4	-9.2±0.8
FROMP	76.2%±0.2	6.1±0.3	**-2.6**±0.4
S-FSVI (ours)	**77.6%**±0.2	**7.3**±0.2	**-2.5**±0.2

a representative coreset of data points from each task, from which a context set can be constructed.

S-FSVI offers two benefits with respect to coresets. First, it is insensitive to which points get included in the coresets. Whereas existing methods often require expensive procedures to select important data points from previous tasks, Figures 3.9b and 3.12 show that S-FSVI achieves strong performance while only using randomly selected coresets. Second, S-FSVI does not require large coresets to perform well. On permuted MNIST, S-FSVI achieves better predictive accuracy than EWC and SI even if the coreset used for S-FSVI consists of only a single data point per class (Table 3.3). On the single-head version of split MNIST, a minimal coreset (one point per class, or two points per task) allows S-FSVI to outperform VCL and FROMP, both with coresets of 40 points per task (Table 3.3). In some multi-head settings, S-FSVI achieves state-of-the-art predictive accuracies with randomly generated noise coresets (Table 3.3 and Figure 3.11).

3.8 Discussion

In this chapter, we presented a tractable approach to function-space variational inference and extended this approach to the continual learning setting. We argued that variational inference in the space of functions induced by a distribution over neural network parameters is more likely to result in an approximate posterior distribution that exhibits effective reversion to the prior predictive distribution for predictions on points in the input space that are meaningfully different from the training data. We discussed how to specify prior distributions over functions that result in a well-defined function-space variational objective and proposed a simple linearization approach to approximate the KL divergence between two distributions over functions induced by a distribution over neural network parameters and evaluated at a finite set of input points.

We carried out an extensive empirical evaluation, which involved assessing the predictive accuracy and uncertainty quantification of neural networks trained via FSVI. To simulate the use of neural networks trained via FSVI in safety-critical settings, we evaluated FSVI on the RETINA benchmark, which emulates an automated

medical diagnosis pipeline where a model is supposed to successfully refer data points on which it is likely to make an incorrect prediction to a human expert. We found that FSVI produces significantly improved uncertainty estimates compared to alternative methods, and even exceeds the performance of deep ensembles in several cases. To assess how effective sequential function-space variational inference is at preventing catastrophic forgetting in continual learning settings, we evaluated S-FSVI on a wide range of well-established and challenging benchmarking tasks. We found that S-FSVI significantly outperforms alternative parameter-space and function-space approaches, even when using very naive coreset selection techniques.

Despite the empirical success of FSVI and S-FSVI showcased in this chapter, several challenges remain. Most importantly, while FSVI scales to datasets with high-dimensional inputs, the computational cost of computing the function-space KL estimate increases linearly with the number of output dimensions, making it expensive to adapt to high-dimensional prediction problems, such as ImageNet classification. Moreover, while we found that S-FSVI exhibits best-in-class performance on continual learning tasks, on some computer vision classification tasks the predictive accuracy of neural networks trained via FSVI was below levels of predictive accuracy that are achievable with commonly used regularization methods for neural networks, such as sharpness-aware minimization (Foret et al., 2021). Finding prior distributions over functions that not only improve neural network models' uncertainty quantification but also improve predictive accuracy remains an open problem.

4

Data-Driven Priors and Variational Inference in Neural Network Models

4.1 Introduction

In this chapter, we design uncertainty-aware priors (UAPs) and group-aware priors (GAPs) for fine-tuning pre-trained neural networks and show that models fine-tuned with such priors achieve significantly improved predictive uncertainty quantification and group robustness, respectively.

While fine-tuning off-the-shelf pre-trained neural networks has become the default approach for most computer vision and natural language processing tasks, standard fine-tuning methods—such as empirical risk minimization (ERM; Vapnik, 1998)—often fall short in providing reliable uncertainty estimates that accurately indicate how confident a model is about its predictions (Tran et al., 2022) and often fail under distribution shifts. Reliable uncertainty quantification, in conjunction with uncertainty-based deferral of predictions to human experts, can be a particularly useful tool in creating scalable oversight for large pre-trained models and help build trust for automated decision-making.

We present a simple addition to standard fine-tuning techniques that enables more reliable uncertainty quantification and improved group robustness in fine-tuned models. More specifically, we develop a family of data-driven priors and specify specific, uncertainty-aware and group-aware prior distributions over neural network parameters designed to lead to improved uncertainty quantification and group robustness, respectively. Prior distributions over model parameters can

incorporate relevant information about the parameter values into Bayesian inference. However, specifying meaningful prior distributions over neural network parameters is challenging since it is difficult to discern which parameter configurations correspond to specific desired behaviors (e.g., good calibration, high predictive uncertainty under semantic shift, low negative log-likelihood, robustness to distribution shifts, etc.).

To tackle this challenge, we use information about the data and the prediction problem to construct a *data-driven uncertainty-aware prior distribution* explicitly designed to encourage improved uncertainty quantification and a *data-driven group-aware prior distribution* explicitly designed to encourage group robustness. The uncertainty-aware data-driven prior is specified by constructing a distribution that assigns high probability density to parameter values that induce functions with high predictive entropy on points that are meaningfully different from the training data, and the group-aware data-driven prior is specified by constructing a distribution that assigns high probability density to parameter values that induce functions that generalize especially well to data points that belong to groups that are underrepresented in the training data.

We show how to use these priors for fine-tuning off-the-shelf pre-trained models using standard mean-field variational inference for neural networks (Graves, 2011; Blundell et al., 2015) and evaluate the fine-tuned models on a wide range of benchmarking tasks—including image classification and text classification. The proposed family of data-driven priors is probabilistically principled, simple, scalable, and easy to implement in practice. Unlike existing data-driven priors that require separate training procedures (e.g., Shwartz-Ziv et al., 2022), the family of data-driven priors proposed here allows for simple end-to-end training without the need for additional prior pre-training.

4.2 Related Work

4.2.1 Informative Priors in Bayesian Deep Learning

Informative priors have the potential to greatly improve model performance. However, certain challenges, such as cold posterior effects and performance concerns compared to traditional neural networks, have been reported in the literature (Wenzel et al., 2020). To address these shortfalls, recent studies have focused on exploring informative priors, including heavy-tailed priors (Fortuin et al., 2022; Izmailov et al., 2021b), noise-contrastive priors (Hafner et al., 2020), and input-dependent priors for domain generalization (Izmailov et al., 2021a; Rudner et al., 2023). Data-driven priors, derived from relevant context data, have been shown to improve group robustness (Rudner et al., 2024), clinical decision-making (Lopez et al., 2023), and out-of-distribution molecular and protein design (Klarner et al., 2024). Furthermore, function-space priors (Sun et al., 2019; Rudner et al., 2022a; Rudner et al., 2022b; Klarner et al.,

2023; Qiu et al., 2023b), sparsity-promoting weight-space priors (Carvalho et al., 2009; Ghosh et al., 2018), structured configuration priors (Louizos et al., 2016), and priors driven by learning algorithms (Khan et al., 2019; Immer et al., 2021) have been developed. Several studies have also explored formulating prior distributions in reference to related tasks. For example, Bayesian meta-learning (Finn et al., 2018; Rothfuss et al., 2021; Pavasovic et al., 2023) can be viewed as a way to learn priors from data, and empirical prior learning has been investigated in Robbins (1992). Fortuin et al. (2022) investigated learning empirical weight distributions using stochastic gradient descent, Wu et al. (2019) applied a moment-matching technique, and Krishnan et al. (2020) proposed to learn the mean of a Gaussian prior distribution from a relevant dataset. To improve transfer learning, Shwartz-Ziv et al. (2022) proposed a method to learn priors from large datasets using the Stochastic Weight Averaging-Gaussian (SWAG) framework (Maddox et al., 2019), which resulted in state-of-the-art predictive performance on computer vision transfer learning tasks (Harvey et al., 2024).

4.2.2 Transfer Learning with Bayesian Principles

In transfer learning, representations learned from one task are adapted to enhance another task. This approach is foundational in deep learning, where large-scale neural networks are pre-trained on extensive datasets and then fine-tuned for specific tasks (Bommasani et al., 2021; Tran et al., 2022). In computer vision, models with vast pre-trained feature extractors are the default starting point for any image classification task (Dai et al., 2021; Dosovitskiy et al., 2021). Similarly, segmentation and object detection models use ImageNet pre-trained CNN or Transformer backbones combined with other modules (Chen et al., 2018; Ren et al., 2015; Dosovitskiy et al., 2021). In natural language processing, text classification is based on fine-tuned transformer models (Devlin et al., 2019; Howard et al., 2018).

Leveraging auxiliary data through Bayesian modeling can lead to improved generalization and robustness. Xuan et al. (2021) explored transfer with probabilistic graphical models, and Karbalayghareh et al. (2018) presented a theoretical framework for optimal Bayes classifiers informed by several domains. Several studies have used Bayesian tools to exploit auxiliary data across domains. For example, Chandra et al. (2020) proposed to learn from multiple domains using round-robin task sampling with a single-layer neural network. Bayesian continual learning methods update the posterior to accommodate new tasks without neglecting previous ones (Ebrahimi et al., 2019; Kapoor et al., 2021; Nguyen et al., 2018; Tseran et al., 2018; Rudner et al., 2022b) or formulate kernels based on previously trained neural networks for Gaussian process inference (Maddox et al., 2021; Pan et al., 2020; Titsias et al., 2020). Semi-supervised algorithms can blend unlabeled data in Bayesian neural network training pipelines, as shown in various studies (Do et al., 2021; Jean et al., 2018; Ravichandran et al., 2021; Shwartz-Ziv et al., 2024).

4.2.3 Robustness to Subpopulation Shifts

Subpopulation shifts are omnipresent in real-world applications, which has led to a large body of literature concerned with reducing the adverse effects of spurious correlations, attribute shifts, and class shifts on model performance through more robust models.

Ensuring robustness to subpopulation shifts is a longstanding challenge in machine learning. A broad range of methods has been developed to mitigate different types of subpopulation shifts, including *data reweighting* (Idrissi et al., 2022), *data augmentation* (Zhang et al., 2018), *domain-invariant feature learning* (Arjovsky et al., 2020; Li et al., 2018a), and *(sub-)group robustness* methods (Sagawa et al., 2020; Liu et al., 2021; Nam et al., 2022; Zhang et al., 2022; Kirichenko et al., 2023). Several methods designed to achieve high worst-group accuracy build on the distributionally robust optimization (DRO) framework (Rahimian et al., 2022), where worst-case accuracy—instead of average-case accuracy—is explicitly maximized during training (Ben-Tal et al., 2013; Hu et al., 2018; Oren et al., 2019; Zhang et al., 2020). Notably, GroupDRO (Sagawa et al., 2020) has become a standard baseline for group robustness.

Group Labeling Models. In most real-world settings, obtaining group information is expensive and only feasible for a small number of data points. To emulate real-world settings where only a small amount of training data with group information is available, existing methods have used the validation sets of benchmarking datasets to achieve high worst-group accuracy. Nam et al. (2022) and Sohoni et al. (2022) train group labeling models on group-labeled validation data, generate group labels for the training dataset, and then train a second model on the full dataset using the generated group labels. In contrast, our proposed method does not require training a group labeling model and instead only requires reweighting the validation set during group-robustness fine-tuning to obtain state-of-the-art performance, outperforming all group labeling techniques.

Last-Layer Retraining. A particularly simple and efficient approach to improving group robustness in settings with limited group label availability is *deep feature reweighting* (DFR; Kirichenko et al., 2023), which first fine-tunes a neural network using empirical risk minimization (ERM) on a training dataset without group information and then retrains the last layer on a group-balanced reweighting dataset using ERM regularized with an L_1-norm penalty on the difference between the current and the previously fine-tuned last-layer parameters (see also Izmailov et al. (2022), Qiu et al. (2023a), and Le et al. (2023) for follow-up works). This work is notable for its marked simplicity and low complexity compared to related methods, as it only requires last-layer retraining on a validation set to reach state-of-the-art performance on some benchmarks and competitive performance on other benchmarks. We present results for our method in the setting where we only retrain the final layer with a

validation set and show that it outperforms DFR on a majority of benchmarking tasks. While Kirichenko et al. (2023) showed that DFR leads to worse performance when group-robustness fine-tuning on the full network, we find that group-robustness fine-tuning the full network with our method leads to significant improvements over last-layer retraining, resulting in state-of-the-art results.

4.3 Training Neural Networks with Data-Driven Priors

In this section, we develop a family of data-driven priors, show how to perform tractable variational inference in neural networks with such priors, and devise uncertainty-aware priors (UAPs) and group-aware priors (GAPs) designed to improve uncertainty quantification and group robustness, respectively.

4.3.1 A Family of Data-Driven Priors

Consider again a parametric observation model $p(y \mid x, \theta; f)$, and let the mapping f be defined by $f(\cdot\,; \theta) \stackrel{\text{def}}{=} h(\cdot\,; \theta_h)\theta_L$, where $h(\cdot\,; \theta_h)$ is the post-activation output of the penultimate layer, Θ_L is the set of stochastic final-layer parameters, Θ_h is the set of stochastic non-final-layer parameters, and $\Theta \stackrel{\text{def}}{=} \{\Theta_h, \Theta_L\}$ is the full set of stochastic parameters. We assume access to pre-trained feature parameters, θ_h^*, and context data that encodes useful information about the downstream tasks. We denote a batch of context inputs with corresponding context labels by $x_\text{c} = \{x_1, ..., x_M\}$ and $y_\text{c} = \{y_1, ..., y_M\}$, respectively, and let p_{X_c, Y_c} be a joint distribution over context batches.

To construct a family of data-driven priors, we begin by specifying a *context inference problem*. We consider a Bernoulli random variable $\check{Z}$ denoting whether a given set of neural network parameters induces predictions that exhibit some desired property (e.g., high uncertainty on a set of evaluation points). Furthermore, we define a *context observation model* $\check{p}_{\check{Z} \mid \Theta}(\check{z} \mid \theta; f, p_{X_c, Y_c})$—which denotes the likelihood of observing a yet-to-be-specified outcome $\check{z}$ under $\check{p}_{\check{Z} \mid \Theta}$ given θ and p_{X_c, Y_c}—and specify a *base prior* over the model parameters, $p(\theta)$. For notational simplicity, we will drop the newly introduced subscripts except when needed for clarity. With this setup, we can now define the context inference problem as finding the conditional distribution over neural network parameters that we *would* obtain if we conditioned on the desired property being satisfied. This conditional distribution will serve as our data-driven prior, and by Bayes' Theorem, we can express it as

$$\check{p}(\theta \mid \check{z}; p_{X_c, Y_c}) = \frac{\check{p}(\check{z} \mid \theta; p_{X_c, Y_c}) p(\theta)}{\check{p}(\check{z}; p_{X_c, Y_c})}. \tag{4.1}$$

To define a family of data-driven priors that place high probability density on neural network parameter values that induce predictive functions with reliable uncertainty estimates, we specify a Bernoulli context observation model $\check{p}_{\check{Z} \mid \Theta}$ in

which $\check{Z} = 1$ denotes the outcome of 'achieving reliable uncertainty quantification' and $\check{p}(\check{z} = 1 \,|\, \theta; p_{X_c,Y_c})$ denotes the likelihood of $\check{z} = 1$ given θ and p_{X_c,Y_c}. More specifically, we define

$$\begin{aligned} \check{p}(\check{z} = 1 \,|\, \theta; p_{X_c,Y_c}) &= \exp(-\lambda \mathbb{E}_{p_{X_c,Y_c}}[c(X_c, Y_c, \theta)]) \\ \check{p}(\check{z} = 0 \,|\, \theta; p_{X_c,Y_c}) &= 1 - \check{p}(\check{z} = 1 \,|\, \theta; p_{X_c,Y_c}), \end{aligned} \tag{4.2}$$

where $c : \mathcal{X} \times \mathcal{Y} \times \mathbb{R}^P \to \mathbb{R}_{\geq 0}$ is a *cost function* and $\lambda > 0$ is a scaling parameter. By specifying the outcome $\check{z} = 1$ along with a distribution p_{X_c,Y_c}, we obtain a conditional distribution $\check{p}(\theta \,|\, \check{z}; f, p_{X_c,Y_c})$—the distribution over neural network parameters that we *would* infer if we observed outcome $\check{z} = 1$ under the base prior and the Bernoulli context observation model defined in Equation (4.2). Naturally, the quality (i.e., the usefulness) of this conditional distribution is determined by the quality of the context observation model $\check{p}_{\check{Z} \,|\, \Theta}$, the data, and the prior. As a result, the primary challenge in designing effective uncertainty-aware priors lies in constructing a context observation model—via a cost function c and a context distribution p_{X_c,Y_c}—that is as well-specified as possible. The better specified the context observation model, the more useful the data-driven prior.

4.3.2 Variational Inference in Models with Data-Driven Priors

In this section, we show how to perform variational inference with data-driven priors. We start by specifying a probabilistic model with a data-driven prior,

$$p(y_{\mathcal{D}}, \theta \,|\, x_{\mathcal{D}}, \check{z}; p_{X_c,Y_c}) = \underbrace{p(y_{\mathcal{D}} \,|\, x_{\mathcal{D}}, \theta; f)}_{\text{Likelihood}} \cdot \underbrace{p(\theta \,|\, \check{z}; p_{X_c,Y_c})}_{\text{Data-driven prior}}. \tag{4.3}$$

To perform variational inference in this model and approximate the posterior distribution over the parameters of interest, we begin by defining a variational distribution,

$$q(\theta) \stackrel{\text{def}}{=} q(\theta_h)\, q(\theta_L), \tag{4.4}$$

where $q(\theta_L) = \mathcal{N}(\theta_L; \mu_L, \Sigma_L)$ and $q(\theta_h) = \mathcal{N}(\theta_h; \mu_h, \Sigma_h)$ with learnable variational parameters $\mu \stackrel{\text{def}}{=} \{\mu_h, \mu_L\}$ and $\Sigma \stackrel{\text{def}}{=} \{\Sigma_h, \Sigma_L\}$, and frame the inference problem of finding the posterior $p(\theta \,|\, x_{\mathcal{D}}, y_{\mathcal{D}}, \check{z})$ variationally as

$$\min_{q_\Theta \in \mathcal{Q}} \mathbb{D}_{\mathrm{KL}}(q_\Theta \parallel p_{\Theta \,|\, X_{\mathcal{D}}, Y_{\mathcal{D}}, \check{Z}}), \tag{4.5}$$

where $\mathcal{Q}$ is a mean-field Gaussian variational family. This variational problem can equivalently be expressed as maximizing the variational objective

$$\bar{\mathcal{F}}(q_\Theta) \stackrel{\text{def}}{=} \mathbb{E}_{q_\Theta}[\log p(y_{\mathcal{D}} \,|\, x_{\mathcal{D}}, \Theta; f)] - \mathbb{D}_{\mathrm{KL}}(q_\Theta \parallel p_{\Theta \,|\, \check{Z}}). \tag{4.6}$$

Unfortunately, this variational objective is intractable since the data-driven prior $\check{p}(\theta \mid \check{z}; p_{X_c,Y_c})$ defined in Equation (4.1)—which is required to compute $\mathbb{D}_{\mathrm{KL}}(q_\Theta \parallel p_{\Theta \mid \check{Z}})$—is not in general tractable.

To overcome this intractability, we make the assumption that

$$p(\theta) \stackrel{\text{def}}{=} p(\theta_h)p(\theta_L) \tag{4.7}$$

and take advantage of the properties of the KL divergence to express $\mathbb{D}_{\mathrm{KL}}(q_\Theta \parallel p_{\Theta \mid \check{Z}})$ as

$$\begin{aligned}&\mathbb{D}_{\mathrm{KL}}(q_\Theta \parallel p_{\Theta \mid \check{Z}}) \\ &= \mathbb{E}_{q_{\Theta_h} q_{\Theta_L}}\left[\log \frac{q(\Theta_h)\, q(\Theta_L)}{p(\Theta_h)\, p(\Theta_L)}\right] - \mathbb{E}_{q_{\Theta_h} q_{\Theta_L}}[\log \check{p}(\check{z} \mid \Theta; p_{X_c,Y_c})] + \log \check{p}(\check{z}; p_{X_c,Y_c}),\end{aligned} \tag{4.8}$$

where the intractable log-marginal likelihood $\log \check{p}(\check{z}; p_{X_c,Y_c})$ was factored out as an additive constant independent of any learnable parameters. Using this result, we can obtain a tractable upper bound

$$\begin{aligned}&\mathbb{D}_{\mathrm{KL}}(q_\Theta \parallel p_{\Theta \mid \check{Z}}) \\ &\leq -\mathbb{E}_{q_{\Theta_h} q_{\Theta_L}}[\log \check{p}(\check{z} \mid \Theta; p_{X_c,Y_c})] + \mathbb{D}_{\mathrm{KL}}(q_{\Theta_h} \parallel p_{\Theta_h}) + \mathbb{D}_{\mathrm{KL}}(q_{\Theta_L} \parallel p_{\Theta_L}),\end{aligned} \tag{4.9}$$

where each KL divergence term can be computed analytically, and we can obtain an unbiased estimator of the expected context cost using simple Monte Carlo estimation.

Variational Objective. Since q_{Θ_h} and q_{Θ_L} are both mean-field Gaussian distributions, we can obtain a doubly lower-bounded variational objective

$$\begin{aligned}\mathcal{F}(\mu, \Sigma) \stackrel{\text{def}}{=} &\underbrace{\mathbb{E}_{q_\Theta}[\log p(y_\mathcal{D} \mid x_\mathcal{D}, \Theta; f)]}_{\text{Expected log-likelihood}} - \underbrace{\mathbb{D}_{\mathrm{KL}}(q_{\Theta_h} \parallel p_{\Theta_h})}_{\text{Pre-training regularization}} - \underbrace{\mathbb{D}_{\mathrm{KL}}(q_{\Theta_L} \parallel p_{\Theta_L})}_{\text{Final-layer regularization}} \\ &- \underbrace{\mathbb{E}_{q_\Theta}[\mathbb{E}_{p_{X_c,Y_c}}[\lambda c(X_c, Y_c, \Theta)]]}_{\text{Context regularization}},\end{aligned} \tag{4.10}$$

where the cost function and context distribution are as defined above. We can estimate all expectations in the objective using simple Monte Carlo estimation, giving the final variational objective

$$\begin{aligned}\hat{\mathcal{F}}(\mu, \Sigma) \stackrel{\text{def}}{=} &\frac{1}{S}\sum_{i=1}^{S} \log p(y_\mathcal{D} \mid x_\mathcal{D}, \check{\Theta}(\mu, \Sigma, \boldsymbol{\epsilon}^{(i)}); f) - \mathbb{D}_{\mathrm{KL}}(q_\Theta \parallel p_\Theta) \\ &- \frac{\lambda}{SJ}\sum_{i=1}^{S}\sum_{j=1}^{J} c(x_c^{(j)}, y_c^{(j)}, \check{\Theta}(\mu, \Sigma, \boldsymbol{\epsilon}^{(i)})),\end{aligned} \tag{4.11}$$

where $\check{\Theta}(\mu, \Sigma, \boldsymbol{\epsilon}^{(i)}) \stackrel{\text{def}}{=} \mu + \Sigma \odot \boldsymbol{\epsilon}^{(i)}$ is a reparameterization of $\Theta \in \mathbb{R}^P$ with $\boldsymbol{\epsilon}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_P)$, $x_c^{(j)} \sim p_{X_c}$, and $y_c^{(j)} \sim p_{Y_c|X_c}$ for $i = 1, ..., S$ and $j = 1, ..., J$. This objective is amenable to mini-batch-based optimization via stochastic variational inference (Hoffman et al., 2013).

When maximizing this variational objective, the uncertainty-aware prior explicitly encourages the predictive distribution induced by the learned variational distribution q_Θ to have high uncertainty on the context inputs while taking into account the covariance under the pre-trained features $h(\cdot; \theta_h^*)$ and pulling μ_h towards the pre-trained parameters θ_h^*.

Computational Complexity. For a batch of context points of size M, computing the uncertainty regularizer requires inverting an M-by-M covariance matrix, which limits the size of the context batches. However, we find that in practice, a small context batch size M (roughly 25% of the mini-batch size) is sufficient, and the main computational expense is the forward pass needed to compute the context likelihood—not the matrix inversion. As the context distribution p_{X_c,Y_c} is defined over context batches (i.e., as defined above, each sample $x_c^{(j)}$ and $y_c^{(j)}$ is a randomly sampled batch of size M), we find that setting $J = 1$ for each stochastic gradient descent step is sufficient in practice, which significantly reduces the computational cost of the nested Monte Carlo estimation in Equation (4.11).

Defining a Suitable Context Distribution. In order for a data-driven prior to favor models that generate reliable uncertainty estimates, the context input distribution must be chosen carefully. We find that distributions over context inputs that retain domain-specific structure work particularly well. For example, when performing image classification, specifying a distribution over images that are meaningfully distinct from the training images is more effective than white noise, and when performing text classification, specifying a distribution over sentences that are meaningfully distinct from the training data but make sense grammatically is more effective than random sequences of words.

4.3.3 Specifying Uncertainty-Aware Data-Driven Priors

In this section, we present a specific instantiation of an uncertainty-aware prior for fine-tuning foundation models. To define a data-driven prior $\check{p}(\theta \mid \check{z}; p_{X_c,Y_c})$ that incorporates useful information from the pre-trained parameters θ_h^* and assigns high probability density to parameter values θ that induce models with reliable uncertainty quantification, we need to specify a suitable context likelihood and suitable layer-specific base priors $p(\theta_h)$ and $p(\theta_L)$. For the base priors, we let $p(\theta_h) = \mathcal{N}(\theta_h; \theta_h^*, \tau_h^{-1} I)$, which assigns high probability to parameters θ_h that are close to the pre-trained parameters θ_h^*, and $p(\theta_L) = \mathcal{N}(\theta_L; \mathbf{0}, \tau_L^{-1} I)$.

To define a context observation model that induces a data-driven prior with desirable properties, we specify a cost function c of the form

$$c(x_c, y_c, \theta) \stackrel{\text{def}}{=} \tau \sum_{k=1}^{K} D^2_{\mathcal{M}}([f(x_c;\theta)]_k, m(x_c, y_c)_k, C(x_c)), \tag{4.12}$$

where $\tau > 0$ is a cost scaling parameter, K is the number of output dimensions,

$$D^2_{\mathcal{M}}([f(x_c;\theta)]_k, m(x_c, y_c)_k, C(x_c)) \stackrel{\text{def}}{=} \mathbf{v}_k^\top C(x_c)^{-1} \mathbf{v}_k \tag{4.13}$$

with $\mathbf{v}_k \stackrel{\text{def}}{=} [f(x_c;\theta)]_k - m(x_c, y_c)_k$, is the squared Mahalanobis distance between model predictions $[f(x_c;\theta)]_k$ and an input-dependent distribution with mean $m(x_c, y_c)_k$ and M-by-M covariance matrix $C(x_c)$. To obtain a data-driven prior that assigns high probability density to parameters θ that induce models with reliable uncertainty estimates, we specify a data-dependent mean function, $m(x_c, y_c)_k \stackrel{\text{def}}{=} [y_c]_k$, and a covariance function

$$C(\cdot) \stackrel{\text{def}}{=} s_1 h(\cdot;\theta_h^*) h(\cdot;\theta_h^*)^\top + s_2 I, \tag{4.14}$$

parameterized by pre-trained model parameters θ_h^* and fixed scaling parameters $s_1 > 0$ and $s_2 > 0$, that reflects structure in the pre-trained model representations $h(\cdot;\theta_h^*)$. Finally, we define the context distribution as $p(x_c, y_c) = p(y_c \,|\, x_c) p(x_c)$, where $p(y_c \,|\, x_c) \stackrel{\text{def}}{=} \delta(\{\mathbf{0}, ..., \mathbf{0}\} - y_c)$ and $p(x_c)$ is an empirical distribution constructed from a larger set of (domain- and task-specific) context inputs.[1]

Under this cost function and context distribution, the data-driven prior defined via Equation (4.2), by design, assigns high probability density to parameters θ that induce predictions $f(x_c;\theta)$ that have high predictive uncertainty on the context inputs. If the distribution over context inputs, p_{X_c}, is specified to place high probability density on context batches which contain input points that are meaningfully distinct from the training inputs, then the data-driven prior favors models that exhibit high predictive uncertainty on such meaningfully distinct inputs.

4.3.4 Specifying Group-Aware Data-Driven Priors

We consider classification problems on data $(x, y) \in \mathcal{X} \times \mathcal{Y}$, where we assume that the data consists of several groups (subpopulations) $g \in \mathcal{G}$, which are often defined by a combination of a label $y \in \mathcal{Y}$ and a spurious attribute $a \in \mathcal{A}$ (sometimes called an *environment*). The attribute $a \in \mathcal{A}$ may or may not be available during training. For instance, in CelebA hair color prediction (Liu et al., 2015), the labels are 'blond' and 'non-blond', and the groups are 'non-blond women' (g_1), 'non-blond men' (g_2), 'blond women' (g_3), and 'blond men' (g_4) with proportions 44%, 41%, 14%, and 1%

[1] Defining $p(y_c \,|\, x_c) \stackrel{\text{def}}{=} \delta(y_c)$ implies that under p_{X_c, Y_c}, all context batch samples have $y_c = \mathbf{0}$, and therefore, we effectively have $m(x_c, y_c)_k \stackrel{\text{def}}{=} \mathbf{0}$ for all context batch samples.

of the data, respectively; the group g_4 is the minority group, and gender serves as the attribute (spurious feature).

We assume that the training data comes from a mixture of group-wise distributions $p_{\text{train}} = \sum_{g\in\mathcal{G}} \alpha_g p_g$, where $\alpha \in \Delta_{|\mathcal{G}|}$ (group weights summing to 1) and p_g are distributions over elements of the group $g \in \mathcal{G}$. We speak of *subpopulation shift* when the test data comes from a differently weighted distribution $p_{\text{test}} = \sum_{g\in\mathcal{G}} \beta_g p_g$, where the weighting $\beta \in \Delta_{|\mathcal{G}|}$ is not known at training-time. This leads to the natural goal of maximizing *worst-group test accuracy*, that is, the lowest accuracy across groups $\mathcal{G}$ represented in the test data.

To specify a practically useful group-aware prior $\check{p}(\theta \mid \check{z}; p_{X_c,Y_c})$ that prefers parameters that induce predictions with high group robustness, we need two ingredients: (i) we need to specify the distribution p_{X_c,Y_c} for which observing $\check{z} = \{1, ..., 1\}$ would be most informative, and (ii) we need to specify a cost function c which—for suitably chosen p_{X_c,Y_c}—is a good proxy for group robustness on unknown test points. Both (i) and (ii) are generally challenging and could be tackled with sophisticated methods—for example, by learning tailored generative models or handcrafting complex cost functions. However, to demonstrate the usefulness of group-aware priors, we instead limit ourselves to a fixed, prespecified distribution p_{X_c,Y_c} and a very simple cost function.

First, to specify a useful context distribution p_{X_c,Y_c}, we assume that we have access to at least a small dataset, D_{val}, with group information and simply upsample the dataset to create a distribution

$$p_{X_c,Y_c} = \sum_{g\in\mathcal{G}} \alpha_g p_g \tag{4.15}$$

with $\alpha_g = \tilde{\alpha}_g / \sum_{g\in\mathcal{G}} \tilde{\alpha}_g$ and $\tilde{\alpha}_g = (|\mathcal{D}_{\text{val}}|/|\mathcal{D}^g_{\text{val}}|)^\gamma$, where $|\mathcal{D}_{\text{val}}|$ is the number of data points in $\mathcal{D}_{\text{val}}$, $|\mathcal{D}^g_{\text{val}}|$ is the number of data points from group g in $\mathcal{D}_{\text{val}}$, and $\gamma \geq 1$ is a scaling parameter. The larger the hyperparameter γ, the stronger rare groups will be upweighted. The smaller $|\mathcal{D}^g_{\text{val}}|$ as a fraction of $|\mathcal{D}_{\text{val}}|$, the larger α_g will be.

Second, to specify a useful cost function, we define

$$c(x_c, y_c, \theta) \stackrel{\text{def}}{=} \ell(y_c, f(x_c; \theta + \rho\epsilon(\theta))) \tag{4.16}$$

where $\rho > 0$ is a scaling parameter and ℓ is a cross-entropy loss function for classification tasks and a mean-squared error loss function for regression tasks. $\epsilon(\theta)$ is a worst-case perturbation to the model parameters proposed in Foret et al. (2021) and given by

$$\epsilon(\theta, x_c, y_c) \stackrel{\text{def}}{=}\perp \frac{\nabla_\theta \ell(y_c, f(x_c; \theta))}{\|\nabla_\theta \ell(y_c, f(x_c; \theta))\|_2}, \tag{4.17}$$

where $\perp$ is the `stop_gradient` operator. The intuition for this cost function is simple: we know that achieving low loss tends to correspond to good generalization, but given that the context dataset has—by assumption—few data points from rare groups, generalizing to test points from these groups is difficult. To design a cost function that leads to a data-driven prior with high probability density on parameters that enable good generalization to points from rare groups, we add a worst-case perturbation defined to lead to a maximum increase in the loss, as proposed in Foret et al. (2021).

It is worth noting the implications of using the `stop_gradient` operator in a prior probability density function. Given that the perturbation $\epsilon(\theta, x_c, y_c)$ is a function of θ and differentiable with respect to θ, applying the `stop_gradient` operator affects the gradient of the context regularization term in the objective given in Equation (4.10). However, since we apply the `stop_gradient` operator, $\epsilon(\theta, x_c, y_c)$ is treated as a constant for the purposes of gradient computation on the context regularization term. As a result, the gradients of $\log \check{p}(\theta \mid \check{z}; p_{X_c,Y_c})$ with respect to θ are different when the `stop_gradient` operator is applied, implying that the application of the `stop_gradient` operator implicitly changes the prior probability density function.

To see how the application of the `stop_gradient` operator changes the prior probability density function, we first note that the gradient of the cost function *without* the `stop_gradient` operator in $\epsilon(\theta, x_c, y_c)$, evaluated at parameters θ_t, is given by

$$\begin{aligned}
&\nabla_\theta \ell(y_c, f(x_c; \theta + \rho\tilde{\epsilon}(\theta)))|_{\theta=\theta_t} \\
&= \frac{\mathrm{d}(\theta + \tilde{\epsilon}(\theta))}{\mathrm{d}\theta}\bigg|_{\theta=\theta_t} \nabla_\theta \ell(y_c, f(x_c; \theta))|_{\theta=\theta_t+\tilde{\epsilon}(\theta_t)} \qquad (4.18) \\
&= \nabla_\theta \ell(y_c, f(x_c; \theta))|_{\theta=\theta_t+\tilde{\epsilon}(\theta_t)} + \frac{\mathrm{d}\tilde{\epsilon}(\theta)}{\mathrm{d}\theta}\bigg|_{\theta=\theta_t} \nabla_\theta \ell(y_c, f(x_c; \theta))|_{\theta=\theta_t+\tilde{\epsilon}(\theta_t)}, \qquad (4.19)
\end{aligned}$$

where

$$\tilde{\epsilon}(\theta) \stackrel{\text{def}}{=} \frac{\nabla_\theta \ell(y_c, f(x_c; \theta))}{\|\nabla_\theta \ell(y_c, f(x_c; \theta))\|_2}. \qquad (4.20)$$

In contrast, *with* the `stop_gradient` operator in $\epsilon(\cdot)$, the gradient of the cost function, evaluated at θ_t, is given by

$$\nabla_\theta \ell(y_c, f(x_c; \theta + \rho\epsilon(\theta, x_c, y_c)))|_{\theta=\theta_t} = \nabla_\theta \ell(y_c, f(x_c; \theta))|_{\theta=\theta_t+\epsilon(\theta_t)}, \qquad (4.21)$$

that is, it does not contain the term $\frac{\mathrm{d}\tilde{\epsilon}(\theta)}{\mathrm{d}\theta}|_{\theta=\theta_t} \nabla_\theta \ell(y_c, f(x_c; \theta))|_{\theta=\theta_t+\tilde{\epsilon}(\theta_t)}$.

In order to obtain this gradient when using the perturbation function $\tilde{\epsilon}(\theta)$ (which does not use the `stop_gradient` operator), we need to include an additive term in the cost function whose gradient is equal to $\frac{\mathrm{d}\tilde{\epsilon}(\theta)}{\mathrm{d}\theta} \nabla_\theta \ell(y_c, f(x_c; \theta))$. To obtain this

term, all we need to do is to take the integral of $\frac{\mathrm{d}\tilde{\epsilon}(\theta)}{\mathrm{d}\theta}\nabla_\theta \ell(y_c, f(x_c;\theta))$ with respect to θ:

$$A(\theta) \stackrel{\text{def}}{=} -\int \frac{\mathrm{d}\tilde{\epsilon}(\theta)}{\mathrm{d}\theta}\nabla_\theta \ell(y_c, f(x_c;\theta))\,\mathrm{d}\theta. \tag{4.22}$$

This integral exists for all evaluation points for which $\tilde{\epsilon}(\cdot)$ and $\ell(y_c, f(x_c;\cdot))$ are differentiable with respect to θ. While this integral may be analytically intractable, we do not need to compute it unless we want to compute the value of the prior probability density for a given parameter configuration, and since we only need the prior density for optimization, we can simply use the `stop_gradient` operator to compute the desired gradients.

4.4 Empirical Evaluation: Reliable Uncertainty Quantification

In this section, we evaluate the usefulness of uncertainty-aware priors for fine-tuning pre-trained models using uncertainty-based evaluation metrics, including the negative log-likelihood, selective prediction accuracy, calibration, and semantic shift detection AUROC. We find that using uncertainty-aware priors leads to significant empirical improvements in uncertainty quantification on a diverse set of computer vision and natural language classification tasks.

4.4.1 Datasets

Training Data. We evaluate our methods on the CIFAR-10 (Krizhevsky et al., 2009), CIFAR-100 (Krizhevsky et al., 2009), and Flowers102 (Nilsback et al., 2008) computer vision datasets, and on the MultiNLI and QNLI natural language datasets (Wang et al., 2018).

Semantic Shift Data. For CIFAR-10 and CIFAR-100, we use the SVHN dataset as an example of semantic shift. For Flowers102, we use the Plantae subset from the iNaturalist dataset (Van Horn et al., 2018) as an example of semantic shift. For QNLI, we use the MultiNLI dataset, and for MultiNLI, we use the Emotions dataset (Saravia et al., 2018) as an example of semantic shift.

Context Distributions. For all image classification tasks, we sample uniformly from the ImageNet dataset (Deng et al., 2009) to construct the context distribution. For MultiNLI and QNLI, we sample uniformly from the MathQA dataset (Amini et al., 2019) as the context.

For illustrative examples of training, semantic shift, and context data points, see Table 4.1.

4.4.2 Experiment Setup

Models. We use a pre-trained ResNet-50 (He et al., 2016) for image classification and a pre-trained BERT-base (uncased) model (Devlin et al., 2019) for text classification tasks.

Training Details. For training details, see Appendix B.1.

Baselines. The default—and most commonly used—method for fine-tuning pre-trained neural networks is empirical risk minimization (ERM) (Chen et al., 2020; Bardes et al., 2022). In addition to ERM, we also compare our method to the *Pre-train Your Loss* (PTYL) method by Shwartz-Ziv et al. (2022), the state of the art for Bayesian transfer learning from pre-trained models. For implementation details, see Appendix B.1.

4.4.3 Accuracy and Negative Log-Likelihood

While having reliable uncertainty estimates is a crucial component of trustworthy models, it is important that efforts to improve uncertainty quantification do not compromise model accuracy. For this reason, we begin by evaluating predictive accuracies and negative log-likelihoods obtained with different methods.

Results. We report the accuracy and negative log-likelihood (NLL) in Tables 4.2 and 4.6 and Figure 4.1. We find that training with uncertainty-aware priors results in similar or improved predictive accuracy and a consistent improvement in the negative log-likelihood, reflecting improved uncertainty quantification without deterioration in generalization.

4.4.4 Selective Prediction

We evaluate selective prediction accuracy across datasets by computing the average selective prediction accuracy on data with an even mixture of in-domain and out-of-distribution samples. In Table 4.3 and Figure 4.2, we show the selective prediction accuracy. The quality of the uncertainty estimates determines the accuracy of the predicted inputs, where in-domain samples are compared to their true label, and any out-of-distribution samples are marked as incorrect. We find that our method consistently outperforms the baseline methods across all downstream tasks, both for image and language classification tasks.

Table 4.1: Training input, context, and semantic shift examples.

Training Input Examples	Context Point Examples	Semantic Shift Examples
CIFAR-10  **Label:** Dog	**ImageNet**  **Label:** Maltese (Dog)	**SVHN**  **Label:** 19
CIFAR-100  **Label:** Sunflower	**ImageNet**  **Label:** Maltese (Dog)	**SVHN**  **Label:** 19
Flowers102  **Label:** Common Dandelion	**ImageNet**  **Label:** Maltese (Dog)	**iNaturalist (Plantae)**  **Label:** Plantae
MultiNLI **Premise**: He started slowly back to the bunkhouse. **Hypothesis**: He returned slowly to the bunkhouse. **Label:** Neutral	**MathQA** **Problem**: the banker ' s gain of a certain sum due 3 years hence at 10 % per annum is rs . 36 . what is the present worth ?	**Emotions** I feel so cold
QNLI **Question**: What was the port known as prior to the Swedish occupation of St. Barts? **Sentence**: Earlier to their occupation, the port was known as "Carénage". **Label:** Not Entailment	**MathQA** **Problem**: the banker ' s gain of a certain sum due 3 years hence at 10 % per annum is rs . 36 . what is the present worth ?	**MultiNLI** **Premise**: He started slowly back to the bunkhouse. **Hypothesis**: He returned slowly to the bunkhouse. **Label:** Neutral

Table 4.2: Negative Log-Likelihood. Lower values are better. Our method consistently achieves the best negative log-likelihood compared to other methods. The mean and standard error are estimated from five trials.

Method	CIFAR-10	CIFAR-100	Flowers102
ERM	0.17±0.01	0.67±0.02	0.48±0.02
PTYL	**0.11**±0.01	0.68±0.01	0.47±0.01
Ours	**0.11**±0.01	**0.62**±0.01	**0.45**±0.02

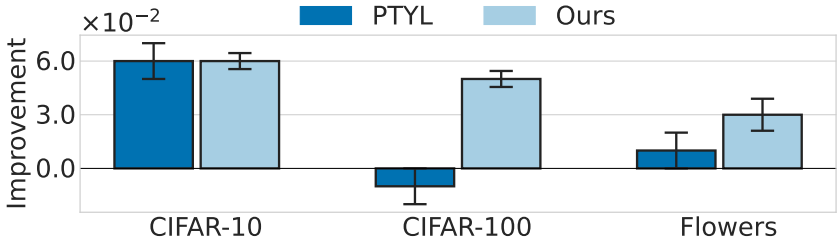


Figure 4.1: Improvement in Negative Log-Likelihood. We plot the difference in the mean negative log-likelihood between ERM and other methods. The mean and standard error are estimated from five trials.

Table 4.3: Selective Prediction Accuracy (in %). Higher values are better. Our method consistently achieves the best compared to other methods. The mean and standard error are estimated from five trials.

Method	CIFAR-10	CIFAR-100	Flowers102
ERM	76.46±0.27	71.99±1.01	78.87±0.62
PTYL	82.48±0.92	72.10±1.69	77.88±0.34
Ours	**83.42**±0.03	**77.61**±0.10	**79.68**±0.08

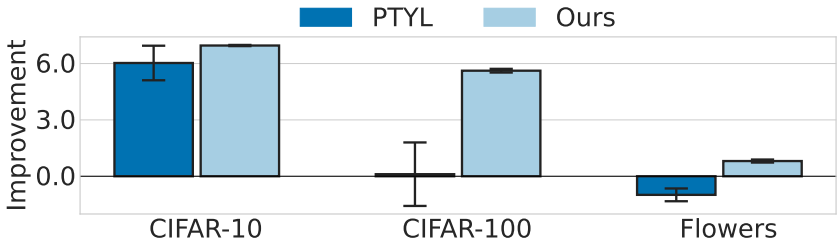


Figure 4.2: Improvement in Selective Prediction Accuracy (in %). We plot the difference in the mean selective accuracy between ERM and other methods. The mean and standard error are estimated from five trials.

Table 4.4: Expected Calibration Error (ECE). Lower values are better. Our method consistently achieves the best ECE compared to other methods. The mean and standard error are estimated from five trials.

Method	CIFAR-10	CIFAR-100	Flowers102
ERM	3.42±0.13	8.61±0.26	11.00±0.47
PTYL	3.10±0.27	8.65±0.56	10.99±0.64
Ours	**1.68**±0.08	**7.22**±0.20	**10.07**±0.18

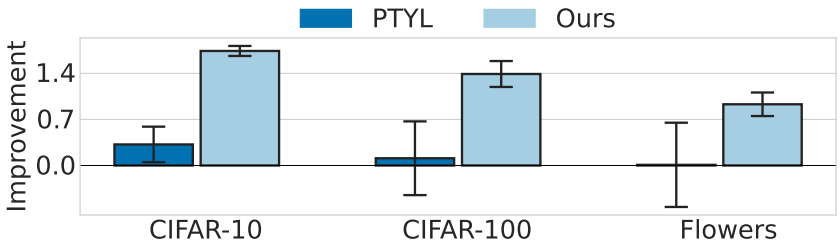


Figure 4.3: Improvement in ECE. Lower values are better. We plot the difference in the mean ECE between ERM and other methods. The mean and standard error are estimated from five trials.

Table 4.5: Semantic Shift Detection AUROC (in %). Higher values are better. Our method consistently achieves the best AUROC compared to other methods. The mean and standard error are estimated from five trials.

Method	CIFAR-10	CIFAR-100	Flowers102
ERM	94.96±0.72	86.57±1.21	95.68±1.85
PTYL	96.94±1.92	88.23±2.98	92.30±0.57
Ours	**99.86**±0.04	**98.93**±0.22	**97.81**±0.10

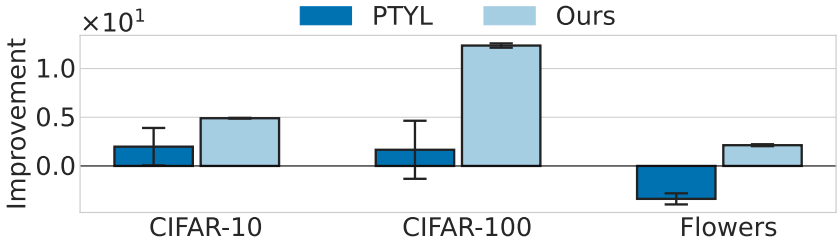


Figure 4.4: Improvement in Semantic Shift Detection (in %). We plot the difference in the mean detection AUROC between ERM and other methods. The mean and standard error are estimated from five trials.

Table 4.6: Predictive Accuracy (in %). Our method achieves competitive accuracy across modalities and datasets. The mean and standard error are estimated from five trials.

(a) Computer Vision Tasks

Method	CIFAR-10	CIFAR-100	Flowers102
ERM	96.45±0.08	**85.76**±0.20	89.64±0.24
PTYL	**97.35**±0.34	**85.82**±**0.23**	89.73±0.51
Ours	**97.19**±0.08	**85.69**±0.13	**90.35**±0.18

(b) Language Tasks

Method	QNLI	MultiNLI
ERM	91.31±0.10	84.65±0.13
PTYL	91.29±0.11	84.60±0.19
Ours	91.28±0.13	84.64±0.06

4.4.5 Model Calibration

We evaluate model calibration on in-domain test data. In Table 4.4 and Figure 4.3, we see that our method significantly improves model calibration (i.e., has lower expected calibration error) on all downstream image tasks. We note that this improvement in performance is dependent on the uncertainty-aware fine-tuning prior, as the prior used by PTYL does not see similar benefits for calibration. For language tasks such as QNLI and MultiNLI, we find our method leads to slight improvements in calibration compared to the ERM baseline (see Appendix B.2).

4.4.6 Semantic Shift Detection

We evaluate uncertainty-based semantic shift detection in vision and language data. Semantic shift in natural language can be particularly subtle, and semantic shifts may occur unexpectedly at deployment, which can lead to catastrophic failures. To assess the reliability of semantic shift detection in language data, we evaluate our method on two different language datasets, the MultiNLI and QNLI datasets. To assess the reliability of semantic shift detection in vision data, we evaluate our method on three different vision datasets, the CIFAR-10, CIFAR-100, and Flowers102 datasets. As can be seen in Table 4.5 and Figure 4.4, we find that our method improves semantic shift detection by a significant margin across all computer vision datasets, and we are able to achieve an AUROC of above 96% for all tasks. This indicates that uncertainty-aware fine-tuning priors enable models to successfully separate in-domain data from semantically shifted data using only uncertainty values at a much higher rate than existing methods. Additionally, as can be seen in Table 4.7, uncertainty-aware fine-tuning priors significantly improve models' ability to detect semantic shifts in natural language, as evidenced by both the selective prediction accuracy and the semantic shift detection AUROC relative to the ERM baseline and PTYL.

Table 4.7: Semantic Shift Detection and Selective Prediction. Our method significantly outperforms ERM at selective prediction with in-domain and semantically shifted data (Selective Acc.) and at uncertainty-based semantic shift detection (Detection AUROC). The mean and standard error are estimated from five trials.

(a) Multi-Genre NLI (MultiNLI)

Method	Selective Acc. (↑)	Det. AUROC (↑)
ERM	72.18±0.42	81.15±0.42
PTYL	73.17±0.23	84.27±0.35
Ours	**79.71**±0.10	**96.90**±0.34

(b) Question-Answering NLI (QNLI)

Method	Selective Acc. (↑)	Det. AUROC (↑)
ERM	75.41±0.13	94.28±0.59
PTYL	75.88±0.15	94.74±0.41
Ours	**76.34**±0.13	**98.17**±0.72

4.5 Empirical Evaluation: Group Robustness

In this section, we evaluate the usefulness of group-aware priors in improving the group robustness of neural network models. We start by describing subpopulation shifts, types of distribution shifts that can cause a lack of group robustness. We then describe the datasets and baselines used in our comparison, as well as our experimental setup. We evaluate our method using both full-network and final-layer-only training, discuss the components that go into group-aware priors, and provide ablation studies to examine their impact.

4.5.1 Subpopulation Shifts

Following the shift taxonomy proposed in Yang et al. (2023), we consider the following three types of subpopulation shifts.

Spurious correlations are present when a label y is correlated with an attribute a in the training distribution but not in the test distribution. For instance, it was shown that thoracic X-ray images contain spurious correlations between labels and attributes such as patient age, scanning position, or text font, which may not be present in the test data (Heaven, 2021; DeGrave et al., 2021).

An *attribute shift* is present when the distribution of an attribute a for a specific label y and context combination differs between the training and test distributions. For instance, MultiNLI has a high prevalence of negation words ('no', 'never')—one of the attributes.

A *class shift* is present when the distribution of classes differs between the training and test distributions. For example, in the CelebA dataset, the 'blond' class in the training set constitutes 15% of the training examples but a larger fraction in the test set.

4.5.2 Datasets

We evaluate our method on both image classification and text datasets that are commonly used to benchmark the performance of group robustness methods.

Waterbirds. Waterbirds is a binary image classification dataset generated synthetically by combining images of birds from the CUB dataset (Wah et al., 2011) and backgrounds from the Places dataset (Zhou et al., 2018); the classes are land- and waterbirds. 73% of images correspond to landbirds on land (the majority group), 22% are waterbirds on water, 4% are landbirds on water, and 1% are waterbirds on land (the minority group). The distribution of backgrounds (land/water) on the validation and test sets is balanced.

CelebA. We consider a binary classification problem ('blond vs. non-blond hair color') with gender serving as the spurious feature. 94% of images with the blond labels show females. Models were trained on pixel intensities to predict binary 'blond vs. not-blond' labels. No individual face characteristics, landmarks, keypoints, facial mapping, metadata, or any other information was used for training.

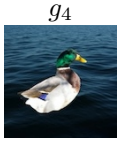
MultiNLI. MultiNLI is a text classification problem where the task is to classify the relationship between a given pair of sentences as a contradiction, entailment, or neither. In this dataset, the presence of negation words (e.g., 'never') in the second sentence is spuriously correlated with the 'contradiction' class.

For illustrative examples and dataset details, see Tables 4.8 and 4.9.

4.5.3 Experiment Setup

Models. For all experiments, we import a pre-trained model and fine-tune all layers of the network using ERM on the target training dataset. For vision datasets, we fine-tune a ResNet-50 (He et al., 2016) pre-trained on ImageNet (Russakovsky et al., 2015) for 30 epochs. For language datasets, we use pre-trained BERT (Devlin et al., 2019) and fine-tune for five epochs.

Table 4.8: Overview of Vision Datasets Used for Evaluating Group Robustness. The images shown below are illustrative only (i.e., they differ from the actual images in the datasets used in this chapter) and are taken from https://unsplash.com (for birds) and https://www.istockphoto.com (for people) under an unlimited, perpetual, nonexclusive, worldwide license.

Waterbirds				
	g_1	g_2	g_3	g_4
Example Image				
Group Description	Landbird on Land	Landbird on Water	Waterbird on Land	Waterbird on Water
Class Label	0	0	1	1
Attribute Label	0	1	0	1
Group Label	0	1	2	3
# Training Data	3,498	184	56	1,057
# Validation Data	467	466	133	133
# Test Data	2,255	2,255	642	642

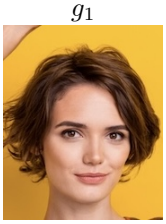
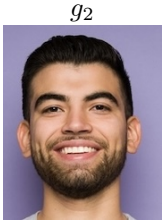
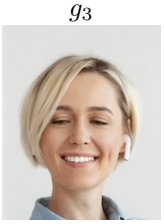
CelebA				
	g_1	g_2	g_3	g_4
Example Image				
Group Description	Non-blond Woman	Non-blond Man	Blond Woman	Blond Man
Class Label	0	0	1	1
Attribute Label	0	1	0	1
Group Label	0	1	2	3
# Training Data	71,629	66,874	22,880	1,387
# Validation Data	8,535	8,276	2,874	182
# Test Data	9,767	7,535	2,480	180

Table 4.9: Overview of Language Datasets Used for Evaluating Group Robustness.

Multi-Genre Natural Language Inference (MultiNLI) corpus			
	g_1	g_2	g_3
Example Text	(P): *if residents are unhappy, they can put wheels on their homes and go someplace else, she said.* (H): *residents are stuck here but they can't go anywhere else.*	(P): *within this conflict of values is a clash about art.* (H): *there is no clash about art.*	(P): *there was something like amusement in the old man's voice.* (H): *the old man showed amusement.*
Group Description	Contradiction without Negations	Contradiction with Negations	Entailment without Negations
Class Label	0	0	1
Attribute Label	0	1	0
Group Label	0	1	2
# Training Data	57,498	11,158	67,376
# Validation Data	22,814	4,634	26,949
# Test Data	34,597	6,655	40,496
	g_4	g_5	g_6
Example Text	(P): *in 1988, the total cost for the postal service was about $36.* (H): *the postal service cost us citizens almost nothing in the late 80's.*	(P): *yeah but even cooking over an open fire is a little more fun isn't it.* (H): *i like the flavour of the food.*	(P): *that's not too bad.* (H): *it's better than nothing.*
Group Description	Entailment with Negations	Neutral without Negations	Neutral with Negations
Class Label	1	2	2
Attribute Label	1	0	1
Group Label	3	4	5
# Training Data	1,521	66,630	1,992
# Validation Data	613	26,655	797
# Test Data	886	39,930	1,148

Table 4.10: Worst-group and average accuracy on the test set of our method against a variety of other baselines in recent literature. We follow Sagawa et al. (2020) and reweight the test accuracy for each group based on their proportion in the training data. The Group Info column details whether a method uses group labels in the training and validation dataset and whether it uses an auxiliary group labeling model. Accuracies of our method are estimated over ten trials. For a description of the baseline methods, see Section 4.5.3. We report the standard error of the mean, which sometimes requires adjusting the error bars of other baselines. The best-performing non-ERM method is highlighted in gray and bolded, and the second-best-performing method is only bolded.

Method	Group Info			Waterbirds		CelebA		MultiNLI	
	Tr.	Val.	Aux.	Worst	Average	Worst	Average	Worst	Average
ERM	N	N	N	74.9±1.0	98.1±0.0	46.9±1.3	95.3±0.0	65.9±0.1	82.8±0.0
JTT	N	Y	Y	86.7	93.3	81.1	88.0	72.6	78.6
CnC	N	Y	Y	88.5±0.2	90.9±0.1	88.8±0.5	89.9±0.3	—	—
SSA	N	Y	Y	89.0±0.3	92.2±0.5	89.8±0.8	**92.8±0.1**	76.6±0.4	79.9±0.5
DFR	N	Y	N	92.9±0.1	94.2±0.2	88.3±0.5	91.3±0.1	74.7±0.3	**82.1±0.1**
SUBG	Y	Y	N	89.1±0.5	—	85.6±1.0	—	68.9±0.4	—
G-DRO	Y	Y	N	91.4	93.5	88.9	**92.9**	**77.7**	81.4
GAP Last Layer	N	Y	N	**93.2±0.2**	**94.6±0.2**	**90.2±0.3**	91.7±0.2	74.3±0.2	81.9±0.0
GAP All Layers	N	Y	N	**93.8±0.1**	**95.6±0.1**	**90.2±0.3**	91.5±0.1	**77.8±0.6**	**82.5±0.1**

Training Details. We assume that the validation dataset $\mathcal{D}_{\text{val}}$ is group-annotated, and we use 85% of it to sample the context distribution $p_{\hat{X},\hat{Y}}$; the remaining 15% are reserved for hyperparameter tuning. The total size of the validation datasets is 19,867 (out of 202,599) for CelebA, 1,199 (out of 11,788) for Waterbirds, and 82,462 (out of 412,349) for MultiNLI. In all cases, we heavily upweight the minority groups in the robust regularization term with $\gamma = 4, 1.5$, and 2 for Waterbirds, CelebA, and MultiNLI, respectively.

Baselines. We consider seven baseline methods that make different assumptions about the availability of group attributes at training time. Empirical risk minimization (ERM) represents conventional training without any procedures for improving worst-group accuracy. Just Train Twice (JTT; Liu et al., 2021) is a method that detects the minority group examples on training data, only using group labels on the validation set to tune hyperparameters. Correct-n-Contrast (CnC; Zhang et al., 2022) detects the minority group examples similarly to JTT and uses a contrastive objective to learn representations robust to spurious correlations. Group DRO (Sagawa et al., 2020) uses group information to train on a worst-group loss objective and is a commonly used baseline. Deep Feature Reweighting (DFR; Kirichenko et al., 2023) uses a network fine-tuned with ERM and performs last-layer feature reweighting. SUBG (Idrissi et al., 2022) is carefully tuned ERM on a random subset of the data where the groups are equally represented. Finally, Spread Spurious Attribute (SSA;

Nam et al., 2022) attempts to fully exploit the group-labeled validation data with a semi-supervised approach that propagates the group labels to the training data.

All baselines use pre-trained models (a ResNet-50, pre-trained on ImageNet1K for Waterbirds and CelebA, and a pre-trained BERT model for MultiNLI), and all of them fine-tune the entire network except for DFR, where fine-tuning is only performed on the last layer (after full-parameter ERM training).

4.5.4 Average and Worst-Group Predictive Accuracy

Table 4.10 presents our results. Using a group-aware prior (GAP) results in state-of-the-art performance for all three benchmarking tasks, demonstrating the value of optimizing with even a very simple data-driven prior. In particular, on Waterbirds and MultiNLI, GAP improves both worst-group and average accuracy even when compared to methods that require training data with group labels or use an auxiliary model to create attribute labels. For CelebA, GAP improves the worst-group accuracy over all baselines, with only marginally lower average accuracy.

Perhaps most remarkably, we achieve state-of-the-art and close-to-state-of-the-art performance even when only retraining the last layer using group-labeled validation data. In particular, GAP applied to the last layer outperforms DFR, another method that only requires last-layer refitting, on the Waterbirds and CelebA tasks, and is competitive with DFR on MultiNLI.

4.5.5 Ablation and Sensitivity Studies

While the group-aware prior constructed in Section 4.3.4 is relatively simple, it adds additional degrees of freedom. Most notably, it involves a scaled parameter perturbation, as can be seen in the robust regularizer of Equation (4.10), meant to favor flatter minima and better generalization to minority groups. Furthermore, the expected cost function is computed under a context distribution, which we construct by upweighting minority groups in the data as per Equation (4.15). Below, we show the impact of these design choices in two ablation studies (for further details, see Appendix B.3).

Ablation on Parameter Perturbation. Table 4.11 compares the performance of GAP (last-layer retraining only) with and without the perturbation of the parameters in the robustness regularization term. As expected, optimizing against a worst-case perturbation in the parameters leads to a small decrease in average accuracy, which is largely compensated for by a significant gain in worst-group accuracy.

Table 4.11: Ablation on Adversarial Perturbation. We ablate the scaling parameter ρ in the GAP robustness regularizer (last-layer retraining only). The table shows worst-group and average accuracies. GAP uses $\rho = 0.15$ for Waterbirds and CelebA and $\rho = 0.1$ for MultiNLI. The $\rho = 0$ setting indicates no parameter perturbation. Means and standard errors are estimated from ten trials.

ρ	Waterbirds		CelebA		MultiNLI	
	Worst	Average	Worst	Average	Worst	Average
GAP	93.2±0.2	94.6±0.2	90.2±0.3	91.7±0.2	74.3±0.2	81.9±0.0
0	92.7±0.2	94.8±0.1	83.2±0.8	94.1±0.1	74.0±0.2	82.2±0.0

Table 4.12: Ablation on Context Distribution. We ablate the upweighting strength in the context distribution $p_{\hat{X},\hat{Y}}$, for exponential upweighting schemes with $\gamma \in \{1, 2, 4\}$ on the Waterbirds dataset using GAP (last-layer retraining). The table shows worst-group and average accuracies. Means and standard errors are estimated from three trials for $\gamma \in \{1, 2\}$ and ten trials for $\gamma = 4$.

Upweight Strength	**Waterbirds**	
	Worst	Average
Balanced ($\gamma = 1$)	90.8±0.6	95.8±0.4
Moderate ($\gamma = 2$)	93.1±0.5	95.1±0.3
Strong ($\gamma = 4$)	**93.2**±0.2	94.6±0.2

Ablation on Context Distribution. We have designed the context distribution $p_{\hat{X},\hat{Y}}$ to upweight the minority groups (see Equation (4.15) and the paragraph following it). In Table 4.12, we compare different exponential weighting schemes from a completely group-balanced setting ($\gamma = 1$) to very strong upweighting ($\gamma = 4$) for Waterbirds (for last-layer retraining). Stronger upweighting of minority groups beyond the balanced setting is beneficial for improved worst-case accuracy.

4.6 Discussion

Reliable uncertainty quantification and group robustness are key ingredients for creating trust in machine learning systems.

We showed that fine-tuning pre-trained models with data-driven priors consistently and, in many cases, significantly improves uncertainty quantification and group robustness across modalities and datasets. Unsurprisingly, the extent to which fine-tuning pre-trained models with data-driven priors improves model reliability depends heavily on the design of the context distribution, which governs the input points on which the prior encourages the model to have a desired behavior, and on the choice of prior hyperparameters. Throughout our experiments, we used fairly naive context distributions: When constructing uncertainty-aware priors, we used ImageNet for computer vision tasks and the MathQA dataset for language tasks to

define domain-specific context distributions that are able to endow fine-tuned models with improved uncertainty quantification. Similarly, when designing group-aware priors, we assumed access to only a small amount of data with group information and constructed a prior using off-the-shelf techniques for improving generalization on these points. We suspect that using more carefully tailored domain-specific context distributions is likely to further improve performance.

One of the key advantages of data-driven priors is that they are complementary to other efforts for improving model performance, such as more sophisticated pre-training techniques or alternative architectures, and can be used with any Bayesian inference method that only requires access to an unnormalized prior density (like SGLD, SG-HMC, or the Laplace approximation). We recommend uncertainty-aware priors as a simple, scalable, and probabilistically principled plug-and-play addition to standard fine-tuning routines. This is an important contrast to function-space variational inference, which is highly prescriptive about the objective design and is not easily compatible with alternative techniques for improving predictive performance of neural networks, such as L_2 or L_1 regularization.

Data-driven priors are probabilistically principled, complementary to alternative approaches, versatile, and can be tailored with specific decision-making tasks in mind. They are a natural choice for prediction tasks where (unlabeled) data that can be used to induce desired behaviors is plentiful, such as molecular and protein design and reinforcement learning, among many others.

5

A Variational Formulation of Reinforcement Learning In Infinite-Horizon MDPs

5.1 Introduction

In this chapter, we develop a variational framework for deriving behavior-driven optimal decision rules for sequential decision problems. In particular, we provide a mathematical justification for learned, dynamic discount factors in KL-regularized reinforcement learning, which have been proposed as an empirically useful tool in recently developed reinforcement learning algorithms (Lu et al., 2023; Rafailov et al., 2021), and establish a rigorous foundation for framing modern reinforcement learning methods as probabilistic inference. Although control as inference has gained in popularity, the treatment of infinite-horizon settings in previous works is ad hoc and not probabilistically well-motivated. With this work, we hope to address this shortcoming and provide a clear formulation of control as inference that carefully disambiguates modeling and inference assumptions.

Levine (2018) and Haarnoja et al. (2018a) presented a framework for framing maximum-entropy reinforcement learning as Bayesian inference in probabilistic models over finite-horizon state–action trajectories. However, most modern reinforcement learning problems are not formulated as finite but as *infinite-horizon* problems (Mnih et al., 2013; Silver et al., 2016). To apply their probabilistic formulation of reinforcement learning to infinite-horizon problems, Levine (2018) and Haarnoja et al. (2018a) introduce a fixed discount factor into their formulation post hoc and without providing a probabilistic justification for doing so. In this chapter, we also show that including a (fixed) discount factor as proposed by Levine (2018) and Haarnoja et al.

(2018a) is a special case of a more general probabilistic framing of the problem, leads to a variational formulation with a loose evidence lower bound, and can provably be improved upon by framing Bayesian variational inference in infinite-horizon MDPs as variational inference in a *transdimensional* probabilistic model.

Instead of focusing on reward maximization as a proxy for desired behaviors, we propose a probabilistic approach that frames reinforcement learning in infinite-horizon MDPs as variational Bayesian inference in transdimensional probabilistic models. More specifically, we consider transdimensional probabilistic models—probabilistic models of random size—and define a random variable denoting whether a given state is on an optimal state trajectory. Under this formulation, each time step becomes a discrete goal-reaching problem, making reinforcement learning a problem of achieving a sequence of interconnected goals.

To derive a learning algorithm that allows us to infer a policy that reflects the behavior encoded in desired state trajectories, we frame the problem of finding an optimal policy as computing an approximation to the conditional distribution over state–action trajectories given state–action trajectories that reflect a desired behavior. We formulate a corresponding probabilistic model and derive tractable variational objectives for finite- and infinite-horizon settings. Based on these results, we define a novel Bellman backup operator and show that for tabular settings, the repeated application of the operator converges to an optimal policy and an optimal dynamic discount factor. Building on this result, we show that fixed-discount-factor KL-regularized reinforcement learning objectives are special cases of the dynamic-discount objectives derived here and demonstrate that variationally learned, dynamic discount factors are optimal in KL-regularized reinforcement learning.

5.1.1 Notation

Standard reinforcement learning (RL) addresses reward maximization in a Markov decision process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, p_{\mathbf{S}_0}, p_d, r, \gamma)$ (Sutton et al., 1998; Szepesvári, 2010), where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action space, respectively, $p_{\mathbf{S}_0}$ denotes the initial state distribution, p_d is a state transition distribution, r is an immediate reward function, and γ is a discount factor. To sample trajectories, an initial state is sampled according to $p_{\mathbf{S}_0}$, and successive states are sampled from the state transition distribution $\mathbf{S}_{t+1} \sim p_d(\cdot \,|\, \mathbf{s}_t, \mathbf{a}_t)$ and actions from a policy $\mathbf{A}_t \sim \pi(\cdot \,|\, \mathbf{s}_t)$. We will write $\boldsymbol{\mathcal{T}}_{0:t} = \{\mathbf{S}_0, \mathbf{A}_0, \mathbf{S}_1, \ldots, \mathbf{S}_t, \mathbf{A}_t\}$ to represent a finite-horizon and $\boldsymbol{\mathcal{T}}_0 \stackrel{\text{def}}{=} \{\mathbf{S}_t, \mathbf{A}_t\}_{t=0}^{\infty}$ to represent an infinite-horizon stochastic state–action trajectory, and write the respective trajectory realizations as $\tau_{0:t} = \{\mathbf{s}_0, \mathbf{a}_0, \mathbf{s}_1, \ldots, \mathbf{s}_t, \mathbf{a}_t\}$ and $\tau_0 \stackrel{\text{def}}{=} \{\mathbf{s}_t, \mathbf{a}_t\}_{t=0}^{\infty}$. Given a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and discount factor $\gamma \in (0, 1)$, the objective in reinforcement learning is to find a policy π that maximizes the returns, defined as $\mathbb{E}_{p_\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t)\right]$, where p_π denotes the distribution of states induced by a policy π.

5.2 Behavior-Driven Reinforcement Learning as Variational Inference

Desired behaviors for artificial agents are often abstract and hard to encode into reward functions. However, in practice, it is often easy to represent desired behaviors via demonstrations. Such demonstrations can be thought of as sample state–action trajectories from a distribution over optimal state–action trajectories. In the remainder of this chapter, we will demonstrate how to use variational Bayesian inference to infer an optimal policy from a set of optimal state–action demonstrations.

To start the exposition, we note that for every state in the environment, there exists a desired, or optimal, behavior that an agent *could* take. We denote this optimal behavior for any given state as the set of state–action trajectories τ^{Ω}. Throughout, we will use the index Ω to denote optimality. Hence, for any state $\mathbf{s} \in \mathcal{S}$, assuming the MDP is ergodic and transition dynamics are deterministic, there exists a set of actions that will set an agent on an optimal state–action trajectory, $\mathcal{A}^{\Omega} \stackrel{\text{def}}{=} \{\mathbf{a} \in \mathcal{A} \,|\, \mathbf{s}' \sim p_d(\mathbf{s}' \,|\, \mathbf{s}, \mathbf{a}) : \mathbf{s}' \in \tau^{\Omega}\}$.

If the transition dynamics are stochastic, each state–action pair will have some probability less than one of transitioning the agent onto an optimal state–action trajectory. Denoting the event of a state being in the optimal state–action trajectory by $\mathbf{s} \in \mathcal{S}^{\Omega}$, where $\mathcal{S}^{\Omega} \stackrel{\text{def}}{=} \{\mathbf{s} \in \tau^{\Omega}\}$, we can define a random variable $\xi(\mathcal{S}^{\Omega}) \stackrel{\text{def}}{=} \mathbb{I}\{\mathbf{s}' \in \mathcal{S}^{\Omega}\}$. We then have that $\xi(\mathcal{S}^{\Omega}) = 1$ if the state $\mathbf{s}'$ into which an agent transitioned after taking action $\mathbf{a}$ in state $\mathbf{s}$ is in the optimal trajectory and $\xi(\mathcal{S}^{\Omega}) = 0$ otherwise. The probability of transitioning into a state on the optimal state–action trajectory at time step $t+1$ is then given by

$$\begin{aligned} &\mathbb{P}(\xi_{t+1}(\mathcal{S}^{\Omega}) = 1 \,|\, \mathbf{s}_t, \mathbf{a}_t) \\ &= \int_{\mathcal{S}^{\Omega}} p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t) d\mathbf{s}_{t+1} = \int_{\mathcal{S}} p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)\, \mathbb{I}\{\mathbf{s}_{t+1} \in \mathcal{S}^{\Omega}\} d\mathbf{s}_{t+1}. \end{aligned} \tag{5.1}$$

In other words, the probability of transitioning into a state on the optimal state–action trajectory corresponds to marginalization over the set of optimal states $\mathcal{S}^{\Omega}$. Equation (5.1) is a likelihood function.

Similarly, by the Markov property, the joint probability of transitioning into a state on the optimal state–action trajectory and staying on it from time step 1 to time step $t^{\star} \stackrel{\text{def}}{=} t+1$, given a state–action trajectory, factorizes and is given by

$$\begin{aligned} &\mathbb{P}(\xi_{1:t+1}(\mathcal{S}^{\Omega}) = 1 \,|\, \mathbf{s}_0, \mathbf{a}_0, ..., \mathbf{s}_t, \mathbf{a}_t) \\ &= \mathbb{P}(\xi_1(\mathcal{S}^{\Omega}) = 1 \,|\, \mathbf{s}_0, \mathbf{a}_0) \cdot ... \cdot \mathbb{P}(\xi_{t+1}(\mathcal{S}^{\Omega}) = 1 \,|\, \mathbf{s}_t, \mathbf{a}_t) \\ &= \prod_{t'=0}^{t} \int_{\mathcal{S}^{\Omega}} p_d(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'})\, d\mathbf{s}_{t'+1} = \prod_{t'=0}^{t} \int_{\mathcal{S}} p_d(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'})\, \mathbb{I}\{\mathbf{s}_{t'+1} \in \mathcal{S}^{\Omega}\}\, d\mathbf{s}_{t'+1}, \end{aligned} \tag{5.2}$$

where we start from $t' = 0$ without loss of generality.

5.2.1 Finite-Horizon Behavior-Driven Reinforcement Learning as Variational Inference

First, we consider the finite-horizon setting. This formulation only diverges slightly from prior work but will help us transition to the transdimensional model formulation for the infinite-horizon setting.

With the notion of trajectory-dependent optimality described in the previous section, we can now specify a model over finite-horizon state–action trajectories and $\xi_{1:t+1}(\mathcal{S}^\Omega)$,

$$\begin{aligned} &p(\tilde{\tau}_{0:t}, \xi^\star_{1:t+1}(\mathcal{S}^\Omega)) \\ &\stackrel{\text{def}}{=} p_{\mathbf{S}_0}(\mathbf{s}_0) \prod_{t'=0}^{t} \mathbb{P}(\xi_{t'+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) p(\mathbf{a}_t \,|\, \mathbf{s}_t) \prod_{t'=0}^{t-1} p_d(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) p(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'}), \end{aligned} \tag{5.3}$$

where $\tilde{\boldsymbol{\mathcal{T}}}_{0:t}$ is a state–action trajectory starting at state $\mathbf{S}_0$ and ending at state $\mathbf{S}_t$, $p(\mathbf{a}_t \,|\, \mathbf{s}_t)$ is a conditional action prior, $p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)$ is the environment's state transition distribution, and $\xi^\star_{1:t+1}(\mathcal{S}^\Omega) \stackrel{\text{def}}{=} \{\xi_{t'}(\mathcal{S}^\Omega) = 1\}_{t'=1}^{t+1}$ is the set of events corresponding to transitioning onto an optimal trajectory. By extension, the probability of transitioning onto an optimal state–action trajectory and remaining on it for $t^\star$ time steps given a state and a prior policy is given by the marginal likelihood

$$\begin{aligned} &\mathbb{P}(\xi_{1:t+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_0) \\ &\quad = \iint_{\mathcal{A}^{t+1}\mathcal{S}^t} p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\tilde{\tau}_{0:t} \,|\, \mathbf{s}_0) \left(\prod_{t'=0}^{t} \int_{\mathcal{S}^\Omega} p_d(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) \, d\mathbf{s}_{t'+1} \right) d\mathbf{s}_{1:t} \, d\mathbf{a}_{0:t} \end{aligned} \tag{5.4}$$

$$\text{where} \qquad p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\tilde{\tau}_{0:t} \,|\, \mathbf{s}_0) \stackrel{\text{def}}{=} p(\mathbf{a}_t \,|\, \mathbf{s}_t) \prod_{t'=0}^{t-1} p_d(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) \, p(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'}) \tag{5.5}$$

is a prior distribution over state–action trajectories. Using an indicator function $\mathbb{I}\{\mathbf{s}_{t+1} \in \mathcal{S}^\Omega\}$ denoting whether the next state is on the desired state–action trajectory, the marginal likelihood in Equation (5.4) can equivalently be expressed as

$$\begin{aligned} &\mathbb{P}(\xi_{1:t+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_0) \\ &= \iint_{\mathcal{A}^{t+1}\mathcal{S}^t} p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\tilde{\tau}_{0:t} \,|\, \mathbf{s}_0) \left(\prod_{t'=0}^{t} \int_{\mathcal{S}} p_d(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) \, \mathbb{I}\{\mathbf{s}_{t'+1} \in \mathcal{S}^\Omega\} \, d\mathbf{s}_{t'+1} \right) d\mathbf{s}_{1:t} \, d\mathbf{a}_{0:t}. \end{aligned} \tag{5.6}$$

This marginalization establishes the connection between the full joint distribution in Equation (5.4) and the likelihood of remaining on an optimal state–action trajectory under a state–action trajectory prior and the likelihood function defined in Equation (5.1).

$\mathbb{P}(\xi_{1:t+1}(\mathcal{S}^\Omega) = 1 \mid \mathbf{s}_0)$ is the marginal likelihood of remaining on the optimal state trajectory from time step 1 to time step $t+1$ under the prior policy $p(\mathbf{a}_t \mid \mathbf{s}_t)$ and the dynamics model $p_d(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)$. Using Bayes' Theorem, we could use the marginal likelihood to compute the posterior distribution over state–action trajectories, $p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t} \mid \xi_{1:t+1}}(\cdot \mid \xi^\star_{1:t+1}(\mathcal{S}^\Omega))$. Unfortunately, the marginal likelihood in Equation (5.6) is intractable for all but the simplest probabilistic models.

To infer an approximate posterior distribution over state–action trajectories instead, we express posterior inference as the variational minimization problem

$$\min_{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}} \in \hat{\mathcal{Q}}} \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\cdot) \parallel p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t} \mid \xi_{1:t+1}}(\cdot \mid \xi^\star_{1:t+1}(\mathcal{S}^\Omega))), \tag{5.7}$$

where $\mathbb{D}_{\mathrm{KL}}(\cdot \parallel \cdot)$ is the KL divergence, and $\hat{\mathcal{Q}}$ denotes the variational family over which to optimize. We consider a family of distributions parameterized by a policy π and defined by

$$q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\tilde{\tau}_{0:t}) \stackrel{\text{def}}{=} p_{\mathbf{S}_0}(\mathbf{s}_0)\pi(\mathbf{a}_t \mid \mathbf{s}_t) \prod_{t'=0}^{t-1} p_d(\mathbf{s}_{t'+1} \mid \mathbf{s}_{t'}, \mathbf{a}_{t'})\pi(\mathbf{a}_{t'} \mid \mathbf{s}_{t'}), \tag{5.8}$$

where $\pi \in \Pi$, a family of policy distributions, and where $p_{\mathbf{S}_0}(\mathbf{s}_0) \prod_{t'=0}^{t-1} p_d(\mathbf{s}_{t'+1} \mid \mathbf{s}_{t'}, \mathbf{a}_{t'})$ is the true state transition distribution up to and including the state transition at t. Under this variational family, the inference problem in Equation (5.7) can be equivalently stated as the problem of maximizing an entropy-regularized expected reward function at every time step, where the reward function is given by the log-likelihood of transitioning onto an optimal state–action trajectory given a state–action pair.

Proposition 5.1. *Given $q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\tilde{\tau}_{0:t})$ from Equation (5.8), an optimal state trajectory $\mathcal{S}^\Omega$, and a horizon length $t^\star$, solving Equation (5.7) is equivalent to maximizing the following objective with respect to $\pi \in \Pi$:*

$$\bar{\mathcal{F}}(\pi, \mathcal{S}^\Omega) \stackrel{\text{def}}{=} \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\tilde{\tau}_{0:t})}\left[\sum_{t'=0}^{t} \log \mathbb{P}(\xi_{t'+1}(\mathcal{S}^\Omega) = 1 \mid \mathbf{s}_{t'}, \mathbf{a}_{t'}) - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \mid \mathbf{s}_{t'}) \,||\, p(\cdot \mid \mathbf{s}_{t'}))\right]. \tag{5.9}$$

Proof. See Appendix C.1.1. □

Replacing $\log \mathbb{P}(\xi_{t+1}(\mathcal{S}^\Omega) = 1 \mid \mathbf{s}_t, \mathbf{a}_t)$ with a reward function yields the optimization objectives obtained by Ziebart et al. (2008), Levine (2018), and Haarnoja et al. (2018a):

Corollary 5.1. *The objective in Equation (5.9) corresponds to KL-regularized reinforcement learning with a time-varying reward function given by*

$$r(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega, t) \stackrel{\text{def}}{=} \log \mathbb{P}(\xi_{t+1}(\mathcal{S}^\Omega) = 1 \mid \mathbf{s}_t, \mathbf{a}_t). \tag{5.10}$$

Proof. See Appendix C.1.1. □

Corollary 5.1 illustrates that—starting from Bayesian principles—we arrive at a very general version of the KL-regularized standard reinforcement learning problem and that, intriguingly, standard KL-regularized reinforcement learning can be viewed as performing approximate Bayesian inference with a heuristically defined reward/likelihood function $\mathbb{P}(\xi_{t+1}(\mathcal{S}^\Omega) = 1 \mid \mathbf{s}_t, \mathbf{a}_t)$.

5.2.2 Infinite-Horizon Behavior-Driven Reinforcement Learning as Variational Inference

To derive an infinite-horizon objective, we modify the probabilistic model used above. To represent the possibility that an agent may stay on the optimal state trajectory for *any* number of time steps, that is, for state–action trajectories of varying lengths, we treat the length of the trajectory itself as a random variable, T, and define the model

$$\begin{aligned} &p(\tilde{\tau}_{0:t}, \xi^\star_{1:t+1}(\mathcal{S}^\Omega), t) \\ &\quad \stackrel{\text{def}}{=} p_T(t) p_{\mathbf{S}_0}(\mathbf{s}_0) \prod_{t'=0}^{t} \mathbb{P}(\xi_{t'+1}(\mathcal{S}^\Omega) = 1 \mid \mathbf{s}_{t'}, \mathbf{a}_{t'}) p_d(\mathbf{s}_{t'+1} \mid \mathbf{s}_{t'}, \mathbf{a}_{t'}) p(\mathbf{a}_{t'} \mid \mathbf{s}_{t'}), \end{aligned} \tag{5.11}$$

where $p_T(t)$ is the probability of remaining on the optimal state trajectory for $t+1$ time steps. Since the trajectory length is itself a random variable, the joint distribution in Equation (5.11) is a *transdimensional* distribution defined on $\biguplus_{t=0}^{\infty}\{t\} \times \mathcal{S}^t \times \mathcal{A}^t$ (Hoffman et al., 2009).

Unlike in the fixed-horizon setting, the variational Bayesian inference problem in the infinite-horizon setting corresponds to finding the posterior distribution over both state–action trajectories *and* the length of the optimal state trajectory T conditioned on the desired behavior $\xi^\star_{1:t+1}(\mathcal{S}^\Omega)$. Analogously to the steps above, we can express this inference problem variationally as

$$\min_{q_{\tilde{\boldsymbol{\tau}}_{0:T},T} \in \mathcal{Q}} \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\tau}}_{0:T},T}(\cdot) \parallel p_{\tilde{\boldsymbol{\tau}}_{0:T},T \mid \xi_{1:T+1}}(\cdot \mid \xi^\star_{1:t+1}(\mathcal{S}^\Omega))), \tag{5.12}$$

where t denotes the time step immediately *before* the outcome is achieved, $\mathcal{Q}$ denotes the variational family. Under this variational distribution, we can obtain an unfactorized variational objective that does not in general lend itself to stochastic gradient-based optimization (and off-policy reinforcement learning). The variational objective is given in Proposition 5.4, but we omit it here for brevity.

To obtain a variational objective amenable to stochastic variational inference and off-policy reinforcement learning, we define the variational family as follows:

$$q_{\tilde{\boldsymbol{\tau}}_{0:T},T}(\tilde{\tau}_{0:t}, t) = q_{\tilde{\boldsymbol{\tau}}_{0:T} \mid T}(\tilde{\tau}_{0:t} \mid t) q_T(t),$$

where q_T is a distribution over T in some variational family $\mathcal{Q}_T$ parameterized by

$$q_T(t) = q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0), \tag{5.13}$$

with Bernoulli random variables Δ_t denoting the event of "remaining on the optimal state trajectory from time step 1 to time step $t+1$," we can equivalently express the variational problem in Equation (5.12) recursively in a way that is tractable and amenable to off-policy optimization:

Theorem 5.1. *Let $q_T(t)$ and $q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|T}(\tilde{\tau}_{0:t} \,|\, t)$ be as defined in Equation (5.8) and Equation (5.13), and define a behavior-driven state value function,*

$$V^\pi(\mathbf{s}_t, \mathcal{S}^\Omega; q_T) \stackrel{\text{def}}{=} \mathbb{E}_{\pi(\mathbf{a}_t \,|\, \mathbf{s}_t)} \Big[Q^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_T) \Big] - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \,\|\, p(\cdot \,|\, \mathbf{s}_t)), \tag{5.14}$$

a behavior-driven state–action value function

$$Q^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_T) \stackrel{\text{def}}{=} r(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_\Delta) + q(\Delta_{t+1} = 0)\, \mathbb{E}_{p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)} \Big[V^\pi(\mathbf{s}_{t+1}, \mathcal{S}^\Omega; q_T) \Big], \tag{5.15}$$

and a behavior-driven reward-like function

$$r(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_\Delta) \stackrel{\text{def}}{=} \log \mathbb{P}(\xi_{t+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_t, \mathbf{a}_t) - q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \,\|\, p_{\Delta_{t+1}}). \tag{5.16}$$

Then given an optimal state trajectory $\mathcal{S}^\Omega$,

$$\mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}, T}(\cdot) \,\|\, p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}, T | \xi_{1:t+1}}(\cdot \,|\, \xi^\star_{1:T+1}(\mathcal{S}^\Omega))) = -\,\mathbb{E}_{p(\mathbf{s}_0)}[V^\pi(\mathbf{s}_0, \mathcal{S}^\Omega; q_T)] + C,$$

where $C \stackrel{\text{def}}{=} \log p(\xi^\star_{1:T+1}(\mathcal{S}^\Omega))$ is independent of π and q_T, and hence maximizing $\mathbb{E}_{p(\mathbf{s}_0)}[V^\pi(\mathbf{s}_0, \mathcal{S}^\Omega; q_T)]$ is equivalent to minimizing Equation (5.12) and hence, the following holds:

$$\begin{aligned} & \arg\min_{\pi \in \Pi, q_T \in \mathcal{Q}_T} \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}, T}(\cdot) \,\|\, p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}, T | \xi_{1:T+1}}(\cdot \,|\, \xi^\star_{1:T+1}(\mathcal{S}^\Omega))) \\ = & \arg\max_{\pi \in \Pi, q_T \in \mathcal{Q}_T} \mathcal{F}(\pi, q_T, \mathcal{S}^\Omega) \\ = & \arg\max_{\pi \in \Pi, q_T \in \mathcal{Q}_T} \mathbb{E}_{p(\mathbf{s}_0)}[V^\pi(\mathbf{s}_0, \mathcal{S}^\Omega; q_T)]. \end{aligned}$$

Proof. See Appendix C.1.2. □

Theorem 5.1 tells us that the solution to the variational problem we started out with (Equation (5.12)) is in fact the solution to an infinite-horizon reinforcement learning problem with a reward function determined by the likelihood of transitioning onto an optimal trajectory, a learned, dynamic discount factor, and KL divergence regularization. In Section 5.2.3, we prove that dynamic-discount factor RL is optimal and preferred over fixed discount factors.

5.2.3 Variational Dynamic-Discount Behavior-Driven Policy Iteration

Building on Theorem 5.1, we will now define a dynamic-discount behavior-driven Bellman backup operator and use it to derive a policy iteration theorem for variational, dynamic-discount reinforcement learning. In particular, we define:

Definition 5.1 (Dynamic-Discount Behavior-Driven Bellman Backup Operator)**.** *Given a function $Q : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, define the operator $\mathcal{T}^\pi$ as*

$$\begin{aligned}&\mathcal{T}^\pi Q(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_T) \\ &\quad \stackrel{\text{def}}{=} r(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_\Delta) + q_{\Delta_{t+1}}(\Delta_{t+1} = 0)\, \mathbb{E}_{p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)} \left[V(\mathbf{s}_{t+1}, \mathcal{S}^\Omega; q_T) \right],\end{aligned} \tag{5.17}$$

where $r(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_\Delta)$ is as defined in Theorem 5.1 and

$$V(\mathbf{s}_t, \mathcal{S}^\Omega; q_T) \stackrel{\text{def}}{=} \mathbb{E}_{\pi(\mathbf{a}_t \,|\, \mathbf{s}_t)} \Big[Q(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_T) \Big] - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \,\|\, p(\cdot \,|\, \mathbf{s}_t)). \tag{5.18}$$

This dynamic-discount, behavior-driven Bellman backup operator is identical to the Bellman backup operator for KL-regularized reinforcement learning (Rudner et al., 2021a) except for the learned, dynamic discount factor, $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$.

In tabular settings, repeated application of this Bellman operator will result in an optimal policy and an optimal dynamic discount factor. More specifically, alternating between policy evaluation and optimization of the variational distribution over the state–action trajectory and the trajectory length converges to an optimal policy.

Theorem 5.2. *Assume $|\mathcal{A}| < \infty$ and that the MDP is ergodic.*

1. *Dynamic-Discount Behavior-Driven Policy Evaluation (BDPE): Given a policy π and a function $Q^0 : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, define $Q^{i+1} = \mathcal{T}^\pi Q^i$. Then the sequence Q^i converges to the lower bound in Theorem 5.1.*

2. *Dynamic-Discount Behavior-Driven Policy Improvement (BDPI): The policy*

$$\pi^+ = \arg\max_{\pi' \in \Pi} \Big\{ \mathbb{E}_{\pi'(\mathbf{a}_t \,|\, \mathbf{s}_t)} \Big[Q^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_T) \Big] - \mathbb{D}_{\mathrm{KL}}(\pi'(\cdot \,|\, \mathbf{s}_t) \,||\, p(\cdot \,|\, \mathbf{s}_t)) \Big\} \tag{5.19}$$

and the variational distribution over T recursively defined in terms of

$$\begin{aligned}&q^+(\Delta_{t+1} = 0 \,|\, \mathbf{s}_0; \pi, Q^\pi) \\ &\quad = \sigma \left(\mathbb{E}_{\pi(\mathbf{a}_{t+1} \,|\, \mathbf{s}_{t+1}) p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)} [Q^\pi(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \mathcal{S}^\Omega; q_T)] + \sigma^{-1} \left(p_{\Delta_{t+1}}(\Delta_{t+1} = 0) \right) \right)\end{aligned} \tag{5.20}$$

improve the variational objective. In other words, $V^{\pi^+}(\mathbf{s}_0, \mathcal{S}^\Omega; q_T) \geq V^\pi(\mathbf{s}_0, \mathcal{S}^\Omega; q_T)$ and $V^\pi(\mathbf{s}_0, \mathcal{S}^\Omega; q_T^+) \geq V^\pi(\mathbf{s}_0, \mathcal{S}^\Omega; q_T)$ for all $\mathbf{s}_0 \in \mathcal{S}$.

3. *Alternating between BDPE and BDPI converges to a policy $\pi^\star$ and a variational distribution over T, $q_T^\star$, such that $Q^{\pi^\star}(\mathbf{s}, \mathbf{a}, \mathcal{S}^\Omega; q_T^\star) \geq Q^\pi(\mathbf{s}, \mathbf{a}, \mathcal{S}^\Omega; q_T)$ for all $(\pi, q_T) \in \Pi \times \mathcal{Q}_T$ and any $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$.*

Proof. See Appendix C.1.4. □

An implication of this result is that an optimal policy found via dynamic-discount behavior-driven policy iteration has at least as high a state value in any state as it would under a fixed discount factor. That is, for q_T given by a fixed geometric distribution with parameter γ, the state–action value function satisfies the standard KL-regularized Bellman recursion,

$$Q^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; p_T) \stackrel{\text{def}}{=} \log \mathbb{P}(\xi_{t+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_t, \mathbf{a}_t) + \gamma \, \mathbb{E}_{p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)} \left[V^\pi(\mathbf{s}_{t+1}, \mathcal{S}^\Omega; \pi, p_T) \right], \tag{5.21}$$

and

$$Q^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_T^*) \geq Q^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; p_T), \tag{5.22}$$

for any $(\mathbf{s}_t, \mathbf{a}_t) \in \mathcal{S} \times \mathcal{A}$. In other words, dynamic discount factors are optimal in KL-regularized reinforcement learning and can be justified using the variational Bayesian inference formulation described above.

5.3 Outcome-Driven Reinforcement Learning as Variational Inference

In Section 5.2, we described a variational procedure for learning optimal decision rules from sets of desired behaviors. While this framework provides an appealing formalism for automated learning of behavioral skills, in practice, specifying a likelihood $\mathbb{P}(\xi_{t+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_t, \mathbf{a}_t)$ is challenging and effectively corresponds to designing a reward function. Although reward functions are used across a wide array of reinforcement learning problems, their design is largely based on heuristics, often lacks theoretical grounding, can make effective learning difficult, and may lead to reward misspecification.

Goal-conditioned reinforcement learning (Kaelbling, 1993), which can be considered a special case of the broader class of stochastic boundary value problems (Aly et al., 1974; Goebel et al., 1990), seeks to circumvent the problem of defining a reward function altogether. Under this framework, the objective is for an agent to reach some pre-specified goal state $\mathbf{g} \in \mathcal{S}$ from some initial state $\mathbf{s}_0$.

Following this intuition, we use the framework introduced in Section 5.2 to express goal-conditioned RL probabilistically as inferring a state–action trajectory distribution conditioned on a *desired future outcome* from a given initial state. Analogous

to the results in Section 5.2, we derive a tractable variational objective, a temporal-difference algorithm that provides a shaping-like effect for effective learning, as well as a reward function that captures the semantics of the underlying decision problem and facilitates effective learning.

We demonstrate that, unlike prior works that proposed inference methods for finding policies that achieve desired outcomes (Attias, 2003; Fu et al., 2018; Hoffman et al., 2009; Toussaint et al., 2006), the resulting algorithm, Outcome-Driven Actor–Critic (ODAC), is amenable to off-policy learning and applicable to complex, high-dimensional continuous control tasks over finite and infinite horizons. The resulting variational algorithm can be interpreted as an automatic shaping method, where each iteration learns a reward function that automatically provides dense rewards. In tabular settings, ODAC is guaranteed to converge to an optimal policy, and in non-tabular settings with linear Gaussian transition dynamics, the derived optimization objective is convex in the policy, facilitating easier learning. In high-dimensional and non-linear domains, our method can be combined with deep neural network function approximators to yield a deep reinforcement learning method that does not require manual specification of rewards and leads to good performance on a range of benchmark tasks.

5.3.1 Finite-Horizon Outcome-Driven Reinforcement Learning as Variational Inference

We, again, first consider a simplified warm-up problem, where the desired outcome is to reach a specific state $\mathbf{g} \in \mathcal{S}$ at a specific time step $t^\star$ when starting from initial state $\mathbf{s}_0$. We can think of the starting state $\mathbf{s}_0$ and the desired outcome $\mathbf{g}$ as boundary conditions, and the goal is to learn a stochastic policy that induces a trajectory from $\mathbf{s}_0$ to $\mathbf{g}$. To derive a control law that solves this stochastic boundary value problem, we frame the problem probabilistically as inferring a state–action trajectory posterior distribution *conditioned on the desired outcome and the initial state*. We will show that, by framing the learning problem this way, we obtain an algorithm for learning outcome-driven policies without needing to manually specify a reward function. We consider a model of the state–action trajectory up to and including the desired outcome $\mathbf{g}$,

$$p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t},\mathbf{S}_{t^\star}}(\tilde{\tau}_{0:t}, \mathbf{g} \,|\, \mathbf{s}_0) \stackrel{\text{def}}{=} p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) p(\mathbf{a}_t \,|\, \mathbf{s}_t) \prod_{t'=0}^{t-1} p_d(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) p(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'}),$$

where $t^\star \stackrel{\text{def}}{=} t+1$, $\tilde{\boldsymbol{\mathcal{T}}}_{0:t}$ is the state–action trajectory up to and including t but excluding $\mathbf{S}_0$, $p(\mathbf{a}_t \,|\, \mathbf{s}_t)$ is a conditional action prior, and $p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)$ is the environment's state transition distribution. If the dynamics are simple (e.g., tabular or Gaussian), the posterior over actions can be computed in closed form (Attias, 2003), but we would like to be able to infer outcome-driven posterior policies in any environment,

including those where exact inference may be intractable. To do so, we start by expressing posterior inference as the variational minimization problem

$$\min_{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0} \in \hat{\mathcal{Q}}} \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\cdot \mid \mathbf{s}_0) \parallel p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0,\mathbf{S}_{t^\star}}(\cdot \mid \mathbf{s}_0, \mathbf{g})), \tag{5.23}$$

where $\mathbb{D}_{\mathrm{KL}}(\cdot \parallel \cdot)$ is the KL divergence, and $\hat{\mathcal{Q}}$ denotes the variational family over which to optimize. We consider a family of distributions parameterized by a policy π and defined by

$$q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\tilde{\tau}_{0:t} \mid \mathbf{s}_0) \stackrel{\text{def}}{=} \pi(\mathbf{a}_t \mid \mathbf{s}_t) \prod_{t'=0}^{t-1} p_d(\mathbf{s}_{t'+1} \mid \mathbf{s}_{t'}, \mathbf{a}_{t'}) \pi(\mathbf{a}_{t'} \mid \mathbf{s}_{t'}), \tag{5.24}$$

where $\pi \in \Pi$, a family of policy distributions, and where $\prod_{t=0}^{t-1} p_d(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)$ is the true action-conditional state transition distribution up to but excluding the state transition at $t^\star - 1$, since $\mathbf{S}_{t^\star} = \mathbf{g}$ is observed. Under this variational family, the inference problem in Equation (5.23) can be equivalently stated as the problem of maximizing the following objective with respect to the policy π:

Proposition 5.2. *Given $q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\tilde{\tau}_{0:t} \mid \mathbf{s}_0)$ from Equation (5.24), any state $\mathbf{s}_0$, termination time $t^\star$, and outcome $\mathbf{g}$, solving Equation (5.23) is equivalent to maximizing this objective with respect to $\pi \in \Pi$:*

$$\bar{\mathcal{F}}(\pi, \mathbf{s}_0, \mathbf{g}) \stackrel{\text{def}}{=} \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\tilde{\tau}_{0:t} \mid \mathbf{s}_0)} \left[\log p_d(\mathbf{g} \mid \mathbf{s}_t, \mathbf{a}_t) - \sum_{t'=0}^{t} \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \mid \mathbf{s}_{t'}) \,||\, p(\cdot \mid \mathbf{s}_{t'})) \right]. \tag{5.25}$$

Proof. See Appendix C.1.5. □

A variational problem of this form—which corresponds to finding a posterior distribution over state–action trajectories—can equivalently be viewed as a reinforcement learning problem:

Corollary 5.2. *The objective in Equation (5.25) corresponds to KL-regularized reinforcement learning with a time-varying reward function given by*

$$r(\mathbf{s}_{t'}, \mathbf{a}_{t'}, \mathbf{g}, t') \stackrel{\text{def}}{=} \mathbb{I}\{t' = t\} \log p_d(\mathbf{g} \mid \mathbf{s}_{t'}, \mathbf{a}_{t'}). \tag{5.26}$$

Proof. See Appendix C.1.5. □

Corollary 5.2 illustrates how a reward function *emerges automatically* from a probabilistic framing of outcome-driven reinforcement learning problems where the sole specification is which variable ($\mathbf{S}_{t^\star}$) should attain which value ($\mathbf{g}$). In particular, Corollary 5.2 suggests that we ought to learn the environment's state-transition distribution, and view the log-likelihood of achieving the desired outcome given a state–action pair as a "reward" that can be used for off-policy learning.

Importantly—and unlike in model-based RL—such a transition model would not have to be accurate beyond single-step predictions, as it would not be used for planning (see Appendix C.3). Instead, $\log p_d(\mathbf{g} \mid \mathbf{s}_t, \mathbf{a}_t)$ only needs to be "well shaped," which we expect to happen for commonly used model classes. For example, when the dynamics are linear-Gaussian, using a conditional Gaussian model (Nagabandi et al., 2018) yields a reward function that is quadratic in $\mathbf{S}_{t+1}$, making the objective convex and thus more amenable to optimization.

5.3.2 Infinite-Horizon Outcome-Driven Reinforcement Learning as Variational Inference

Thus far, we have assumed that the time at which the outcome is achieved is given. In many problem settings, we do not know (or care) when an outcome is achieved. In this section, we present a variational inference perspective on achieving desired outcomes in settings where *no reward function* and *no termination time* are given, but only a desired outcome is provided. As in the previous section, we derive a variational objective and show that a principled algorithm and reward function emerge automatically when framing the problem as variational inference.

To derive such an objective, we modify the probabilistic model used in the previous section to model that the time at which the outcome is achieved is not known. As before, we define an "outcome" as a point in the state space, but instead of defining the event of "achieving a desired outcome" as a realization $\mathbf{S}_{t^\star} = \mathbf{g}$ for a known $t^\star$, we define it as a realization $\mathbf{S}_{T^\star} = \mathbf{g}$ for an *unknown* termination time $T^\star$ at which the outcome is achieved. Specifically, we model the distribution over the trajectory and the unknown termination time with

$$p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}, \mathbf{S}_{T^\star}, T \mid \mathbf{S}_0}(\tilde{\tau}_{0:t}, \mathbf{g}, t \mid \mathbf{s}_0) = p_T(t) p_d(\mathbf{g} \mid \mathbf{s}_t, \mathbf{a}_t) p(\mathbf{a}_t \mid \mathbf{s}_t) \prod_{t'=0}^{t-1} p_d(\mathbf{s}_{t'+1} \mid \mathbf{s}_{t'}, \mathbf{a}_{t'}) p(\mathbf{a}_{t'} \mid \mathbf{s}_{t'}), \tag{5.27}$$

where $p_T(t)$ is the probability of reaching the outcome at $t+1$. Since the trajectory length is itself a random variable, the joint distribution in Equation (5.27) is a *transdimensional* distribution defined on $\biguplus_{t=0}^{\infty} \{t\} \times \mathcal{S}^{t+1} \times \mathcal{A}^{t+1}$ (Hoffman et al., 2009).

Unlike in the warm-up, the problem of finding an outcome-driven policy that eventually achieves the desired outcome corresponds to finding the posterior distribution over state–action trajectories *and* the termination time T conditioned on the desired outcome $\mathbf{S}_{T^\star}$ and a starting state. Analogously to Section 5.3.1, we can express this inference problem variationally as

$$\min_{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}, T \mid \mathbf{S}_0} \in \mathcal{Q}} \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}, T \mid \mathbf{S}_0}(\cdot \mid \mathbf{s}_0) \parallel p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}, T \mid \mathbf{S}_0, \mathbf{S}_{T^\star}}(\cdot \mid \mathbf{s}_0, \mathbf{g})), \tag{5.28}$$

where t denotes the time immediately *before* the outcome is achieved, and $\mathcal{Q}$ denotes the variational family. In general, solving this variational problem in closed form is challenging, but by choosing a variational family

$$q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T|\mathbf{S}_0}(\tilde{\tau}_{0:t}, t \,|\, \mathbf{s}_0) = q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T,\mathbf{S}_0}(\tilde{\tau}_{0:t} \,|\, t, \mathbf{s}_0) q_T(t), \tag{5.29}$$

where q_T is a distribution over T in some variational family $\mathcal{Q}_T$ parameterized by

$$q_T(t) = q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0), \tag{5.30}$$

with Bernoulli random variables Δ_t denoting the event of "reaching $\mathbf{g}$ at time t given that $\mathbf{g}$ has not yet been reached by time $t-1$," we can equivalently express the variational problem in Equation (5.28) in a way that is tractable and amenable to off-policy optimization:

Theorem 5.3. *Let $q_T(t)$ and $q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T,\mathbf{S}_0}(\tilde{\tau}_{0:t} \,|\, t, \mathbf{s}_0)$ be as defined before, and define an outcome-driven state value function,*

$$V^\pi(\mathbf{s}_t, \mathbf{g}; q_T) \stackrel{\text{def}}{=} \mathbb{E}_{\pi(\mathbf{a}_t \,|\, \mathbf{s}_t)}[Q^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T)] - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \,\|\, p(\cdot \,|\, \mathbf{s}_t)) \tag{5.31}$$

an outcome-driven state–action value function

$$Q^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T) \stackrel{\text{def}}{=} r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) + q_{\Delta_{t+1}}(\Delta_{t+1} = 0)\, \mathbb{E}_{p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)}[V^\pi(\mathbf{s}_{t+1}, \mathbf{g}; q_T)] \tag{5.32}$$

and an outcome-driven reward-like function

$$r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) \stackrel{\text{def}}{=} q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) - \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \,\|\, p_{\Delta_{t+1}}). \tag{5.33}$$

Then given any initial state $\mathbf{s}_0$ and outcome $\mathbf{g}$,

$$\mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T|\mathbf{S}_0}(\cdot \,|\, \mathbf{s}_0) \,\|\, p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T|\mathbf{S}_0,\mathbf{S}_{T^*}}(\cdot \,|\, \mathbf{s}_0, \mathbf{g})) = -V^\pi(\mathbf{s}_0, \mathbf{g}; q_T) + \log p(\mathbf{g} \,|\, \mathbf{s}_0),$$

where $\log p(\mathbf{g} \,|\, \mathbf{s}_0)$ is independent of π and q_T and hence maximizing $V^\pi(\mathbf{s}_0, \mathbf{g}; \pi, q_T)$ is equivalent to minimizing Equation (5.28).

Proof. See Appendix C.1.6. □

This theorem tells us that the maximizer of $V^\pi(\mathbf{s}_t, \mathbf{g}; q_T)$ is equal to the minimizer of Equation (5.28). In other words, Theorem 5.3 presents a variational objective with dense reward functions defined solely in terms of the desired outcome and the environment dynamics, which we can learn directly from environment interactions. It

further makes precise that the variational objective, $V^\pi(\mathbf{s}_0, \mathbf{g}; q_T)$, is a lower bound on the log-marginal likelihood, that is, $\log p(\mathbf{g} \mid \mathbf{s}_0) \geq V^\pi(\mathbf{s}_0, \mathbf{g}; q_T)$, where

$$\begin{aligned} &V^\pi(\mathbf{s}_0, \mathbf{g}; q_T) \\ &= \mathbb{E}\left[\sum_{t=0}^{\infty}\left(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'}=0)\right)\Big(r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \mid \mathbf{s}_t) \parallel p(\cdot \mid \mathbf{s}_t))\Big)\right], \end{aligned} \tag{5.34}$$

with the expectation taken with respect to the infinite-horizon trajectory distribution $q_{\tilde{\boldsymbol{\mathcal{T}}}_0|\mathbf{s}_0}(\tilde{\tau}_0 \mid \mathbf{s}_0)$. Thanks to the recursive expression of the variational objective, we can find the optimal variational distribution over T as a function of the current policy and Q-function analytically:

Proposition 5.3. *The optimal distribution $q_T^\star$ with respect to Equation (5.31) is*

$$q^\star_{\Delta_{t+1}}(\Delta_{t+1}=0; \pi, Q^\pi) = \sigma\left(\Lambda(\mathbf{s}_t, \pi, q_T, Q^\pi) + \sigma^{-1}\left(p_{\Delta_{t+1}}(\Delta_{t+1}=0)\right)\right), \tag{5.35}$$

where

$$\begin{aligned} &\Lambda(\mathbf{s}_t, \pi, q_T, Q^\pi) \\ &\stackrel{\text{def}}{=} \mathbb{E}_{\pi(\mathbf{a}_{t+1} \mid \mathbf{s}_{t+1}) p_d(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t) \pi(\mathbf{a}_t \mid \mathbf{s}_t)}[Q^\pi(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \mathbf{g}; q_T) - \log p_d(\mathbf{g} \mid \mathbf{s}_t, \mathbf{a}_t)] \end{aligned} \tag{5.36}$$

and $\sigma(\cdot)$ is the sigmoid function, that is, $\sigma(x) = \frac{1}{e^{-x}+1}$ and $\sigma^{-1}(x) = \log \frac{x}{1-x}$.

Proof. See Appendix C.1.7. □

Alternatively, if instead of learning q_T variationally, we fix q_T to the prior p_T, we recover the more conventional fixed-discount-factor objective (Galashov et al., 2019; Haarnoja et al., 2018a; Rudner et al., 2021a):

Corollary 5.3. *Let $q_T = p_T$, and assume that p_T is a geometric distribution with parameter $\gamma \in (0, 1)$. Then the inference problem in Equation (5.28) of finding an outcome-driven variational trajectory distribution simplifies to maximizing the following recursively defined variational objective with respect to π:*

$$\bar{V}^\pi(\mathbf{s}_0, \mathbf{g}; \gamma) \stackrel{\text{def}}{=} \mathbb{E}_{\pi(\mathbf{a}_0 \mid \mathbf{s}_0)}\left[Q(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; \gamma)\right] - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \mid \mathbf{s}_0) \,\|\, p(\cdot \mid \mathbf{s}_0))), \tag{5.37}$$

where

$$\bar{Q}^\pi(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; \gamma) \stackrel{\text{def}}{=} (1-\gamma)\log p_d(\mathbf{g} \mid \mathbf{s}_0, \mathbf{a}_0) + \gamma\, \mathbb{E}_{p_d(\mathbf{s}_1 \mid \mathbf{s}_0, \mathbf{a}_0)}\left[V^\pi(\mathbf{s}_1, \mathbf{g}; \gamma)\right]. \tag{5.38}$$

In the next section, we derive a temporal-difference algorithm and discuss how we can learn the Q-function in Theorem 5.3 using off-policy transitions.

5.3.3 Variational Dynamic-Discount Outcome-Driven Policy Iteration

To develop an outcome-directed off-policy algorithm, we define the following Bellman operator:

Definition 5.2. *Given a function $Q : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, define the operator $\mathcal{T}^\pi$ as*

$$\mathcal{T}^\pi Q(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T) \stackrel{\text{def}}{=} r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) + q_{\Delta_{t+1}}(\Delta_{t+1} = 0)\, \mathbb{E}_{p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)} \left[V(\mathbf{s}_{t+1}, \mathbf{g}; q_T) \right], \tag{5.39}$$

where $r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta)$ is from Theorem 5.3 and

$$V(\mathbf{s}_t, \mathbf{g}; q_T) \stackrel{\text{def}}{=} \mathbb{E}_{\pi(\mathbf{a}_t \,|\, \mathbf{s}_t)} \left[Q(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T) \right] - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \,\|\, p(\cdot \,|\, \mathbf{s}_t)). \tag{5.40}$$

Unlike the standard Bellman operator, the above operator has a varying weight factor $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$, with the optimal weight factor given by Equation (5.35). From Equation (5.35), we see that as the outcome likelihood $p_d(\mathbf{g} \,|\, \mathbf{s}, \mathbf{a})$ becomes large relative to the Q-function, the weight factor automatically adjusts the target to rely more on the rewards.

Below, we show that repeatedly applying the operator $\mathcal{T}^\pi$ (policy evaluation) and optimizing π with respect to Q^π (policy improvement) converges to a policy that maximizes the objective in Theorem 5.3.

Theorem 5.4. *Assume $|\mathcal{A}| < \infty$ and that the MDP is ergodic.*

1. *Outcome-Driven Policy Evaluation (ODPE): Given a policy π and a function $Q^0 : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, define $Q^{i+1} = \mathcal{T}^\pi Q^i$. Then the sequence Q^i converges to the lower bound in Theorem 5.3.*

2. *Outcome-Driven Policy Improvement (ODPI): The policy*

$$\pi^+ = \arg\max_{\pi' \in \Pi} \{ \mathbb{E}_{\pi'(\mathbf{a}_t \,|\, \mathbf{s}_t)} \left[Q^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T) \right] - \mathbb{D}_{\mathrm{KL}}(\pi'(\cdot \,|\, \mathbf{s}_t) \,||\, p(\cdot \,|\, \mathbf{s}_t)) \} \tag{5.41}$$

   and the variational distribution over T defined in Equation (5.35) improve the variational objective, that is, $\mathcal{F}(\pi^+, q_T, \mathbf{s}_0) \geq \mathcal{F}(\pi, q_T, \mathbf{s}_0)$ and $\mathcal{F}(\pi, q_T^+, \mathbf{s}_0) \geq \mathcal{F}(\pi, q_T, \mathbf{s}_0)$ for all $\mathbf{s}_0, \pi, q_T$.

3. *Alternating between ODPE and ODPI converges to a policy $\pi^\star$ and a variational distribution over T, q_T, such that $Q^{\pi^\star}(\mathbf{s}, \mathbf{a}, \mathbf{g}; q_T^\star) \geq Q^\pi(\mathbf{s}, \mathbf{a}, \mathbf{g}; q_T)$ for all $(\pi, q_T) \in \Pi \times \mathcal{Q}_T$ and any $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$.*

Proof. See Appendix C.1.8. □

This result tells us that alternating between applying the outcome-driven Bellman operator in Definition 5.2 and optimizing the bound in Theorem 5.3 using the resulting Q-function, which can equivalently be viewed as expectation maximization, will lead to a policy that induces an outcome-driven trajectory and solves the inference problem in Equation (5.28). This implies that Variational Dynamic-Discount Outcome-Driven Policy Iteration is theoretically at least as good as or better than standard policy iteration for KL-regularized objectives:

Corollary 5.4. *Variational Dynamic-Discount Outcome-Driven Policy Iteration on* $(\pi, q_T) \in \Pi \times \mathcal{Q}_T$ *results in an optimal policy at least as good as or better than any optimal policy attainable from policy iteration on* $\pi \in \Pi$ *alone.*

5.3.4 Outcome-Driven Actor–Critic

We now build on previous sections to develop Outcome-Driven Actor–Critic (ODAC), a practical algorithm that handles large and continuous domains. In such domains, the expectation in the Bellman operator in Definition 5.2 is intractable, and so we approximate the policy π_θ and Q-function Q_ϕ with neural networks parameterized by parameters θ and ϕ, respectively. We train the Q-function to minimize

$$\mathcal{F}_Q(\phi) = \mathbb{E}\left[\left(Q_\phi(\mathbf{s}, \mathbf{a}, \mathbf{g}) - (r(\mathbf{s}, \mathbf{a}, \mathbf{g}; q_\Delta) + q_{\Delta_{t+1}}(\Delta_{t+1} = 0)\, \hat{V}(\mathbf{s}', \mathbf{g}))\right)^2\right], \quad (5.42)$$

where the expectation is taken with respect to $(\mathbf{s}, \mathbf{a}, \mathbf{g}, \mathbf{s}')$ sampled from a replay buffer, $\mathcal{D}$, of data collected by a policy. We approximate the $\hat{V}$-function using a target Q-function $Q_{\bar{\phi}}$: $\hat{V}(\mathbf{s}', \mathbf{g}) \approx Q_{\bar{\phi}}(\mathbf{s}', \mathbf{a}', \mathbf{g}) - \log \pi(\mathbf{a}' \,|\, \mathbf{s}'; \mathbf{g})$, where $\mathbf{a}'$ are samples from the amortized variational policy $\pi_\theta(\cdot \,|\, \mathbf{s}'; \mathbf{g})$. We further assume a uniform prior policy $p(\cdot \,|\, \mathbf{s}_t)$ in all of our experiments. The parameters $\bar{\phi}$ slowly track the parameters of ϕ at each time step via the standard update $\bar{\phi} \leftarrow \tau\bar{\phi} + (1 - \tau)\phi$ (Lillicrap et al., 2016). We then train the policy to maximize the approximate Q-function by performing gradient descent on

$$\mathcal{F}_\pi(\theta) = -\,\mathbb{E}_{\mathbf{s}\sim\mathcal{D}, \mathbf{a}\sim\pi_\theta(\cdot \,|\, \mathbf{s}; \mathbf{g})}\left[Q_\phi(\mathbf{s}, \mathbf{a}, \mathbf{g}) - \log \pi_\theta(\mathbf{a} \,|\, \mathbf{s}; \mathbf{g})\right]. \quad (5.43)$$

We estimate $\hat{q}_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ with a Monte Carlo estimate of Equation (5.35) obtained via a single Monte Carlo sample $(\mathbf{s}, \mathbf{a}, \mathbf{s}', \mathbf{a}', \mathbf{g})$ from the replay buffer. In practice, a value of $q_{\Delta_{t+1}}(\Delta_{t+1} = 0) = 1$ can lead to numerical instabilities with bootstrapping, and so we also upper bound the estimated $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ by the prior distribution $p_{\Delta_{t+1}}(\Delta_{t+1} = 0)$.

To compute the rewards, we need to compute the likelihood of achieving the desired outcome. If the transition dynamics are unknown, we learn a dynamics model from environment interactions by training a neural network p_ψ that parameterizes

Algorithm 1 ODAC: Outcome-Driven Actor–Critic

1: Initialize policy π_θ, replay buffer $\mathcal{R}$, Q-function Q_ϕ, and dynamics model p_ψ.
2: **for** iteration $i = 1, 2, \ldots$ **do**
3: Collect on-policy samples to add to $\mathcal{R}$ by sampling $\mathbf{g}$ from environment and executing π.
4: Sample batch $(\mathbf{s}, \mathbf{a}, \mathbf{s}', \mathbf{g})$ from $\mathcal{R}$.
5: Compute approximate reward and optimal weights with Equation (5.45) and Equation (5.35).
6: Update Q_ϕ with Equation (5.42), π_θ with Equation (5.43), and p_ψ with Equation (5.44).
7: **end for**

the mean and scale of a factorized Laplace distribution. We train this model by minimizing the negative log-likelihood of the data collected by the policy,

$$\mathcal{F}_p(\psi) = -\,\mathbb{E}_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim\mathcal{D}}[\log p_\psi(\mathbf{s}' \,|\, \mathbf{s}, \mathbf{a})], \tag{5.44}$$

and use it to compute the rewards

$$\hat{r}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) \stackrel{\text{def}}{=} \hat{q}_{\Delta_{t+1}}(\Delta_{t+1} = 1) \log p_\psi(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) - \mathbb{D}_{\text{KL}}(q_{\Delta_t} \parallel p_{\Delta_t}). \tag{5.45}$$

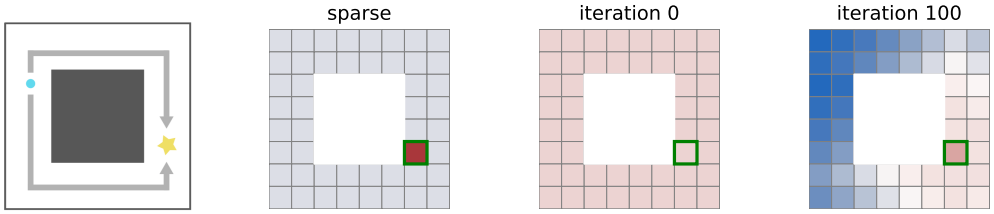
The complete algorithm is presented in Algorithm 1 and consists of alternating between collecting data via policy π and minimizing Equations (5.42), (5.43), and (5.44) via gradient descent. This algorithm alternates between approximating the lower bound in Equation (5.31) by repeatedly applying the outcome-driven Bellman operator to an approximate Q-function, and maximizing this lower bound by performing approximate policy optimization on Equation (5.43).

5.3.5 Empirical Evaluation

Our experiments compare ODAC to prior methods for learning goal-conditioned policies and evaluate how the components of our variational objective impact the performance of the method. Specifically, we compare ODAC to prior methods on a wide range of manipulation and locomotion tasks that require achieving a desired outcome. Additionally, we conduct several ablation studies and visualize the behavior of $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$. In our experiments, we use a uniform action prior $p(\mathbf{a})$, and the time prior p_T is geometric with parameter 0.01, that is, $p_{\Delta_{t+1}}(\Delta_{t+1} = 0) = 0.99$. We begin by describing prior methods and environments used for the experiments.

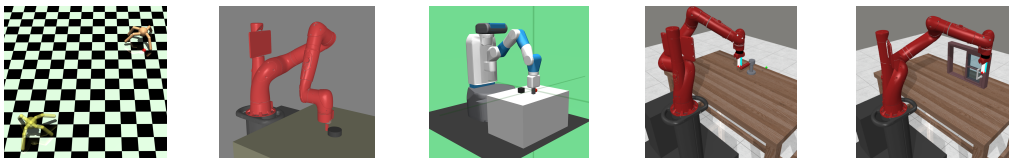
5.3.6 Learning to Achieve Desired Outcomes

We compare different methods across several robot morphologies and tasks, all illustrated in Figure 5.2.



(a) **(b)** **(c)** **(d)**

Figure 5.1: Visualization of the shaping effect of the reward function derived from the outcome-driven variational inference objective on the Box2D environment. **(a)** A 2-dimensional grid world with a desired outcome marked by a star. **(b)** The corresponding sparse reward function provides little shaping. **(c)** The reward function derived from our variational inference formulation at initialization. **(d)** The derived reward function after training. We see that the derived reward learns to provide a dense reward signal everywhere in the state space.



Ant Sawyer Push Fetch Push Faucet Window

Figure 5.2: Illustration of Environments Used to Evaluate the Outcome-Driven Actor–Critic Algorithm. From left to right, we evaluate on: a locomotion task in which a quadruped robot must match a location and pose (yellow), and four manipulation tasks in which the robot must push objects, rotate a faucet valve, or open a window.

Environments. We compare ODAC to prior work on a simple 2D navigation task, in which an agent must take non-greedy actions to move around a box, as well as the *Ant*, *Sawyer Push*, and *Fetch Push* simulated robot domains, which have each been studied in prior work on reinforcement learning for reaching goals (Andrychowicz et al., 2017; Nachum et al., 2018; Nair et al., 2018; Pong et al., 2019; Schroecker et al., 2020). For the Ant and Sawyer tasks, desired outcomes correspond to full states (that is, desired positions and joints). For the Fetch task, we use the same goal representation as in prior work (Andrychowicz et al., 2017) and only represent $\mathbf{g}$ with the position of the object. Lastly, we demonstrate the feasibility of replacing manually designed rewards with our outcome-driven paradigm by evaluating the methods on the *Sawyer Window* and *Sawyer Faucet* tasks from the MetaWorld benchmark (Yu et al., 2020). These tasks come with manually designed reward functions, which we replace by simply specifying a desired outcome $\mathbf{g}$. We plot the mean and standard deviation of the final Euclidean distance to the desired outcome across four random seeds. We normalize the distance to be 1 at the start of training. For further details, see Appendix C.2.1.

Table 5.1: Ablation results, showing mean final normalized distance (×100) at the end of training across four seeds. Standard errors are shown in parentheses. ODAC is not sensitive to the dynamics model $\hat{p}_d$ but benefits from the dynamic q_T variant.

Env	ODAC	fixed $\hat{p}_d$	fixed q_T	fixed q_T, $\hat{p}_d$
2D	1.7 (1.20)	1.2 (0.14)	1.0 (0.24)	1.3 (0.29)
Ant	9 (0.48)	11 (0.57)	12 (0.41)	13 (0.20)
Push	35 (2.7)	34 (1.5)	37 (1.5)	38 (3.1)
Fetch	19 (6)	15 (3)	53 (13)	66 (15)
Window	5.4 (0.62)	5.0 (0.62)	7.9 (0.71)	6.0 (0.12)
Faucet	13 (4.2)	15 (3.3)	37 (8.3)	38 (7.2)

Goal Sampling. In all tasks, rather than assuming that we are able to perform oracle goal sampling (which can be difficult or even impossible in complex real-world environments), a fixed desired outcome is commanded as the exploration goal in each episode. During training, the goals are relabeled using the future-style relabeling scheme from Andrychowicz et al. (2017). Unlike oracle goal sampling, this approach does not assume knowledge of the set of admissible states in the environment, and as such, it is more realistic and presents a more challenging exploration problem. To challenge the methods, we choose the desired goal to be far from the starting state.

Baselines and Prior Work. We compare our method to hindsight experience replay (HER) (Andrychowicz et al., 2017), a goal-conditioned method, where the learner receives a reward of -1 if it is more than an ϵ distance from the goal, and 0 otherwise, universal value density estimation (UVD) (Schroecker et al., 2020), which also uses sparse rewards as well as a generative model of the future occupancy measure to estimate the Q-values, and DISCERN (Warde-Farley et al., 2019), which learns a reward function by training a discriminator and using a clipped log-likelihood as the reward. Lastly, we include an oracle Soft Actor–Critic (SAC) baseline that uses a manually designed reward. For the MetaWorld tasks, this baseline uses the benchmark reward for each task. For the remaining environments, this baseline uses the Euclidean distance between the agent's current state and the desired outcome for the reward.

Results. In Figure 5.3, we see that ODAC outperforms virtually every method on all tasks, consistently learning faster and often reaching a final distance that is orders of magnitude closer to the desired outcome. The only exception is that the hand-crafted reward learns slightly faster on the 2D task, but this gap is closed within a few tens of thousands of steps.

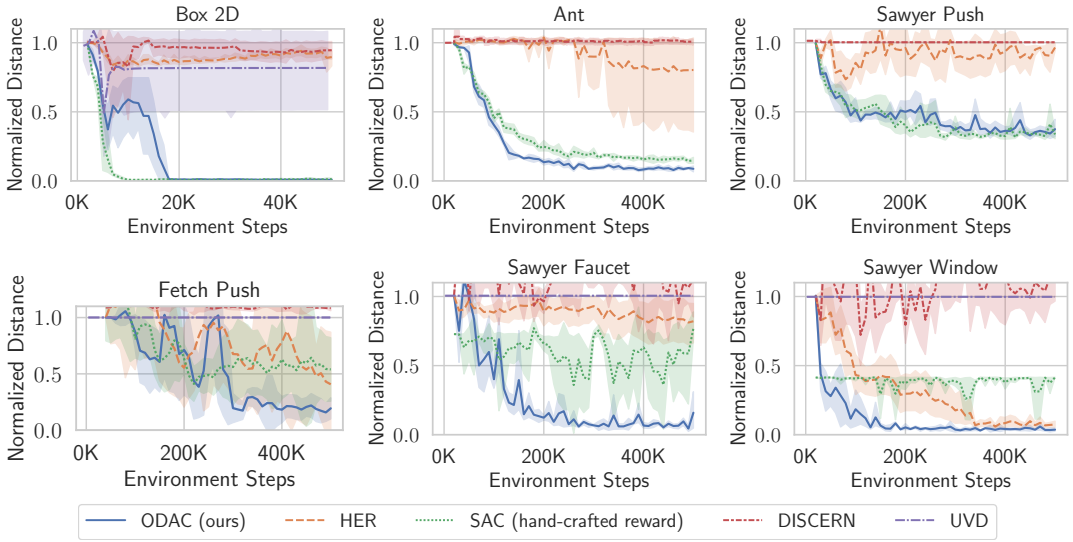


Figure 5.3: Learning curves showing final distance vs. environment steps across all six environments. Only ODAC consistently performs well on all six tasks. Prior methods struggle to learn, especially in the absence of uniform goal sampling. See text for details.

5.3.7 The Effect of a Variational Discount Factor

Next, we study the importance of the dynamic discount factor $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ and the sensitivity of our method to the dynamics model. On all tasks, we evaluate the performance when the posterior exactly matches the prior, that is, $q_{\Delta_{t+1}}(\Delta_{t+1} = 0) = 0.99$ (labeled "fixed q_T" in Table 5.1). Our analysis in Appendix C.1.7 suggests that this setting is suboptimal, and this ablation empirically evaluates the cost of making it. We also measure how the algorithm's performance depends on the accuracy of the learned dynamics model used for the reward in ODAC. To do this, we evaluate ODAC with the dynamics model fixed to a multivariate Laplace distribution with a fixed variance, centered at the previous state (labeled "fixed $\hat{p}_d$" in Table 5.1). This ablation represents an extremely crude model, and good performance with such a model would indicate that our method does not depend on obtaining a particularly accurate model.

In Table 5.1, we see that fixing the distribution q_T to the prior p_T as described in Corollary 5.3 deteriorates performance, and that both a learned and a fixed model perform relatively well. These results suggest that the derived optimal variational distribution $q^\star_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ given in Proposition 5.3 is better not only in theory but also in practice, and that ODAC is not sensitive to the accuracy of the dynamics model. Moreover, we note that a more expressive dynamics model—which could lead to a tighter variational bound—may not necessarily lead to a better variational policy if the functional form of the log-density under the dynamics model does not also provide favorable shaping.

In Appendix C.3, we provide the full learning curves for this ablation study and present further experiments. For example, we compare ODAC to a variant in which we use the learned dynamics model for model-based planning. We find that using the dynamics model only to compute rewards significantly outperforms the variant where it is used for both computing rewards and model-based planning. This result suggests that ODAC does not require learning a dynamics model that is accurate enough for planning, and that the derived Bellman updates are sufficient for obtaining policies that can achieve desired outcomes. In Figure C.2, we also visualize q_T and find that as the policy reaches an irrecoverable state, $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ drops in value, suggesting that ODAC *automatically* learns a dynamic discount factor that terminates an episode when an irrecoverable state is reached.

5.4 Discussion

In this chapter, we developed a variational formulation of reinforcement learning in infinite-horizon Markov decision processes. We first derived a probabilistic framework that casts behavior-driven reinforcement learning as variational inference. More specifically, we showed that performing variational inference to infer a state–action trajectory distribution conditioned on a set of desired behaviors corresponds to KL-regularized reinforcement learning with an environment-informed reward function and dynamic discount factor. We then built on this framework to consider settings where only a desired outcome and not a set of desired behaviors is available. We showed that by framing the problem of achieving desired outcomes as variational inference, we can derive an off-policy temporal-difference algorithm, a reward function learnable from environment interactions, and a novel Bellman backup that contains a state–action-dependent dynamic discount factor for the reward and bootstrap term. Our experimental results demonstrate that the resulting algorithm, outcome-driven actor–critic, leads to efficient outcome-driven behaviors and works well even for simple reward models. However, given the dependence of the method on a useful signal from the learned reward function, a fruitful direction for future research would be to use probabilistic methods to learn reward functions that are robust to distribution shifts and provide reliable uncertainty estimates that can guide policy learning.

6

Conclusion

In this concluding chapter, we reflect on the contributions made in this thesis. The focus of this thesis was to develop probabilistic methods that improve the reliability of neural network models used for decision-making tasks.

In Chapter 3, we introduced a variational inference method for Bayesian neural networks, transitioning from parameter-space to function-space variational inference. This function-space approach allowed us to address several limitations of traditional parameter-space methods, particularly in uncertainty quantification and continual learning. We demonstrated that by taking a function-space perspective, we can perform approximate inference in function space. This resulted in improved performance in settings involving distribution shifts and continual learning, as evidenced by our empirical evaluations.

However, a key limitation of this approach lies in its computational complexity. While the function-space variational objective can improve predictive accuracy and uncertainty quantification, scaling this method to more complex, high-dimensional datasets remains challenging. Applying function-space methods to large-scale datasets like ImageNet would be computationally expensive. Furthermore, although we saw improvements in uncertainty estimation, function-space variational inference does not consistently achieve the highest predictive accuracy compared to non-Bayesian methods. A promising avenue for future research would be to explore ways to reduce the computational overhead associated with function-space inference, potentially by developing more efficient approximations of the function-space KL divergence or by improving sampling techniques. We also see potential in hybrid approaches that

combine the strengths of function-space variational inference with other regularization techniques to further improve predictive performance.

In Chapter 4, we addressed the challenge of specifying meaningful priors in stochastic neural networks. We proposed a method for deriving data-driven priors, showing how domain-specific information can be incorporated into the training process to improve model reliability and robustness. Our introduction of uncertainty-aware and group-aware priors represents a significant advancement, allowing neural networks to better handle distribution shifts and perform selective prediction with increased robustness.

A notable limitation of this work is its reliance on the availability of domain-specific information to construct meaningful priors. In real-world applications, obtaining such information can be difficult, which may limit the practical application of data-driven priors. While we showed that data-driven priors improve robustness to distribution shifts and improve uncertainty quantification, balancing interpretability and performance remains a challenge. A promising avenue for future research would be to use self-supervised learning or meta-learning to infer priors from related tasks or to use hierarchical or multi-level priors that can adaptively adjust as new data becomes available.

In Chapter 5, we presented a variational framework for reinforcement learning. By reframing both behavior-driven and outcome-driven RL within a probabilistic framework, we demonstrated that existing RL algorithms can be derived as special cases of this general approach. This probabilistic perspective enabled us to derive new RL algorithms that incorporate learned reward functions and time-dependent discount factors, expanding the scope of RL methods to handle more complex and dynamic environments.

Despite these advancements, a key limitation of the temporal-difference algorithm derived from the variational framework is that it requires a significant amount of environment interactions to learn an effective reward function. A promising avenue for future work would be the integration of self-supervised learning methods for learning the reward model.

This thesis has made contributions to probabilistic machine learning and neural network-based decision-making, and our empirical results demonstrate that these methods offer improvements in areas such as distribution shift detection, continual learning, and safety-critical decision-making. We hope that future advances in probabilistic machine learning will adapt the methods developed in this thesis to generative models used for computer vision and language modeling.

Bibliography

[1] Hongjoon Ahn, Sungmin Cha, Donggyu Lee, and Taesup Moon. "Uncertainty-based Continual Learning with Adaptive Regularization". In: *Advances in Neural Information Processing Systems*. 2019.

[2] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. "Memory Aware Synapses: Learning What (not) to Forget". In: *European Conference on Computer Vision*. 2018.

[3] G.M Aly and W.C Chan. "Numerical computation of optimal control problems with unknown final time". In: *Journal of Mathematical Analysis and Applications* 45.2 (1974), pp. 274–284.

[4] Joost van Amersfoort, Lewis Smith, Andrew Jesson, Oscar Key, and Yarin Gal. *Variational Deterministic Uncertainty Quantification*. 2021.

[5] Joost van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. "Uncertainty Estimation Using a Single Deep Deterministic Neural Network". In: *International Conference on Machine Learning*. 2020.

[6] Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. "MathQA: Towards Interpretable Math Word Problem Solving with Operation-Based Formalisms". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 2357–2367.

[7] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. *Concrete Problems in AI Safety*. 2016. eprint: arXiv:1606.06565.

[8] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob Mcgrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. "Hindsight Experience Replay". In: *Neural Information Processing Systems (NeurIPS)*. 2017. arXiv: 1707.01495.

[9] APTOS. *APTOS 2019 Blindness Detection Dataset*. 2019.

[10] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. *Invariant Risk Minimization*. 2020. arXiv: 1907.02893 [stat.ML].

[11] H. Attias. "Planning by Probabilistic Inference". In: *Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics*. 2003.

[12] Neil Band, Tim G. J. Rudner, Qixuan Feng, Angelos Filos, Zachary Nado, Michael W. Dusenberry, Ghassen Jerfel, Dustin Tran, and Yarin Gal. "Benchmarking Bayesian Deep Learning on Diabetic Retinopathy Detection Tasks". In: *Advances in Neural Information Processing Systems 34*. 2021.

[13] Adrien Bardes, Jean Ponce, and Yann LeCun. “VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning”. In: *International Conference on Learning Representations*. 2022.

[14] Aharon Ben-Tal, Dick Den Hertog, Anja De Waegenaere, Bertrand Melenberg, and Gijs Rennen. “Robust solutions of optimization problems affected by uncertain probabilities”. In: *Management Science* 59.2 (2013), pp. 341–357.

[15] Ari Benjamin, David Rolnick, and Konrad Kording. “Measuring and regularizing networks in function space”. In: *International Conference on Learning Representations*. 2019.

[16] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. “Variational Inference: A Review for Statisticians”. In: *Journal of the American Statistical Association* 112.518 (2017), pp. 859–877.

[17] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. “Weight Uncertainty in Neural Network”. In: ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 1613–1622.

[18] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. “On the opportunities and risks of foundation models”. In: *arXiv preprint arXiv:2108.07258* (2021).

[19] John S. Bridle. “Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition”. In: *Neurocomputing*. Ed. by Françoise Fogelman Soulié and Jeanny Hérault. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990, pp. 227–236.

[20] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. “Openai gym”. In: *arXiv preprint arXiv:1606.01540* (2016).

[21] Thang D Bui, Cuong Nguyen, and Richard E Turner. “Streaming Sparse Gaussian Process Approximations”. In: *Advances in Neural Information Processing Systems*. 2017.

[22] David R. Burt, Sebastian W. Ober, Adrià Garriga-Alonso, and Mark van der Wilk. “Understanding Variational Inference in Function-Space”. In: *Third Symposium on Advances in Approximate Bayesian Inference*. 2021.

[23] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. “Dark Experience for General Continual Learning: a Strong, Simple Baseline”. In: *Advances in Neural Information Processing Systems*. 2020.

[24] Carlos M. Carvalho, Nicholas G. Polson, and James G. Scott. “Handling Sparsity via the Horseshoe”. In: *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*. Ed. by David van Dyk and Max Welling. Vol. 5. Proceedings of Machine Learning Research. Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA: PMLR, Apr. 2009, pp. 73–80.

[25] Eduardo D. C. Carvalho, Ronald Clark, Andrea Nicastro, and Paul H. J. Kelly. “Scalable Uncertainty for Computer Vision With Functional Variational Inference”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[26] Rohitash Chandra and Arpit Kapoor. “Bayesian neural multi-source transfer learning”. In: *Neurocomputing* 378 (2020), pp. 54–64.

[27] Arslan Chaudhry, P. Dokania, Thalaiyasingam Ajanthan, and P. Torr. “Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence”. In: *European Conference on Computer Vision*. 2018.

[28] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. “Encoder-decoder with atrous separable convolution for semantic image segmentation”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 801–818.

[29] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. “A Simple Framework for Contrastive Learning of Visual Representations”. In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 1597–1607.

[30] Ching-An Cheng and Byron Boots. “Incremental Variational Sparse Gaussian Process Regression”. In: *Advances in Neural Information Processing Systems*. 2016.

[31] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. New York: Wiley, 1991.

[32] Lehel Csató. “Gaussian processes: iterative sparse approximations”. PhD thesis. Aston University, 2002.

[33] Lehel Csató and Manfred Opper. “Sparse On-Line Gaussian Processes”. In: *Neural Computation* (2002).

[34] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. “CoAtNet: Marrying Convolution and Attention for All Data Sizes”. In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., 2021, pp. 3965–3977.

[35] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars. “A continual learning survey: defying forgetting in classification tasks”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[36] Alex J. DeGrave, Joseph D. Janizek, and Su-In Lee. “AI for radiographic COVID-19 detection selects shortcuts over signal”. In: *Nature Machine Intelligence* (2021).

[37] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. “ImageNet: A Large-Scale Hierarchical Image Database”. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Ieee. 2009, pp. 248–255.

[38] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186.

[39] Kien Do, Truyen Tran, and Svetha Venkatesh. “Semi-Supervised Learning with Variational Bayesian Inference and Maximum Uncertainty Regularization”. In: *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 2021, pp. 7236–7244.

[40] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *International Conference on Learning Representations*. 2021.

[41] Sayna Ebrahimi, Mohamed Elhoseiny, Trevor Darrell, and Marcus Rohrbach. “Uncertainty-Guided Continual Learning in Bayesian Neural Networks - Extended Abstract”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2019.

[42] EyePACS. *Diabetic Retinopathy Detection Dataset*. 2015.

[43] Sebastian Farquhar and Yarin Gal. “Towards Robust Evaluations of Continual Learning”. In: *ICML Workshop on Lifelong Learning: A Reinforcement Learning Approach* (2018).

[44] Sebastian Farquhar, Michael A. Osborne, and Yarin Gal. “Radial Bayesian Neural Networks: Beyond Discrete Support In Large-Scale Bayesian Deep Learning”. In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Ed. by Silvia Chiappa and Roberto Calandra. Vol. 108. Proceedings of Machine Learning Research. PMLR, Aug. 2020, pp. 1352–1362.

[45] Sebastian Farquhar, Lewis Smith, and Yarin Gal. “Liberty or Depth: Deep Bayesian Neural Nets Do Not Need Complex Weight Posterior Approximations”. In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin. 2020.

[46] Angelos Filos, Sebastian Farquhar, Aidan N. Gomez, Tim G. J. Rudner, Zachary Kenton, Lewis Smith, Milad Alizadeh, Arnoud de Kroon, and Yarin Gal. *A Systematic Comparison of Bayesian Deep Learning Robustness in Diabetic Retinopathy Tasks*. 2019. eprint: arXiv:1912.10481.

[47] Chelsea Finn, Kelvin Xu, and Sergey Levine. “Probabilistic model-agnostic meta-learning”. In: *Advances in neural information processing systems* 31 (2018).

[48] Andrew Y. K. Foong, David R. Burt, Yingzhen Li, and Richard E. Turner. “On the Expressiveness of Approximate Inference in Bayesian Neural Networks”. In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin. 2020.

[49] Andrew Y. K. Foong, Yingzhen Li, José Miguel Hernández-Lobato, and Richard E. Turner. *’In-Between’ Uncertainty in Bayesian Neural Networks*. 2019. eprint: arXiv:1906.11537.

[50] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. "Sharpness-aware Minimization for Efficiently Improving Generalization". In: *International Conference on Learning Representations*. 2021.

[51] Vincent Fortuin, Adrià Garriga-Alonso, Sebastian W. Ober, Florian Wenzel, Gunnar Ratsch, Richard E Turner, Mark van der Wilk, and Laurence Aitchison. "Bayesian Neural Network Priors Revisited". In: *International Conference on Learning Representations*. 2022.

[52] Justin Fu, Avi Singh, Dibya Ghosh, Larry Yang, and Sergey Levine. "Variational Inverse Control with Events: A General Framework for Data-Driven Reward Definition". In: *Neural Information Processing Systems (NeurIPS)*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. 2018, pp. 8538–8547.

[53] Scott Fujimoto, Herke van Hoof, and David Meger. "Addressing Function Approximation Error in Actor-Critic Methods". In: *International Conference on Machine Learning (ICML)* (2018).

[54] Yarin Gal and Zoubin Ghahramani. "Dropout As a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML 2016. New York, NY, USA, 2016, pp. 1050–1059.

[55] Alexandre Galashov, Siddhant M. Jayakumar, Leonard Hasenclever, Dhruva Tirumala, Jonathan Schwarz, Guillaume Desjardins, Wojciech M. Czarnecki, Yee Whye Teh, Razvan Pascanu, and Nicolas Heess. "Information asymmetry in KL-regularized RL". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. 2019.

[56] Soumya Ghosh, Jiayu Yao, and Finale Doshi-Velez. "Structured variational learning of Bayesian neural networks with horseshoe priors". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1744–1753.

[57] Manfred Goebel and Uldis Raitums. "Optimal control of two point boundary value problems". In: *System Modelling and Optimization*. Ed. by H. -J. Sebastian and K. Tammer. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990, pp. 281–290.

[58] Alex Graves. "Practical Variational Inference for Neural Networks". In: *Proceedings of the 24th International Conference on Neural Information Processing Systems*. NIPS'11. Granada, Spain: Curran Associates Inc., 2011, pp. 2348–2356.

[59] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor". In: *International Conference on Machine Learning*. 2018. arXiv: arXiv:1801.01290v2.

[60] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. *Soft Actor-Critic Algorithms and Applications*. 2018. eprint: arXiv:1812.05905.

[61] Danijar Hafner, Dustin Tran, Timothy Lillicrap, Alex Irpan, and James Davidson. "Noise contrastive priors for functional uncertainty". In: *Uncertainty in Artificial Intelligence*. PMLR. 2020, pp. 905–914.

[62] Ethan Harvey, Mikhail Petrov, and Michael C Hughes. “Transfer Learning with Informative Priors: Simple Baselines Better than Previously Reported”. In: *Transactions on Machine Learning Research*. 2024.

[63] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778. arXiv: 1512.03385v1.

[64] Will Douglas Heaven. *Hundreds of AI tools have been built to catch COVID. None of them helped.* 2021.

[65] Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. *Unsolved Problems in ML Safety*. 2021. eprint: arXiv:2109.13916.

[66] Geoffrey E. Hinton and Drew van Camp. “Keeping the Neural Networks Simple by Minimizing the Description Length of the Weights”. In: *Proceedings of the Sixth Annual Conference on Computational Learning Theory*. COLT ’93. Citeseer. Santa Cruz, California, USA: Association for Computing Machinery, 1993, pp. 5–13.

[67] Matthew Hoffman, Nando Freitas, Arnaud Doucet, and Jan Peters. “An expectation maximization algorithm for continuous Markov decision processes with arbitrary reward”. In: *Artificial intelligence and statistics*. 2009, pp. 232–239.

[68] Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. “Stochastic Variational Inference”. In: *Journal of Machine Learning Research* 14.1 (May 2013), pp. 1303–1347.

[69] Jeremy Howard and Sebastian Ruder. “Universal Language Model Fine-tuning for Text Classification”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Iryna Gurevych and Yusuke Miyao. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 328–339.

[70] Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. “Does distributionally robust supervised learning give robust classifiers?” In: *International Conference on Machine Learning*. PMLR. 2018, pp. 2029–2037.

[71] Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. “Simple data balancing achieves competitive worst-group-accuracy”. In: *Proceedings of the First Conference on Causal Learning and Reasoning*. Ed. by Bernhard Schölkopf, Caroline Uhler, and Kun Zhang. Vol. 177. Proceedings of Machine Learning Research. PMLR, Apr. 2022, pp. 336–351.

[72] Alexander Immer, Matthias Bauer, Vincent Fortuin, Gunnar Rätsch, and Khan Mohammad Emtiyaz. “Scalable marginal likelihood estimation for model selection in deep learning”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 4563–4573.

[73] Alexander Immer, Maciej Korzepa, and Matthias Bauer. *Improving predictions of Bayesian neural networks via local linearization*. 2020. eprint: arXiv:2008.08400.

[74] Pavel Izmailov, Polina Kirichenko, Nate Gruver, and Andrew G Wilson. “On feature learning in the presence of spurious correlations”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 38516–38532.

[75] Pavel Izmailov, Wesley J. Maddox, Polina Kirichenko, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. "Subspace Inference for Bayesian Deep Learning". In: *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*. Ed. by Ryan P. Adams and Vibhav Gogate. Vol. 115. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 1169–1179.

[76] Pavel Izmailov, Patrick Nicholson, Sanae Lotfi, and Andrew Gordon Wilson. "Dangers of Bayesian Model Averaging under Covariate Shift". In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan. 2021.

[77] Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Gordon Wilson. "What Are Bayesian Neural Network Posteriors Really Like?" In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 4629–4640.

[78] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. "When to Trust Your Model: Model-Based Policy Optimization". In: *Advances in Neural Information Processing Systems*. 2019.

[79] E. T. Jaynes. *Probability theory: The logic of science*. Cambridge: Cambridge University Press, 2003.

[80] Neal Jean, Sang Michael Xie, and Stefano Ermon. "Semi-supervised deep kernel learning: Regression with unlabeled data by minimizing predictive variance". In: *Advances in Neural Information Processing Systems* 31 (2018).

[81] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Zidek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (2021), pp. 583–589.

[82] Heechul Jung, Jeongwoo Ju, Minju Jung, and Junmo Kim. "Less-forgetful Learning for Domain Expansion in Deep Neural Networks". In: *AAAI Conference on Artificial Intelligence*. 2018.

[83] Leslie P Kaelbling. "Learning to achieve goals". In: *International Joint Conference on Artificial Intelligence (IJCAI)*. Vol. vol.2. 1993, pp. 1094–8.

[84] Sanyam Kapoor, Theofanis Karaletsos, and Thang D Bui. "Variational auto-regressive gaussian processes for continual learning". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 5290–5300.

[85] Alireza Karbalayghareh, Xiaoning Qian, and Edward R Dougherty. "Optimal Bayesian transfer learning". In: *IEEE Transactions on Signal Processing* 66.14 (2018), pp. 3724–3739.

[86] Mohammad Emtiyaz E Khan, Alexander Immer, Ehsan Abedi, and Maciej Korzepa. "Approximate Inference Turns Deep Networks into Gaussian Processes". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., 2019, pp. 3094–3104.

[87] Hyo-Eun Kim, Seungwook Kim, and Jaehwan Lee. “Keep and Learn: continual Learning by Constraining the Latent Space for Knowledge Preservation in Neural Networks”. In: *Medical Image Computing and Computer Assisted Intervention*. 2018.

[88] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. “Last Layer Re-Training is Sufficient for Robustness to Spurious Correlations”. In: *The Eleventh International Conference on Learning Representations*. 2023.

[89] J. Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, J. Veness, G. Desjardins, Andrei A. Rusu, K. Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. “Overcoming catastrophic forgetting in neural networks”. In: *Proceedings of the National Academy of Sciences* 114 (2017), pp. 3521–3526.

[90] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. “Overcoming catastrophic forgetting in neural networks”. In: *Proceedings of the national academy of sciences* 114.13 (2017), pp. 3521–3526.

[91] Leo Klarner, Tim G. J. Rudner, and Yee Whye Teh. Garrett M. Morris Charlotte Deane. “Domain-Aware Guidance for Out-of-Distribution Molecular and Protein Design”. In: *Proceedings of the 41th International Conference on Machine Learning*. Proceedings of Machine Learning Research. PMLR, 2024.

[92] Leo Klarner, Tim G. J. Rudner, Michael Reutlinger, Torsten Schindler, Garrett M. Morris, Charlotte Deane, and Yee Whye Teh. “Drug Discovery under Covariate Shift with Domain-Informed Prior Distributions over Functions”. In: *Proceedings of the 40th International Conference on Machine Learning*. Proceedings of Machine Learning Research. PMLR, 2023.

[93] Ranganath Krishnan, Mahesh Subedar, and Omesh Tickoo. “Specifying weight priors in Bayesian deep neural networks with empirical Bayes”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 04. 2020, pp. 4477–4484.

[94] Alex Krizhevsky, Geoffrey Hinton, et al. “Learning multiple layers of features from tiny images”. In: (2009).

[95] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems 25:* 2012, pp. 1106–1114.

[96] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. “Human-level concept learning through probabilistic program induction”. In: *Science* (2015).

[97] Phuong Quynh Le, Jörg Schlötterer, and Christin Seifert. *Is Last Layer Re-Training Truly Sufficient for Robustness to Spurious Correlations?* 2023. arXiv: 2308.00473 [cs.LG].

[98] Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. “Overcoming Catastrophic Forgetting by Incremental Moment Matching”. In: *Advances in Neural Information Processing Systems*. 2017.

[99] Christian Leibig, Vaneeda Allken, Murat Seçkin Ayhan, Philipp Berens, and Siegfried Wahl. “Leveraging Uncertainty Information From Deep Neural Networks for Disease Detection”. In: *Nature Scientific Reports* 7.1 (2017), p. 17816.

[100] Sergey Levine. “Reinforcement Learning and Control as Probabilistic Inference: Tutorial and Review”. In: (2018). arXiv: arXiv:1805.00909v2.

[101] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C. Kot. “Domain Generalization with Adversarial Feature Learning”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5400–5409.

[102] Zhizhong Li and Derek Hoiem. “Learning without Forgetting”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018).

[103] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. “Continuous control with deep reinforcement learning”. In: *International Conference on Learning Representations (ICLR)*. 2016. arXiv: 9605103 [cs].

[104] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. “Just Train Twice: Improving Group Robustness without Training Group Information”. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 6781–6792.

[105] X. Liu, Marc Masana, Luis Herranz, Joost van de Weijer, Antonio M. López, and Andrew D. Bagdanov. “Rotate your Networks: better Weight Consolidation and Less Catastrophic Forgetting”. In: *International Conference on Pattern Recognition* (2018).

[106] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. “Deep Learning Face Attributes in the Wild”. In: *Proceedings of International Conference on Computer Vision (ICCV)*. Dec. 2015.

[107] Noel Loo, S. Swaroop, and Richard E. Turner. “Generalized Variational Continual Learning”. In: *International Conference on Learning Representations*. 2020.

[108] Julian Lechuga Lopez, Tim G. J. Rudner, and Farah Shamout. “Informative Priors Improve the Reliability of Multimodal Clinical Data Classification”. In: *Machine Learning for Health Symposium Findings*. 2023.

[109] Ilya Loshchilov and Frank Hutter. “Decoupled weight decay regularization”. In: *arXiv preprint arXiv:1711.05101* (2017).

[110] Christos Louizos and Max Welling. “Structured and efficient variational deep learning with matrix gaussian posteriors”. In: *International conference on machine learning*. PMLR. 2016, pp. 1708–1716.

[111] Cong Lu, Philip J. Ball, Tim G. J. Rudner, Jack Parker-Holder, Michael A. Osborne, and Yee Whye Teh. “Challenges and Opportunities in Offline Reinforcement Learning from Visual Observations”. In: *Transactions on Machine Learning Research*. 2023.

[112] Chao Ma and José Miguel Hernández-Lobato. “Functional Variational Inference based on Stochastic Process Generators”. In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan. 2021.

[113] Chao Ma, Yingzhen Li, and Jose Miguel Hernandez-Lobato. “Variational Implicit Processes”. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, June 2019, pp. 4222–4233.

[114] David J. C. MacKay. “A Practical Bayesian Framework for Backpropagation Networks”. In: *Neural Comput.* 4.3 (May 1992), pp. 448–472.

[115] Wesley Maddox, Shuai Tang, Pablo Moreno, Andrew Gordon Wilson, and Andreas Damianou. “Fast adaptation with linearized neural networks”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 2737–2745.

[116] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. “A simple baseline for bayesian uncertainty in deep learning”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 13153–13164.

[117] Alexander G. de G. Matthews, James Hensman, Richard Turner, and Zoubin Ghahramani. “On Sparse Variational Methods and the Kullback-Leibler Divergence between Stochastic Processes”. In: *International Conference on Artificial Intelligence and Statistics*. Ed. by Arthur Gretton and Christian C. Robert. Vol. 51. Proceedings of Machine Learning Research. Cadiz, Spain: PMLR, May 2016, pp. 231–239.

[118] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. “Playing Atari with Deep Reinforcement Learning”. In: *NIPS Workshop on Deep Learning*. 2013, pp. 1–9. arXiv: 1312.5602.

[119] Pablo Moreno-Muñoz, Antonio Artés-Rodríguez, and Mauricio A. Álvarez. “Continual Multi-task Gaussian Processes”. In: *arXiv* (2019).

[120] Kevin P. Murphy. *Machine learning : a probabilistic perspective*. Cambridge, Mass. [u.a.]: MIT Press, 2013.

[121] Ofir Nachum, Google Brain, Shixiang Gu, Honglak Lee, and Sergey Levine. “Data-Efficient Hierarchical Reinforcement Learning”. In: *Neural Information Processing Systems (NeurIPS)*. 2018. arXiv: 1805.08296v4.

[122] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. “Obtaining well calibrated probabilities using Bayesian binning”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 29. 1. 2015.

[123] Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. “Neural Network Dynamics for Model-Based Deep Reinforcement Learning with Model-Free Fine-Tuning”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2018.

[124] Ashvin Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. “Visual Reinforcement Learning with Imagined Goals”. In: *Neural Information Processing Systems (NeurIPS)*. 2018. arXiv: arXiv:1807.04742v1.

[125] Junhyun Nam, Jaehyung Kim, Jaeho Lee, and Jinwoo Shin. “Spread Spurious Attribute: Improving Worst-group Accuracy with Spurious Attribute Estimation”. In: *International Conference on Learning Representations*. 2022.

[126] Radford M Neal. “Bayesian Learning for Neural Networks”. In: (1996).

[127] Cuong V Nguyen, Yingzhen Li, Thang D. Bui, and Richard E. Turner. “Variational Continual Learning”. In: *International Conference on Learning Representations*. 2018.

[128] Maria-Elena Nilsback and Andrew Zisserman. “Automated flower classification over a large number of classes”. In: *2008 Sixth Indian conference on computer vision, graphics & image processing*. IEEE. 2008, pp. 722–729.

[129] Sebastian W. Ober and Laurence Aitchison. *Global inducing point variational posteriors for Bayesian neural networks and deep Gaussian processes*. 2020. eprint: arXiv:2005.08140.

[130] Yonatan Oren, Shiori Sagawa, Tatsunori B Hashimoto, and Percy Liang. "Distributionally robust language modeling". In: *arXiv preprint arXiv:1909.02060* (2019).

[131] Kazuki Osawa, Siddharth Swaroop, Mohammad Emtiyaz E Khan, Anirudh Jain, Runa Eschenhagen, Richard E Turner, and Rio Yokota. "Practical Deep Learning with Bayesian Principles". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019, pp. 4287–4299.

[132] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. "Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift". In: *Advances in Neural Information Processing Systems 32*. 2019.

[133] Pingbo Pan, Siddharth Swaroop, Alexander Immer, Runa Eschenhagen, Richard Turner, and Mohammad Emtiyaz E Khan. "Continual Deep Learning by Functional Regularisation of Memorable Past". In: *Advances in Neural Information Processing Systems*. 2020.

[134] G. I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and S. Wermter. "Continual Lifelong Learning with Neural Networks: a Review". In: *Neural Networks* (2019).

[135] Dongmin Park, Seokil Hong, B. Han, and Kyoung Mu Lee. "Continual Learning by Asymmetric Loss Approximation With Single-Side Overestimation". In: *International Conference on Computer Vision*. 2019.

[136] Krunoslav Lehman Pavasovic, Jonas Rothfuss, and Andreas Krause. "MARS: Meta-learning as Score Matching in the Function Space". In: *The Eleventh International Conference on Learning Representations*. 2023.

[137] Yury Polyanskiy and Yihong Wu. "Strong Data-Processing Inequalities for Channels and Bayesian Networks". In: *Convexity and Concentration*. Ed. by Eric Carlen, Mokshay Madiman, and Elisabeth M. Werner. New York, NY: Springer New York, 2017, pp. 211–249.

[138] Vitchyr H. Pong, Murtaza Dalal, Steven Lin, Ashvin Nair, Shikhar Bahl, and Sergey Levine. "Skew-Fit: State-Covering Self-Supervised Reinforcement Learning". In: *arXiv preprint arXiv:1903.03698* abs/1903.03698 (2019). arXiv: 1903.03698.

[139] Shikai Qiu, Andres Potapczynski, Pavel Izmailov, and Andrew Gordon Wilson. "Simple and Fast Group Robustness by Automatic Feature Reweighting". In: *International Conference on Machine Learning (ICML)* (2023).

[140] Shikai Qiu, Tim G. J. Rudner, Sanyam Kapoor, and Andrew Gordon Wilson. "Should We Learn Most Likely Functions or Parameters?" In: *Advances in Neural Information Processing Systems 36*. 2023.

[141] Alec Radford, Jeff Wu, Rewon Child, D. Luan, Dario Amodei, and Ilya Sutskever. *Language models are unsupervised multitask learners*. 2019.

[142] Rafael Rafailov, Tianhe Yu, Aravind Rajeswaran, and Chelsea Finn. "Offline Reinforcement Learning from Images with Latent Space Models". In: *Proceedings of the 3rd Conference on Learning for Dynamics and Control*. Ed. by Ali Jadbabaie, John Lygeros, George J. Pappas, Pablo A. Parrilo, Benjamin Recht, Claire J. Tomlin, and Melanie N. Zeilinger. Vol. 144. Proceedings of Machine Learning Research. PMLR, June 2021, pp. 1154–1168.

[143] Hamed Rahimian and Sanjay Mehrotra. "Frameworks and Results in Distributionally Robust Optimization". In: *Open Journal of Mathematical Optimization* 3 (July 2022), pp. 1–85.

[144] Naresh Balaji Ravichandran, Anders Lansner, and Pawel Herman. "Semi-supervised learning with Bayesian confidence propagation neural network". In: *arXiv preprint arXiv:2106.15546* (2021).

[145] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015), pp. 91–99.

[146] Hippolyt Ritter, Aleksandar Botev, and David Barber. "Online Structured Laplace Approximations for Overcoming Catastrophic Forgetting". In: *Advances in Neural Information Processing Systems*. 2018.

[147] Herbert E. Robbins. "An Empirical Bayes Approach to Statistics". In: *Breakthroughs in Statistics: Foundations and Basic Theory*. Ed. by Samuel Kotz and Norman L. Johnson. New York, NY: Springer New York, 1992, pp. 388–394.

[148] Jonas Rothfuss, Vincent Fortuin, Martin Josifoski, and Andreas Krause. "PACOH: Bayes-optimal meta-learning with PAC-guarantees". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 9116–9126.

[149] Tim G. J. Rudner, Zonghao Chen, Yee Whye Teh, and Yarin Gal. "Tractable Function-Space Variational Inference in Bayesian Neural Networks". In: *Advances in Neural Information Processing Systems 35*. 2022.

[150] Tim G. J. Rudner, Sanyam Kapoor, Shikai Qiu, and Andrew Gordon Wilson. "Function-Space Regularization in Neural Networks: A Probabilistic Perspective". In: *Proceedings of the 40th International Conference on Machine Learning*. 2023.

[151] Tim G. J. Rudner, Cong Lu, Michael A. Osborne, Yarin Gal, and Yee Whye Teh. "On Pathologies in KL-Regularized Reinforcement Learning from Expert Demonstrations". In: *Advances in Neural Information Processing Systems 34*. 2021.

[152] Tim G. J. Rudner, Freddie Bickford Smith, Qixuan Feng, Yee Whye Teh, and Yarin Gal. "Continual Learning via Sequential Function-Space Variational Inference". In: *Proceedings of the 39th International Conference on Machine Learning*. 2022.

[153] Tim G. J. Rudner and Helen Toner. "Key Concepts in AI Safety: An Overview". In: *CSET Issue Briefs*. 2021.

[154] Tim G. J. Rudner and Helen Toner. "Key Concepts in AI Safety: Robustness and Adversarial Examples". In: *CSET Issue Briefs*. 2021.

[155] Tim G. J. Rudner, Ya Shi Zhang, Andrew Gordon Wilson, and Julia Kempe. "Mind the GAP: Improving Robustness to Subpopulation Shifts with Group-Aware Priors". In: *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*. 2024.

[156] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252.

[157] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. "Distributionally Robust Neural Networks". In: *International Conference on Learning Representations*. 2020.

[158] Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. "CARER: Contextualized Affect Representations for Emotion Recognition". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 3687–3697.

[159] M. J. Schervish. *Theory of Statistics*. New York, NY: Springer-Verlag, 1995.

[160] Yannick Schroecker and Charles Isbell. "Universal value density estimation for imitation learning and goal-conditioned reinforcement learning". In: *arXiv preprint arXiv:2002.06473* (2020).

[161] Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. "Progress & compress: a scalable framework for continual learning". In: *International Conference on Machine Learning*. 2018.

[162] Claude E. Shannon and Warren Weaver. *The Mathematical Theory of Communication*. Urbana and Chicago: University of Illinois Press, 1949.

[163] Ravid Shwartz-Ziv, Micah Goldblum, Hossein Souri, Sanyam Kapoor, Chen Zhu, Yann LeCun, and Andrew G Wilson. "Pre-train your loss: Easy Bayesian transfer learning with informative priors". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 27706–27715.

[164] Ravid Shwartz-Ziv and Yann LeCun. "To Compress or Not to Compress—Self-Supervised Learning and Information Theory: A Review". In: *Entropy* 26.3 (2024).

[165] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529.7587 (Jan. 2016), pp. 484–489. arXiv: 1610.00633.

[166] Edward Snelson and Zoubin Ghahramani. "Sparse Gaussian Processes using Pseudo-inputs". In: *Advances in Neural Information Processing Systems 18*. Ed. by Y. Weiss, B. Schölkopf, and J. C. Platt. MIT Press, 2006, pp. 1257–1264.

[167] Nimit Sharad Sohoni, Maziar Sanjabi, Nicolas Ballas, Aditya Grover, Shaoliang Nie, Hamed Firooz, and Christopher Re. "BARACK: Partially Supervised Group Robustness With Guarantees". In: *ICML 2022: Workshop on Spurious Correlations, Invariance and Stability*. 2022.

[168] Shengyang Sun, Guodong Zhang, Jiaxin Shi, and Roger B. Grosse. "Functional variational Bayesian Neural Networks". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[169] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. 1998.

[170] S. Swaroop, Cuong V Nguyen, Thang D. Bui, and Richard E. Turner. "Improving and Understanding Variational Continual Learning". In: *NeurIPS Workshop on Continual Learning*. 2019.

[171] Csaba Szepesvári. *Algorithms for Reinforcement Learning*. Vol. 4. 1. 2010, pp. 1–103.

[172] Michalis K. Titsias, Jonathan Schwarz, Alexander G. de G. Matthews, Razvan Pascanu, and Yee Whye Teh. "Functional Regularisation for Continual Learning with Gaussian Processes". In: *International Conference on Learning Representations*. 2020.

[173] Marc Toussaint, Stefan Harmeling, and Amos Storkey. *Probabilistic inference for solving (PO)MDPs*. Tech. rep. 2006.

[174] Dustin Tran, Jeremiah Liu, Michael W. Dusenberry, Du Phan, Mark Collier, Jie Ren, Kehang Han, Zi Wang, Zelda Mariet, Huiyi Hu, Neil Band, Tim G. J. Rudner, Karan Singhal, Zachary Nado, Joost van Amersfoort andAndreas Kirsch, Rodolphe Jenatton, Nithum Thain, Honglin Yuan, Kelly Buchanan, Kevin Murphy, D. Sculley, Yarin Gal, Zoubin Ghahramani, Jasper Snoek, and Balaji Lakshminarayanan. "Plex: Towards Reliability Using Pretrained Large Model Extensions". In: *ICML 2022 Workshop on Pre-training: Perspectives, Pitfalls, and Paths Forward*. 2022.

[175] Hanna Tseran, Mohammad Emtiyaz Khan, Tatsuya Harada, and Thang D Bui. "Natural variational continual learning". In: *Continual Learning Workshop@ NeurIPS*. Vol. 2. 2018.

[176] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. "The INaturalist Species Classification and Detection Dataset". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

[177] Vladimir Vapnik. *Statistical learning theory*. Wiley, 1998.

[178] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. *The Caltech-UCSD Birds-200-2011 Dataset*. July 2011.

[179] Martin J Wainwright and Michael I Jordan. *Graphical Models, Exponential Families, and Variational Inference*. Hanover, MA, USA: Now Publishers Inc., 2008.

[180] Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding". In: *arXiv preprint arXiv:1804.07461* (2018).

[181] Ziyu Wang, Tongzheng Ren, Jun Zhu, and Bo Zhang. "Function Space Particle Optimization for Bayesian Neural Networks". In: *International Conference on Learning Representations*. 2019.

[182] David Warde-Farley, Tom Van De Wiele, Tejas Kulkarni, Catalin Ionescu, Steven Hansen, & Volodymyr, and Mnih Deepmind. "Unsupervised Control Through Non-Parametric Discriminative Rewards". In: *International Conference on Learning Representations (ICLR)*. 2019. arXiv: 1811.11359v1.

[183] Florian Wenzel, Kevin Roth, Bastiaan Veeling, Jakub Swiatkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. "How Good is the Bayes Posterior in Deep Neural Networks Really?" In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 10248–10259.

[184] Martha White. "Unifying task specification in reinforcement learning". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 3742–3750.

[185] D. T. S. Widdowson. "The Management of Grading Quality: Good Practice in the Quality Assurance of Grading". In: *Tech. Rep.* (2016).

[186] David H. Wolpert. "Bayesian Backpropagation Over I-O Functions Rather Than Weights". In: *Advances in Neural Information Processing Systems*. Ed. by J. Cowan, G. Tesauro, and J. Alspector. Vol. 6. Morgan-Kaufmann, 1993.

[187] Anqi Wu, Sebastian Nowozin, Edward Meeds, Richard E Turner, José Miguel Hernández-Lobato, and Alexander L Gaunt. "Deterministic Variational Inference for Robust Bayesian Neural Networks". In: *International Conference on Learning Representations* (2019).

[188] Junyu Xuan, Jie Lu, and Guangquan Zhang. "Bayesian Transfer Learning: An Overview of Probabilistic Graphical Models for Transfer Learning". In: *arXiv preprint arXiv:2109.13233* (2021).

[189] Le Yang, Ke Wang, and Lyudmila S. Mihaylova. "Online Sparse Multi-Output Gaussian Process Regression and Learning". In: *IEEE Transactions on Signal and Information Processing over Networks* (2019).

[190] Yuzhe Yang, Haoran Zhang, Dina Katabi, and Marzyeh Ghassemi. "Change is Hard: A Closer Look at Subpopulation Shift". In: *International Conference on Machine Learning*. 2023.

[191] Ran El-Yaniv et al. "On the Foundations of Noise-free Selective Classification." In: *Journal of Machine Learning Research* 11.5 (2010).

[192] Dong Yin, Mehrdad Farajtabar, and Ang Li. "SOLA: continual Learning with Second-Order Loss Approximation". In: *arXiv* (2020).

[193] Dong Yin, Mehrdad Farajtabar, Ang Li, N. Levine, and A. Mott. "Optimization and Generalization of Regularization-Based Continual Learning: a Loss Approximation Viewpoint." In: *arXiv* (2020).

[194] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning". In: *Conference on Robot Learning*. PMLR. 2020, pp. 1094–1100.

[195] Friedemann Zenke, Ben Poole, and Surya Ganguli. "Continual Learning Through Synaptic Intelligence". In: *International Conference on Machine Learning*. 2017.

[196] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. "mixup: Beyond Empirical Risk Minimization". In: *International Conference on Learning Representations*. 2018.

[197] Jingzhao Zhang, Aditya Menon, Andreas Veit, Srinadh Bhojanapalli, Sanjiv Kumar, and Suvrit Sra. "Coping with label shift via distributionally robust optimisation". In: *arXiv preprint arXiv:2010.12230* (2020).

[198] Michael Zhang, Nimit S Sohoni, Hongyang R Zhang, Chelsea Finn, and Christopher Re. "Correct-N-Contrast: a Contrastive Approach for Improving Robustness to Spurious Correlations". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato. Vol. 162. Proceedings of Machine Learning Research. PMLR, July 2022, pp. 26484–26516.

[199] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. "Places: A 10 Million Image Database for Scene Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.6 (2018), pp. 1452–1464.

[200] Brian D Ziebart, Andrew Maas, J Andrew Bagnell, and Anind K Dey. "Maximum Entropy Inverse Reinforcement Learning." In: *AAAI Conference on Artificial Intelligence*. 2008, pp. 1433–1438. arXiv: arXiv:1507.04888v2.

A

Appendix A: Appendix to Chapter 3

A.1 Proofs & Derivations

Proposition 3.1. *For a stochastic function $f(\cdot\,;\Theta)$ defined in terms of stochastic parameters Θ distributed according to distribution g_Θ with $\mathbf{m} \in \mathbb{R}^P$ and $\mathbf{S} \stackrel{\text{def}}{=} \mathrm{Cov}_{g_\Theta}[\Theta]$, denote the linearization of the stochastic function $f(\cdot\,;\Theta)$ about $\mathbf{m}$ by*

$$f(\cdot\,;\Theta) \approx \tilde{f}(\cdot\,;\mathbf{m},\Theta) \stackrel{\text{def}}{=} f(\cdot\,;\mathbf{m}) + \mathcal{J}(\cdot\,;\mathbf{m})(\Theta - \mathbf{m}),$$

where $\mathcal{J}(\cdot\,;\mathbf{m}) \stackrel{\text{def}}{=} (\partial f(\cdot\,;\Theta)/\partial\Theta)|_{\Theta=\mathbf{m}}$ is the Jacobian of $f(\cdot\,;\Theta)$ evaluated at $\Theta = \mathbf{m}$. Then the mean and covariance of the distribution over the linearized mapping $\tilde{f}$ at $X, X' \in \mathcal{X}$ are given by

$$\begin{aligned}\mathbb{E}[\tilde{f}(X;\Theta)] &= f(X;\mathbf{m}) \\ \mathrm{Cov}[\tilde{f}(X;\Theta), \tilde{f}(X';\Theta)] &= \mathcal{J}(X;\mathbf{m})\mathbf{S}\mathcal{J}(X';\mathbf{m})^\top.\end{aligned}$$

Proof. We wish to find $\mathbb{E}[\tilde{f}(X;\mathbf{m},\Theta)]$ and

$$\begin{aligned}&\mathrm{Cov}(\tilde{f}(X;\mathbf{m},\Theta), \tilde{f}(X';\mathbf{m},\Theta)) \\ &= \mathbb{E}[(\tilde{f}(X;\mathbf{m},\Theta) - \mathbb{E}[\tilde{f}(X;\mathbf{m},\Theta)])\,(\tilde{f}(X';\mathbf{m},\Theta) - \mathbb{E}[\tilde{f}(X';\mathbf{m},\Theta)])^\top].\end{aligned} \tag{A.1}$$

To see that $\mathbb{E}[\tilde{f}(X;\mathbf{m},\theta)] = f(X;\mathbf{m})$, note that, by linearity of expectation, we have

$$\begin{aligned}\mathbb{E}[\tilde{f}(X;\mathbf{m},\theta)] &= \mathbb{E}[f(X;\mathbf{m}) + \mathcal{J}(X;\mathbf{m})(\Theta - \mathbf{m})] \\ &= f(X;\mathbf{m}) + \mathcal{J}(X;\mathbf{m})(\mathbb{E}[\Theta] - \mathbf{m}) = f(X;\mathbf{m}).\end{aligned} \tag{A.2}$$

To see that $\mathrm{Cov}(\tilde{f}(X;\mathbf{m},\theta),\tilde{f}(X';\mathbf{m},\theta)) = \mathcal{J}(X;\mathbf{m})\mathbf{S}\mathcal{J}(X';\mathbf{m})^\top$, note that in general, for a multivariate random variable $\mathbf{Z}$, $\mathrm{Cov}(\mathbf{Z},\mathbf{Z}) = \mathbb{E}[\mathbf{Z}\mathbf{Z}^\top] - \mathbb{E}[\mathbf{Z}]\mathbb{E}[\mathbf{Z}]^\top$, and hence,

$$\begin{aligned}&\mathrm{Cov}(\tilde{f}(X;\mathbf{m},\Theta),\tilde{f}(X';\mathbf{m},\Theta))\\&= \mathbb{E}[\tilde{f}(X;\mathbf{m},\Theta)\tilde{f}(X';\mathbf{m},\Theta)^\top] - \mathbb{E}[\tilde{f}(X;\mathbf{m},\Theta)]\mathbb{E}[\tilde{f}(X';\mathbf{m},\Theta)]^\top.\end{aligned} \tag{A.3}$$

We already know that $\mathbb{E}[\tilde{f}(X;\Theta)] = f(X;\mathbf{m})$, so we only need to find $\mathbb{E}[\tilde{f}(X;\Theta)\tilde{f}(X';\Theta)^\top]$:

$$\begin{aligned}&\mathbb{E}_{g_\Theta}[\tilde{f}(X;\mathbf{m},\Theta)\tilde{f}(X';\mathbf{m},\Theta)^\top]\\&=\mathbb{E}_{g_\Theta}[(f(X;\mathbf{m}) + \mathcal{J}(X;\mathbf{m})(\Theta-\mathbf{m}))(f(X';\mathbf{m}) + \mathcal{J}(X';\mathbf{m})(\Theta-\mathbf{m}))^\top]\end{aligned} \tag{A.4}$$

$$\begin{aligned}&=\mathbb{E}_{g_\Theta}[f(X;\mathbf{m})f(X';\mathbf{m})^\top + (\mathcal{J}(X;\mathbf{m})(\Theta-\mathbf{m}))(\mathcal{J}(X';\mathbf{m})(\Theta-\mathbf{m}))^\top\\&\qquad + f(X;\mathbf{m})(\mathcal{J}(X';\mathbf{m})(\Theta-\mathbf{m}))^\top + \mathcal{J}(X;\mathbf{m})(\Theta-\mathbf{m})f(X';\mathbf{m})^\top]\end{aligned} \tag{A.5}$$

$$\begin{aligned}&=\mathbb{E}_{g_\Theta}[f(X;\mathbf{m})f(X';\mathbf{m})^\top + \mathcal{J}(X;\mathbf{m})(\Theta-\mathbf{m})(\Theta-\mathbf{m})^\top\mathcal{J}(X';\mathbf{m})^\top\\&\qquad + f(X;\mathbf{m})(\mathcal{J}(X';\mathbf{m})(\Theta-\mathbf{m}))^\top + \mathcal{J}(X;\mathbf{m})(\Theta-\mathbf{m})f(X';\mathbf{m})^\top]\end{aligned} \tag{A.6}$$

$$\begin{aligned}&=f(X;\mathbf{m})f(X';\mathbf{m})^\top + \mathcal{J}(X;\mathbf{m})\mathbb{E}_{g_\Theta}[(\Theta-\mathbf{m})(\Theta-\mathbf{m})^\top]\mathcal{J}(X';\mathbf{m})^\top\\&\qquad + f(X;\mathbf{m})(\mathcal{J}(X';\mathbf{m})(\underbrace{\mathbb{E}_{g_\Theta}[\Theta]-\mathbf{m}}_{=0}))^\top + \mathcal{J}(X;\mathbf{m})(\underbrace{\mathbb{E}_{g_\Theta}[\Theta]-\mathbf{m}}_{=0})f(X';\mathbf{m})^\top,\end{aligned} \tag{A.7}$$

where the last line follows from the definition of g_Θ. By definition of the covariance, we then obtain

$$\begin{aligned}&\mathbb{E}_{g_\Theta}[\tilde{f}(X;\mathbf{m},\Theta)\tilde{f}(X';\mathbf{m},\Theta)^\top]\\&= f(X;\mathbf{m})f(X';\mathbf{m})^\top + \mathcal{J}(X;\mathbf{m})\mathbb{E}_{g_\Theta}[(\Theta-\mathbf{m})(\Theta-\mathbf{m})^\top]\mathcal{J}(X';\mathbf{m})^\top\end{aligned} \tag{A.8}$$

$$= f(X;\mathbf{m})f(X';\mathbf{m})^\top + \mathcal{J}(X;\mathbf{m})\mathrm{Cov}(\Theta)\mathcal{J}(X';\mathbf{m})^\top. \tag{A.9}$$

With this result, we obtain the covariance function

$$\begin{aligned}&\mathrm{Cov}(\tilde{f}(X;\mathbf{m},\Theta),\tilde{f}(X';\mathbf{m},\Theta))\\&= \mathbb{E}[\tilde{f}(X;\mathbf{m},\Theta)\tilde{f}(X';\mathbf{m},\Theta)^\top] - \mathbb{E}[\tilde{f}(X;\mathbf{m},\Theta)]\mathbb{E}[\tilde{f}(X';\mathbf{m},\Theta)]^\top\end{aligned} \tag{A.10}$$

$$= f(X;\mathbf{m})f(X';\mathbf{m})^\top + \mathcal{J}(X;\mathbf{m})\mathrm{Cov}(\Theta)\mathcal{J}(X';\mathbf{m})^\top - \mathbb{E}[\tilde{f}(X;\mathbf{m},\Theta)]\mathbb{E}[\tilde{f}(X';\mathbf{m},\Theta)]^\top \tag{A.11}$$

$$= f(X;\mathbf{m})f(X';\mathbf{m})^\top + \mathcal{J}(X;\mathbf{m})\mathrm{Cov}(\Theta)\mathcal{J}(X';\mathbf{m})^\top - f(X;\mathbf{m})f(X';\mathbf{m})^\top \tag{A.12}$$

$$= \mathcal{J}(X;\mathbf{m})\mathrm{Cov}(\Theta)\mathcal{J}(X';\mathbf{m})^\top. \tag{A.13}$$

Finally, $\mathrm{Cov}(\Theta) = \mathbf{S}$ yields $\mathrm{Cov}(\tilde{f}(X;\mathbf{m},\Theta),\tilde{f}(X';\mathbf{m},\Theta)) = \mathcal{J}(X;\mathbf{m})\mathbf{S}\mathcal{J}(X';\mathbf{m})^\top$. This concludes the proof. □

Proposition 3.2. *For a stochastic function $f(\cdot\,;\Theta)$ defined in terms of stochastic parameters Θ distributed according to distribution $g_\Theta = \mathcal{N}(\mathbf{m}, \mathbf{S})$, denote the linearization of the stochastic function $f(\cdot\,;\Theta)$ about $\mathbf{m}$ by*

$$f(\cdot\,;\Theta) \approx \tilde{f}(\cdot\,;\mathbf{m},\Theta) \stackrel{\text{def}}{=} f(\cdot\,;\mathbf{m}) + \mathcal{J}(\cdot\,;\mathbf{m})(\Theta - \mathbf{m}),$$

where $\mathcal{J}(\cdot\,;\mathbf{m}) \stackrel{\text{def}}{=} (\partial f(\cdot\,;\Theta)/\partial\Theta)|_{\Theta=\mathbf{m}}$ is the Jacobian of $f(\cdot\,;\Theta)$ evaluated at $\Theta = \mathbf{m}$. Then, for a partition of the set of parameters into sets α and β, a distribution $g_\Theta = \mathcal{N}(\mathbf{m}, \mathbf{S})$ with $\Theta_\alpha \perp \Theta_\beta$, the distribution $\tilde{g}_{\tilde{f}(X;\Theta)}$ can be approximated via the Monte Carlo estimator

$$\hat{\tilde{g}}_{\tilde{f}(X;\mathbf{m},\Theta)} = \frac{1}{R}\sum_{j=1}^{R} \mathcal{N}\Big(f(X;\mathbf{m}) + \tilde{f}_\alpha(X;\mathbf{m},\Theta_\alpha)^{(j)}, \mathcal{J}_\beta(X;\mathbf{m})\mathbf{S}_\beta\mathcal{J}_\beta(X;\mathbf{m})^\top\Big), \tag{A.14}$$

where $g_{\Theta_\alpha} = \mathcal{N}(\mathbf{m}_\alpha, \mathbf{S}_\alpha)$, $g_{\Theta_\beta} = \mathcal{N}(\mathbf{m}_\beta, \mathbf{S}_\beta)$, and

$$\tilde{f}_\alpha(\cdot\,;\mathbf{m},\Theta_\alpha) \stackrel{\text{def}}{=} \mathcal{J}_\alpha(\cdot\,;\mathbf{m})(\Theta_\alpha - \mathbf{m}_\alpha), \tag{A.15}$$

with $\mathcal{J}_\alpha(\cdot\,;\mathbf{m})$ denoting the columns of the Jacobian matrix corresponding to the set of parameters α, and with $\tilde{f}_\alpha(X;\mathbf{m},\Theta_\alpha)^{(j)}$, for $j = 1,\ldots,R$, obtained by sampling parameters from the distribution $g_{\Theta_\alpha} = \mathcal{N}(\mathbf{m}_\alpha, \mathbf{S}_\alpha)$.

Proof. Consider a partition of the set of parameters into sets α and β and express the linearized mapping as

$$\tilde{f}(\cdot\,;\mathbf{m},\Theta) = \tilde{f}_\alpha(\cdot\,;\mathbf{m},\Theta_\alpha) + \tilde{f}_\beta(\cdot\,;\mathbf{m},\Theta_\beta), \tag{A.16}$$

with

$$\tilde{f}_\alpha(\cdot\,;\mathbf{m},\Theta_\alpha) \stackrel{\text{def}}{=} \mathcal{J}_\alpha(\cdot\,;\mathbf{m})(\Theta_\alpha - \mathbf{m}_\alpha), \tag{A.17}$$

and

$$\tilde{f}_\beta(\cdot\,;\mathbf{m},\Theta_\beta) \stackrel{\text{def}}{=} f(\cdot\,;\mathbf{m}) + \mathcal{J}_\beta(\cdot\,;\mathbf{m})(\Theta_\beta - \mathbf{m}_\beta), \tag{A.18}$$

where $\mathcal{J}_\alpha(\cdot\,;\mathbf{m})$ and $\mathcal{J}_\beta(\cdot\,;\mathbf{m})$ are the columns of the Jacobian matrix corresponding to the sets of parameters α and β, respectively, and Θ_α and Θ_β are the corresponding random parameter vectors.

Noting that Equation (A.16) expresses $\tilde{f}$ as a sum of (affine transformations of) random variables, we can use the fact that for independent Gaussian random variables $\mathbf{U}$ and $\mathbf{V}$, the distribution $h_\mathbf{Z}$ of $\mathbf{Z} = \mathbf{U} + \mathbf{V}$ is equal to the convolution of the distributions h_X and $h_\mathbf{Y}$ to obtain an approximation to $\tilde{f}$. In particular, for $\mathbf{Z} = \mathbf{U} + \mathbf{V}$, with

$$f_\mathbf{Z}(\mathbf{z}) = \int_{-\infty}^{\infty} f_\mathbf{V}(\mathbf{z} - \mathbf{u}) f_\mathbf{U}(\mathbf{u})\,\mathrm{d}\mathbf{u}. \tag{A.19}$$

Letting $\mathbf{U} = \tilde{f}_\alpha(X;\mathbf{m},\Theta_\alpha)$, $\mathbf{V} = \tilde{f}_\beta(X;\mathbf{m},\Theta_\beta)$, and $\mathbf{Z} = \tilde{f}(X;\mathbf{m},\Theta)$, we can write

$$\begin{aligned}
&\tilde{g}_{\tilde{f}(X;\mathbf{m},\Theta)}(\tilde{f}(X;\mathbf{m},\theta)) \\
&= \int_{-\infty}^{\infty} \tilde{g}_{\tilde{f}_\beta(X;\mathbf{m},\Theta_\beta)}(\tilde{f}(X;\mathbf{m},\theta) - \tilde{f}_\alpha(X;\mathbf{m},\theta_\alpha))\tilde{g}_{\tilde{f}_\alpha(X;\mathbf{m},\Theta_\alpha)}(\tilde{f}_\alpha(X;\mathbf{m},\theta_\alpha))\,\mathrm{d}\mathbf{u}, \qquad \text{(A.20)} \\
&= \int_{-\infty}^{\infty} \mathcal{N}(\tilde{f}(X;\mathbf{m},\theta)\,;\boldsymbol{\mu}(X;\mathbf{m},\theta_\alpha,\tilde{f}_\alpha),\boldsymbol{\Sigma}(X;\mathbf{m},\mathbf{S}_\beta,\mathcal{J}_\beta))\tilde{g}_{\tilde{f}_\alpha(X;\mathbf{m},\Theta_\alpha)}(\tilde{f}_\alpha(X;\mathbf{m},\theta_\alpha))\,\mathrm{d}X, \qquad \text{(A.21)}
\end{aligned}$$

with

$$\boldsymbol{\mu}(X;\mathbf{m},\theta_\alpha,\tilde{f}_\alpha) = f(X;\mathbf{m}) + \tilde{f}_\alpha(X;\mathbf{m},\theta_\alpha) \qquad \text{(A.22)}$$

and

$$\boldsymbol{\Sigma}(X;\mathbf{m},\mathbf{S}_\beta,\mathcal{J}_\beta) = \mathcal{J}_\beta(X;\mathbf{m})\mathbf{S}_\beta\mathcal{J}_\beta(X;\mathbf{m})^\top, \qquad \text{(A.23)}$$

where we have used the fact that for a Gaussian distribution with mean m and covariance S, $\mathcal{N}(z-y;m,S) = \mathcal{N}(z;m+y,S)$. We can then approximate the probability density function $\tilde{g}_{\tilde{f}(X;\mathbf{m},\Theta)}(\tilde{f}(X;\theta))$ via the Monte Carlo estimator

$$\begin{aligned}
&\hat{\tilde{g}}_{\tilde{f}(X;\mathbf{m},\Theta)}(\tilde{f}(X;\mathbf{m},\theta)) \\
&= \frac{1}{R}\sum\nolimits_{j=1}^{R} \mathcal{N}(\tilde{f}(X;\mathbf{m},\theta)\,;\boldsymbol{\mu}(X;\mathbf{m},\theta_\alpha^{(j)},\tilde{f}_\alpha),\boldsymbol{\Sigma}(X;\mathbf{m},\mathbf{S}_\beta,\mathcal{J}_\beta))
\end{aligned} \qquad \text{(A.24)}$$

with $\tilde{f}_\alpha(X;\mathbf{m},\theta_\alpha)^{(j)} \sim \tilde{g}_{\tilde{f}_\alpha(X;\mathbf{m},\Theta_\alpha)}$. Finally, we can express the distribution $\hat{\tilde{g}}_{\tilde{f}(X;\mathbf{m},\Theta)}$ as

$$\hat{\tilde{g}}_{\tilde{f}(X;\mathbf{m},\Theta)} = \frac{1}{R}\sum\nolimits_{j=1}^{R} \mathcal{N}\Big(f(X;\mathbf{m}) + \tilde{f}_\alpha(X;\mathbf{m},\Theta_\alpha)^{(j)}, \mathcal{J}_\beta(X;\mathbf{m})\mathbf{S}_\beta\mathcal{J}_\beta(X;\mathbf{m})^\top\Big), \qquad \text{(A.25)}$$

where $g_{\Theta_\beta} = \mathcal{N}(\mathbf{m}_\beta,\mathbf{S}_\beta)$ and samples $\tilde{f}_\alpha(X;\mathbf{m},\Theta_\alpha)^{(j)}$ are obtained by sampling parameters from the distribution $g_{\Theta_\alpha} = \mathcal{N}(\mathbf{m}_\alpha,\mathbf{S}_\alpha)$. This concludes the proof. □

A.2 Experimental Details: Robustness to Distribution Shifts

A.2.1 Hyperparameter Selection Protocol

For FSVI, we used a holdout validation set (10% of the training set) to conduct a hyperparameter search over the prior variance, the number of context points used to evaluate the KL divergence, the context distribution, and the number of Monte Carlo samples used to evaluate the expected log-likelihood. We selected the set of hyperparameters that yielded the highest validation log-likelihood for all experiments. We state the hyperparameters selected for the different datasets below.

For other methods, we used a holdout validation set of the same size and selected the best-performing hyperparameters. We used implementations provided by the authors of MFVI (radial) and SWAG. All other methods were implemented from scratch unless stated otherwise.

A.2.2 FashionMNIST vs. MNIST/NotMNIST

We train all models on the FashionMNIST dataset and evaluate the models' predictive uncertainty performance on out-of-distribution data on the MNIST dataset. Both datasets consist of images of size 28×28 pixels. The FashionMNIST dataset is normalized to have zero mean and a standard deviation of one. The MNIST dataset is normalized with the same transformation, that is, using the same mean and standard deviation used for the in-distribution data. We chose FashionMNIST/MNIST instead of MNIST/NotMNIST because the latter is notably easier than the former.

In this experiment, a network architecture with two convolutional layers of 32 and 64 3×3 filters and a fully connected final layer of 128 hidden units is used. A max pooling operation is placed after each convolutional layer and ReLU activations are used. We do not use batch normalization. All models are trained for 30 epochs with a mini-batch size of 128 using SGD with a learning rate of 5×10^{-3}, momentum (with momentum parameter 0.9), and a cosine learning rate schedule with parameter 0.05.

For FSVI with p_{X_c} =random monochrome, we sampled 50% of the context points for each gradient step from the mini-batch and the other 50% according to the method described in Appendix A.2.7. For FSVI with p_{X_c}= KMNIST, we used the KMNIST dataset.

A.2.3 CIFAR-10 vs. SVHN

We train all models on the CIFAR-10 dataset and evaluate the models' predictive uncertainty performance on out-of-distribution data on the SVHN dataset. Both datasets consist of images of size $32 \times 32 \times 3$, with RGB channels. The CIFAR-10 dataset is normalized to have zero mean and a standard deviation of one. The SVHN dataset is normalized with the same transformation, that is, using the same mean and standard deviation used for the in-distribution data. The training data is augmented with random horizontal flips (with a probability of 0.5) and random crops (4 zero pixels on all sides).

In this experiment, a standard ResNet-18 network architecture was used. All models are trained for 200 epochs with a mini-batch size of 128 using SGD with a learning rate of 5×10^{-3}, momentum (with momentum parameter 0.9), and a cosine learning rate schedule with parameter 0.05.

For FSVI with p_{X_c} =random monochrome, we sampled 50% of the context points for each gradient step from the mini-batch and the other 50% according to the method described in Appendix A.2.7. For FSVI with p_{X_c}= CIFAR-100, we used the CIFAR-100 dataset.

A.2.4 Diabetic Retinopathy Diagnosis

Prediction and Expert Referral. In real-world settings where the evaluation data may be sampled from a shifted distribution, incorrect predictions may become increasingly likely. To account for that possibility, predictive uncertainty estimates can be used to identify datapoints where the likelihood of an incorrect prediction is particularly high and refer them for further review. We consider a corresponding selective prediction task, where the predictive performance of a given model is evaluated for varying expert referral rates. That is, for a given referral rate of $\gamma \in [0, 1]$, a model's predictive uncertainty is used to identify the γ proportion of images in the evaluation set for which the model's predictions are most uncertain. Those images are referred to a medical professional for further review, and the model is assessed on its predictions on the remaining $(1 - \gamma)$ proportion of images. By repeating this process for all possible referral rates and assessing the model's predictive performance on the retained images, we estimate how reliable it would be in a safety-critical downstream task, where predictive uncertainty estimates are used in conjunction with human expertise to avoid harmful predictions. Importantly, selective prediction tolerates out-of-distribution examples. For example, even if unfamiliar features appear in certain images, a model with reliable uncertainty estimates will perform better in selective prediction by assigning these images high epistemic (and predictive) uncertainty, therefore referring them to an expert at a lower γ.

For all methods, experiments are performed using a ResNet-50 network architecture. Training and evaluation scripts as well as model checkpoints can be found at

`github.com/google/uncertainty-baselines/.../diabetic_retinopathy_detection`.

A.2.5 Two Moons and 1D Regression

For the *Two-Moons* experiment, we use a multi-layer perceptron (MLP) consisting of two fully-connected layers with 30 hidden units each and tanh activations. We train all models with a learning rate of 10^{-3}.

For FSVI, we sampled context points uniformly from $[-10, 10] \times [-10, 10]$.

For the *1D regression* experiment, we use a multi-layer perceptron (MLP) consisting of two fully-connected layers with 100 hidden units each and ReLU activations.

For FSVI, we sampled context points uniformly from $[-10, 10]$.

A.2.6 Further Implementation Details

We use the Adam optimizer with settings of $\beta_1 = 0.9$, $\beta_2 = 0.99$ and $\epsilon = 10^{-8}$ for all experiments. The deterministic neural networks that were used for the ensemble were trained with a weight decay of $\lambda = 10^{-1}$. MFVI (tempered) was trained with a KL scaling factor of 0.1 to obtain a cold posterior.

A.2.7 Selection of Context Distribution

We estimate the supremum at every gradient step by sampling a set of context points X_c from a distribution p_{X_c}. For tasks with image inputs, we construct a distribution p_{X_c}, defined as a uniform distribution over images with monochromatic channels. To generate a sample from this "monochrome images" distribution, we first take all images in the training data, flatten each channel, and stack the flattened image channels into a single vector each. We then draw a random element (i.e., a pixel) from each channel vector and then use these pixels to generate a monochrome image of a given resolution by setting every channel equal to the value of the pixel that was drawn. For regression tasks with a D-dimensional input space, p_{X_c} is defined as a uniform distribution with lower and upper bounds set to the empirical lower and upper bounds of the training data.

A.3 Further Empirical Results: Robustness to Distribution Shifts

A.3.1 Tabular Results for Diabetic Retinopathy Diagnosis Tasks

Table A.1: Country Shift. Prediction and uncertainty quality of baseline methods in terms of the area under the receiver operating characteristic curve (AUC) and classification accuracy, as a function of the proportion of data referred to a medical expert. All methods are tuned on in-domain validation AUC, and ensembles have $K = 3$ constituent models (true for all subsequent tables unless specified otherwise). On in-domain data, MC DROPOUT performs best across all thresholds. On distributionally shifted data, no method consistently performs best.

	No Referral		50% Data Referred		70% Data Referred	
Method	**AUC (%) ↑**	**Accuracy (%) ↑**	**AUC (%) ↑**	**Accuracy (%) ↑**	**AUC (%) ↑**	**Accuracy (%) ↑**
EyePACS Dataset (In-Domain)						
MAP (Deterministic)	87.4±1.3	88.6±0.7	91.1±1.8	95.9±0.4	94.9±1.1	96.5±0.3
MFVI	83.3±0.2	85.7±0.1	85.5±0.7	94.5±0.1	88.2±0.7	95.9±0.1
RADIAL-MFVI	83.2±0.5	74.2±5.0	88.9±0.9	81.8±6.0	91.2±1.3	83.8±5.5
FSVI	88.5±0.1	89.8±0.0	91.0±0.4	96.4±0.0	94.3±0.3	97.2±0.1
MC DROPOUT	91.4±0.2	90.9±0.1	95.3±0.2	97.4±0.1	97.4±0.1	98.1±0.0
RANK-1	85.6±1.4	87.7±0.8	87.1±2.3	95.3±0.5	90.9±2.0	96.4±0.4
DEEP ENSEMBLE	90.3±0.2	90.3±0.3	91.7±0.6	97.2±0.0	95.0±0.5	97.9±0.0
MFVI ENSEMBLE	85.4±0.0	87.8±0.0	86.3±0.4	95.4±0.0	89.2±0.4	96.7±0.1
RADIAL-MFVI ENSEMBLE	84.9±0.1	74.2±1.5	91.4±0.2	83.4±1.7	93.3±0.3	85.9±1.6
FSVI ENSEMBLE	90.3±0.1	90.6±0.0	92.1±0.2	97.1±0.0	95.2±0.2	97.8±0.1
MC DROPOUT ENSEMBLE	**92.5±0.0**	**91.6±0.0**	**95.8±0.1**	**97.8±0.0**	**97.7±0.1**	**98.4±0.0**
RANK-1 ENSEMBLE	89.5±0.8	89.3±0.4	88.5±1.3	96.9±0.3	91.6±1.2	97.6±0.3
APTOS 2019 Dataset (Population Shift)						
MAP (Deterministic)	92.2±0.2	86.2±0.6	80.1±3.6	87.6±1.5	55.4±4.3	85.4±1.2
MFVI	91.4±0.2	84.1±0.3	93.8±0.4	92.1±0.5	93.0±0.6	92.7±0.5
RADIAL-MFVI	90.7±0.7	71.8±4.6	82.0±2.5	81.5±2.7	66.4±2.1	85.9±1.0
FSVI	94.1±0.1	87.6±0.5	90.6±0.9	90.7±0.7	77.2±4.6	89.8±0.3
MC DROPOUT	94.0±0.2	86.8±0.2	87.4±0.3	88.1±0.2	65.3±1.7	88.2±0.4
RANK-1	92.5±0.3	86.2±0.5	90.1±2.5	91.4±1.1	75.1±7.8	89.5±1.5
DEEP ENSEMBLE	94.2±0.2	87.5±0.1	91.2±1.9	92.4±0.9	67.4±7.3	90.1±1.2
MFVI ENSEMBLE	93.2±0.1	87.0±0.2	**94.9±0.3**	**93.7±0.3**	**94.2±0.3**	**94.0±0.3**
RADIAL-MFVI ENSEMBLE	91.8±0.2	69.0±1.9	78.6±0.6	79.8±0.9	60.9±0.3	86.7±0.2
FSVI ENSEMBLE	**94.6±0.1**	**88.9±0.2**	90.7±0.5	91.1±0.6	74.1±3.4	89.8±0.2
MC DROPOUT ENSEMBLE	94.1±0.1	87.6±0.1	86.8±0.2	88.0±0.2	62.3±0.4	87.7±0.2
RANK-1 ENSEMBLE	94.1±0.2	88.3±0.2	**94.9±0.4**	93.5±0.3	92.4±1.5	93.8±0.3

A.3.2 UCI Regression

Table A.2: This table compares the predictive performance between the method proposed here and the method proposed by Sun et al. (2019) on six datasets from the UCI database. We followed the same training protocol as Sun et al. (2019) and used the code provided by the authors to load and process the data. The same network architecture was used (one hidden layer with 50 hidden units). We report the results for the best set of hyperparameters, computed over ten random seeds. Lower RMSE and higher log-likelihood are better. Best results are shaded in gray. The first five rows are small-scale UCI experiments, and the sixth row ("Protein") is a larger-scale experiment (45,740 data points).

	RMSE		Log-Likelihood	
	Sun et al. (2019)	**Ours**	Sun et al. (2019)	**Ours**
Boston	$\mathbf{2.378 \pm 0.104}$	3.632 ± 0.515	$\mathbf{-2.301 \pm 0.038}$	-3.150 ± 0.495
Concrete	4.935 ± 0.180	$\mathbf{4.177 \pm 0.443}$	-3.096 ± 0.016	$\mathbf{-2.855 \pm 0.116}$
Energy	0.412 ± 0.017	$\mathbf{0.409 \pm 0.060}$	-0.684 ± 0.020	$\mathbf{-0.539 \pm 0.138}$
Wine	0.673 ± 0.014	$\mathbf{0.615 \pm 0.033}$	-1.040 ± 0.013	$\mathbf{-0.959 \pm 0.034}$
Yacht	0.607 ± 0.068	$\mathbf{0.514 \pm 0.242}$	-1.033 ± 0.033	$\mathbf{-0.888 \pm 0.334}$
Protein	4.326 ± 0.019	$\mathbf{4.248 \pm 0.043}$	-2.892 ± 0.004	$\mathbf{-2.866 \pm 0.009}$

A.4 Experimental Details: Continual Learning

Our empirical evaluation centers on six sequences of classification tasks: a synthetic sequence of binary-classification tasks with 2D inputs; split MNIST; split Fashion MNIST; permuted MNIST; split CIFAR; and sequential Omniglot. With the exception of permuted MNIST, each of these task sequences can be tackled by a neural network with either a multi-head setup (MH) or a single-head setup (SH). In a multi-head setup, the neural network has a separate output layer (or head) for each task, and task identifiers are provided at test time in order to select the appropriate head. In a single-head setup, the neural network has just one output layer shared across all tasks, and task identifiers are not provided. In our experiments, we use multi-head setups for split Fashion MNIST, split CIFAR, and sequential Omniglot, and single-head setups for the synthetic task sequence along with permuted MNIST. For split MNIST, we run both setups.

A.4.1 Illustrative Example

The task sequence shown in Figure 3.7 was created by Pan et al. (2020). Each of the five tasks in this sequence involves binary classification on 2D inputs, where the number of training examples per task is 3,600. Following Pan et al. (2020), we use a fully connected neural network with an input layer of size 2, two hidden layers of size 20, and an output layer of size 2. When running S-FSVI, we set the prior covariance as $\Sigma_0 = 0.1$ and train the neural network for 250 epochs on each task. We use the Adam optimizer with an initial learning rate of 0.0005 ($\beta_1 = 0.9, \beta_2 = 0.999$) and a batch size of 128. The coreset is constructed by choosing 40 samples from the training data for each task. To evaluate the KL divergence between the posterior and the prior distributions over functions, for each previous task we sample 20 input points from the context set and generate another 30 samples by sampling each pixel uniformly from the range $[-4, 4]$. For example, when we train the model on task $t \in \{1, 2, 3, \ldots\}$, we use $20(t-1)$ samples chosen from the context set and $30t$ white-noise samples. The noise samples encourage the neural network to preserve high predictive uncertainty in regions far from the training data.

A.4.2 Task Sequences Based on (Fashion) MNIST

Split MNIST consists of five tasks, where each task is binary classification on a pair of MNIST classes. Split Fashion MNIST has the same form but uses data from Fashion MNIST. Permuted MNIST comprises ten tasks, where each task involves classifying images into the ten MNIST classes after the image pixels have been randomly reordered. Unless specified otherwise, the following setups apply to Figures 3.10 to 3.12, Figure A.1, and Table 3.3.

Dataset. In all cases, 60,000 data samples are used for training and 10,000 data samples are used for testing. The input images are converted to floating-point numbers with values in the range $[0, 1]$.

Neural-Network Size & Coreset Size. To ensure a fair comparison, all methods in Table 3.3 (unless where explicitly indicated otherwise) use the same neural-network size and (where applicable) coreset size. As in prior work (Pan et al., 2020; Titsias et al., 2020), we use fully connected neural networks, with two hidden layers of size 100 for permuted MNIST and two hidden layers of size 256 for split (Fashion) MNIST. In all cases, the ReLU activation function is applied to non-output units. For single-head setups, we use 200 coreset points; for multi-head setups, we use 40 points.

Coreset Selection. For S-FSVI with a coreset, when training on the first task, 40 context points are generated by sampling each pixel uniformly from the range $[0, 1]$; during training on subsequent tasks, 40 context points are chosen randomly from the context set. For S-FSVI without a coreset, 40 context points are chosen uniformly randomly from the training data of the current task (corresponding to the "Random" label in Figure 3.12).

Prior Distribution. For the first task, S-FSVI uses a prior distribution over functions with fixed mean and diagonal covariance. When using a coreset, the prior distribution is assumed to be Gaussian with zero mean and a diagonal covariance of magnitude 0.001. When not using a coreset, the prior distribution is assumed to be Gaussian with zero mean and a diagonal covariance of magnitude 100. The prior variance is optimized via hyperparameter selection on a validation set.

Optimization. We use the Adam optimizer with an initial learning rate of 0.0005 ($\beta_1 = 0.9, \beta_2 = 0.999$). The number of epochs on each task is 60 for split MNIST (MH), 60 for split Fashion MNIST (MH), 10 for permuted MNIST (SH), and 80 for split MNIST (SH). The batch size is 128.

Prediction. The predictive distribution used for computing the expected log-likelihood is estimated using five Monte Carlo samples.

Hyperparameter Selection. For "S-FSVI (optimized)" in Table 3.3, we used the optimized hyperparameters chosen on a validation set after exploring the configurations shown in Table A.3. For cases where no configuration is significantly better than the rest, the default value given in Appendix A.4.2 is used.

Table A.3: Hyperparameter selection. Optimal values (in bold) were chosen based on validation-set accuracy.

Task Sequences	Number of Layers & Units	Magnitude of Prior Variance	Number of Epochs
Split MNIST (MH)	{**1**, 2} * {100, 200, 300, **400**}	{**0.001**, 0.01, 0.1, 1, 10, 100}	{40, **60**, 80, 120, 160}
Split Fashion MNIST (MH)	{**4**} * {50, **200**, 300, 400}	{**0.001**, 0.01, 0.1, 1, 10, 100}	{40, **60**, 80, 120, 160}
Permuted MNIST (SH)	{**2**} * {100, 200, 400, **500**}	{**0.001**, 0.01, 0.1, 1, 10, 100}	{10, **20**, 40, 60, 80}
Split MNIST (SH)	{**1**, 2} * {100, 200, 300, **400**}	{**0.001**, 0.01, 0.1, 1, 10, 100}	{60, **80**, 120, 160, 240}

A.4.3 Split CIFAR

Split CIFAR, as described in Pan et al. (2020), consists of six tasks. The first is ten-way classification on the full CIFAR-10 dataset. Each of the following five is also ten-way classification, with classes drawn from CIFAR-100. Following Pan et al. (2020), we use a neural network with four convolutional layers followed by two fully connected layers followed by multiple output heads (one for each task). For S-FSVI, we use the following setup: Adam optimizer with learning rate 0.0005, prior with covariance 0.01, random coreset selection, 200 coreset points per task, and 50 context points at each task. We also use this setup (and a training duration of 2,000 epochs) when training individual neural networks for the "separate" baseline.

A.4.4 Sequential Omniglot

Sequential Omniglot, as described in Schwarz et al. (2018), comprises 50 classification tasks. Each task is associated with an alphabet, and the number of characters (classes) varies between alphabets. Following Schwarz et al. (2018), we use a neural network with four convolutional layers followed by one fully connected layer. For S-FSVI, we use two coreset points per character, as used by Titsias et al. (2020). The coreset points are sampled from the training set with probability proportional to the entropy of the neural network's posterior predictive distribution. To limit memory usage, we draw no more than 25 context points from the context set at each gradient step after task 25. We use a learning rate of 0.001 and a prior covariance of 1.0. For the first task, the neural network trains for 200 epochs; for subsequent tasks, it trains for ten epochs per task. We use the same data augmentation and train-test split as Titsias et al. (2020).

A.4.5 Coreset-Selection Methods

We consider different distributions from which to sample points to be added to the coreset. For each of the scoring methods below, we use the scores to create a probability mass function from which points can be sampled.

Random. Points are sampled uniformly from the training data.

Predictive-Entropy Scoring. Points are scored according to the total predictive uncertainty (i.e., the predictive entropy) of the model. For a model with stochastic parameters Θ, pre-likelihood outputs $f(X;\theta)$, and a likelihood function $p(y \mid f(X;\theta))$,

the predictive entropy is given by $\mathcal{H}(\mathbb{E}[p(y \mid f(X;\theta))])$ (Cover et al., 1991; Shannon et al., 1949). The expectation is taken with respect to the model parameters.

Evidence-Lower-Bound Scoring. Points are scored according to the value of the evidence lower bound (ELBO) given in Equation (3.43).

Kullback–Leibler-Divergence Scoring. Points are scored according to the value of the approximation to the function-space KL divergence, that is, the second term in Equation (3.43).

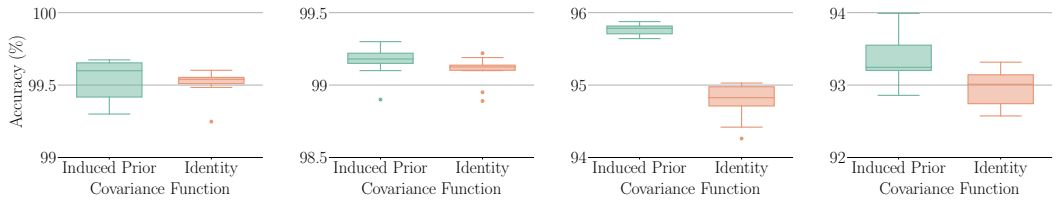
Score-Based Distributions. After scoring with the above methods, points are added to the coreset by sampling from one of the following probability mass functions:

$$\textbf{Lowest:} \quad \mathbb{P}(i) \stackrel{\text{def}}{=} \frac{\bar{s}_i}{\sum_{j=1}^{N} \bar{s}_j} \qquad \text{and} \qquad \textbf{Highest:} \quad \mathbb{P}(i) \stackrel{\text{def}}{=} \frac{s_i}{\sum_{j=1}^{N} s_j}, \tag{A.26}$$

where s_i is the score of the i-th point, $\bar{s}_i = \max_{j=1}^{N} s_j - s_i$, and N is the number of candidate points.

A.5 Further Empirical Results: Continual Learning

A.5.1 Model Design



(a) S-MNIST (MH) **(b)** S-FMNIST (MH) **(c)** P-MNIST (SH) **(d)** S-MNIST (SH)

Figure A.1: Effect of Empirical Prior Covariance on Performance under Sequential Function-Space Variational Inference. Comparison of predictive performance under the induced prior covariance function $\mathbf{K}^{p_t} = \text{diag}\left(\mathcal{J}_{\mu_{t-1}}(x)\Sigma_{t-1}\mathcal{J}_{\mu_{t-1}}(x')^\top\right)$ (left) vs. an identity covariance function (right).

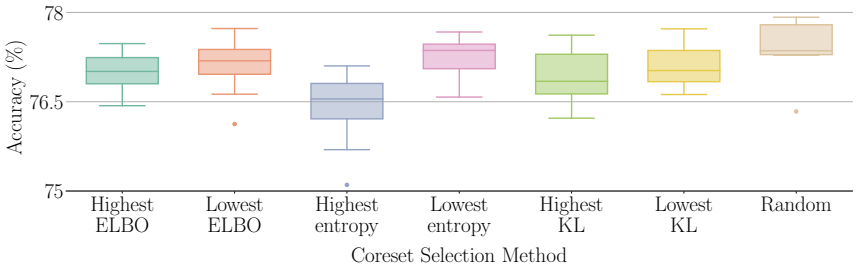
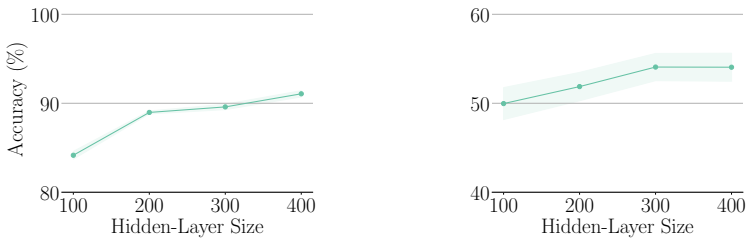
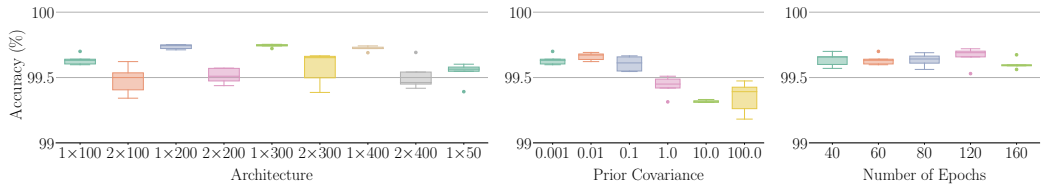


Figure A.2: Effect of Different Coreset-Selection Methods on the Performance of Sequential Function-Space Variational Inference on the Split CIFAR Benchmark. For score-based coreset-selection methods, we first score each coreset point—using Equation (3.43) for ELBO scoring, the predictive entropy for entropy scoring, and the KL divergence in Equation (3.43) for KL scoring—then sample context points from the coreset according to the probability mass function defined in Equation (A.26).

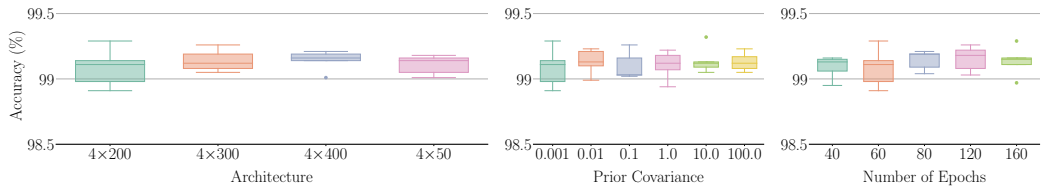


(a) Permuted MNIST (SH) **(b)** Split MNIST (SH)

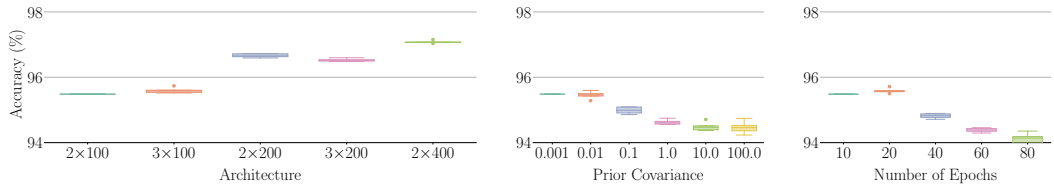
Figure A.3: Effect of Neural Network Size on the Performance of Sequential Function-Space Variational Inference with Minimal Coresets. Predictive accuracy under S-FSVI on permuted MNIST (SH) and split MNIST (SH) as a function of network width, using only a minimal coreset of one sample per class, selected randomly.



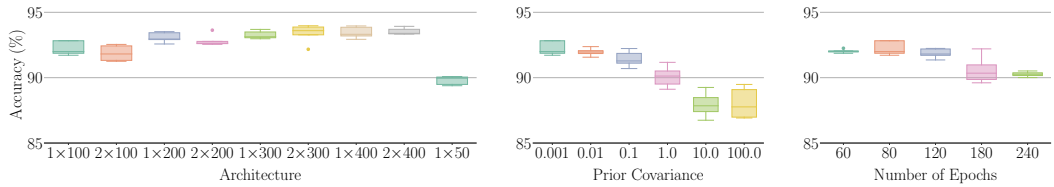
(a) S-MNIST (MH)



(b) S-FMNIST (MH)



(c) P-MNIST (SH)



(d) S-MNIST (SH)

Figure A.4: Effect of Neural-Network Size, First-Task Prior Covariance, and the Number of Training Epochs on Sequential Function-Space Variational Inference. We explore settings of neural-network size (e.g., 2×100 means a fully connected neural network with two hidden layers of size 100), initial prior covariance, and the number of training epochs for each task. To limit the computational resources required, we vary the values of one hyperparameter at a time instead of carrying out a full grid search.

A.5.2 Hyperparameter Search

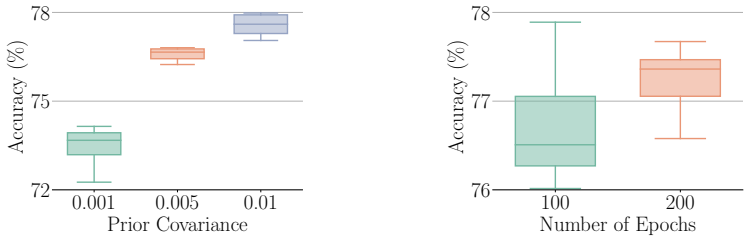


Figure A.5: Sensitivity Study into the Effect of Different Hyperparameters on the Performance of Sequential Function-Space Variational Inference Evaluated on Split CIFAR We explore different settings of the initial first-task prior covariance and the number of epochs for the first task. To limit the computational resources required, we vary the values of one hyperparameter at a time instead of carrying out a full grid search.

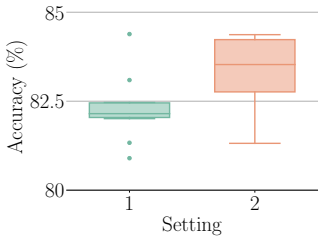


Figure A.6: Hyperparameter Search on Sequential Omniglot. We compare two settings. In the first, we always sample one context point for each previous task from the context set at each gradient step. In the second, we sample a larger number of context points (with a budget of 60 samples per gradient step) from the context set when learning on the first 25 tasks.

B

Appendix B: Appendix to Chapter 4

B.1 Experimental Details: Reliable Uncertainty Quantification

B.1.1 Training Details

Image Tasks. We use the SGD optimizer with momentum of 0.9 and a learning rate of 0.005 with a batch size of 128 and a context batch size of 32 for methods with a context dataset. We train all models for 50 epochs with a cosine annealing learning rate scheduler and use the parameters at the last epoch to evaluate the models.

Language Tasks. We use the AdamW optimizer with a learning rate of 0.0001 with a batch size of 32 and a context batch size of 8 for methods with context datasets. We train our models for three epochs with a linear learning rate scheduler and use the parameters at the last epoch to evaluate our models.

Shared Details. We use Monte Carlo sampling with one sample during training and ten samples during evaluation for all non-deterministic methods.

Sweep Protocol. We keep the same training setting for ERM training as in the state-of-the-art results reported by Shwartz-Ziv et al. (2022). For our method, we sweep the covariance scale from 1×10^{-5} to 1×10^{-3} and report the best hyperparameter as our result.

B.2 Further Empirical Results: Reliable Uncertainty Quantification

B.2.1 Tabular Results

Dataset	Method	Accuracy(↑)	Sel. Acc.(↑)	NLL(↓)	ECE(↓)	Det. AUROC(↑)
CIFAR-10	ERM	96.45±0.08	76.46±0.27	0.17±0.01	3.42±0.13	94.96±0.72
	PTYL	**97.35**±0.34	82.48±0.92	**0.11**±0.01	3.10±0.27	96.94±1.92
	Ours	**97.19**±0.08	**83.42**±0.03	**0.11**±0.01	**1.68**±0.08	**99.86**±0.04
CIFAR-100	ERM	**85.76**±0.20	71.99±1.01	0.67±0.02	8.61±0.26	86.57±1.21
	PTYL	**85.82**±0.23	72.10±1.69	0.68±0.01	8.65±0.56	88.23±2.98
	Ours	**85.69**±0.13	**77.61**±0.10	**0.62**±0.01	**7.22**±0.20	**98.93**±0.22
Flowers	ERM	89.64±0.24	78.87±0.62	0.48±0.02	11.00±0.47	95.68±1.85
	PTYL	89.73±0.51	77.88±0.34	0.47±0.01	10.99±0.64	92.30±0.57
	Ours	**90.35**±0.18	**79.68**±0.08	**0.45**±0.02	**10.07**±0.18	**97.81**±0.10
MultiNLI	ERM	**84.65**±0.13	72.18±0.42	**0.34**±0.01	**6.14**±0.11	81.15±0.42
	PTYL	**84.60**±0.19	73.17±0.23	**0.35**±0.01	**6.15**±0.09	84.27±0.35
	Ours	**84.64**±0.06	**79.71**±0.10	**0.34**±0.01	**6.08**±0.06	**96.90**±0.34
QNLI	ERM	**91.31**±0.10	75.41±0.13	**0.51**±0.01	**8.44**±0.20	94.28±0.59
	PTYL	**91.29**±0.11	75.88±0.15	**0.52**±0.02	**8.48**±0.31	94.74±0.41
	Ours	**91.28**±0.13	**76.34**±0.13	**0.51**±0.01	**8.37**±0.20	**98.17**±0.72

Table B.1: Comparison of Predictive Performance Across Datasets. Our method consistently outperforms existing methods on calibration, selective prediction and semantic shift detection across downstream image and language tasks, and remains competitive on accuracy and negative log-likelihood. We achieved improved results for Expected Calibration Error (ECE), Selective Prediction Accuracy (Sel. Acc.), and Semantic Shift Detection AUROC on all five datasets. Results within one standard error of the best are shaded in gray and printed in boldface. We show the mean and standard error over five trials.

B.2.2 Ablation Study

To better understand the difference between using our uncertainty-aware fine-tuning prior and a standard Gaussian prior, we show in Figure B.1 how the KL divergence between the variational distribution learned under our prior and the variational distribution learned under a standard Gaussian prior grows over the course of fine-tuning. Writing $\mu_{\text{ours}}, \sigma_{\text{ours}}$ and $\mu_{\text{standard}}, \sigma_{\text{standard}}$ for the means and standard deviations of the two variational distributions, we compute the KL divergence between the two distributions as follows:

$$D_{\text{KL}}(q_{\text{ours}}||q_{\text{standard}}) = \frac{1}{2}\left(\log \frac{\sigma^2_{\text{standard}}}{\sigma^2_{\text{ours}}} + \frac{\sigma^2_{\text{ours}} + (\mu_{\text{ours}} - \mu_{\text{standard}})^2}{\sigma^2_{\text{standard}}} - 1\right).$$

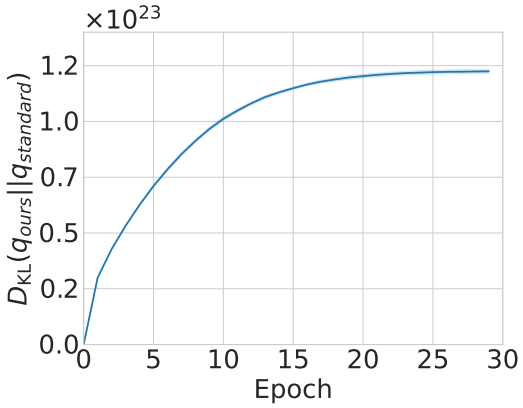
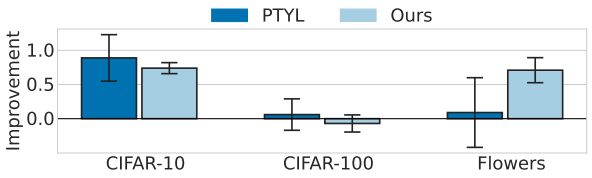
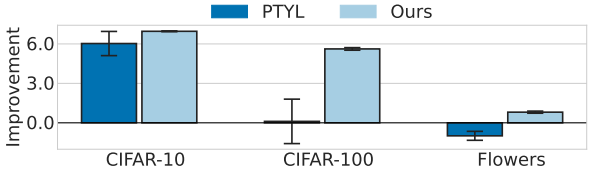


Figure B.1: Comparison of Learned Variational Distributions Under Different Priors. The plot shows the KL divergence from the variational distribution learned with an uncertainty-aware prior to the variational distribution learned with an uninformative Gaussian prior. The variational distribution learned under the uncertainty-aware prior differs significantly from the variational distribution learned with an uninformative Gaussian prior. The plot shows the mean and the standard error of the KL divergence on CIFAR-10 with a ResNet-18 over five trials. We fixed the parameter initialization and the stochasticity in the data loader to ensure comparability.

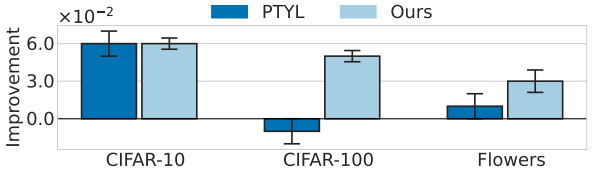
B.2.3 Improvement Over Empirical Risk Minimization



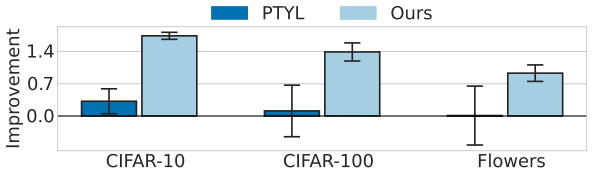
(a) Accuracy



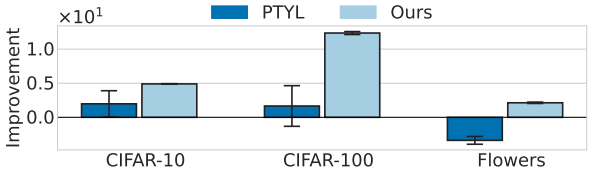
(b) Selective Prediction Accuracy



(c) Negative Log-Likelihood



(d) Expected Calibration Error



(e) Semantic Shift Detection AUROC

Figure B.2: On downstream image tasks, our method improves on all uncertainty metrics compared to the ERM baseline while maintaining similar or improved levels of accuracy. For each metric, we plot the improvement over ERM, where positive values indicate the metric has changed in the preferred direction (e.g., increased accuracy or decreased ECE). Means and standard errors are calculated over five trials.

B.3 Experimental Details: Group Robustness

B.3.1 Neural Network Architecture and Optimization

We follow the precedents set by Sagawa et al. (2020), Yang et al. (2023), and others. That is, all image datasets use a pre-trained ResNet-50 (He et al., 2016), and language datasets use a pre-trained BERT (Devlin et al., 2019). Images are resized and cropped in the center to 224×224 pixels. For image datasets, we use SGD with momentum 0.9. For language datasets, we use AdamW with default parameters (Loshchilov et al., 2017). Table B.2 shows the hyperparameters we use for the ERM-training step. Table B.3 shows the hyperparameters we use for last-layer fine-tuning, and Table B.4 shows the hyperparameters we use for all-layer fine-tuning. We note that in the fine-tuning step with a group-aware prior, we use a portion of the validation set for both data-dependent terms of the loss in Equation (4.10), that is, both terms are evaluated on points from the context set, which is chosen to be the validation dataset in our case. In principle, we could use the training set for the first term in Equation (4.10), as it does not require group labels, but we have chosen to evaluate both terms on the context distribution for simplicity.

B.3.2 Parameters for Ablation Studies

For Table 4.11, we use the hyperparameters detailed in Table B.3 and set the relevant parameter to its 'trivial' realization. That is, we set $\rho = 0$. In this ablation study, we average our results over ten trials. For Table 4.12, we also use a 'trivial' parameter choice $\gamma = 1$, but also investigate results under different strengths of upweighting. In this study, we estimate our mean and standard errors from three trials, except for when the parameter choice coincides with our parameter choice from Table B.3 (i.e., $\gamma = 4$), in which case we use ten.

Table B.2: Table of Hyperparameter Choice for Initial ERM Fine-Tuning. For all of our experiments that use a pre-trained network fine-tuned on the target task with ERM, we detail the hyperparameters used. Initial LR means the initial learning rate input into the learning rate scheduler. Note that α is the minimum multiplier value for adjusting the learning rate for the cosine decay scheduler.

Hyperparameter	Waterbirds	CelebA	MultiNLI
Epochs	30	5	3
Initial LR	0.005	0.005	0.00002
LR Scheduler	Cosine Decay	Cosine Decay	Linear Decay
α	0.01	0.001	—
Batch Size	128	128	32
Weight Decay	0	0	0.01

Table B.3: Table of Hyperparameter Choice for Last-Layer Retraining. For our experiments involving last-layer retraining on our group-aware prior, we detail the hyperparameters used. Epochs specifically indicates the number of epochs for which we retrain our last layer, and does not include the previous ERM fine-tuning steps. Initial LR means the initial learning rate input into the learning rate scheduler. Note that α is the minimum multiplier value for adjusting the learning rate for the cosine decay scheduler. The parameters λ, γ, and ρ are hyperparameters specific to our method.

Hyperparameter	**Waterbirds**	**CelebA**	**MultiNLI**
Epochs	40	20	5
Initial LR	0.001	0.0001	0.00004
LR Scheduler	Cosine Decay	Cosine Decay	Linear Decay
α	1	1	—
Batch Size	128	128	32
Weight Decay	0	0	0
λ	1	30	10
γ	4	1.5	2
ρ	0.15	0.15	0.1

Table B.4: Table of Hyperparameter Choice for All-Layer Fine-Tuning. For our experiments involving all-layer fine-tuning on our group-aware prior, we detail the hyperparameters used. Epochs specifically indicates the number of epochs for which we fine-tune the entire network, and does not include the previous ERM fine-tuning steps. Initial LR means the initial learning rate input into the learning rate scheduler. The parameters λ, γ, and ρ are hyperparameters specific to our method.

Hyperparameter	**Waterbirds**	**CelebA**	**MultiNLI**
Epochs	40	10	1
Initial LR	0.001	0.0001	0.000005
LR Scheduler	Linear Decay	Linear Decay	Linear Decay
Batch Size	128	128	32
Weight Decay	0	0	0
λ	15	200	10
γ	4	4	4
ρ	0.15	0.1	0.1

C

Appendix C: Appendix to Chapter 5

C.1 Proofs & Derivations

C.1.1 Finite- and Infinite-Horizon Behavior-Driven Variational Objectives

In this section, we present detailed derivations and proofs for the results in the main text.

Proposition 5.1. *Let the variational distribution $q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\tilde{\tau}_{0:t})$ be as defined in Equation (5.8). Then, given a horizon length $t^\star = t+1$ and an optimal state trajectory $\mathcal{S}^\Omega$,*

$$\mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\cdot) \,\|\, p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\xi_{1:t+1}}(\cdot \,|\, \xi^\star_{1:t+1}(\mathcal{S}^\Omega))) = \log p(\xi^\star_{1:t+1}(\mathcal{S}^\Omega)) - \bar{\mathcal{F}}(\pi, \mathcal{S}^\Omega), \tag{C.1}$$

where

$$\bar{\mathcal{F}}(\pi, \mathcal{S}^\Omega) \stackrel{\text{def}}{=} \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\tilde{\tau}_{0:t})}\Bigg[\sum_{t'=0}^{t} \log \mathbb{P}(\xi_{t'+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_{t'}) \,||\, p(\cdot \,|\, \mathbf{s}_{t'}))\Bigg], \tag{C.2}$$

and since $\log p(\xi^\star_{1:t+1}(\mathcal{S}^\Omega))$ is constant in π,

$$\arg\min_{\pi \in \Pi} \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\cdot) \,\|\, p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\xi_{1:t+1}}(\cdot \,|\, \xi^\star_{1:t+1}(\mathcal{S}^\Omega))) = \arg\max_{\pi \in \Pi} \bar{\mathcal{F}}(\pi, \mathcal{S}^\Omega). \tag{C.3}$$

Proof. To find an approximation to the posterior $p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\xi_{1:t+1}}(\cdot \,|\, \xi^\star_{1:t+1}(\mathcal{S}^\Omega)))$, we can use variational inference. To do so, we consider the trajectory distribution under $p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\xi_{1:t+1}}(\cdot \,|\, \xi^\star_{1:t+1}(\mathcal{S}^\Omega)))$, which by Bayes' Theorem is given by

$$\begin{aligned} &p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\xi_{1:t+1}}(\cdot \,|\, \xi^\star_{1:t+1}(\mathcal{S}^\Omega))) \\ &= \frac{p_{\mathbf{S}_0}(\mathbf{s}_0) \prod_{t'=0}^{t} \mathbb{P}(\xi_{t'+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) p(\mathbf{a}_t \,|\, \mathbf{s}_t) \prod_{t'=0}^{t-1} p_d(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) p(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'})}{p(\xi^\star_{1:t+1}(\mathcal{S}^\Omega))}. \end{aligned} \tag{C.4}$$

Inferring an approximation to the posterior distribution $p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\xi_{1:t+1}}(\cdot \,|\, \xi^\star_{1:t+1}(\mathcal{S}^\Omega))$ then becomes equivalent to finding a variational distribution $q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|\mathbf{S}_0}(\cdot \,|\, \mathbf{s}_0)$, which induces a trajectory distribution $q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\cdot)$ that minimizes the KL divergence from $q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\cdot)$ to $p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\xi_{1:t+1}}(\cdot \,|\, \xi^\star_{1:t+1}(\mathcal{S}^\Omega))$:

$$\min_{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}} \in \hat{\mathcal{Q}}} \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\cdot) \,\|\, p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\xi_{1:t+1}}(\cdot \,|\, \xi^\star_{1:t+1}(\mathcal{S}^\Omega))). \tag{C.5}$$

If we find a distribution $q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\cdot)$ for which the resulting KL divergence is zero, then $q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\cdot)$ is the exact posterior. If the KL divergence is positive, then $q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\cdot)$ is an

approximate posterior. To solve the variational problem in Equation (C.5), we can define a factorized variational family

$$q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\tilde{\tau}_{0:t}) \stackrel{\text{def}}{=} p_{\mathbf{S}_0}(\mathbf{s}_0)\pi(\mathbf{a}_t \,|\, \mathbf{s}_t) \prod_{t'=0}^{t-1} q_{\mathbf{S}_{t'+1}|\mathbf{S}_{t'},\mathbf{A}_{t'}}(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'})\pi(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'}), \tag{C.6}$$

where $\mathbf{A}_{0:t}$ and $\mathbf{S}_{0:t}$ are latent variables over which to infer an approximate posterior distribution. Returning to the variational problem in Equation (C.5), we can now write

$$\begin{aligned}
&\mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\cdot) \parallel p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\xi_{1:t+1}}(\cdot \,|\, \xi^{\star}_{1:t+1}(\mathcal{S}^{\Omega}))) \\
&= \int_{\mathcal{A}^{t+1}} \int_{\mathcal{S}^{t+1}} q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\tilde{\tau}_{0:t}) \log \frac{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\tilde{\tau}_{0:t})}{p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\xi_{1:t+1}}(\tilde{\tau}_{0:t} \,|\, \xi^{\star}_{1:t+1}(\mathcal{S}^{\Omega}))} d\mathbf{s}_{0:t} d\mathbf{a}_{0:t} \\
&= -\bar{\mathcal{F}}(\pi, \mathcal{S}^{\Omega}) + \log p(\xi^{\star}_{1:t+1}(\mathcal{S}^{\Omega})),
\end{aligned} \tag{C.7}$$

where

$$\begin{aligned}
\bar{\mathcal{F}}(\pi, \mathcal{S}^{\Omega}) \stackrel{\text{def}}{=} \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\tilde{\tau}_{0:t})} \Bigg[&\sum_{t'=0}^{t} \log \mathbb{P}(\xi_{t'+1}(\mathcal{S}^{\Omega}) = 1 \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) \\
&+ \log p(\mathbf{a}_t \,|\, \mathbf{s}_t) - \log \pi(\mathbf{a}_t \,|\, \mathbf{s}_t) + \sum_{t'=0}^{t-1} \log p(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'}) - \log \pi(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'}) \\
&+ \sum_{t'=0}^{t-1} \log p_d(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) - \log q_{\mathbf{S}_{t'+1}|\mathbf{S}_{t'},\mathbf{A}_{t'}}(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) \Bigg]
\end{aligned} \tag{C.8}$$

and

$$\begin{aligned}
&\log p(\xi^{\star}_{1:t+1}(\mathcal{S}^{\Omega})) \\
&= \log \int \int_{\mathcal{A}^{t+1}\mathcal{S}^{t+1}} \mathbb{P}(\xi_{1:t+1} = 1 \,|\, \mathbf{s}_0, \mathbf{a}_0, ..., \mathbf{s}_t, \mathbf{a}_t) p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\tilde{\tau}_{0:t}) d\mathbf{s}_{0:t} d\mathbf{a}_{0:t}
\end{aligned} \tag{C.9}$$

$$= \log \int \int_{\mathcal{A}^{t+1}\mathcal{S}^{t+1}} \left(\prod_{t'=0}^{t} \int_{\mathcal{S}} p_d(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) \, \mathbb{I}\{\mathbf{s}_{t'+1} \in \mathcal{S}^{\Omega}\} \, d\mathbf{s}_{t'+1} \right) p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\tilde{\tau}_{0:t}) d\mathbf{s}_{0:t} d\mathbf{a}_{0:t} \tag{C.10}$$

is a log-marginal likelihood. Following Haarnoja et al. (2018a), we define the variational distribution over next states as the true transition dynamics, that is,

$$q_{\mathbf{S}_{t+1}|\mathbf{S}_t,\mathbf{A}_t}(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t) = p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t), \tag{C.11}$$

so that

$$q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\tilde{\tau}_{0:t}) \stackrel{\text{def}}{=} p_{\mathbf{S}_0}(\mathbf{s}_0)\pi(\mathbf{a}_t \,|\, \mathbf{s}_t) \prod_{t'=0}^{t-1} p_d(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'})\pi(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'}). \tag{C.12}$$

We can then simplify $\bar{\mathcal{F}}(\pi, \mathcal{S}^\Omega)$ to

$$\bar{\mathcal{F}}(\pi, \mathcal{S}^\Omega) = \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\tilde{\tau}_{0:t})}\left[\sum_{t'=0}^{t} \log \mathbb{P}(\xi_{t'+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_{t'}) \,\|\, p(\cdot \,|\, \mathbf{s}_{t'}))\right]. \tag{C.13}$$

Since $\log p(\xi^\star_{1:t+1}(\mathcal{S}^\Omega))$ is constant in π, solving the variational optimization problem in Equation (C.5) is equivalent to maximizing the variational objective with respect to $\pi \in \Pi$, where Π is a family of policy distributions. □

Corollary 5.1. *The objective in Equation (C.13) corresponds to KL-regularized reinforcement learning with a time-varying reward function given by*

$$r(\mathbf{s}_{t'}, \mathbf{a}_{t'}, \mathcal{S}^\Omega) \stackrel{\text{def}}{=} \log \mathbb{P}(\xi_{t'+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}).$$

Proof. Let

$$r(\mathbf{s}_{t'}, \mathbf{a}_{t'}, \mathcal{S}^\Omega) \stackrel{\text{def}}{=} \log \mathbb{P}(\xi_{t'+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}). \tag{C.14}$$

and note that the objective

$$\bar{\mathcal{F}}(\pi, \mathcal{S}^\Omega) = \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\tilde{\tau}_{0:t})}\left[\sum_{t'=0}^{t} \log \mathbb{P}(\xi_{t'+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_{t'}) \,\|\, p(\cdot \,|\, \mathbf{s}_{t'}))\right]. \tag{C.15}$$

can equivalently be written as

$$\bar{\mathcal{F}}(\pi, \mathcal{S}^\Omega) = \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}}(\tilde{\tau}_{0:t})}\left[\sum_{t'=0}^{t} r(\mathbf{s}_{t'}, \mathbf{a}_{t'}, \mathcal{S}^\Omega) + \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_{t'}) \,\|\, p(\cdot \,|\, \mathbf{s}_{t'}))\right], \tag{C.16}$$

which, as shown in Haarnoja et al. (2018a), can be written in the form of a conventional Q-function,

$$Q^\pi(\mathbf{s}_0, \mathbf{a}_0) = r(\mathbf{s}_0, \mathbf{a}_0) + \gamma\, \mathbb{E}_{q_{\tilde{\mathcal{T}}_1}}[Q^\pi(\mathbf{s}_1, \mathbf{a}_1) \,|\, \mathbf{S}_0 = \mathbf{s}_0, \mathbf{A}_0 = \mathbf{a}_0], \tag{C.17}$$

□

Proposition 5.4. *Let*

$$q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}, T}(\tilde{\tau}_{0:t}, t) = q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t} \,|\, t) q_T(t), \tag{C.18}$$

let $q_T(t)$ be a variational distribution defined on $t \in \mathbb{N}_0$, and let $q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t} \,|\, t)$ be as defined in Equation (5.8). Then, given an optimal state trajectory $\mathcal{S}^\Omega$, we have that

$$\mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}, T}(\cdot) \,\|\, p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}, T|\xi_{1:T+1}}(\cdot \,|\, \xi^\star_{1:T+1}(\mathcal{S}^\Omega))) = \log p(\xi^\star_{1:T+1}(\mathcal{S}^\Omega)) - \mathcal{F}(\pi, q_T, \mathcal{S}^\Omega), \tag{C.19}$$

where

$$\mathcal{F}(\pi, q_T, \mathcal{S}^\Omega) \stackrel{\text{def}}{=} \sum_{t=0}^{\infty} q_T(t)\, \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)} \Big[\sum_{t'=0}^{t} \log \mathbb{P}(\xi_{t'+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) - \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T}(\cdot)\,||\,p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T}(\cdot)) \Big] \tag{C.20}$$

and $\log p(\xi^\star_{1:T+1}(\mathcal{S}^\Omega))$ *is constant in* π *and* q_T*.*

Proof. In general, solving the variational problem

$$\min_{q \in \mathcal{Q}} \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T}(\cdot) \parallel p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T|\xi_{1:T+1}}(\cdot \,|\, \xi^\star_{1:T+1}(\mathcal{S}^\Omega))) \tag{C.21}$$

is challenging, but as in the fixed-time setting, we can take advantage of the fact that, by choosing a variational family parameterized by

$$q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t) \stackrel{\text{def}}{=} \pi(\mathbf{a}_t \,|\, \mathbf{s}_t) \prod_{t'=0}^{t-1} p_d(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'})\, \pi(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'}), \tag{C.22}$$

with $\pi \in \Pi$, we can follow the same steps as in the proof for Proposition 5.1 and show that given an optimal state trajectory $\mathcal{S}^\Omega$,

$$\mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T}(\cdot) \parallel p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T|\xi_{1:T+1}}(\cdot \,|\, \xi^\star_{1:T+1}(\mathcal{S}^\Omega)))) = \log p(\xi^\star_{1:T+1}(\mathcal{S}^\Omega)) - \mathcal{F}(\pi, q_T, \mathcal{S}^\Omega), \tag{C.23}$$

where

$$\mathcal{F}(\pi, q_T, \mathcal{S}^\Omega) \stackrel{\text{def}}{=} \sum_{t=0}^{\infty} q_T(t)\, \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)} \Big[\sum_{t'=0}^{t} \log \mathbb{P}(\xi_{t'+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) - \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T}(\cdot) \parallel p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T}(\cdot)) \Big], \tag{C.24}$$

where $q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T}(\tilde{\tau}_{0:t}, t) \stackrel{\text{def}}{=} q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t) q_T(t)$, and hence, solving the variational problem in Equation (5.12) is equivalent to maximizing $\mathcal{F}(\pi, q_T, \mathcal{S}^\Omega)$ with respect to π and q_T. □

C.1.2 Recursive Behavior-Driven Variational Objective & Behavior-Driven Bellman Backup Operator

Proposition 5.5. *Let the variational distribution factorize as*

$$q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T}(\tilde{\tau}_{0:t}, t) = q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t) q_T(t), \tag{C.25}$$

let

$$q_T(t) = q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \tag{C.26}$$

be a variational distribution defined on $t \in \mathbb{N}_0$, and let $q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t} \,|\, t)$ be as defined in Equation (5.8). Then, given an optimal state trajectory $\mathcal{S}^\Omega$, Equation (C.20) can be rewritten as

$$\begin{aligned}
&\mathcal{F}(\pi, q_T, \mathcal{S}^\Omega) \\
&= \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_0}(\tilde{\tau}_0)} \left[\sum_{t=0}^{\infty} \left(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \right) \Big(r(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_\Delta) - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \,\|\, p(\cdot \,|\, \mathbf{s}_t)) \Big) \right]
\end{aligned} \tag{C.27}$$

where

$$r(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_\Delta) \overset{\text{def}}{=} \log \mathbb{P}(\xi_{t+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_t, \mathbf{a}_t) - q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \,\|\, p_{\Delta_{t+1}}). \tag{C.28}$$

Proof. Consider the variational objective $\mathcal{F}(\pi, q_T, \mathcal{S}^\Omega)$ in Equation (C.20):

$$\begin{aligned}
&\mathcal{F}(\pi, q_T, \mathcal{S}^\Omega) \\
&= \sum_{t=0}^{\infty} q_T(t) \, \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t} \,|\, t)} \Big[\sum_{t'=0}^{t} \log \mathbb{P}(\xi_{t'+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) \\
&\qquad\qquad - \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T}(\cdot) \parallel p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T}(\cdot)) \Big]
\end{aligned} \tag{C.29}$$

$$\begin{aligned}
&= \sum_{t=0}^{\infty} q_T(t) \, \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t} \,|\, t)} \Big[\sum_{t'=0}^{t} \log \mathbb{P}(\xi_{t'+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) \\
&\qquad\qquad - \log \frac{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t} \,|\, t) q_T(t)}{p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t} \,|\, t) p_T(t)} d\tilde{\tau}_{0:t} \Big]
\end{aligned} \tag{C.30}$$

$$\begin{aligned}
&= \sum_{t=0}^{\infty} q_T(t) \, \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t} \,|\, t)} \left[\sum_{t'=0}^{t} \log \mathbb{P}(\xi_{t'+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) - \log \frac{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t} \,|\, t)}{p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t} \,|\, t)} \right] \\
&\quad - \sum_{t=0}^{\infty} q_T(t) \log \frac{q_T(t)}{p_T(t)}.
\end{aligned} \tag{C.31}$$

Noting that $\sum_{t=0}^{\infty} q_T(t) \log \frac{q_T(t)}{p_T(t)} = \mathbb{D}_{\mathrm{KL}}(q_T \parallel p_T)$, we can write

$$\begin{aligned}
&\mathcal{F}(\pi, q_T, \mathcal{S}^{\Omega}) \\
&= \sum_{t=0}^{\infty} q_T(t)\, \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)} \left[\sum_{t'=0}^{t} \log \mathbb{P}(\xi_{t'+1}(\mathcal{S}^{\Omega}) = 1 \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) - \log \frac{q_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)}{p_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)} \right] \\
&\qquad - \mathbb{D}_{\mathrm{KL}}(q_T \parallel p_T) \qquad \text{(C.32)} \\
&= \sum_{t=0}^{\infty} q_T(t)\, \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)} \Big[\sum_{t'=0}^{t} \log \mathbb{P}(\xi_{t'+1}(\mathcal{S}^{\Omega}) = 1 \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) \Big] \\
&\qquad - \sum_{t=0}^{\infty} q_T(t)\, \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)} \left[\log \frac{q_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)}{p_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)} \right] - \mathbb{D}_{\mathrm{KL}}(q_T \parallel p_T). \qquad \text{(C.33)}
\end{aligned}$$

Further noting that for an infinite-horizon trajectory distribution

$$q_{\tilde{\boldsymbol{\tau}}_{t'}|\mathbf{S}_{t'}}(\tilde{\tau}_{t'} \,|\, \mathbf{s}_{t'}) \stackrel{\text{def}}{=} \prod_{t=t'}^{\infty} p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t) \pi(\mathbf{a}_t \,|\, \mathbf{s}_t), \qquad \text{(C.34)}$$

trajectory realization $\tilde{\tau}_{t+1} \stackrel{\text{def}}{=} \{\tau_{t'}\}_{t'=t+1}^{\infty}$, and any joint probability density $f(\mathbf{s}_t, \mathbf{a}_t)$,

$$\begin{aligned}
&\sum_{t=0}^{\infty} q_T(t)\, \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)} \Big[f(\mathbf{s}_t, \mathbf{a}_t) \Big] \qquad \text{(C.35)} \\
&= \sum_{t=0}^{\infty} \left(\int q_{\tilde{\boldsymbol{\tau}}_{T+1}}(\tilde{\tau}_{t+1}) \left(\int_{\mathcal{S}^t \times \mathcal{A}^t} q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\tau}_{0:t}) q_T(t) f(\mathbf{s}_t, \mathbf{a}_t) d\tilde{\tau}_{0:t} \right) d\tilde{\tau}_{t+1} \right) \\
&= \sum_{t=0}^{\infty} \left(\mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)} \Big[q_T(t) f(\mathbf{s}_t, \mathbf{a}_t) \Big] \cdot \underbrace{\left(\int q_{\tilde{\boldsymbol{\tau}}_{T+1}}(\tilde{\tau}_{t+1}) d\tilde{\tau}_{t+1} \right)}_{=1} \right) \qquad \text{(C.36)} \\
&= \sum_{t=0}^{\infty} \left(\left(\int_{\mathcal{S}^t \times \mathcal{A}^t} q(\tilde{\tau}_{0:t}) q_T(t) f(\mathbf{s}_t, \mathbf{a}_t) d\tilde{\tau}_{0:t} \right) \cdot \underbrace{\left(\int q_{\tilde{\boldsymbol{\tau}}_{T+1}}(\tilde{\tau}_{t+1}) d\tilde{\tau}_{t+1} \right)}_{=1} \right) \qquad \text{(C.37)} \\
&= \sum_{t=0}^{\infty} \int q_{\tilde{\boldsymbol{\tau}}_0}(\tilde{\tau}_0) q_T(t) f(\mathbf{s}_t, \mathbf{a}_t) d\tilde{\tau}_0 \qquad \text{(C.38)} \\
&= \int q_{\tilde{\boldsymbol{\tau}}_0}(\tilde{\tau}_0) \sum_{t=0}^{\infty} q_T(t) f(\mathbf{s}_t, \mathbf{a}_t) d\tilde{\tau}_0, \qquad \text{(C.39)}
\end{aligned}$$

we can express Equation (C.33) in terms of the infinite-horizon state–action trajectory

$$q_{\tilde{\boldsymbol{\tau}}_0}(\tilde{\tau}_0) \stackrel{\text{def}}{=} \prod_{t=0}^{\infty} p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t) \pi(\mathbf{a}_t \,|\, \mathbf{s}_t) \qquad \text{(C.40)}$$

as

$$\mathcal{F}(\pi, q_T, \mathcal{S}^\Omega) = \int q_{\tilde{\boldsymbol{\mathcal{T}}}_0}(\tilde{\tau}_0) \sum_{t=0}^{\infty} q_T(t) \sum_{t'=0}^{t} \log \mathbb{P}(\xi_{t'+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) d\tilde{\tau} \\ - \sum_{t=0}^{\infty} q_T(t) \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\cdot \,|\, t) \parallel p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\cdot \,|\, t)) - \mathbb{D}_{\mathrm{KL}}(q_T \parallel p_T) \tag{C.41}$$

$$= \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_0}(\tilde{\tau}_0)} \Bigg[\sum_{t=0}^{\infty} q_T(t) \Big(\sum_{t'=0}^{t} \log \mathbb{P}(\xi_{t'+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) \\ - \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\cdot \,|\, t) \parallel p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\cdot \,|\, t)) \Big) \Bigg] - \mathbb{D}_{\mathrm{KL}}(q_T \parallel p_T). \tag{C.42}$$

Using Lemma C.5 and the definition of $q_T(t)$ in Equation (5.30), we can rewrite this objective as

$$\mathcal{F}(\pi, q_T, \mathcal{S}^\Omega) \\ = \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_0}(\tilde{\tau}_0)} \Bigg[\sum_{t=0}^{\infty} \Big(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \Big) q_{\Delta_{t'}}(\Delta_{t'} = 1) \Big(\sum_{t'=0}^{t} \log \mathbb{P}(\xi_{t'+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) \\ - \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\cdot \,|\, t) \parallel p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\cdot \,|\, t)) \Big) \Bigg] - \sum_{t=0}^{\infty} \Big(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \Big) \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} || p_{\Delta_{t+1}}) \tag{C.43}$$

$$= \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_0}(\tilde{\tau}_0)} \Bigg[\sum_{t=0}^{\infty} \Bigg(\prod_{t'=1}^{t} q(\Delta_{t'} = 0) \Bigg) \\ \cdot \Big(q(\Delta_{t+1} = 1) (\sum_{t'=0}^{t} \log \mathbb{P}(\xi_{t'+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) - \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\cdot \,|\, t) \parallel p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\cdot \,|\, t))) \\ - \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \parallel p_{\Delta_{t+1}}) \Big) \Bigg], \tag{C.44}$$

with

$$\mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \parallel p_{\Delta_{t+1}}) = q_{\Delta_{t+1}}(\Delta_{t+1} = 0) \log \frac{q_{\Delta_{t+1}}(\Delta_{t+1} = 0)}{p_{\Delta_{t+1}}(\Delta_{t+1} = 0)} \\ + (1 - q_{\Delta_{t+1}}(\Delta_{t+1} = 0)) \log \frac{1 - q_{\Delta_{t+1}}(\Delta_{t+1} = 0)}{1 - p_{\Delta_{t+1}}(\Delta_{t+1} = 0)}. \tag{C.45}$$

Next, to re-express $\mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\cdot \,|\, t) \parallel p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\cdot \,|\, t))$ as a sum over Kullback-Leibler

divergences between distributions over single action random variables, we note that

$$\mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\cdot\,|\,t)\parallel p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\cdot\,|\,t)) = \int_{\mathcal{S}^t\times\mathcal{A}^t} q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)\log\frac{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)}{p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)}d\tilde{\tau}_{0:t} \tag{C.46}$$

$$= \int_{\mathcal{S}^t\times\mathcal{A}^t} q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)\log\frac{\prod_{t'=1}^{t}\pi(\mathbf{a}_{t'}\,|\,\mathbf{s}_{t'})}{\prod_{t'=1}^{t}p(\mathbf{a}_{t'}\,|\,\mathbf{s}_{t'})}d\tilde{\tau}_{0:t} \tag{C.47}$$

$$= \int_{\mathcal{S}^t\times\mathcal{A}^t} q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)\sum_{t'=0}^{t}\log\frac{\pi(\mathbf{a}_{t'}\,|\,\mathbf{s}_{t'})}{p(\mathbf{a}_{t'}\,|\,\mathbf{s}_{t'})}d\tilde{\tau}_{0:t} \tag{C.48}$$

$$= \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_0}(\tilde{\tau}_0)}\left[\sum_{t'=0}^{t}\int_{\mathcal{A}}\pi(\mathbf{a}_{t'}\,|\,\mathbf{s}_{t'})\log\frac{\pi(\mathbf{a}_{t'}\,|\,\mathbf{s}_{t'})}{p(\mathbf{a}_{t'}\,|\,\mathbf{s}_{t'})}d\mathbf{a}_{t'}\right] \tag{C.49}$$

$$= \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_0}(\tilde{\tau}_0)}\left[\sum_{t'=0}^{t}\mathbb{D}_{\mathrm{KL}}(\pi(\cdot\,|\,\mathbf{s}_{t'})\parallel p(\cdot\,|\,\mathbf{s}_{t'}))\right], \tag{C.50}$$

where we have used the same marginalization trick as above to express the expression in terms of an infinite-horizon trajectory distribution, which allows us to express Equation (C.44) as

$$\begin{aligned}
&\mathcal{F}(\pi, q_T, \mathcal{S}^\Omega)\\
&= \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_0}(\tilde{\tau}_0)}\left[\sum_{t=0}^{\infty}\left(\prod_{t'=1}^{t}q(\Delta_{t'}=0)\right)\cdot\Bigg(q_{\Delta_{t+1}}(\Delta_{t+1}=1)\Bigg(\sum_{t'=0}^{t}\log\mathbb{P}(\xi_{t'+1}(\mathcal{S}^\Omega)=1\,|\,\mathbf{s}_{t'},\mathbf{a}_{t'})\right.\right.\\
&\qquad\left.\left.-\,\mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_0}(\tilde{\tau}_0)}\left[\sum_{t'=0}^{t}\mathbb{D}_{\mathrm{KL}}(\pi(\cdot\,|\,\mathbf{s}_{t'})\parallel p(\cdot\,|\,\mathbf{s}_{t'}))\right]\Bigg)-\mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}}\parallel p_{\Delta_{t+1}})\Bigg)\right]\\
&= \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_0}(\tilde{\tau}_0)}\left[\sum_{t=0}^{\infty}\left(\prod_{t'=1}^{t}q(\Delta_{t'}=0)\right)\right.\\
&\quad\cdot\Bigg(q_{\Delta_{t+1}}(\Delta_{t+1}=1)\Bigg(\mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_0}(\tilde{\tau}_0)}\left[\sum_{t'=0}^{t}\log\mathbb{P}(\xi_{t'+1}(\mathcal{S}^\Omega)=1\,|\,\mathbf{s}_{t'},\mathbf{a}_{t'})\right.\\
&\qquad\left.\left.\left.-\,\mathbb{D}_{\mathrm{KL}}(\pi(\cdot\,|\,\mathbf{s}_{t'})\parallel p(\cdot\,|\,\mathbf{s}_{t'}))\right]\Bigg)-\mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}}\parallel p_{\Delta_{t+1}})\Bigg)\right].
\end{aligned} \tag{C.51}$$

Rearranging and dropping redundant expectation operators, we can now express the

objective as

$$
\begin{aligned}
&\mathcal{F}(\pi, q_T, \mathcal{S}^\Omega) \\
&\quad = \mathbb{E}_{q_{\tilde{\boldsymbol{T}}_0}(\tilde{\tau}_0)}\Bigg[-\sum_{t=0}^{\infty}\left(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'}=0)\right)\left(q_{\Delta_{t+1}}(\Delta_{t+1}=1)\mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \,\|\, p_{\Delta_{t+1}})\right)\Bigg] \\
&\qquad\qquad + \sum_{t=0}^{\infty}\underbrace{\left(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'}=0) q_{\Delta_{t+1}}(\Delta_{t+1}=1)\right)}_{=q_T(t)} \\
&\qquad\qquad\qquad \cdot \mathbb{E}_{q_{\tilde{\boldsymbol{T}}_0}(\tilde{\tau}_0)}\Bigg[\sum_{t'=0}^{t} \log \mathbb{P}(\xi_{t'+1}(\mathcal{S}^\Omega)=1 \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) \\
&\qquad\qquad\qquad\qquad - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_{t'}) \,\|\, p(\cdot \,|\, \mathbf{s}_{t'}))\Bigg],
\end{aligned}
\tag{C.52}
$$

whereupon we note that the last term can be expressed as

$$
\begin{aligned}
&\sum_{t=0}^{\infty} q_T(t)\, \mathbb{E}_{q_{\tilde{\boldsymbol{T}}_0}(\tilde{\tau}_0)}\left[\sum_{t'=0}^{t} \log \mathbb{P}(\xi_{t'+1}(\mathcal{S}^\Omega)=1 \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_{t'}) \,\|\, p(\cdot \,|\, \mathbf{s}_{t'}))\right] \\
&= \mathbb{E}_{q_{\tilde{\boldsymbol{T}}_0}(\tilde{\tau}_0)}\left[\sum_{t=0}^{\infty}\sum_{t'=0}^{t} q_T(t)(\log \mathbb{P}(\xi_{t'+1}(\mathcal{S}^\Omega)=1 \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_{t'}) \,\|\, p(\cdot \,|\, \mathbf{s}_{t'})))\right]
\end{aligned}
\tag{C.53}
$$

$$
= \mathbb{E}_{q_{\tilde{\boldsymbol{T}}_0}(\tilde{\tau}_0)}\left[\sum_{t=0}^{\infty} q(T \geq t)(\log \mathbb{P}(\xi_{t+1}(\mathcal{S}^\Omega)=1 \,|\, \mathbf{s}_t, \mathbf{a}_t) - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \,\|\, p(\cdot \,|\, \mathbf{s}_t)))\right] \tag{C.54}
$$

$$
\begin{aligned}
&= \mathbb{E}_{q_{\tilde{\boldsymbol{T}}_0}(\tilde{\tau}_0)}\Bigg[\sum_{t=0}^{\infty}\underbrace{\left(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'}=0)\right)}_{\text{(by Lemma C.2)}}(\log \mathbb{P}(\xi_{t+1}(\mathcal{S}^\Omega)=1 \,|\, \mathbf{s}_t, \mathbf{a}_t) \\
&\qquad - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \,\|\, p(\cdot \,|\, \mathbf{s}_t)))\Bigg],
\end{aligned}
\tag{C.55}
$$

where the second line follows from expanding the sums and regrouping terms. By substituting the expression in Equation (C.55) into Equation (C.52), we obtain an objective expressed entirely in terms of distributions over single-index random

variables:

$$
\begin{aligned}
&\mathcal{F}(\pi, q_T, \mathcal{S}^\Omega) \\
&= \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_0}(\tilde{\tau}_0)} \left[\sum_{t=0}^{\infty} \left(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \right) \right. \\
&\qquad \left. \cdot \Big(\log \mathbb{P}(\xi_{t+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_t, \mathbf{a}_t) - q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \parallel p_{\Delta_{t+1}}) \right. \right. \qquad \text{(C.56)} \\
&\qquad\qquad \left. \left. - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)) \Big) \right] \\
&= \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_0}(\tilde{\tau}_0)} \left[\sum_{t=0}^{\infty} \left(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \right) \Big(r(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_\Delta) \right. \qquad \text{(C.57)} \\
&\qquad \left. - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)) \Big) \right], \qquad \text{(C.58)}
\end{aligned}
$$

where we defined

$$
r(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_\Delta) \stackrel{\text{def}}{=} \log \mathbb{P}(\xi_{t+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_t, \mathbf{a}_t) - q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \parallel p_{\Delta_{t+1}}), \tag{C.59}
$$

which concludes the proof. □

Theorem 5.1. *Let $q_T(t)$ and $q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|T}(\tilde{\tau}_{0:t} \,|\, t)$ be as defined in Equation (5.8) and Equation (5.30), and define a behavior-driven state value function,*

$$
V^\pi(\mathbf{s}_t, \mathcal{S}^\Omega; q_T) \stackrel{\text{def}}{=} \mathbb{E}_{\pi(\mathbf{a}_t \,|\, \mathbf{s}_t)} \Big[Q^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_T) \Big] - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \,\|\, p(\cdot \,|\, \mathbf{s}_t)), \tag{C.60}
$$

a behavior-driven state–action value function

$$
Q^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_T) \stackrel{\text{def}}{=} r(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_\Delta) + q(\Delta_{t+1} = 0) \, \mathbb{E}_{p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)} \Big[V^\pi(\mathbf{s}_{t+1}, \mathcal{S}^\Omega; q_T) \Big], \tag{C.61}
$$

and a behavior-driven reward-like function

$$
r(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_\Delta) \stackrel{\text{def}}{=} \log \mathbb{P}(\xi_{t+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_t, \mathbf{a}_t) - q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \,\|\, p_{\Delta_{t+1}}). \tag{C.62}
$$

Then given an optimal state trajectory $\mathcal{S}^\Omega$,

$$
\mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}, T}(\cdot) \,\|\, p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}, T | \xi_{1:t+1}}(\cdot \,|\, \xi^\star_{1:t+1}(\mathcal{S}^\Omega))) - C = -\mathcal{F}(\pi, q_T, \mathcal{S}^\Omega) = -V^\pi(\mathbf{s}_0, \mathcal{S}^\Omega; q_T),
$$

where $C \stackrel{\text{def}}{=} \log p(\xi^\star_{1:T+1}(\mathcal{S}^\Omega))$ is independent of π and q_T, and hence maximizing $V^\pi(\mathcal{S}^\Omega; \pi, q_T)$ is equivalent to minimizing Equation (5.12) and hence, the following

holds:

$$\begin{aligned}
&\arg\min_{\pi\in\Pi, q_T\in\mathcal{Q}_T} \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T}(\cdot)\,\|\,p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T|\xi_{1:T+1}}(\cdot\,|\,\xi^\star_{1:T+1}(\mathcal{S}^\Omega)))\\
=&\arg\max_{\pi\in\Pi, q_T\in\mathcal{Q}_T} \mathcal{F}(\pi, q_T, \mathcal{S}^\Omega)\\
=&\arg\max_{\pi\in\Pi, q_T\in\mathcal{Q}_T} \mathbb{E}_{p(\mathbf{s}_0)}[V^\pi(\mathbf{s}_0, \mathcal{S}^\Omega; q_T)].
\end{aligned}$$

Proof. Consider the objective derived in Proposition 5.5,

$$\begin{aligned}
&\mathcal{F}(\pi, q_T, \mathcal{S}^\Omega)\\
&= \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_0}(\tilde{\tau}_0)}\Bigg[\sum_{t=0}^{\infty}\Bigg(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'}=0)\Bigg)\\
&\quad\cdot\underbrace{\Big(q_{\Delta_{t+1}}(\Delta_{t+1}=1)\log\mathbb{P}(\xi_{t+1}(\mathcal{S}^\Omega)=1\,|\,\mathbf{s}_t,\mathbf{a}_t) - \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}}\,\|\,p_{\Delta_{t+1}})\Big)}_{\stackrel{\text{def}}{=} r(\mathbf{s}_t,\mathbf{a}_t,\mathcal{S}^\Omega;q_\Delta)}\\
&\quad - \mathbb{D}_{\mathrm{KL}}(\pi(\mathbf{a}_t\,|\,\mathbf{s}_t)\,\|\,p(\mathbf{a}_t|\mathbf{s}_t))\Bigg],
\end{aligned}\tag{C.63}$$

and recall that, by Proposition 5.4,

$$\mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T}(\cdot)\,\|\,p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T|\xi_{1:T+1}}(\cdot\,|\,\xi^\star_{1:T+1}(\mathcal{S}^\Omega))) = -\mathcal{F}(\pi, q_T, \mathcal{S}^\Omega) + \log p(\xi^\star_{1:T+1}(\mathcal{S}^\Omega)).\tag{C.64}$$

Therefore, to prove the result that

$$\mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T}(\cdot)\,\|\,p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T|\xi_{1:T+1}}(\cdot\,|\,\xi^\star_{1:T+1}(\mathcal{S}^\Omega))) = -V^\pi(\mathbf{s}_0, \mathcal{S}^\Omega; q_T) + \log p(\xi^\star_{1:T+1}(\mathcal{S}^\Omega)),$$

we just need to show that $\mathcal{F}(\pi, q_T, \mathcal{S}^\Omega) = V^\pi(\mathbf{s}_0, \mathcal{S}^\Omega; q_T)$ for $V^\pi(\mathbf{s}_0, \mathcal{S}^\Omega; q_T)$ as defined in the theorem statement. To do so, we start from the objective $\mathcal{F}(\pi, q_T, \mathcal{S}^\Omega)$ and unroll it for $t = 0$:

$$\begin{aligned}
&\mathcal{F}(\pi, q_T, \mathcal{S}^\Omega)\\
&= \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_0}(\tilde{\tau}_0)}\left[\sum_{t=0}^{\infty}\left(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'}=0)\right) r(\mathbf{s}_t,\mathbf{a}_t,\mathcal{S}^\Omega;q_\Delta) - \mathbb{D}_{\mathrm{KL}}(\pi(\mathbf{a}_t\,|\,\mathbf{s}_t)\,\|\,p(\mathbf{a}_t|\mathbf{s}_t))\right]
\end{aligned}\tag{C.65}$$

$$\begin{aligned}
&= \mathbb{E}_{\pi(\mathbf{a}_0\,|\,\mathbf{s}_0)p(\mathbf{s}_0)}\Bigg[r(\mathbf{s}_0,\mathbf{a}_0,\mathcal{S}^\Omega;q_\Delta) + \mathbb{E}_{q(\tau_1\,|\,\mathbf{s}_0,\mathbf{a}_0)}\Bigg[\sum_{t=1}^{\infty}\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'}=0)\Big(r(\mathbf{s}_t,\mathbf{a}_t,\mathcal{S}^\Omega;q_\Delta)\\
&\quad - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot\,|\,\mathbf{s}_t)\,\|\,p(\cdot\,|\,\mathbf{s}_t))\Big)\Bigg]\Bigg] - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot\,|\,\mathbf{s}_0)\,\|\,p(\cdot\,|\,\mathbf{s}_0)).
\end{aligned}\tag{C.66}$$

With this expression at hand, we now define

$$
\begin{aligned}
&Q^{\pi}_{\text{sum}}(\mathbf{s}_0, \mathbf{a}_0, \mathcal{S}^{\Omega}; q_T) \\
&\stackrel{\text{def}}{=} r(\mathbf{s}_0, \mathbf{a}_0, \mathcal{S}^{\Omega}; q_{\Delta}) + \mathbb{E}_{q(\tau_1 | \mathbf{s}_0, \mathbf{a}_0)} \Bigg[\sum_{t=1}^{\infty} \prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \Big(r(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^{\Omega}; q_{\Delta}) \\
&\qquad\qquad - \mathbb{D}_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)) \Big) \Bigg],
\end{aligned}
\tag{C.67}
$$

and note that

$$
\begin{aligned}
\mathcal{F}(\pi, q_T, \mathcal{S}^{\Omega}) &= \mathbb{E}_{\pi(\mathbf{a}_0 \,|\, \mathbf{s}_0) p(\mathbf{s}_0)} [Q^{\pi}_{\text{sum}}(\mathbf{s}_0, \mathbf{a}_0, \mathcal{S}^{\Omega}; q_T) - \mathbb{D}_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_0) \,\|\, p(\cdot \,|\, \mathbf{s}_0))] \\
&= \mathbb{E}_{p(\mathbf{s}_0)} [V^{\pi}(\mathbf{s}_0, \mathcal{S}^{\Omega}; q_T)],
\end{aligned}
\tag{C.68}
$$

as per the definition of $\mathbb{E}_{p(\mathbf{s}_0)}[V^{\pi}(\mathbf{s}_0, \mathcal{S}^{\Omega}; q_T)]$. To prove the theorem from this intermediate result, we now have to show that $Q^{\pi}_{\text{sum}}(\mathbf{s}_0, \mathbf{a}_0, \mathcal{S}^{\Omega}; q_T)$ as defined in Equation (C.67) can in fact be expressed recursively as $Q^{\pi}_{\text{sum}}(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^{\Omega}; q_T) = Q^{\pi}(\mathbf{s}_0, \mathbf{a}_0, \mathcal{S}^{\Omega}; q_T)$ with

$$
Q^{\pi}(\mathbf{s}_0, \mathbf{a}_0, \mathcal{S}^{\Omega}; q_T) = r(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^{\Omega}; q_{\Delta}) + q(\Delta_{t+1} = 0) \, \mathbb{E}_{p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)} \Big[V^{\pi}(\mathbf{s}_{t+1}, \mathcal{S}^{\Omega}; q_T) \Big].
\tag{C.69}
$$

To see that this is the case, first, unroll $Q^{\pi}(\mathbf{s}_0, \mathbf{a}_0, \mathcal{S}^{\Omega}; q_T)$ for $t = 1$,

$$
\begin{aligned}
&Q^{\pi}_{\text{sum}}(\mathbf{s}_0, \mathbf{a}_0, \mathcal{S}^{\Omega}; q_T) \\
&= r(\mathbf{s}_0, \mathbf{a}_0, \mathcal{S}^{\Omega}; q_{\Delta}) + \mathbb{E}_{q(\tau_1 | \mathbf{s}_0, \mathbf{a}_0)} \Bigg[\sum_{t=1}^{\infty} \prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \Big(r(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^{\Omega}; q_{\Delta}) \\
&\qquad - \mathbb{D}_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)) \Big) \Bigg]
\end{aligned}
\tag{C.70}
$$

$$
\begin{aligned}
&= r(\mathbf{s}_0, \mathbf{a}_0, \mathcal{S}^{\Omega}; q_{\Delta}) + \mathbb{E}_{p_d(\mathbf{s}_1 | \mathbf{s}_0, \mathbf{a}_0)} \Bigg[\mathbb{E}_{q(\tau_1 | \mathbf{s}_0, \mathbf{a}_0)} \Bigg[\sum_{t=1}^{\infty} \prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \Big(r(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^{\Omega}; q_{\Delta}) \\
&\qquad - \mathbb{D}_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)) \Big) \Bigg] \Bigg]
\end{aligned}
\tag{C.71}
$$

$$
\begin{aligned}
&= r(\mathbf{s}_0, \mathbf{a}_0, \mathcal{S}^{\Omega}; q_{\Delta}) \\
&\quad + \mathbb{E}_{p_d(\mathbf{s}_1 | \mathbf{s}_0, \mathbf{a}_0)} \Bigg[\mathbb{E}_{\pi(\mathbf{a}_1 \,|\, \mathbf{s}_1)} \Bigg[q_{\Delta_1}(\Delta_1 = 0) \Big(r(\mathbf{s}_1, \mathbf{a}_1, \mathcal{S}^{\Omega}; q_{\Delta}) - \mathbb{D}_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_1) \parallel p(\cdot \,|\, \mathbf{s}_1)) \Big) \\
&\qquad + \mathbb{E}_{q(\tau_2 | \mathbf{s}_1, \mathbf{a}_1)} \Bigg[\sum_{t=2}^{\infty} \prod_{t'=2}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \Big(r(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^{\Omega}; q_{\Delta}) - \mathbb{D}_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)) \Big) \Bigg] \Bigg] \Bigg],
\end{aligned}
\tag{C.72}
$$

and note that we can rearrange this expression to obtain the recursive relationship

$$
\begin{aligned}
&Q^{\pi}_{\text{sum}}(\mathbf{s}_0, \mathbf{a}_0, \mathcal{S}^{\Omega}; q_T) \\
&\quad = r(\mathbf{s}_0, \mathbf{a}_0, \mathcal{S}^{\Omega}; q_{\Delta}) + q_{\Delta_1}(\Delta_1 = 0)\, \mathbb{E}_{p_d(\mathbf{s}_{0+1} \,|\, \mathbf{s}_0, \mathbf{a}_0)} \Bigg[- \mathbb{D}_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_1) \parallel p(\cdot \,|\, \mathbf{s}_1)) \\
&\qquad + \mathbb{E}_{\pi(\mathbf{a}_1 \,|\, \mathbf{s}_1)} \Bigg[r(\mathbf{s}_1, \mathbf{a}_1, \mathcal{S}^{\Omega}; q_{\Delta}) + \mathbb{E} \Bigg[\sum_{t=2}^{\infty} \Bigg(\prod_{t'=2}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \Bigg) \Big(r(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^{\Omega}; q_{\Delta}) \\
&\qquad\quad - \mathbb{D}_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}))\Big) \Bigg] \Bigg] \Bigg],
\end{aligned}
\tag{C.73}
$$

where the innermost expectation is taken with respect to $q(\tau_2|\mathbf{s}_1, \mathbf{a}_1)$. With this result and noting that

$$
\begin{aligned}
&Q^{\pi}_{\text{sum}}(\mathbf{s}_1, \mathbf{a}_1, \mathcal{S}^{\Omega}; q_T) \\
&= r(\mathbf{s}_1, \mathbf{a}_1, \mathcal{S}^{\Omega}; q_{\Delta}) \\
&\qquad + \mathbb{E} \Bigg[\sum_{t=2}^{\infty} \Bigg(\prod_{t'=2}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \Bigg) \Big(r(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^{\Omega}; q_{\Delta}) - \mathbb{D}_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}))\Big) \Bigg],
\end{aligned}
\tag{C.74}
$$

where the expectation is again taken with respect to $q(\tau_2|\mathbf{s}_1, \mathbf{a}_1)$, we see that

$$
\begin{aligned}
&Q^{\pi}_{\text{sum}}(\mathbf{s}_0, \mathbf{a}_0, \mathcal{S}^{\Omega}; q_T) \\
&\quad = r(\mathbf{s}_0, \mathbf{a}_0, \mathcal{S}^{\Omega}; q_{\Delta}) + q_{\Delta_1}(\Delta_1 = 0)\, \mathbb{E}_{p_d(\mathbf{s}_{0+1} \,|\, \mathbf{s}_0, \mathbf{a}_0)} \Big[\mathbb{E}_{\pi(\mathbf{a}_1|\mathbf{s}_1)} \Big[Q^{\pi}_{\text{sum}}(\mathbf{s}_1, \mathbf{a}_1, \mathcal{S}^{\Omega}; q_T) \Big] \\
&\qquad - \mathbb{D}_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_1) \parallel p(\cdot \,|\, \mathbf{s}_1)) \Big]
\end{aligned}
\tag{C.75}
$$

$$
= r(\mathbf{s}_0, \mathbf{a}_0, \mathcal{S}^{\Omega}; q_{\Delta}) + q_{\Delta_1}(\Delta_1 = 0)\, \mathbb{E}_{p_d(\mathbf{s}_1|\mathbf{s}_0, \mathbf{a}_0)} \Big[V^{\pi}(\mathbf{s}_1, \mathcal{S}^{\Omega}; q_T) \Big], \tag{C.76}
$$

for $V(\mathbf{s}_{t+1}, \mathcal{S}^{\Omega}; q_T)$ as defined above, as desired. In other words, we have that

$$
\begin{aligned}
\mathcal{F}(\pi, q_T, \mathcal{S}^{\Omega}) &= \mathbb{E}_{\pi(\mathbf{a}_0 \,|\, \mathbf{s}_0) p(\mathbf{s}_0)} [Q^{\pi}_{\text{sum}}(\mathbf{s}_0, \mathbf{a}_0, \mathcal{S}^{\Omega}; q_T) - \mathbb{D}_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_0) \,\|\, p(\cdot \,|\, \mathbf{s}_0))] \\
&= \mathbb{E}_{p(\mathbf{s}_0)} [V^{\pi}(\mathbf{s}_0, \mathcal{S}^{\Omega}; q_T)].
\end{aligned}
\tag{C.77}
$$

Combining this result with Proposition 5.4 and Proposition 5.5, we finally conclude that

$$
\begin{aligned}
&\mathbb{D}_{\text{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}, T}(\cdot) \,\|\, p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}, T | \xi_{1:T+1}}(\cdot \,|\, \xi^{\star}_{1:T+1}(\mathcal{S}^{\Omega}))) \\
&= - \mathcal{F}(\pi, q_T, \mathcal{S}^{\Omega}) + C \\
&= - \mathbb{E}_{p(\mathbf{s}_0)} [V^{\pi}(\mathbf{s}_0, \mathcal{S}^{\Omega}; q_T)] + C,
\end{aligned}
\tag{C.78}
$$

where $C \stackrel{\text{def}}{=} \log p(\xi^\star_{1:T+1}(\mathcal{S}^\Omega))$ is independent of π and q_T. Hence, maximizing $V^\pi(\mathbf{s}_0, \mathcal{S}^\Omega; q_T)$ is equivalent to minimizing the objective in Equation (5.12). In other words,

$$\begin{aligned} &\arg\min_{\pi\in\Pi, q_T\in\mathcal{Q}_T} \{\mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T}(\cdot) \,\|\, p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T|\xi_{1:T+1}}(\cdot \,|\, \xi^\star_{1:T+1}(\mathcal{S}^\Omega)))\} \\ =\ &\arg\max_{\pi\in\Pi, q_T\in\mathcal{Q}_T} \mathcal{F}(\pi, q_T, \mathcal{S}^\Omega) \\ =\ &\arg\max_{\pi\in\Pi, q_T\in\mathcal{Q}_T} \mathbb{E}_{p(\mathbf{s}_0)}[V^\pi(\mathbf{s}_0, \mathcal{S}^\Omega; q_T)]. \end{aligned} \tag{C.79}$$

This concludes the proof. □

C.1.3 Optimal Behavior-Driven Variational Posterior over T

Proposition 5.6. *The optimal variational distribution $q^\star_T$ with respect to Equation (5.14) is defined recursively in terms of $q^\star_{\Delta_{t+1}}(\Delta_{t+1}=0) \forall t \in \mathbb{N}_0$ by*

$$q^\star_{\Delta_{t+1}}(\Delta_{t+1}=0; \pi, Q^\pi) = \sigma\left(\Lambda(\mathbf{s}_t, \pi, q_T, Q^\pi) + \sigma^{-1}\left(p_{\Delta_{t+1}}(\Delta_{t+1}=0)\right)\right), \tag{C.80}$$

where

$$\begin{aligned} &\Lambda(\mathbf{s}_t, \pi, q_T, Q^\pi) \\ &\quad\stackrel{\text{def}}{=} \mathbb{E}_{\pi(\mathbf{a}_{t+1}\,|\,\mathbf{s}_{t+1}) p_d(\mathbf{s}_{t+1}\,|\,\mathbf{s}_t,\mathbf{a}_t)\pi(\mathbf{a}_t\,|\,\mathbf{s}_t)}[Q^\pi(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \mathcal{S}^\Omega; q_T) - \log \mathbb{P}(\xi_{t+1}(\mathcal{S}^\Omega)=1 \,|\, \mathbf{s}_t, \mathbf{a}_t)] \end{aligned}$$

and $\sigma(\cdot)$ is the sigmoid function, that is, $\sigma(x) = \frac{1}{e^{-x}+1}$ and $\sigma^{-1}(x) = \log \frac{x}{1-x}$.

Proof. Consider $\mathcal{F}(\pi, q_T, \mathcal{S}^\Omega)$:

$$\begin{aligned} \mathcal{F}(\pi, q_T, \mathcal{S}^\Omega) &= \mathbb{E}_{\pi(\mathbf{a}_t|\mathbf{s}_t)}[Q^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_T)] \\ &= \mathbb{E}_{\pi(\mathbf{a}_t|\mathbf{s}_t)}[r(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_\Delta) + q_{\Delta_{t+1}}(\Delta_{t+1}=0)\, \mathbb{E}\left[V(\mathbf{s}_{t+1}, \mathcal{S}^\Omega; q_T)\right]]. \end{aligned} \tag{C.81}$$

Since the variational objective $\mathcal{F}(\pi, q_T, \mathcal{S}^\Omega)$ can be expressed recursively as

$$V^\pi(\mathbf{s}_t, \mathcal{S}^\Omega; q_T) \stackrel{\text{def}}{=} \mathbb{E}_{\pi(\mathbf{a}_t\,|\,\mathbf{s}_t)}\left[Q(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_T)\right] - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot\,|\,\mathbf{s}_t) \,\|\, p(\cdot\,|\,\mathbf{s}_t)),$$

with

$$\begin{aligned} Q^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_T) &\stackrel{\text{def}}{=} r(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_\Delta) + q_{\Delta_{t+1}}(\Delta_{t+1}=0)\, \mathbb{E}_{p_d(\mathbf{s}_{t+1}\,|\,\mathbf{s}_t,\mathbf{a}_t)}\left[V^\pi(\mathbf{s}_{t+1}, \mathcal{S}^\Omega; q_T)\right], \\ r(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_\Delta) &\stackrel{\text{def}}{=} \log \mathbb{P}(\xi_{t+1}(\mathcal{S}^\Omega)=1 \,|\, \mathbf{s}_t, \mathbf{a}_t) - q_{\Delta_{t+1}}(\Delta_{t+1}=1)\mathbb{D}_{KL}(q_{\Delta_{t+1}} \,\|\, p_{\Delta_{t+1}}), \end{aligned}$$

and since $\mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \,\|\, p_{\Delta_{t+1}})$ is strictly convex in $q_{\Delta_{t+1}}(\Delta_{t+1}=0)$, we can find the globally optimal Bernoulli distribution parameters $q_{\Delta_{t+1}}(\Delta_{t+1}=0)$ for all $t \in \mathbb{N}_0$ recursively. That is, it is sufficient to solve the problem

$$\begin{aligned} q^\star_{\Delta_{t+1}}(\Delta_{t+1}=0) &\stackrel{\text{def}}{=} \arg\max_{q_{\Delta_{t+1}}(\Delta_{t+1}=0)} \left\{\mathcal{F}(\pi, q_T, \mathcal{S}^\Omega)\right\} \\ &= \arg\max_{q_{\Delta_{t+1}}(\Delta_{t+1}=0)} \left\{\mathcal{F}(\pi, \{q_{\Delta_1}, \dots, q_{\Delta_{t+1}}, \dots\}, \mathbf{s}_0, \mathcal{S}^\Omega)\right\} \end{aligned} \tag{C.82}$$

where we have written q_T out in terms of its Bernoulli factors, for a fixed $t+1$. To do so, we take the derivative of $\mathcal{F}(\pi, \{q_{\Delta_1}, \dots, q_{\Delta_{t+1}}, \dots\}, \mathbf{s}_0, \mathcal{S}^\Omega)$, which—defined recursively—is given by

$$\begin{aligned}
&\mathbb{E}_{\pi(\mathbf{a}_t \,|\, \mathbf{s}_t)}\Big[Q(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_T\Big] - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)) \\
&= \mathbb{E}_{\pi(\mathbf{a}_t \,|\, \mathbf{s}_t)}\Big[r(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_\Delta) + q_{\Delta_{t+1}}(\Delta_{t+1} = 0)\, \mathbb{E}_{p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)}\Big[V^\pi(\mathbf{s}_{t+1}, \mathcal{S}^\Omega; q_T)\Big]\Big] \\
&\qquad - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t))
\end{aligned} \tag{C.83}$$

$$\begin{aligned}
&= \mathbb{E}_{\pi(\mathbf{a}_t \,|\, \mathbf{s}_t)}\Bigg[\log \mathbb{P}(\xi_{t+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_t, \mathbf{a}_t) - \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \parallel p_{\Delta_{t+1}}) \\
&\qquad + q_{\Delta_{t+1}}(\Delta_{t+1} = 0)\, \mathbb{E}_{p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)}\Big[V^\pi(\mathbf{s}_{t+1}, \mathcal{S}^\Omega; q_T)\Big]\Bigg] - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t))
\end{aligned} \tag{C.84}$$

$$\begin{aligned}
&= \mathbb{E}_{\pi(\mathbf{a}_t \,|\, \mathbf{s}_t)}\Bigg[\log \mathbb{P}(\xi_{t+1}(\mathcal{S}^\Omega) = 1 \,|\, \mathbf{s}_t, \mathbf{a}_t) - \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \parallel p_{\Delta_{t+1}}) \\
&\qquad + q_{\Delta_{t+1}}(\Delta_{t+1} = 0)\, \mathbb{E}_{p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)}\Big[V^\pi(\mathbf{s}_{t+1}, \mathcal{S}^\Omega; q_T)\Big]\Bigg] - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)),
\end{aligned} \tag{C.85}$$

with respect to $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ and set it to zero, which yields

$$\begin{aligned}
0 = &- \mathbb{E}_{\pi(\mathbf{a}_t \,|\, \mathbf{s}_t)}\Big[\mathbb{E}_{\pi(\mathbf{a}_{t+1} \,|\, \mathbf{s}_{t+1}) p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)}[Q^\pi(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \mathcal{S}^\Omega; q_T)]\Big] \\
&+ \log \frac{1 - q^\star_{\Delta_{t+1}}(\Delta_{t+1} = 0)}{1 - p_{\Delta_{t+1}}(\Delta_{t+1} = 0)} - \log \frac{q^\star_{\Delta_{t+1}}(\Delta_{t+1} = 0)}{p_{\Delta_{t+1}}(\Delta_{t+1} = 0)}.
\end{aligned} \tag{C.86}$$

Rearranging, we get

$$\frac{q^\star_{\Delta_{t+1}}(\Delta_{t+1} = 0)}{1 - q^\star_{\Delta_{t+1}}(\Delta_{t+1} = 0)} = \exp\left(\mathbb{E}[Q^\pi(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \mathcal{S}^\Omega; q_T)] + \log \frac{p_{\Delta_{t+1}}(\Delta_{t+1} = 0)}{1 - p_{\Delta_{t+1}}(\Delta_{t+1} = 0)}\right), \tag{C.87}$$

where the expectation is taken with respect to $\pi(\mathbf{a}_{t+1} \,|\, \mathbf{s}_{t+1}) p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t) \pi(\mathbf{a}_t \,|\, \mathbf{s}_t)$ and the Q-function depends on $q(\Delta_{t'})$ with $t' > t$, but not on $q^\star_{\Delta_{t+1}}(\Delta_{t+1} = 0)$. Solving for $q^\star_{\Delta_{t+1}}(\Delta_{t+1} = 0)$. Solving for $q^\star_{\Delta_{t+1}}(\Delta_{t+1} = 0)$, we obtain

$$q^\star_{\Delta_{t+1}}(\Delta_{t+1} = 0) = \frac{\exp(\mathbb{E}_{p_{\pi p_d} \pi(\mathbf{a}_t \,|\, \mathbf{s}_t)}[Q^\pi(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \mathcal{S}^\Omega; q_T)] + \log \frac{p_{\Delta_{t+1}}(\Delta_{t+1}=0)}{1-p_{\Delta_{t+1}}(\Delta_{t+1}=0)})}{1 + \exp(\mathbb{E}_{p_{\pi p_d} \pi(\mathbf{a}_t \,|\, \mathbf{s}_t)}[Q^\pi(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \mathcal{S}^\Omega; q_T)] + \log \frac{p_{\Delta_{t+1}}(\Delta_{t+1}=0)}{1-p_{\Delta_{t+1}}(\Delta_{t+1}=0)})} \tag{C.88}$$

$$= \sigma\Big(\mathbb{E}_{p_{\pi p_d}}[Q^\pi(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \mathcal{S}^\Omega; q_T)] + \sigma^{-1}\left(p_{\Delta_{t+1}}(\Delta_{t+1} = 0)\right)\Big), \tag{C.89}$$

where $p_{\pi p_d} \stackrel{\text{def}}{=} \pi(\mathbf{a}_{t+1} \mid \mathbf{s}_{t+1}) p_d(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)$, $\sigma(\cdot)$ is the sigmoid function with $\sigma(x) = \frac{1}{e^{-x}+1}$ and $\sigma^{-1}(x) = \log \frac{x}{1-x}$. This concludes the proof. □

Remark 1. *As can be seen from Proposition 5.6, the optimal approximation to the posterior over T trades off short-term rewards via $\mathbb{E}_{\pi(\mathbf{a}_t \mid \mathbf{s}_t)}[r(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_\Delta)]$, long-term rewards via $\mathbb{E}_{\pi(\mathbf{a}_{t+1} \mid \mathbf{s}_{t+1}) p_d(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)}[Q^\pi(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \mathcal{S}^\Omega; q_T)]$, and the prior log-odds of not achieving the behavior at a given point in time conditioned on the behavior not having been achieved yet, $\frac{p_{\Delta_{t+1}}(\Delta_{t+1}=0)}{1-p_{\Delta_{t+1}}(\Delta_{t+1}=0)}$.*

C.1.4 Dynamic-Discount Behavior-Driven Policy Iteration

Theorem 5.2. *Assume $|\mathcal{A}| < \infty$ and that the MDP is ergodic.*

1. *Dynamic-Discount Behavior-Driven Policy Evaluation (D2BD-PE): Given a policy π and a function $Q^0 : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, define $Q^{i+1} = \mathcal{T}^\pi Q^i$. Then the sequence Q^i converges to the lower bound in Theorem 5.3.*

2. *Dynamic-Discount Behavior-Driven Policy Improvement (D2BD-PI): The policy*

$$\pi^+ = \arg\max_{\pi' \in \Pi} \left\{ \mathbb{E}_{\pi'(\mathbf{a}_t \mid \mathbf{s}_t)} \left[Q^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_T) \right] - \mathbb{D}_{\mathrm{KL}}(\pi'(\cdot \mid \mathbf{s}_t) \,||\, p(\cdot \mid \mathbf{s}_t)) \right\} \tag{C.90}$$

and the variational distribution over T recursively defined in terms of

$$\begin{aligned} &q^+(\Delta_{t+1} = 0; \pi, Q^\pi) \\ &\quad = \sigma\Big(\mathbb{E}_{\pi(\mathbf{a}_{t+1} \mid \mathbf{s}_{t+1}) p_d(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)}[Q^\pi(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \mathcal{S}^\Omega; q_T)] + \sigma^{-1}\left(p_{\Delta_{t+1}}(\Delta_{t+1} = 0)\right) \Big) \end{aligned} \tag{C.91}$$

improve the variational objective. In other words, we have that $V^{\pi^+}(\mathbf{s}_0, \mathcal{S}^\Omega; q_T) \geq V^\pi(\mathbf{s}_0, \mathcal{S}^\Omega; q_T)$ and $V^\pi(\mathbf{s}_0, \mathcal{S}^\Omega; q_T^+) \geq V^\pi(\mathbf{s}_0, \mathcal{S}^\Omega; q_T)$ for all $\mathbf{s}_0 \in \mathcal{S}$.

3. *Alternating between D2BD-PE and D2BD-PI converges to a policy $\pi^\star$ and a variational distribution over T, $q_T^\star$, such that $Q^{\pi^\star}(\mathbf{s}, \mathbf{a}, \mathcal{S}^\Omega; q_T^\star) \geq Q^\pi(\mathbf{s}, \mathbf{a}, \mathcal{S}^\Omega; q_T)$ for all $(\pi, q_T) \in \Pi \times \mathcal{Q}_T$ and any $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$.*

Proof. Parts of this proof are adapted from the proof given in Haarnoja et al. (2018a), modified for the Bellman operator proposed in Definition 5.1.

1. Dynamic-Discount Behavior-Driven Policy Evaluation (D2BD-PE): Instead of absorbing the KL term into the Q-function, we can define an KL-regularized reward as

$$\begin{aligned} r^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_\Delta) \stackrel{\text{def}}{=} &\log \mathbb{P}(\xi_{t+1}(\mathcal{S}^\Omega) = 1 \mid \mathbf{s}_t, \mathbf{a}_t) - \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \parallel p_{\Delta_{t+1}}) \\ &- q_{\Delta_{t+1}}(\Delta_{t+1} = 0)\, \mathbb{E}_{p_d(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)}[\mathbb{D}_{\mathrm{KL}}(\pi(\cdot \mid \mathbf{s}_{t+1}) \parallel p(\cdot \mid \mathbf{s}_{t+1}))]. \end{aligned} \tag{C.92}$$

We can then write an update rule according to Definition 5.1 as

$$\tilde{Q}(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_T) \leftarrow r^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_\Delta) + q_{\Delta_{t+1}}(\Delta_{t+1} = 0)\, \mathbb{E}[\tilde{Q}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \mathcal{S}^\Omega; q_T)], \tag{C.93}$$

where $q_{\Delta_{t+1}}(\Delta_{t+1} = 0) \leq 1$ and the expectation is computed under the conditional state-action distribution $\pi(\mathbf{a}_{t+1} \,|\, \mathbf{s}_{t+1}) p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)$. This update is similar to a Bellman update (Sutton et al., 1998), but with a discount factor given by $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$. In general, this discount factor $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ can be computed dynamically based on the current state and action, such as in Equation (C.80). As discussed in White (2017), this Bellman operator is still a contraction mapping so long as the Markov chain induced by the current policy is ergodic and there exists a state such that $q_{\Delta_{t+1}}(\Delta_{t+1} = 0) < 1$. The first condition is true by assumption. The second condition is true since $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ is given by Equation (C.80), which is always strictly between 0 and 1. Therefore, we apply convergence results for policy evaluation with transition-dependent discount factors (White, 2017) to this contraction mapping, and the result immediately follows.

2. Dynamic-Discount Behavior-Driven Policy Improvement (D2BD-PI): Let $\pi_{\text{old}} \in \Pi$ and let $Q^{\pi_{\text{old}}}$ and $V^{\pi_{\text{old}}}$ be the behavior-driven state and state-action value functions from Definition 5.1, let q_T be some variational distribution over T, and let π_{new} be given by

$$\pi_{\text{new}}(\mathbf{a}_t|\mathbf{s}_t) = \arg\max_{\pi' \in \Pi} \left\{ \mathbb{E}_{\pi'(\mathbf{a}_t \,|\, \mathbf{s}_t)} \left[Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_T) \right] - \mathbb{D}_{\text{KL}}(\pi'(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)) \right\} \tag{C.94}$$

$$\stackrel{\text{def}}{=} \arg\max_{\pi' \in \Pi} \mathcal{J}_{\pi_{\text{old}}}(\pi'(\mathbf{a}_t \,|\, \mathbf{s}_t); q_T). \tag{C.95}$$

Then, it must be true that $\mathcal{J}_{\pi_{\text{old}}}(\pi_{\text{old}}(\mathbf{a}_t|\mathbf{s}_t); q_T) \leq \mathcal{J}_{\pi_{\text{old}}}(\pi_{\text{new}}(\mathbf{a}_t|\mathbf{s}_t); q_T)$, since one could set $\pi_{\text{new}} = \pi_{\text{old}} \in \Pi$. Thus,

$$\begin{aligned} &\mathbb{E}_{\pi_{\text{new}}(\mathbf{a}_t|\mathbf{s}_t)} \left[Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_T) \right] - \mathbb{D}_{\text{KL}}(\pi_{\text{new}}(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)) \\ &\quad \geq \mathbb{E}_{\pi_{\text{old}}(\mathbf{a}_t|\mathbf{s}_t)} \left[Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_T) \right] - \mathbb{D}_{\text{KL}}(\pi_{\text{old}}(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)), \end{aligned} \tag{C.96}$$

and since

$$V^{\pi_{\text{old}}}(\mathbf{s}_t, \mathcal{S}^\Omega; q_T) = \mathbb{E}_{\pi_{\text{old}}(\mathbf{a}_t|\mathbf{s}_t)} \left[Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_T) \right] - \mathbb{D}_{\text{KL}}(\pi_{\text{old}}(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)), \tag{C.97}$$

we get

$$\mathbb{E}_{\pi_{\text{new}}(\mathbf{a}_t|\mathbf{s}_t)} \left[Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^\Omega; q_T) \right] - \mathbb{D}_{\text{KL}}(\pi_{\text{new}}(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)) \geq V^{\pi_{\text{old}}}(\mathbf{s}_t, \mathcal{S}^\Omega; q_T). \tag{C.98}$$

We can now write the Bellman equation as

$$
\begin{aligned}
& Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^{\Omega}; q_T) \\
& = q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \log \mathbb{P}(\xi_{t+1}(\mathcal{S}^{\Omega}) = 1 \,|\, \mathbf{s}_t, \mathbf{a}_t) + q_{\Delta_{t+1}}(\Delta_{t+1} = 0) \, \mathbb{E}_{p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)}[V^{\pi_{\text{old}}}(\mathbf{s}_{t+1}, \mathcal{S}^{\Omega}; q_T)] \qquad \text{(C.99)} \\
& \leq q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \log \mathbb{P}(\xi_{t+1}(\mathcal{S}^{\Omega}) = 1 \,|\, \mathbf{s}_t, \mathbf{a}_t) \\
& \qquad + q_{\Delta_{t+1}}(\Delta_{t+1} = 0) \, \mathbb{E}_{p_d(\mathbf{s}_{t'}|\mathbf{s}_t, \mathbf{a}_t)}[\mathbb{E}_{\pi_{\text{new}}(\mathbf{a}_{t'}|\mathbf{s}_{t'})} \left[Q^{\pi_{\text{old}}}(\mathbf{s}_{t'}, \mathbf{a}_{t'}, \mathcal{S}^{\Omega}; q_T) \right] \qquad \text{(C.100)} \\
& \qquad\qquad - \mathbb{D}_{\text{KL}}(\pi_{\text{new}}(\cdot \,|\, \mathbf{s}_{t'}) \parallel p(\cdot \,|\, \mathbf{s}_{t'}))], \\
& \vdots \\
& \leq Q^{\pi_{\text{new}}}(\mathbf{s}_t, \mathbf{a}_t, \mathcal{S}^{\Omega}; q_T) \qquad \text{(C.101)}
\end{aligned}
$$

where we defined $t' \stackrel{\text{def}}{=} t + 1$, repeatedly applied the Bellman backup operator defined in Definition 5.1 and used the bound in Equation (C.98). Convergence follows from Dynamic-Discount Behavior-Driven Policy Evaluation above.

3. Locally Optimal Variational Dynamic-Discount Behavior-Driven Policy Iteration: Define π^i to be a policy at iteration i. By D2BD-PI for a given q_T, the sequence of state-action value functions $\{Q^{\pi^i}(q_T)\}_{i=1}^{\infty}$ is monotonically increasing in i. Since the reward is finite and the negative KL divergence is upper bounded by zero, $Q^{\pi}(q_T)$ is upper bounded for $\pi \in \Pi$ and the sequence $\{\pi^i\}_{i=1}^{\infty}$ converges to some $\pi^{\star}$. To see that $\pi^{\star}$ is an optimal policy, note that it must be the case that $\mathcal{J}_{\pi^{\star}}(\pi^{\star}(\mathbf{a}_t|\mathbf{s}_t); q_T) > \mathcal{J}_{\pi^{\star}}(\pi(\mathbf{a}_t \,|\, \mathbf{s}_t); q_T)$ for any $\pi \in \Pi$ with $\pi \neq \pi^{\star}$. By the argument used in D2BD-PI above, it must be the case that the behavior-driven state-action value of the converged policy is higher than that of any other non-converged policy in Π, that is, $Q^{\pi^{\star}}(\mathbf{s}_t, \mathbf{a}_t; q_T) > Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t; q_T)$ for all $\pi \in \Pi$ and any $q_T^i \in \mathcal{Q}_T$ and $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$. Therefore, given q_T, $\pi^{\star}$ must be optimal in Π, which concludes the proof.

4. Globally Optimal Variational Dynamic-Discount Behavior-Driven Policy Iteration: Let π^i be a policy and let q_T^i be variational distributions over T at iteration i. By Locally Optimal Variational Dynamic-Discount Behavior-Driven Policy Iteration, for a *fixed* q_T^i with $q_T^i = q_T^j \forall i, j \in \mathbb{N}_0$, the sequence of $\{(\pi^i, q_T^i)\}_{i=1}^{\infty}$ increases the objective in Equation (C.19) at each iteration and converges to a stationary point in π^i, where $Q^{\pi^{\star}}(\mathbf{s}_t, \mathbf{a}_t; q_T^i) > Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t; q_T^i)$ for all $\pi \in \Pi$ and any $q_T^i \in \mathcal{Q}_T$ and $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$. Since the objective in Equation (C.19) is concave in q_T, it must be the case that for $q_T^{\star^i} \in \mathcal{Q}_T$, the optimal variational distribution over T at iteration i, defined recursively by

$$
\begin{aligned}
& q^{\star^i}(\Delta_{t+1} = 0; \pi^i, Q^{\pi^i}) \\
& = \sigma\Big(\mathbb{E}_{\pi(\mathbf{a}_{t+1} \,|\, \mathbf{s}_{t+1}) p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)}[Q^{\pi^i}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \mathcal{S}^{\Omega}; q_T(\pi^i, Q^{\pi^i}))] + \sigma^{-1}(p_{\Delta_{t+1}}(\Delta_{t+1} = 0)) \Big),
\end{aligned}
$$

for $t \in \mathbb{N}_0$, $Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t; q_T^{\star}) > Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t; q_T)$ for all $\pi \in \Pi$ and any $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$. Note that q_T is defined implicitly in terms of π^i and Q^{π^i}, that is, the optimal variational distribution over T at iteration i is defined as a function of the policy and Q-function at iteration i. Hence, it must then be true that $Q^{\pi^\star}(\mathbf{s}_t, \mathbf{a}_t; q_T^{\star}) > Q^{\pi^\star}(\mathbf{s}_t, \mathbf{a}_t; q_T)$ for all $q_T^{\star}(\pi^\star, Q^{\pi^\star}) \in \mathcal{Q}_T$ and for any $\pi^\star \in \Pi$ and $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$. In other words, for an optimal policy and corresponding Q-function, there exists an optimal variational distribution over T that maximizes the Q-function, given the optimal policy. Repeating locally optimal variational behavior-driven policy iteration under the new variational distribution $q_T^{\star}(\pi^\star, Q^{\pi^\star})$ will yield an optimal policy $\pi^{\star\star}$ and computing the corresponding optimal variational distribution, $q_T^{\star\star}(\pi^{\star\star}, Q^{\pi^{\star\star}})$ will further increase the variational objective such that for $\pi^{\star\star} \in \Pi$ and $q_T^{\star\star}(\pi^{\star\star}, Q^{\pi^{\star\star}}) \in \mathcal{Q}_T$, we have that

$$Q^{\pi^{\star\star}}(\mathbf{s}_t, \mathbf{a}_t; q_T^{\star\star}) > Q^{\pi^{\star\star}}(\mathbf{s}_t, \mathbf{a}_t; q_T^{\star}) > Q^{\pi^\star}(\mathbf{s}_t, \mathbf{a}_t; q_T^{\star}) > Q^{\pi^\star}(\mathbf{s}_t, \mathbf{a}_t; q_T) \quad \text{(C.102)}$$

for any $\pi^\star \in \Pi$ and $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$. Hence, globally optimal variational behavior-driven policy iteration increases the variational objective at every step. Since the objective is upper bounded (by virtue of the rewards being finite and the negative KL divergence being upper bounded by zero) and the sequence of $\{(\pi^i, q_T^i)\}_{i=1}^{\infty}$ increases the objective Equation (C.19) at each iteration, by the monotone convergence theorem, the objective value converges to a supremum and since the objective function is concave the supremum is unique. Hence, since the supremum is unique and obtained via globally optimal variational behavior-driven policy iteration on $(\pi, q_T) \in \Pi \times \mathcal{Q}_T$, the sequence of $\{(\pi^i, q_T^i)\}_{i=1}^{\infty}$ converges to a unique stationary point $(\pi^\star, q_T^{\star}) \in \Pi \times \mathcal{Q}_T$, where $Q^{\pi^\star}(\mathbf{s}_t, \mathbf{a}_t; q_T^{\star}) > Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t; q_T^i)$ for all $\pi \in \Pi$ and any $q_T^i \in \mathcal{Q}_T$ and $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$.

□

Corollary 5.5. *(Optimality of Variational Behavior-Driven Policy Iteration) Variational Dynamic-Discount Behavior-Driven Policy Iteration on $(\pi, q_T) \in \Pi \times \mathcal{Q}_T$ results in an optimal policy at least as good as or better than any optimal policy attainable from policy iteration on $\pi \in \Pi$ alone.*

Remark 2. *The convergence proof of D2BD-PE assumes a transition-dependent discount factor (White, 2017), because the variational distribution used in Equation (C.80) depends on the next state and action as well as on the desired behavior.*

C.1.5 Finite- and Infinite-Horizon Outcome-Driven Variational Objectives

In this section, we present detailed derivations and proofs for the results in Sections 5.3.1 and Section 5.3.2.

Proposition 5.2. *Let $q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\tilde{\tau}_{0:t} \mid \mathbf{s}_0)$ be as defined in Equation (5.24). Then, given any initial state $\mathbf{s}_0$, termination time $t^\star = t+1$, and outcome $\mathbf{g}$,*

$$\mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\cdot \mid \mathbf{s}_0) \,\|\, p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0,\mathbf{S}_{t^\star}}(\cdot \mid \mathbf{s}_0, \mathbf{g})) = \log p(\mathbf{g} \mid \mathbf{s}_0) - \bar{\mathcal{F}}(\pi, \mathbf{s}_0, \mathbf{g}), \tag{C.103}$$

where

$$\bar{\mathcal{F}}(\pi, \mathbf{s}_0, \mathbf{g}) \stackrel{\text{def}}{=} \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\tilde{\tau}_{0:t} \mid \mathbf{s}_0)}\Big[\log p_d(\mathbf{g} \mid \mathbf{s}_t, \mathbf{a}_t) - \sum_{t'=0}^{t} \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \mid \mathbf{s}_{t'}) \,||\, p(\cdot \mid \mathbf{s}_{t'}))\Big], \tag{C.104}$$

and since $\log p(\mathbf{g}|\mathbf{s}_0)$ is constant in π,

$$\arg\min_{\pi \in \Pi} \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\cdot \mid \mathbf{s}_0) \,\|\, p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0,\mathbf{S}_{t^\star}}(\cdot \mid \mathbf{s}_0, \mathbf{g})) = \arg\max_{\pi \in \Pi} \bar{\mathcal{F}}(\pi, \mathbf{s}_0, \mathbf{g}). \tag{C.105}$$

Proof. To find an approximation to the posterior $p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0,\mathbf{S}_{t^\star}}(\cdot \mid \mathbf{s}_0, \mathbf{g})$, we can use variational inference. To do so, we consider the trajectory distribution under $p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0,\mathbf{S}_{t^\star}}(\cdot \mid \mathbf{s}_0, \mathbf{g})$, which by Bayes' Theorem is given by

$$p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0,\mathbf{S}_{t^\star}}(\tilde{\tau}_{0:t} \mid \mathbf{s}_0, \mathbf{g}) = \frac{p_d(\mathbf{g} \mid \mathbf{s}_t, \mathbf{a}_t) p(\mathbf{a}_t \mid \mathbf{s}_t) \prod_{t'=0}^{t-1} p_d(\mathbf{s}_{t'+1} \mid \mathbf{s}_{t'}, \mathbf{a}_{t'}) p(\mathbf{a}_{t'} \mid \mathbf{s}_{t'})}{p(\mathbf{g} \mid \mathbf{s}_0)}, \tag{C.106}$$

where $t = t^\star - 1$, and we denote the state–action trajectory realization from action $\mathbf{a}_0$ to $\mathbf{a}_t$ by $\tilde{\tau}_{0:t} \stackrel{\text{def}}{=} \{\mathbf{a}_0, \mathbf{s}_1, \mathbf{a}_1, ..., \mathbf{s}_t, \mathbf{a}_t\}$. Inferring an approximation to the posterior distribution $p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0,\mathbf{S}_{t^\star}}(\cdot \mid \mathbf{s}_0, \mathbf{g})$ then becomes equivalent to finding a variational distribution $q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|\mathbf{S}_0}(\cdot \mid \mathbf{s}_0)$, which induces a trajectory distribution $q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\cdot \mid \mathbf{s}_0)$ that minimizes the KL divergence from $q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\cdot \mid \mathbf{s}_0)$ to $p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0,\mathbf{S}_{t^\star}}(\cdot \mid \mathbf{s}_0, \mathbf{g})$:

$$\min_{q \in \bar{\mathcal{Q}}} \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\cdot \mid \mathbf{s}_0) \parallel p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0,\mathbf{S}_{t^\star}}(\cdot \mid \mathbf{s}_0, \mathbf{g})). \tag{C.107}$$

If we find a distribution $q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\cdot \mid \mathbf{s}_0)$ for which the resulting KL divergence is zero, then $q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\cdot \mid \mathbf{s}_0)$ is the exact posterior. If the KL divergence is positive, then $q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\cdot \mid \mathbf{s}_0)$ is an approximate posterior. To solve the variational problem in Equation (C.107), we can define a factorized variational family

$$q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\tilde{\tau}_{0:t} \mid \mathbf{s}_0) \stackrel{\text{def}}{=} \pi(\mathbf{a}_t \mid \mathbf{s}_t) \prod_{t'=0}^{t-1} q_{\mathbf{S}_{t'+1}|\mathbf{S}_{t'},\mathbf{A}_{t'}}(\mathbf{s}_{t'+1} \mid \mathbf{s}_{t'}, \mathbf{a}_{t'}) \pi(\mathbf{a}_{t'} \mid \mathbf{s}_{t'}), \tag{C.108}$$

where $\mathbf{A}_{0:t}$ and $\mathbf{S}_{1:t}$ are latent variables over which to infer an approximate posterior distribution, and the product is from $t = 0$ to $t = t^\star - 1$ to exclude the conditional distribution over the (observed) state $\mathbf{S}_{t+1} = \mathbf{g}$ from the variational distribution.

Returning to the variational problem in Equation (C.107), we can now write

$$\begin{aligned}
&\mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\cdot \,|\, \mathbf{s}_0) \parallel p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0,\mathbf{S}_{t^\star}}(\cdot \,|\, \mathbf{s}_0, \mathbf{g})) \\
&= \int_{\mathcal{A}^{t+1}} \int_{\mathcal{S}^t} q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\tilde{\tau}_{0:t} \,|\, \mathbf{s}_0) \log \frac{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\tilde{\tau}_{0:t} \,|\, \mathbf{s}_0)}{p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0,\mathbf{S}_{t^\star}}(\tilde{\tau}_{0:t}|\mathbf{s}_0, \mathbf{g})} d\mathbf{s}_{1:t} d\mathbf{a}_{0:t} \\
&= -\bar{\mathcal{F}}(\pi, \mathbf{s}_0, \mathbf{g}) + \log p(\mathbf{g}|\mathbf{s}_0),
\end{aligned} \tag{C.109}$$

where

$$\begin{aligned}
&\bar{\mathcal{F}}(\pi, \mathbf{s}_0, \mathbf{g}) \\
&\stackrel{\text{def}}{=} \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\tilde{\tau}_{0:t} \,|\, \mathbf{s}_0)} \Bigg[\log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) + \log p(\mathbf{a}_t \,|\, \mathbf{s}_t) - \log \pi(\mathbf{a}_t \,|\, \mathbf{s}_t) \\
&\qquad + \sum_{t'=0}^{t-1} \log p(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'}) + \log p_d(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) \\
&\qquad - \log \pi(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'}) - \log q_{\mathbf{S}_{t'+1}|\mathbf{S}_{t'},\mathbf{A}_{t'}}(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) \Bigg]
\end{aligned} \tag{C.110}$$

and

$$\log p(\mathbf{g}|\mathbf{s}_0) = \log \int_{\mathcal{A}^{t+1}} \int_{\mathcal{S}^t} p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\tilde{\tau}_{0:t} \,|\, \mathbf{s}_0) d\mathbf{s}_{1:t} d\mathbf{a}_{0:t} \tag{C.111}$$

is a log-marginal likelihood. Following Haarnoja et al. (2018a), we define the variational distribution over next states to be the true transition dynamics, that is, $q_{\mathbf{S}_{t+1}|\mathbf{S}_t,\mathbf{A}_t}(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t) = p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)$, so that

$$q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\tilde{\tau}_{0:t} \,|\, \mathbf{s}_0) \stackrel{\text{def}}{=} \pi(\mathbf{a}_t \,|\, \mathbf{s}_t) \prod_{t'=0}^{t-1} p_d(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) \pi(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'}). \tag{C.112}$$

We can then simplify $\bar{\mathcal{F}}(\pi, \mathbf{s}_0, \mathbf{g})$ to

$$\bar{\mathcal{F}}(\pi, \mathbf{s}_0, \mathbf{g}) = \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\tilde{\tau}_{0:t} \,|\, \mathbf{s}_0)} \left[\log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) - \sum_{t'=0}^{t} \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_{t'}) \parallel p(\cdot \,|\, \mathbf{s}_{t'})) \right]. \tag{C.113}$$

Since $\log p(\mathbf{g}|\mathbf{s}_0)$ is constant in π, solving the variational optimization problem in Equation (C.107) is equivalent to maximizing the variational objective with respect to $\pi \in \Pi$, where Π is a family of policy distributions. □

Corollary 5.2. *The objective in Equation (5.25) corresponds to KL-regularized reinforcement learning with a time-varying reward function given by*

$$r(\mathbf{s}_{t'}, \mathbf{a}_{t'}, \mathbf{g}, t') \stackrel{\text{def}}{=} \mathbb{I}\{t' = t\} \log p_d(\mathbf{g} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}).$$

Proof. Let

$$r(\mathbf{s}_{t'}, \mathbf{a}_{t'}, \mathbf{g}, t') \stackrel{\text{def}}{=} \mathbb{I}\{t' = t\} \log p_d(\mathbf{g} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) \tag{C.114}$$

and note that the objective

$$\bar{\mathcal{F}}(\pi, \mathbf{s}_0, \mathbf{g}) = \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:T}|\mathbf{s}_0}(\cdot \,|\, \mathbf{s}_0)} \left[\log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) - \sum_{t=0}^{t} \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \,\|\, p(\cdot \,|\, \mathbf{s}_t)) \right] \tag{C.115}$$

can equivalently be written as

$$\bar{\mathcal{F}}(\pi, \mathbf{s}_0, \mathbf{g}) = \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:T}|\mathbf{s}_0}(\cdot \,|\, \mathbf{s}_0)} \left[\sum_{t'=0}^{t} r(\mathbf{s}_{t'}, \mathbf{a}_{t'}, \mathbf{g}, t') - \sum_{t'=0}^{t} \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_{t'}) \parallel p(\cdot \,|\, \mathbf{s}_{t'})) \right] \tag{C.116}$$

$$= \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:T}|\mathbf{s}_0}(\cdot \,|\, \mathbf{s}_0)} \left[\sum_{t'=0}^{t} r(\mathbf{s}_{t'}, \mathbf{a}_{t'}, \mathbf{g}, t') - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_{t'}) \parallel p(\cdot \,|\, \mathbf{s}_{t'})) \right], \tag{C.117}$$

which, as shown in Haarnoja et al. (2018a), can be written in the form of a conventional Q-function for KL-regularized RL. □

Proposition 5.7. *Let*

$$q_{\tilde{\boldsymbol{\tau}}_{0:T}, T|\mathbf{s}_0}(\tilde{\tau}_{0:t}, t \,|\, \mathbf{s}_0) = q_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t} \,|\, t) q_T(t), \tag{C.118}$$

let $q_T(t)$ be a variational distribution defined on $t \in \mathbb{N}_0$, and let $q_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t} \,|\, t)$ be as defined in Equation (5.24). Then, given any initial state $\mathbf{s}_0$ and outcome $\mathbf{g}$, we have that

$$\mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\tau}}_{0:T}, T|\mathbf{s}_0}(\cdot \,|\, \mathbf{s}_0) \,\|\, p_{\tilde{\boldsymbol{\tau}}_{0:T}, T|\mathbf{s}_0, \mathbf{s}_{T^\star}}(\cdot \,|\, \mathbf{s}_0, \mathbf{g})) = \log p(\mathbf{g}|\mathbf{s}_0) - \mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}), \tag{C.119}$$

where

$$\begin{aligned} \mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) \stackrel{\text{def}}{=} \sum_{t=0}^{\infty} q_T(t)\, \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t} \,|\, t)} \Big[& \log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) \\ & - \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\tau}}_{0:T}, T|\mathbf{s}_0}(\cdot \,|\, \mathbf{s}_0) \,||\, p_{\tilde{\boldsymbol{\tau}}_{0:T}, T|\mathbf{s}_0}(\cdot \,|\, \mathbf{s}_0)) \Big] \end{aligned} \tag{C.120}$$

and $\log p(\mathbf{g} \,|\, \mathbf{s}_0)$ is constant in π and q_T.

Proof. In general, solving the variational problem

$$\min_{q \in \mathcal{Q}} \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\tau}}_{0:T}, T|\mathbf{s}_0}(\cdot \,|\, \mathbf{s}_0) \parallel p_{\tilde{\boldsymbol{\tau}}_{0:T}, T|\mathbf{s}_0, \mathbf{s}_{T^\star}}(\cdot \,|\, \mathbf{s}_0, \mathbf{g})) \tag{C.121}$$

from Section 5.3.2 in closed form is challenging, but as in the fixed-time setting, we can take advantage of the fact that, by choosing a variational family parameterized by

$$q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t} \,|\, t) \stackrel{\text{def}}{=} \pi(\mathbf{a}_t \,|\, \mathbf{s}_t) \prod_{t'=0}^{t-1} p_d(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'})\, \pi(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'}), \tag{C.122}$$

with $\pi \in \Pi$, we can follow the same steps as in the proof for Proposition 5.2 and show that given any initial state $\mathbf{s}_0$ and outcome $\mathbf{g}$,

$$\mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T|\mathbf{S}_0}(\cdot \,|\, \mathbf{s}_0) \parallel p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T|\mathbf{S}_0,\mathbf{S}_{T^\star}}(\cdot \,|\, \mathbf{s}_0, \mathbf{g})) = \log p(\mathbf{g} \,|\, \mathbf{s}_0) - \mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}), \tag{C.123}$$

where

$$\begin{aligned} \mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) \stackrel{\text{def}}{=} \sum_{t=0}^{\infty} q_T(t)\, \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t} \,|\, t)} \Big[& \log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) \\ & - \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T|\mathbf{S}_0}(\cdot \,|\, \mathbf{s}_0) \parallel p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T|\mathbf{S}_0}(\cdot \,|\, \mathbf{s}_0)) \Big], \end{aligned} \tag{C.124}$$

where $q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T|\mathbf{S}_0}(\tilde{\tau}_{0:t}, t \,|\, \mathbf{s}_0) \stackrel{\text{def}}{=} q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t} \,|\, t) q_T(t)$, and hence, solving the variational problem in Equation (5.28) is equivalent to maximizing $\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g})$ with respect to π and q_T. □

C.1.6 Recursive Outcome-Driven Variational Objective & Outcome-Driven Bellman Backup Operator

Proposition 5.8. *Let the variational distribution factorize as*

$$q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T}(\tilde{\tau}_{0:t}, t \,|\, \mathbf{s}_0) = q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t} \,|\, t) q_T(t), \tag{C.125}$$

let

$$q_T(t) = q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \tag{C.126}$$

be a variational distribution defined on $t \in \mathbb{N}_0$, and let $q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t} \,|\, t)$ be as defined in Equation (5.24). Then, given any initial state $\mathbf{s}_0$ and outcome $\mathbf{g}$, Equation (C.120) can be rewritten as

$$\begin{aligned} & \mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) \\ & = \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_0|\mathbf{S}_0}(\tilde{\tau}_0 \,|\, \mathbf{s}_0)} \left[\sum_{t=0}^{\infty} \left(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \right) \Big(r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \,\|\, p(\cdot \,|\, \mathbf{s}_t)) \Big) \right] \end{aligned} \tag{C.127}$$

where

$$r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) \stackrel{\text{def}}{=} q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) - \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \,\|\, p_{\Delta_{t+1}}). \tag{C.128}$$

Proof. Consider the variational objective $\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g})$ in Equation (C.120):

$$
\begin{aligned}
&\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) \\
&= \sum_{t=0}^{\infty} q_T(t)\, \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)} \Big[\log p_d(\mathbf{g}\,|\,\mathbf{s}_t, \mathbf{a}_t) \\
&\qquad - \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\tau}}_{0:T},T|\mathbf{S}_0}(\cdot\,|\,\mathbf{s}_0) \parallel p_{\tilde{\boldsymbol{\tau}}_{0:T},T|\mathbf{S}_0}(\cdot\,|\,\mathbf{s}_0)) \Big] && \text{(C.129)} \\
&= \sum_{t=0}^{\infty} q_T(t)\, \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)} \left[\log p_d(\mathbf{g}\,|\,\mathbf{s}_t, \mathbf{a}_t) - \log \frac{q_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t) q_T(t)}{p_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t) p_T(t)} d\tilde{\tau}_{0:t} \right] && \text{(C.130)} \\
&= \sum_{t=0}^{\infty} q_T(t)\, \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)} \left[\log p_d(\mathbf{g}\,|\,\mathbf{s}_t, \mathbf{a}_t) - \log \frac{q_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)}{p_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)} \right] \\
&\qquad - \sum_{t=0}^{\infty} q_T(t) \log \frac{q_T(t)}{q_T(t)}. && \text{(C.131)}
\end{aligned}
$$

Noting that $\sum_{t=0}^{\infty} q_T(t) \log \frac{q_T(t)}{q_T(t)} = \mathbb{D}_{\mathrm{KL}}(q_T \parallel p_T)$, we can write

$$
\begin{aligned}
&\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) \\
&= \sum_{t=0}^{\infty} q_T(t)\, \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)} \left[\log p_d(\mathbf{g}\,|\,\mathbf{s}_t, \mathbf{a}_t) - \log \frac{q_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)}{p_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)} \right] - \mathbb{D}_{\mathrm{KL}}(q_T \parallel p_T) && \text{(C.132)} \\
&= \sum_{t=0}^{\infty} q_T(t)\, \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)} \Big[\log p_d(\mathbf{g}\,|\,\mathbf{s}_t, \mathbf{a}_t) \Big] \\
&\qquad - \sum_{t=0}^{\infty} q_T(t)\, \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)} \left[\log \frac{q_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)}{p_{\tilde{\boldsymbol{\tau}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)} \right] && \text{(C.133)} \\
&\qquad - \mathbb{D}_{\mathrm{KL}}(q_T \parallel p_T).
\end{aligned}
$$

Further noting that for an infinite-horizon trajectory distribution

$$
q_{\tilde{\boldsymbol{\tau}}_{t'}|\mathbf{S}_{t'}}(\tilde{\tau}_{t'}\,|\,\mathbf{s}_{t'}) \stackrel{\text{def}}{=} \prod_{t=t'}^{\infty} p_d(\mathbf{s}_{t+1}\,|\,\mathbf{s}_t, \mathbf{a}_t) \pi(\mathbf{a}_t\,|\,\mathbf{s}_t), \tag{C.134}
$$

trajectory realization $\tilde{\tau}_{t+1} \stackrel{\text{def}}{=} \{\tau_{t'}\}_{t'=t+1}^{\infty}$, and any joint probability density $f(\mathbf{s}_t, \mathbf{a}_t)$,

$$\sum_{t=0}^{\infty} q_T(t)\, \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)} \Big[f(\mathbf{s}_t, \mathbf{a}_t) \Big] \tag{C.135}$$

$$= \sum_{t=0}^{\infty} \left(\int q_{\tilde{\boldsymbol{\mathcal{T}}}_{T+1}|\mathbf{S}_0}(\tilde{\tau}_{t+1}\,|\,\mathbf{s}_0) \left(\int_{\mathcal{S}^t \times \mathcal{A}^{t+1}} q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:t}|\mathbf{S}_0}(\tilde{\tau}_{0:t}\,|\,\mathbf{s}_0) q_T(t) f(\mathbf{s}_t, \mathbf{a}_t) d\tilde{\tau}_{0:t} \right) d\tilde{\tau}_{t+1} \right), \tag{C.136}$$

$$= \sum_{t=0}^{\infty} \left(\mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t}\,|\,t)} \Big[q_T(t) f(\mathbf{s}_t, \mathbf{a}_t) \Big] \cdot \underbrace{\left(\int q_{\tilde{\boldsymbol{\mathcal{T}}}_{T+1}|\mathbf{S}_0}(\tilde{\tau}_{t+1}\,|\,\mathbf{s}_0) d\tilde{\tau}_{t+1} \right)}_{=1} \right) \tag{C.137}$$

$$= \sum_{t=0}^{\infty} \left(\left(\int_{\mathcal{S}^t \times \mathcal{A}^{t+1}} q(\tilde{\tau}_{0:t}\,|\,\mathbf{s}_0) q_T(t) f(\mathbf{s}_t, \mathbf{a}_t) d\tilde{\tau}_{0:t} \right) \cdot \underbrace{\left(\int q_{\tilde{\boldsymbol{\mathcal{T}}}_{T+1}|\mathbf{S}_0}(\tilde{\tau}_{t+1}\,|\,\mathbf{s}_0) d\tilde{\tau}_{t+1} \right)}_{=1} \right) \tag{C.138}$$

$$= \sum_{t=0}^{\infty} \int q_{\tilde{\boldsymbol{\mathcal{T}}}_0|\mathbf{S}_0}(\tilde{\tau}_0\,|\,\mathbf{s}_0) q_T(t) f(\mathbf{s}_t, \mathbf{a}_t) d\tilde{\tau}_0 \tag{C.139}$$

$$= \int q_{\tilde{\boldsymbol{\mathcal{T}}}_0|\mathbf{S}_0}(\tilde{\tau}_0\,|\,\mathbf{s}_0) \sum_{t=0}^{\infty} q_T(t) f(\mathbf{s}_t, \mathbf{a}_t) d\tilde{\tau}_0, \tag{C.140}$$

we can express Equation (C.133) in terms of the infinite-horizon state–action trajectory $q_{\tilde{\boldsymbol{\mathcal{T}}}_0|\mathbf{S}_0}(\tilde{\tau}_0\,|\,\mathbf{s}_0) \stackrel{\text{def}}{=} \prod_{t=0}^{\infty} p_d(\mathbf{s}_{t+1}\,|\,\mathbf{s}_t, \mathbf{a}_t) \pi(\mathbf{a}_t\,|\,\mathbf{s}_t)$ as

$$\begin{aligned} &\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) \\ &= \int q_{\tilde{\boldsymbol{\mathcal{T}}}_0|\mathbf{S}_0}(\tilde{\tau}_0\,|\,\mathbf{s}_0) \sum_{t=0}^{\infty} q_T(t) \log p(\mathbf{g}\,|\,\mathbf{s}_t, \mathbf{a}_t) d\tilde{\tau} \\ &\qquad - \sum_{t=0}^{\infty} q_T(t) \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\cdot\,|\,t) \parallel p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\cdot\,|\,t)) - \mathbb{D}_{\mathrm{KL}}(q_T \parallel p_T) \end{aligned} \tag{C.141}$$

$$\begin{aligned} &= \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_0|\mathbf{S}_0}(\tilde{\tau}_0\,|\,\mathbf{s}_0)} \Bigg[\sum_{t=0}^{\infty} q_T(t) \Big(\log p(\mathbf{g}\,|\,\mathbf{s}_t, \mathbf{a}_t) \\ &\qquad - \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\cdot\,|\,t) \parallel p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\cdot\,|\,t)) \Big) \Bigg] - \mathbb{D}_{\mathrm{KL}}(q_T \parallel p_T). \end{aligned} \tag{C.142}$$

Using Lemma C.5 and the definition of $q_T(t)$ in Equation (5.30), we can rewrite this

objective as

$$\begin{aligned}
&\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) \\
&= \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_0|\mathbf{S}_0}(\tilde{\tau}_0 \,|\, \mathbf{s}_0)} \Bigg[\sum_{t=0}^{\infty} \Big(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \Big) q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \Big(\log p(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) \\
&\qquad - \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\cdot \,|\, t) \parallel p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\cdot \,|\, t)) \Big) \Bigg] - \sum_{t=0}^{\infty} \Big(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \Big) \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \,||\, p_{\Delta_{t+1}})
\end{aligned} \tag{C.143}$$

$$\begin{aligned}
&= \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_0|\mathbf{S}_0}(\tilde{\tau}_0 \,|\, \mathbf{s}_0)} \Bigg[\sum_{t=0}^{\infty} \Bigg(\prod_{t'=1}^{t} q(\Delta_{t'} = 0) \Bigg) \\
&\qquad \cdot \Big(q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \big(\log p(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) - \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\cdot \,|\, t) \parallel p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\cdot \,|\, t))) \\
&\qquad\qquad - \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \parallel p_{\Delta_{t+1}}) \Big) \Bigg],
\end{aligned} \tag{C.144}$$

with

$$\begin{aligned}
&\mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \parallel p_{\Delta_{t+1}}) \\
&= q_{\Delta_{t+1}}(\Delta_{t+1} = 0) \log \frac{q_{\Delta_{t+1}}(\Delta_{t+1} = 0)}{p_{\Delta_{t+1}}(\Delta_{t+1} = 0)} + (1 - q_{\Delta_{t+1}}(\Delta_{t+1} = 0)) \log \frac{1 - q_{\Delta_{t+1}}(\Delta_{t+1} = 0)}{1 - p_{\Delta_{t+1}}(\Delta_{t+1} = 0)}.
\end{aligned} \tag{C.145}$$

Next, to re-express $\mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\cdot \,|\, t) \parallel p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\cdot \,|\, t))$ as a sum over Kullback-Leibler divergences between distributions over single action random variables, we note that

$$\begin{aligned}
&\mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\cdot \,|\, t) \parallel p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\cdot \,|\, t)) \\
&= \int_{\mathcal{S}^t \times \mathcal{A}^{t+1}} q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t} \,|\, t) \log \frac{q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t} \,|\, t)}{p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t} \,|\, t)} d\tilde{\tau}_{0:t} && \text{(C.146)} \\
&= \int_{\mathcal{S}^t \times \mathcal{A}^{t+1}} q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t} \,|\, t) \log \frac{\prod_{t'=0}^{t} \pi(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'})}{\prod_{t'=0}^{t} p(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'})} d\tilde{\tau}_{0:t} && \text{(C.147)} \\
&= \int_{\mathcal{S}^t \times \mathcal{A}^{t+1}} q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}|T}(\tilde{\tau}_{0:t} \,|\, t) \sum_{t'=0}^{t} \log \frac{\pi(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'})}{p(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'})} d\tilde{\tau}_{0:t} && \text{(C.148)} \\
&= \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_0|\mathbf{S}_0}(\tilde{\tau}_0 \,|\, \mathbf{s}_0)} \left[\sum_{t'=0}^{t} \int_{\mathcal{A}} \pi(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'}) \log \frac{\pi(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'})}{p(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'})} d\mathbf{a}_{t'} \right] && \text{(C.149)} \\
&= \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_0|\mathbf{S}_0}(\tilde{\tau}_0 \,|\, \mathbf{s}_0)} \left[\sum_{t'=0}^{t} \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_{t'}) \parallel p(\cdot \,|\, \mathbf{s}_{t'})) \right], && \text{(C.150)}
\end{aligned}$$

where we have used the same marginalization trick as above to express the expression in terms of an infinite-horizon trajectory distribution, which allows us to

express Equation (C.144) as

$$
\begin{aligned}
&\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) \\
&= \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_0|\mathbf{s}_0}(\tilde{\tau}_0 \,|\, \mathbf{s}_0)} \Bigg[\sum_{t=0}^{\infty} \left(\prod_{t'=1}^{t} q(\Delta_{t'} = 0) \right) \\
&\qquad \cdot \Bigg(q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \left(\log p(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) - \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_0|\mathbf{s}_0}(\tilde{\tau}_0 \,|\, \mathbf{s}_0)} \left[\sum_{t'=0}^{t} \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_{t'}) \parallel p(\cdot \,|\, \mathbf{s}_{t'})) \right] \right) \\
&\qquad\qquad - \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \parallel p_{\Delta_{t+1}}) \Bigg) \Bigg].
\end{aligned}
\tag{C.151}
$$

Rearranging and dropping redundant expectation operators, we can now express the objective as

$$
\begin{aligned}
&\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) \\
&= \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_0|\mathbf{s}_0}(\tilde{\tau}_0 \,|\, \mathbf{s}_0)} \Bigg[\sum_{t=0}^{\infty} \left(\prod_{t'=1}^{t} q_{\Delta_{t+1}}(\Delta_{t'} = 0) \right) \\
&\qquad \cdot \Bigg(q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \left(\log p(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) - \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_0|\mathbf{s}_0}(\tilde{\tau}_0 \,|\, \mathbf{s}_0)} \left[\sum_{t'=0}^{t} \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_{t'}) \parallel p(\cdot \,|\, \mathbf{s}_{t'})) \right] \right) \\
&\qquad\qquad - \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \parallel p_{\Delta_{t+1}}) \Bigg) \Bigg].
\end{aligned}
\tag{C.152}
$$

$$
\begin{aligned}
&= \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_0|\mathbf{s}_0}(\tilde{\tau}_0 \,|\, \mathbf{s}_0)} \Bigg[\sum_{t=0}^{\infty} \left(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \right) \\
&\qquad \cdot \Big(q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \log p(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) - \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \parallel p_{\Delta_{t+1}}) \Big) \Bigg] \\
&\qquad\qquad - \sum_{t=0}^{\infty} \underbrace{\left(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \right)}_{= q_T(t)} \\
&\qquad\qquad\qquad \cdot \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_0|\mathbf{s}_0}(\tilde{\tau}_0 \,|\, \mathbf{s}_0)} \left[\sum_{t'=0}^{t} \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_{t'}) \parallel p(\cdot \,|\, \mathbf{s}_{t'})) \right],
\end{aligned}
\tag{C.153}
$$

whereupon we note that the last term can be expressed as

$$\sum_{t=0}^{\infty} q_T(t)\, \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_0|\mathbf{s}_0}(\tilde{\tau}_0\,|\,\mathbf{s}_0)} \left[\sum_{t'=0}^{t} \mathbb{D}_{\mathrm{KL}}(\pi(\cdot\,|\,\mathbf{s}_{t'}) \parallel p(\cdot\,|\,\mathbf{s}_{t'})) \right] \tag{C.154}$$

$$= \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_0|\mathbf{s}_0}(\tilde{\tau}_0\,|\,\mathbf{s}_0)} \left[\sum_{t=0}^{\infty} \sum_{t'=0}^{t} q_T(t) \mathbb{D}_{\mathrm{KL}}(\pi(\cdot\,|\,\mathbf{s}_{t'}) \parallel p(\cdot\,|\,\mathbf{s}_{t'})) \right]$$

$$= \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_0|\mathbf{s}_0}(\tilde{\tau}_0\,|\,\mathbf{s}_0)} \left[\sum_{t=0}^{\infty} q(T \geq t) \mathbb{D}_{\mathrm{KL}}(\pi(\cdot\,|\,\mathbf{s}_t) \parallel p(\cdot\,|\,\mathbf{s}_t)) \right] \tag{C.155}$$

$$= \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_0|\mathbf{s}_0}(\tilde{\tau}_0\,|\,\mathbf{s}_0)} \left[\sum_{t=0}^{\infty} \underbrace{\left(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \right)}_{\text{(by Lemma C.2)}} \mathbb{D}_{\mathrm{KL}}(\pi(\cdot\,|\,\mathbf{s}_t) \parallel p(\cdot\,|\,\mathbf{s}_t)) \right], \tag{C.156}$$

where the second line follows from expanding the sums and regrouping terms. By substituting the expression in Equation (C.156) into Equation (C.153), we obtain an objective expressed entirely in terms of distributions over single-index random variables:

$$\begin{aligned} &\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) \\ &= \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_0|\mathbf{s}_0}(\tilde{\tau}_0\,|\,\mathbf{s}_0)} \left[\sum_{t=0}^{\infty} \left(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \right) \right. \\ &\qquad \cdot \Big(q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \log p_d(\mathbf{g}\,|\,\mathbf{s}_t, \mathbf{a}_t) - \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \parallel p_{\Delta_{t+1}}) \\ &\qquad \left. - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot\,|\,\mathbf{s}_t) \parallel p(\cdot\,|\,\mathbf{s}_t)) \Big) \right] \end{aligned} \tag{C.157}$$

$$= \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_0|\mathbf{s}_0}(\tilde{\tau}_0\,|\,\mathbf{s}_0)} \left[\sum_{t=0}^{\infty} \left(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \right) \Big(r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot\,|\,\mathbf{s}_t) \parallel p(\cdot\,|\,\mathbf{s}_t)) \Big) \right], \tag{C.158}$$

where we defined

$$r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) \stackrel{\text{def}}{=} q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \log p_d(\mathbf{g}\,|\,\mathbf{s}_t, \mathbf{a}_t) - \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \parallel p_{\Delta_{t+1}}), \tag{C.159}$$

which concludes the proof. □

Theorem 5.3. *Let $q_T(t)$ and $q_{\tilde{\boldsymbol{\tau}}_{0:t}|T}(\tilde{\tau}_{0:t}\,|\,t, \mathbf{s}_0)$ be as defined in Equation (5.24) and Equation (5.30), and define*

$$V^\pi(\mathbf{s}_t, \mathbf{g}; q_T) \stackrel{\text{def}}{=} \mathbb{E}_{\pi(\mathbf{a}_t\,|\,\mathbf{s}_t)}[Q^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T)] - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot\,|\,\mathbf{s}_t) \,\|\, p(\cdot\,|\,\mathbf{s}_t)), \tag{C.160}$$

$$Q^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T) \stackrel{\text{def}}{=} r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) + q(\Delta_{t+1} = 0)\, \mathbb{E}_{p_d(\mathbf{s}_{t+1}\,|\,\mathbf{s}_t, \mathbf{a}_t)}[V^\pi(\mathbf{s}_{t+1}, \mathbf{g}; q_T)], \tag{C.161}$$

$$r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) \stackrel{\text{def}}{=} q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \log p_d(\mathbf{g}\,|\,\mathbf{s}_t, \mathbf{a}_t) - \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \,\|\, p_{\Delta_{t+1}}). \tag{C.162}$$

Then given any initial state $\mathbf{s}_0$ *and outcome* $\mathbf{g}$*,*

$$\mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T|\mathbf{S}_0}(\cdot\,|\,\mathbf{s}_0)\,\|\,p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T|\mathbf{S}_0,\mathbf{S}_{T^\star}}(\cdot\,|\,\mathbf{s}_0,\mathbf{g})) = -\mathcal{F}(\pi,q_T,\mathbf{s}_0,\mathbf{g}) + C = -V^\pi(\mathbf{s}_0,\mathbf{g};q_T) + C,$$

where $C \stackrel{\text{def}}{=} \log p(\mathbf{g}\,|\,\mathbf{s}_0)$ *is independent of* π *and* q_T*, and hence maximizing* $V^\pi(\mathbf{s}_0,\mathbf{g};\pi,q_T)$ *is equivalent to minimizing Equation (5.28). In other words,*

$$\begin{aligned}
&\arg\min_{\pi\in\Pi,q_T\in\mathcal{Q}_T} \{\mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T|\mathbf{S}_0}(\cdot\,|\,\mathbf{s}_0)\,\|\,p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T|\mathbf{S}_0,\mathbf{S}_{T^\star}}(\cdot\,|\,\mathbf{s}_0,\mathbf{g}))\}\\
=&\arg\max_{\pi\in\Pi,q_T\in\mathcal{Q}_T} \mathcal{F}(\pi,q_T,\mathbf{s}_0,\mathbf{g})\\
=&\arg\max_{\pi\in\Pi,q_T\in\mathcal{Q}_T} V^\pi(\mathbf{s}_0,\mathbf{g};q_T).
\end{aligned}$$

Proof. Consider the objective derived in Proposition 5.8,

$$\begin{aligned}
&\mathcal{F}(\pi,q_T,\mathbf{s}_0,\mathbf{g})\\
&= \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_0|\mathbf{S}_0}(\tilde{\tau}_0\,|\,\mathbf{s}_0)}\Bigg[\sum_{t=0}^{\infty}\Bigg(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'}=0)\Bigg)\\
&\quad\cdot\underbrace{\left(q_{\Delta_{t+1}}(\Delta_{t+1}=1)\log p_d(\mathbf{g}\,|\,\mathbf{s}_t,\mathbf{a}_t) - \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}}\,\|\,p_{\Delta_{t+1}})\right)}_{\stackrel{\text{def}}{=} r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_\Delta)}\\
&\qquad - \mathbb{D}_{\mathrm{KL}}(\pi(\mathbf{a}_t\,|\,\mathbf{s}_t)\,\|\,p(\mathbf{a}_t|\mathbf{s}_t))\Bigg],
\end{aligned} \tag{C.163}$$

and recall that, by Proposition 5.7,

$$\mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T|\mathbf{S}_0}(\cdot\,|\,\mathbf{s}_0)\,\|\,p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T|\mathbf{S}_0,\mathbf{S}_{T^\star}}(\cdot\,|\,\mathbf{s}_0,\mathbf{g})) = -\mathcal{F}(\pi,q_T,\mathbf{s}_0,\mathbf{g}) + \log p(\mathbf{g}|\mathbf{s}_0). \tag{C.164}$$

Therefore, to prove the result that

$$\mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T|\mathbf{S}_0}(\cdot\,|\,\mathbf{s}_0)\,\|\,p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T},T|\mathbf{S}_0,\mathbf{S}_{T^\star}}(\cdot\,|\,\mathbf{s}_0,\mathbf{g})) = -V^\pi(\mathbf{s}_0,\mathbf{g};q_T) + \log p(\mathbf{g}|\mathbf{s}_0),$$

we just need to show that $\mathcal{F}(\pi,q_T,\mathbf{s}_0,\mathbf{g}) = V^\pi(\mathbf{s}_0,\mathbf{g};q_T)$ for $V^\pi(\mathbf{s}_0,\mathbf{g};q_T)$ as defined in the theorem. To do so, we start from the objective $\mathcal{F}(\pi,q_T,\mathbf{s}_0,\mathbf{g})$ and unroll it for $t=0$:

$$\begin{aligned}
&\mathcal{F}(\pi,q_T,\mathbf{s}_0,\mathbf{g})\\
&= \mathbb{E}_{q_{\tilde{\boldsymbol{\mathcal{T}}}_0|\mathbf{S}_0}(\tilde{\tau}_0\,|\,\mathbf{s}_0)}\left[\sum_{t=0}^{\infty}\left(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'}=0)\right) r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_\Delta) - \mathbb{D}_{\mathrm{KL}}(\pi(\mathbf{a}_t\,|\,\mathbf{s}_t)\,\|\,p(\mathbf{a}_t|\mathbf{s}_t))\right]
\end{aligned} \tag{C.165}$$

$$\begin{aligned}
&= \mathbb{E}_{\pi(\mathbf{a}_0\,|\,\mathbf{s}_0)}\Bigg[r(\mathbf{s}_0,\mathbf{a}_0,\mathbf{g};q_\Delta) + \mathbb{E}_{q(\tau_1\,|\,\mathbf{s}_0,\mathbf{a}_0)}\Bigg[\sum_{t=1}^{\infty}\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'}=0)\Big(r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_\Delta)\\
&\qquad - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot\,|\,\mathbf{s}_t)\,\|\,p(\cdot\,|\,\mathbf{s}_t))\Big)\Bigg]\Bigg] - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot\,|\,\mathbf{s}_0)\,\|\,p(\cdot\,|\,\mathbf{s}_0)).
\end{aligned} \tag{C.166}$$

With this expression at hand, we now define

$$
\begin{aligned}
&Q^{\pi}_{\text{sum}}(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_T) \\
&\overset{\text{def}}{=} r(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_\Delta) \\
&\qquad + \mathbb{E}_{q(\tau_1 | \mathbf{s}_0, \mathbf{a}_0)} \left[\sum_{t=1}^{\infty} \prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \Big(r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) - \mathbb{D}_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)) \Big) \right],
\end{aligned}
\tag{C.167}
$$

and note that

$$
\begin{aligned}
\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) =& \mathbb{E}_{\pi(\mathbf{a}_0 \,|\, \mathbf{s}_0)}[Q^{\pi}_{\text{sum}}(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_T)] - \mathbb{D}_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_0) \,\|\, p(\cdot \,|\, \mathbf{s}_0)) \\
=& V^{\pi}(\mathbf{s}_0, \mathbf{g}; q_T)
\end{aligned}
\tag{C.168}
$$

as per the definition of $V^{\pi}(\mathbf{s}_0, \mathbf{g}; q_T)$. To prove the theorem from this intermediate result, we now have to show that $Q^{\pi}_{\text{sum}}(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_T)$ as defined in Equation (C.167) can in fact be expressed recursively as $Q^{\pi}_{\text{sum}}(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_T) = Q^{\pi}(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_T)$ with

$$
Q^{\pi}(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_T) = r(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_\Delta) + q(\Delta_1 = 0)\, \mathbb{E}_{p_d(\mathbf{s}_1 \,|\, \mathbf{s}_0, \mathbf{a}_0)}[V^{\pi}(\mathbf{s}_1, \mathbf{g}; q_T)]\,. \tag{C.169}
$$

To see that this is the case, first, unroll $Q^{\pi}(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_T)$ for $t = 1$,

$$
\begin{aligned}
&Q^{\pi}_{\text{sum}}(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_T) \\
&= r(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_\Delta) + \mathbb{E}_{q(\tau_1 | \mathbf{s}_0, \mathbf{a}_0)} \left[\sum_{t=1}^{\infty} \prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \Big(r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) \right. \\
&\qquad \left. - \mathbb{D}_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)) \Big) \right]
\end{aligned}
\tag{C.170}
$$

$$
\begin{aligned}
&= r(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_\Delta) + \mathbb{E}_{p_d(\mathbf{s}_1 | \mathbf{s}_0, \mathbf{a}_0)} \left[\mathbb{E}_{q(\tau_1 | \mathbf{s}_0, \mathbf{a}_0)} \left[\sum_{t=1}^{\infty} \prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \Big(r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) \right. \right. \\
&\qquad \left. \left. - \mathbb{D}_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)) \Big) \right] \right]
\end{aligned}
\tag{C.171}
$$

$$
\begin{aligned}
&= r(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_\Delta) + \mathbb{E}_{p_d(\mathbf{s}_1 | \mathbf{s}_0, \mathbf{a}_0)} \left[\mathbb{E}_{\pi(\mathbf{a}_1 \,|\, \mathbf{s}_1)} \left[q_{\Delta_1}(\Delta_1 = 0) \Big(r(\mathbf{s}_1, \mathbf{a}_1, \mathbf{g}; q_\Delta) \right. \right. \\
&\qquad \left. - \mathbb{D}_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_1) \parallel p(\cdot \,|\, \mathbf{s}_1)) \Big) \right. \\
&\qquad\quad \left. \left. + \mathbb{E}_{q(\tau_2 | \mathbf{s}_1, \mathbf{a}_1)} \left[\sum_{t=2}^{\infty} \prod_{t'=2}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \Big(r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) - \mathbb{D}_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)) \Big) \right] \right] \right],
\end{aligned}
\tag{C.172}
$$

and note that we can rearrange this expression to obtain the recursive relationship

$$
\begin{aligned}
&Q^{\pi}_{\text{sum}}(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_T) \\
&\quad = r(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_\Delta) + q_{\Delta_1}(\Delta_1 = 0)\, \mathbb{E}_{p_d(\mathbf{s}_{0+1} \,|\, \mathbf{s}_0, \mathbf{a}_0)} \Bigg[- \mathbb{D}_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_1) \parallel p(\cdot \,|\, \mathbf{s}_1)) \\
&\qquad + \mathbb{E}_{\pi(\mathbf{a}_1 \,|\, \mathbf{s}_1)} \Bigg[r(\mathbf{s}_1, \mathbf{a}_1, \mathbf{g}; q_\Delta) + \mathbb{E} \Bigg[\sum_{t=2}^{\infty} \Bigg(\prod_{t'=2}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \Bigg) \Big(r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) \\
&\qquad\quad - \mathbb{D}_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)) \Big) \Bigg] \Bigg] \Bigg],
\end{aligned}
\tag{C.173}
$$

where the innermost expectation is taken with respect to $q(\tau_2|\mathbf{s}_1, \mathbf{a}_1)$. With this result and noting that

$$
\begin{aligned}
&Q^{\pi}_{\text{sum}}(\mathbf{s}_1, \mathbf{a}_1, \mathbf{g}; q_T) \\
&\quad = r(\mathbf{s}_1, \mathbf{a}_1, \mathbf{g}; q_\Delta) \\
&\qquad + \mathbb{E} \Bigg[\sum_{t=2}^{\infty} \Bigg(\prod_{t'=2}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \Bigg) \Big(r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) - \mathbb{D}_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s})) \Big) \Bigg],
\end{aligned}
\tag{C.174}
$$

where the expectation is again taken with respect to $q(\tau_2|\mathbf{s}_1, \mathbf{a}_1)$, we see that

$$
\begin{aligned}
&Q^{\pi}_{\text{sum}}(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_T) \\
&\quad = r(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_\Delta) + q_{\Delta_1}(\Delta_1 = 0)\, \mathbb{E}_{p_d(\mathbf{s}_{0+1} \,|\, \mathbf{s}_0, \mathbf{a}_0)} \Big[\mathbb{E}_{\pi(\mathbf{a}_1|\mathbf{s}_1)} \left[Q^{\pi}_{\text{sum}}(\mathbf{s}_1, \mathbf{a}_1, \mathbf{g}; q_T) \right] \\
&\qquad - \mathbb{D}_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_1) \parallel p(\cdot \,|\, \mathbf{s}_1)) \Big]
\end{aligned}
\tag{C.175}
$$

$$
= r(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_\Delta) + q_{\Delta_1}(\Delta_1 = 0)\, \mathbb{E}_{p_d(\mathbf{s}_1|\mathbf{s}_0, \mathbf{a}_0)} \Big[V^{\pi}(\mathbf{s}_1, \mathbf{g}; q_T) \Big], \tag{C.176}
$$

for $V^{\pi}(\mathbf{s}_{t+1}, \mathbf{g}; q_T)$ as defined above, as desired. In other words, we have that

$$
\begin{aligned}
&\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) \\
&\quad = \mathbb{E}_{\pi(\mathbf{a}_0 \,|\, \mathbf{s}_0)}[Q^{\pi}_{\text{sum}}(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_T)] - \mathbb{D}_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_0) \,\|\, p(\cdot \,|\, \mathbf{s}_0)) = V^{\pi}(\mathbf{s}_0, \mathbf{g}; q_T).
\end{aligned}
\tag{C.177}
$$

Combining this result with Proposition 5.7 and Proposition 5.8, we finally conclude that

$$
\begin{aligned}
&\mathbb{D}_{\text{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}, T|\mathbf{S}_0}(\cdot \,|\, \mathbf{s}_0) \,\|\, p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}, T|\mathbf{S}_0, \mathbf{S}_{T^\star}}(\cdot \,|\, \mathbf{s}_0, \mathbf{g})) \\
&= - \mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) + C \\
&= - V^{\pi}(\mathbf{s}_0, \mathbf{g}; q_T) + C,
\end{aligned}
\tag{C.178}
$$

where $C \stackrel{\text{def}}{=} \log p(\mathbf{g} \mid \mathbf{s}_0)$ is independent of π and q_T. Hence, maximizing $V^\pi(\mathbf{s}_0, \mathbf{g}; \pi, q_T)$ is equivalent to minimizing the objective in Equation (5.28). In other words,

$$\begin{aligned} &\arg\min_{\pi \in \Pi, q_T \in \mathcal{Q}_T} \mathbb{D}_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}, T \mid \mathbf{S}_0}(\cdot \mid \mathbf{s}_0) \,\|\, p_{\tilde{\boldsymbol{\mathcal{T}}}_{0:T}, T \mid \mathbf{S}_0, \mathbf{S}_{T^\star}}(\cdot \mid \mathbf{s}_0, \mathbf{g})) \\ &= \arg\max_{\pi \in \Pi, q_T \in \mathcal{Q}_T} \mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) \\ &= \arg\max_{\pi \in \Pi, q_T \in \mathcal{Q}_T} V^\pi(\mathbf{s}_0, \mathbf{g}; q_T). \end{aligned} \tag{C.179}$$

This concludes the proof. □

Corollary 5.3. *Let $q_T = p_T$, assume that p_T is a geometric distribution with parameter $\gamma \in (0, 1)$. Then the inference problem in Equation (5.28) of finding an outcome-driven variational trajectory distribution simplifies to maximizing the following recursively defined variational objective with respect to π:*

$$\bar{V}^\pi(\mathbf{s}_0, \mathbf{g}; \gamma) \stackrel{\text{def}}{=} \mathbb{E}_{\pi(\mathbf{a}_0 \mid \mathbf{s}_0)}\left[Q^\pi(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; \gamma)\right] - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \mid \mathbf{s}_0) \,\|\, p(\cdot \mid \mathbf{s}_0))), \tag{C.180}$$

where

$$\bar{Q}^\pi(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; \gamma) \stackrel{\text{def}}{=} (1-\gamma) \log p_d(\mathbf{g} \mid \mathbf{s}_0, \mathbf{a}_0) + \gamma\, \mathbb{E}_{p_d(\mathbf{s}_1 \mid \mathbf{s}_0, \mathbf{a}_0)}\left[V^\pi(\mathbf{s}_1, \mathbf{g}; \gamma)\right]. \tag{C.181}$$

Proof. The result follows immediately when replacing q_Δ in Theorem 5.3 by p_Δ and noting that $\mathbb{D}_{\mathrm{KL}}(p_\Delta \,\|\, p_\Delta) = 0$. □

C.1.7 Optimal Outcome-Driven Variational Posterior over T

Proposition 5.3. *The optimal variational distribution $q_T^\star$ with respect to Equation (5.31) is defined recursively in terms of $q^\star_{\Delta_{t+1}}(\Delta_{t+1} = 0) \forall t \in \mathbb{N}_0$ by*

$$q^\star_{\Delta_{t+1}}(\Delta_{t+1} = 0; \pi, Q^\pi) = \sigma\left(\Lambda(\mathbf{s}_t, \pi, q_T, Q^\pi) + \sigma^{-1}\left(p_{\Delta_{t+1}}(\Delta_{t+1} = 0)\right)\right), \tag{C.182}$$

where

$$\Lambda(\mathbf{s}_t, \pi, q_T, Q^\pi) \stackrel{\text{def}}{=} \mathbb{E}_{\pi(\mathbf{a}_{t+1} \mid \mathbf{s}_{t+1}) p_d(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t) \pi(\mathbf{a}_t \mid \mathbf{s}_t)}[Q^\pi(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \mathbf{g}; q_T) - \log p_d(\mathbf{g} \mid \mathbf{s}_t, \mathbf{a}_t)]$$

and $\sigma(\cdot)$ is the sigmoid function, that is, $\sigma(x) = \frac{1}{e^{-x}+1}$ and $\sigma^{-1}(x) = \log \frac{x}{1-x}$.

Proof. Consider $\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g})$:

$$\begin{aligned} \mathcal{F}(\pi, q_T, \mathbf{s}_t, \mathbf{g}) &= \mathbb{E}_{\pi(\mathbf{a}_t \mid \mathbf{s}_t)}[Q^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T)] \\ &= \mathbb{E}_{\pi(\mathbf{a}_t \mid \mathbf{s}_t)}[r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) + q_{\Delta_{t+1}}(\Delta_{t+1} = 0)\, \mathbb{E}\left[V(\mathbf{s}_{t+1}, \mathbf{g}; q_T)\right]]. \end{aligned} \tag{C.183}$$

Since the variational objective $\mathcal{F}(\pi, q_T, \mathbf{s}_t, \mathbf{g})$ can be expressed recursively as

$$V^\pi(\mathbf{s}_t, \mathbf{g}; q_T) \stackrel{\text{def}}{=} \mathbb{E}_{\pi(\mathbf{a}_t \mid \mathbf{s}_t)}\left[Q(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T)\right] - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \mid \mathbf{s}_t) \,\|\, p(\cdot \mid \mathbf{s}_t)),$$

with

$$Q^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T) = r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) + q_{\Delta_{t+1}}(\Delta_{t+1} = 0)\, \mathbb{E}_{p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)}[V^\pi(\mathbf{s}_{t+1}, \mathbf{g}; q_T)]\,,$$
$$r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) = q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) - \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \parallel p_{\Delta_{t+1}}),$$

and since $\mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \parallel p_{\Delta_{t+1}})$ is strictly convex in $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$, we can find the globally optimal Bernoulli distribution parameters $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ for all $t \in \mathbb{N}_0$ recursively. That is, it is sufficient to solve the problem

$$\begin{aligned} & q^\star_{\Delta_{t+1}}(\Delta_{t+1} = 0) \\ \overset{\text{def}}{=} & \operatorname*{arg\,max}_{q_{\Delta_{t+1}}(\Delta_{t+1}=0)} \{\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g})\} \\ = & \operatorname*{arg\,max}_{q_{\Delta_{t+1}}(\Delta_{t+1}=0)} \{\mathcal{F}(\pi, \{q_{\Delta_1}, \dots, q_{\Delta_{t+1}}, \dots\}, \mathbf{s}_0, \mathbf{g})\}\,, \end{aligned} \tag{C.184}$$

where we have written q_T out in terms of its Bernoulli factors, for a fixed $t+1$. To do so, we take the derivative of $\mathcal{F}(\pi, \{q_{\Delta_1}, \dots, q_{\Delta_{t+1}}, \dots\}, \mathbf{s}_0, \mathbf{g})$, which—defined recursively—is given by

$$\begin{aligned} & \mathbb{E}_{\pi(\mathbf{a}_t \,|\, \mathbf{s}_t)}[Q(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T)] - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)) \\ = & \mathbb{E}_{\pi(\mathbf{a}_t \,|\, \mathbf{s}_t)}\Big[r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) + q_{\Delta_{t+1}}(\Delta_{t+1} = 0)\, \mathbb{E}_{p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)}[V^\pi(\mathbf{s}_{t+1}, \mathbf{g}; q_T)]\Big] \\ & \quad - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)) \end{aligned} \tag{C.185}$$

$$\begin{aligned} = & \mathbb{E}_{\pi(\mathbf{a}_t \,|\, \mathbf{s}_t)}\Big[q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) - \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \parallel p_{\Delta_{t+1}}) \\ & \quad + q_{\Delta_{t+1}}(\Delta_{t+1} = 0)\, \mathbb{E}_{p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)}[V^\pi(\mathbf{s}_{t+1}, \mathbf{g}; q_T)]\Big] - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)) \end{aligned} \tag{C.186}$$

$$\begin{aligned} = & \mathbb{E}_{\pi(\mathbf{a}_t \,|\, \mathbf{s}_t)}\Big[(1 - q_{\Delta_{t+1}}(\Delta_{t+1} = 0)) \log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) - \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \parallel p_{\Delta_{t+1}}) \\ & \quad + q_{\Delta_{t+1}}(\Delta_{t+1} = 0)\, \mathbb{E}_{p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)}[V^\pi(\mathbf{s}_{t+1}, \mathbf{g}; q_T)]\Big] - \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)), \end{aligned} \tag{C.187}$$

with respect to $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ and set it to zero, which yields

$$\begin{aligned} 0 = & - \mathbb{E}_{\pi(\mathbf{a}_t \,|\, \mathbf{s}_t)}\Big[\log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) + \mathbb{E}_{\pi(\mathbf{a}_{t+1} \,|\, \mathbf{s}_{t+1}) p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)}[Q^\pi(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \mathbf{g}; q_T)]\Big] \\ & + \log \frac{1 - q^\star_{\Delta_{t+1}}(\Delta_{t+1} = 0)}{1 - p_{\Delta_{t+1}}(\Delta_{t+1} = 0)} - \log \frac{q^\star_{\Delta_{t+1}}(\Delta_{t+1} = 0)}{p_{\Delta_{t+1}}(\Delta_{t+1} = 0)}. \end{aligned} \tag{C.188}$$

Rearranging, we get

$$\begin{aligned}
&\frac{q^\star_{\Delta_{t+1}}(\Delta_{t+1}=0)}{1-q^\star_{\Delta_{t+1}}(\Delta_{t+1}=0)} \\
&= \exp\Bigg(\mathbb{E}[Q^\pi(\mathbf{s}_{t+1},\mathbf{a}_{t+1},\mathbf{g};q_T) - \log p_d(\mathbf{g}\,|\,\mathbf{s}_t,\mathbf{a}_t)] \\
&\qquad + \log\frac{p_{\Delta_{t+1}}(\Delta_{t+1}=0)}{1-p_{\Delta_{t+1}}(\Delta_{t+1}=0)}\Bigg),
\end{aligned} \tag{C.189}$$

where the expectation is taken with respect to $\pi(\mathbf{a}_{t+1}\,|\,\mathbf{s}_{t+1})p_d(\mathbf{s}_{t+1}\,|\,\mathbf{s}_t,\mathbf{a}_t)\pi(\mathbf{a}_t\,|\,\mathbf{s}_t)$ and the Q-function depends on $q(\Delta_{t'})$ with $t' > t$, but not on $q^\star_{\Delta_{t+1}}(\Delta_{t+1}=0)$. Solving for $q^\star_{\Delta_{t+1}}(\Delta_{t+1}=0)$, we obtain

$$q^\star_{\Delta_{t+1}}(\Delta_{t+1}=0) \tag{C.190}$$

$$= \sigma\Big(\mathbb{E}_{p_{\pi p_d}}[Q^\pi(\mathbf{s}_{t+1},\mathbf{a}_{t+1},\mathbf{g};q_T)] - \mathbb{E}_{\pi(\mathbf{a}_t\,|\,\mathbf{s}_t)}[\log p_d(\mathbf{g}\,|\,\mathbf{s}_t,\mathbf{a}_t)] \tag{C.191}$$

$$+ \sigma^{-1}\left(p_{\Delta_{t+1}}(\Delta_{t+1}=0)\right)\Big), \tag{C.192}$$

where $p_{\pi p_d} \overset{\text{def}}{=} \pi(\mathbf{a}_{t+1}\,|\,\mathbf{s}_{t+1})p_d(\mathbf{s}_{t+1}\,|\,\mathbf{s}_t,\mathbf{a}_t)$, $\sigma(\cdot)$ is the sigmoid function with $\sigma(x) = \frac{1}{e^{-x}+1}$ and $\sigma^{-1}(x) = \log\frac{x}{1-x}$. This concludes the proof. □

Remark 3. *As can be seen from Proposition 5.3, the optimal approximation to the posterior over T trades off short-term rewards via $\mathbb{E}_{\pi(\mathbf{a}_t\,|\,\mathbf{s}_t)}[r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_\Delta)]$, long-term rewards via $\mathbb{E}_{\pi(\mathbf{a}_{t+1}\,|\,\mathbf{s}_{t+1})p_d(\mathbf{s}_{t+1}\,|\,\mathbf{s}_t,\mathbf{a}_t)}[Q^\pi(\mathbf{s}_{t+1},\mathbf{a}_{t+1},\mathbf{g};q_T)]$, and the prior log-odds of not achieving the outcome at a given point in time conditioned on the outcome not having been achieved yet, $\frac{p_{\Delta_{t+1}}(\Delta_{t+1}=0)}{1-p_{\Delta_{t+1}}(\Delta_{t+1}=0)}$.*

C.1.8 Dynamic-Discount Outcome-Driven Policy Iteration

Theorem 5.4. *Assume $|\mathcal{A}| < \infty$ and that the MDP is ergodic.*

1. *Outcome-Driven Policy Evaluation (ODPE): Given a policy π and a function $Q^0 : \mathcal{S}\times\mathcal{A}\times\mathcal{S}\to\mathbb{R}$, define $Q^{i+1} = \mathcal{T}^\pi Q^i$. Then the sequence Q^i converges to the lower bound in Theorem 5.3.*

2. *Outcome-Driven Policy Improvement (ODPI): The policy*

$$\pi^+ = \arg\max_{\pi'\in\Pi}\Big\{\mathbb{E}_{\pi'(\mathbf{a}_t\,|\,\mathbf{s}_t)}\left[Q^\pi(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_T)\right] - \mathbb{D}_{\mathrm{KL}}(\pi'(\cdot\,|\,\mathbf{s}_t)\,||\,p(\cdot\,|\,\mathbf{s}_t))\Big\} \tag{C.193}$$

and the variational distribution over T recursively defined in terms of

$$\begin{aligned}
&q^+(\Delta_{t+1}=0\,|\,\mathbf{s}_0;\pi,Q^\pi) \\
&= \sigma\Big(\mathbb{E}_{\pi(\mathbf{a}_{t+1}\,|\,\mathbf{s}_{t+1})p_d(\mathbf{s}_{t+1}\,|\,\mathbf{s}_t,\mathbf{a}_t)}[Q^\pi(\mathbf{s}_{t+1},\mathbf{a}_{t+1},\mathbf{g};q_T)] \\
&\qquad - \mathbb{E}_{\pi(\mathbf{a}_t\,|\,\mathbf{s}_t)}[\log p_d(\mathbf{g}\,|\,\mathbf{s}_t,\mathbf{a}_t)] + \sigma^{-1}\left(p_{\Delta_{t+1}}(\Delta_{t+1}=0)\right)\Big)
\end{aligned} \tag{C.194}$$

improve the variational objective. In other words, $\mathcal{F}(\pi^+, q_T, \mathbf{s}_0, \mathbf{g}) \geq \mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g})$ *and* $\mathcal{F}(\pi, q_T^+, \mathbf{s}_0) \geq \mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g})$ *for all* $\mathbf{s}_0 \in \mathcal{S}$.

3. *Alternating between ODPE and ODPI converges to a policy* $\pi^\star$ *and a variational distribution over* T, $q_T^\star$, *such that* $Q^{\pi^\star}(\mathbf{s}, \mathbf{a}, \mathbf{g}; q_T^\star) \geq Q^\pi(\mathbf{s}, \mathbf{a}, \mathbf{g}; q_T)$ *for all* $(\pi, q_T) \in \Pi \times \mathcal{Q}_T$ *and any* $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$.

Proof. Parts of this proof are adapted from the proof given in Haarnoja et al. (2018a), modified for the Bellman operator proposed in Definition 5.2.

1. Outcome-Driven Policy Evaluation (ODPE): Instead of absorbing the KL term into the Q-function, we can define an KL-regularized reward as

$$r^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) \stackrel{\text{def}}{=} q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \log p_d(\mathbf{g} \mid \mathbf{s}_t, \mathbf{a}_t) - \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \parallel p_{\Delta_{t+1}}) \\ - q_{\Delta_{t+1}}(\Delta_{t+1} = 0)\, \mathbb{E}_{p_d(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)}[\mathbb{D}_{\mathrm{KL}}(\pi(\cdot \mid \mathbf{s}_{t+1}) \parallel p(\cdot \mid \mathbf{s}_{t+1}))]. \tag{C.195}$$

   We can then write an update rule according to Definition 5.2 as

$$\tilde{Q}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T) \leftarrow r^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) \\ + q_{\Delta_{t+1}}(\Delta_{t+1} = 0)\, \mathbb{E}_{\pi(\mathbf{a}_{t+1} \mid \mathbf{s}_{t+1}) p_d(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)}[\tilde{Q}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \mathbf{g}; q_T)], \tag{C.196}$$

   where $q_{\Delta_{t+1}}(\Delta_{t+1} = 0) \leq 1$. This update is similar to a Bellman update (Sutton et al., 1998), but with a discount factor given by $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$. In general, this discount factor $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ can be computed dynamically based on the current state and action, such as in Equation (5.35). As discussed in White (2017), this Bellman operator is still a contraction mapping so long as the Markov chain induced by the current policy is ergodic and there exists a state such that $q_{\Delta_{t+1}}(\Delta_{t+1} = 0) < 1$. The first condition is true by assumption. The second condition is true since $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ is given by Equation (5.35), which is always strictly between 0 and 1. Therefore, we apply convergence results for policy evaluation with transition-dependent discount factors (White, 2017) to this contraction mapping, and the result immediately follows.

2. Outcome-Driven Policy Improvement (ODPI): Let $\pi_{\text{old}} \in \Pi$ and let $Q^{\pi_{\text{old}}}$ and $V^{\pi_{\text{old}}}$ be the outcome-driven state and state-action value functions from Definition 5.2, let q_T be some variational distribution over T, and let π_{new} be given by

$$\pi_{\text{new}}(\mathbf{a}_t|\mathbf{s}_t) = \arg\max_{\pi' \in \Pi} \left\{ \mathbb{E}_{\pi'(\mathbf{a}_t \mid \mathbf{s}_t)} \left[Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T) \right] - \mathbb{D}_{\mathrm{KL}}(\pi'(\cdot \mid \mathbf{s}_t) \parallel p(\cdot \mid \mathbf{s}_t)) \right\} \tag{C.197}$$

$$\stackrel{\text{def}}{=} \arg\max_{\pi' \in \Pi} \mathcal{J}_{\pi_{\text{old}}}(\pi'(\mathbf{a}_t \mid \mathbf{s}_t); q_T). \tag{C.198}$$

Then, it must be true that $\mathcal{J}_{\pi_{\text{old}}}(\pi_{\text{old}}(\mathbf{a}_t|\mathbf{s}_t); q_T) \leq \mathcal{J}_{\pi_{\text{old}}}(\pi_{\text{new}}(\mathbf{a}_t|\mathbf{s}_t); q_T)$, since one could set $\pi_{\text{new}} = \pi_{\text{old}} \in \Pi$. Thus,

$$\begin{aligned}&\mathbb{E}_{\pi_{\text{new}}(\mathbf{a}_t|\mathbf{s}_t)}\left[Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T)\right] - \mathbb{D}_{\text{KL}}(\pi_{\text{new}}(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)) \\ &\quad \geq \mathbb{E}_{\pi_{\text{old}}(\mathbf{a}_t|\mathbf{s}_t)}\left[Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T)\right] - \mathbb{D}_{\text{KL}}(\pi_{\text{old}}(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)),\end{aligned} \tag{C.199}$$

and since

$$V^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{g}; q_T) = \mathbb{E}_{\pi_{\text{old}}(\mathbf{a}_t|\mathbf{s}_t)}\left[Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T)\right] - \mathbb{D}_{\text{KL}}(\pi_{\text{old}}(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)), \tag{C.200}$$

we get

$$\mathbb{E}_{\pi_{\text{new}}(\mathbf{a}_t|\mathbf{s}_t)}\left[Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T)\right] - \mathbb{D}_{\text{KL}}(\pi_{\text{new}}(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)) \geq V^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{g}; q_T). \tag{C.201}$$

We can now write the Bellman equation as

$$\begin{aligned}&Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T) \\ &\quad = q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) \qquad \text{(C.202)} \\ &\qquad + q_{\Delta_{t+1}}(\Delta_{t+1} = 0)\, \mathbb{E}_{p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)}[V^{\pi_{\text{old}}}(\mathbf{s}_{t+1}, \mathbf{g}; q_T)] \\ &\quad \leq q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) \\ &\qquad + q_{\Delta_{t+1}}(\Delta_{t+1} = 0)\, \mathbb{E}_{p_d(\mathbf{s}_{t'}|\mathbf{s}_t, \mathbf{a}_t)}[\mathbb{E}_{\pi_{\text{new}}(\mathbf{a}_{t'}|\mathbf{s}_{t'})}\left[Q^{\pi_{\text{old}}}(\mathbf{s}_{t'}, \mathbf{a}_{t'}, \mathbf{g}; q_T)\right] \\ &\qquad\quad - \mathbb{D}_{\text{KL}}(\pi_{\text{new}}(\cdot \,|\, \mathbf{s}_{t'}) \parallel p(\cdot \,|\, \mathbf{s}_{t'}))], \qquad \text{(C.203)} \\ &\quad \vdots \\ &\quad \leq Q^{\pi_{\text{new}}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T) \qquad \text{(C.204)}\end{aligned}$$

where we defined $t' \stackrel{\text{def}}{=} t + 1$, repeatedly applied the Bellman backup operator defined in Definition 5.2 and used the bound in Equation (C.201). Convergence follows from Outcome-Driven Policy Evaluation above.

3. Locally Optimal Variational Outcome-Driven Policy Iteration: Define π^i to be a policy at iteration i. By ODPI for a given q_T, the sequence of state-action value functions $\{Q^{\pi^i}(q_T)\}_{i=1}^{\infty}$ is monotonically increasing in i. Since the reward is finite and the negative KL divergence is upper bounded by zero, $Q^{\pi}(q_T)$ is upper bounded for $\pi \in \Pi$ and the sequence $\{\pi^i\}_{i=1}^{\infty}$ converges to some $\pi^\star$. To see that $\pi^\star$ is an optimal policy, note that it must be the case that $\mathcal{J}_{\pi^\star}(\pi^\star(\mathbf{a}_t|\mathbf{s}_t); q_T) > \mathcal{J}_{\pi^\star}(\pi(\mathbf{a}_t \,|\, \mathbf{s}_t); q_T)$ for any $\pi \in \Pi$ with $\pi \neq \pi^\star$. By the argument used in ODPI above, it must be the case that the outcome-driven state-action value of the converged policy is higher than that of any other non-converged policy in Π, that is, $Q^{\pi^\star}(\mathbf{s}_t, \mathbf{a}_t; q_T) > Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t; q_T)$ for all $\pi \in \Pi$ and any $q_T^i \in \mathcal{Q}_T$ and $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$. Therefore, given q_T, $\pi^\star$ must be optimal in Π, which concludes the proof.

4. Globally Optimal Variational Outcome-Driven Policy Iteration: Let π^i be a policy and let q_T^i be variational distributions over T at iteration i. By Locally Optimal Variational Outcome-Driven Policy Iteration, for a *fixed* q_T^i with $q_T^i = q_T^j \forall i, j \in \mathbb{N}_0$, the sequence of $\{(\pi^i, q_T^i)\}_{i=1}^{\infty}$ increases the objective in Equation (C.119) at each iteration and converges to a stationary point in π^i, where $Q^{\pi^\star}(\mathbf{s}_t, \mathbf{a}_t; q_T^i) > Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t; q_T^i)$ for all $\pi \in \Pi$ and any $q_T^i \in \mathcal{Q}_T$ and $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$. Since the objective in Equation (C.119) is concave in q_T, it must be the case that for $q_T^{\star^i} \in \mathcal{Q}_T$, the optimal variational distribution over T at iteration i, defined recursively by

$$
\begin{aligned}
&q^{\star^i}(\Delta_{t+1} = 0; \pi^i, Q^{\pi^i}) \\
&\quad = \sigma\Big(\mathbb{E}_{\pi(\mathbf{a}_{t+1} \,|\, \mathbf{s}_{t+1}) p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)}[Q^{\pi^i}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \mathbf{g}; q_T(\pi^i, Q^{\pi^i}))] \\
&\qquad - \mathbb{E}_{\pi(\mathbf{a}_t \,|\, \mathbf{s}_t)}[\log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t)] + \sigma^{-1}(p_{\Delta_{t+1}}(\Delta_{t+1} = 0)) \Big),
\end{aligned}
$$

for $t \in \mathbb{N}_0$, $Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t; q_T^\star) > Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t; q_T)$ for all $\pi \in \Pi$ and any $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$. Note that q_T is defined implicitly in terms of π^i and Q^{π^i}, that is, the optimal variational distribution over T at iteration i is defined as a function of the policy and Q-function at iteration i. Hence, it must then be true that $Q^{\pi^\star}(\mathbf{s}_t, \mathbf{a}_t; q_T^\star) > Q^{\pi^\star}(\mathbf{s}_t, \mathbf{a}_t; q_T)$ for all $q_T^\star(\pi^\star, Q^{\pi^\star}) \in \mathcal{Q}_T$ and for any $\pi^\star \in \Pi$ and $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$. In other words, for an optimal policy and corresponding Q-function, there exists an optimal variational distribution over T that maximizes the Q-function, given the optimal policy. Repeating locally optimal variational outcome-driven policy iteration under the new variational distribution $q_T^\star(\pi^\star, Q^{\pi^\star})$ will yield an optimal policy $\pi^{\star\star}$ and computing the corresponding optimal variational distribution, $q_T^{\star\star}(\pi^{\star\star}, Q^{\pi^{\star\star}})$ will further increase the variational objective such that for $\pi^{\star\star} \in \Pi$ and $q_T^{\star\star}(\pi^{\star\star}, Q^{\pi^{\star\star}}) \in \mathcal{Q}_T$, we have that

$$
Q^{\pi^{\star\star}}(\mathbf{s}_t, \mathbf{a}_t; q_T^{\star\star}) > Q^{\pi^{\star\star}}(\mathbf{s}_t, \mathbf{a}_t; q_T^\star) > Q^{\pi^\star}(\mathbf{s}_t, \mathbf{a}_t; q_T^\star) > Q^{\pi^\star}(\mathbf{s}_t, \mathbf{a}_t; q_T) \quad \text{(C.205)}
$$

for any $\pi^\star \in \Pi$ and $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$. Hence, globally optimal variational outcome-driven policy iteration increases the variational objective at every step. Since the objective is upper bounded (by virtue of the rewards being finite and the negative KL divergence being upper bounded by zero) and the sequence of $\{(\pi^i, q_T^i)\}_{i=1}^{\infty}$ increases the objective in Equation (C.119) at each iteration, by the monotone convergence theorem, the objective value converges to a supremum and since the objective function is concave the supremum is unique. Hence, since the supremum is unique and obtained via globally optimal variational outcome-driven policy iteration on $(\pi, q_T) \in \Pi \times \mathcal{Q}_T$, the sequence of $\{(\pi^i, q_T^i)\}_{i=1}^{\infty}$ converges to a unique stationary point $(\pi^\star, q_T^\star) \in \Pi \times \mathcal{Q}_T$, where $Q^{\pi^\star}(\mathbf{s}_t, \mathbf{a}_t; q_T^\star) > Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t; q_T^i)$ for all $\pi \in \Pi$ and any $q_T^i \in \mathcal{Q}_T$ and $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$.

□

Corollary 5.4. *Variational Outcome-Driven Policy Iteration on* $(\pi, q_T) \in \Pi \times \mathcal{Q}_T$ *results in an optimal policy at least as good as or better than any optimal policy attainable from policy iteration on* $\pi \in \Pi$ *alone.*

Remark 4. *The convergence proof of ODPE assumes a transition-dependent discount factor (White, 2017), because the variational distribution used in Equation (5.35) depends on the next state and action as well as on the desired outcome.*

C.1.9 Lemmas

Lemma C.1. *Let* $q(T = t) \stackrel{\text{def}}{=} q(T = t|T \geq t) \prod_{i=1}^{t} q(T \neq i-1|T \geq i-1)$ *be a discrete probability distribution with support* $\mathbb{N}_0$*. Then for any* $t \in \mathbb{N}_0$*, we have that*

$$q(T \geq t) = \sum_{i=t}^{\infty} q(T = i|T \geq i) \prod_{j=1}^{i} q(T \neq j-1|T \geq j-1) = \prod_{i=1}^{t} q(T \neq i-1|T \geq i-1). \tag{C.206}$$

Proof. We prove the statement by induction on t.

Base case: For $t = 0$, $q(T \geq 0) = 1$ by definition of the empty product.

Inductive case: Note that $q(T \leq t) = \prod_{i=1}^{t} q(T = i-1|T \geq i-1)$. Show that

$$q(T \geq t) = \prod_{i=1}^{t} q(T \neq i-1|T \geq i-1) \implies q(T \geq t+1) = \prod_{i=1}^{t+1} q(T \neq i-1|T \geq i-1). \tag{C.207}$$

Consider $q(T \geq t+1) = \sum_{i=t+1}^{\infty} q(T = i|T \geq i) \prod_{j=1}^{i} q(T \neq j-1|T \geq j-1)$. To prove the inductive hypothesis, we need to show that the following equality is true:

$$\sum_{i=t+1}^{\infty} q(T = i|T \geq i) \prod_{j=1}^{i} q(T \neq j-1|T \geq j-1) = \prod_{i=1}^{t+1} q(T \neq i-1|T \geq i-1) \tag{C.208}$$

$$\begin{aligned} \iff & \sum_{i=t}^{\infty} q(T = i|T \geq i) \prod_{j=1}^{i} q(T \neq j-1|T \geq j-1) \\ & \quad - q(T = t|T \geq t) \prod_{j=1}^{t} q(T \neq j-1|T \geq j-1) \\ & = q(T \neq t|T \geq t) \prod_{i=1}^{t} q(T \neq i-1|T \geq i-1). \end{aligned} \tag{C.209}$$

By the inductive hypothesis,

$$q(T \geq t) = \sum_{i=t}^{\infty} q(T = i | T \geq i) \prod_{j=1}^{i} q(T \neq j-1 | T \geq j-1) = \prod_{i=1}^{t} q(T \neq i-1 | T \geq i-1), \tag{C.210}$$

and so

$$\text{Equation (C.209)} \iff \prod_{j=1}^{t} q(T \neq j | T \geq j) - q(T \neq t+1 | T \geq t+1) \prod_{j=1}^{t} q(T = j | T \geq j) \tag{C.211}$$

$$= q(T \neq t | T \geq t) \prod_{i=1}^{t} q(T \neq i-1 | T \geq i-1). \tag{C.212}$$

Factoring out $\prod_{i=1}^{t} q(T \neq i-1 | T \geq i-1)$, we get

$$\begin{aligned} &\iff \prod_{j=1}^{t} q(T \neq j-1 | T \geq j-1) \underbrace{(1 - q(T = t | T \geq t))}_{= q(T \neq t | T \geq t)} \\ &= q(T \neq t | T \geq t) \prod_{j=1}^{t} q(T = j-1 | T \geq j-1) \end{aligned} \tag{C.213}$$

$$\begin{aligned} &\iff q(T \neq t | T \geq t) \prod_{j=1}^{t} q(T \neq j-1 | T \geq j-1) \\ &= q(T \neq t | T \geq t) \prod_{j=1}^{t} q(T \neq j-1 | T \geq j-1), \end{aligned} \tag{C.214}$$

which proves the inductive hypothesis. □

Lemma C.2. *Let $q_T(t)$ and $p_T(t)$ be discrete probability distributions with support $\mathbb{N}_0$, let Δ_t be a Bernoulli random variable, with success defined as $T = t+1$ given that $T \geq t$, and let q_{Δ_t} be a discrete probability distribution over Δ_t for $t \in \mathbb{N} \backslash \{0\}$, so that*

$$\begin{aligned} q_{\Delta_{t+1}}(\Delta_{t+1} = 0) &\stackrel{\text{def}}{=} q(T \neq t \,|\, T \geq t) \\ q_{\Delta_{t+1}}(\Delta_{t+1} = 1) &\stackrel{\text{def}}{=} q(T = t \,|\, T \geq t). \end{aligned} \tag{C.215}$$

Then we can write $q(T = t) = q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \prod_{i=1}^{t} q_{\Delta_i}(\Delta_i = 0)$ for any $t \in \mathbb{N}_0$ and have that

$$q(T \geq t) = \sum_{i=t}^{\infty} q_{\Delta_{i+1}}(\Delta_{i+1} = 1) \prod_{j=1}^{i} q_{\Delta_j}(\Delta_j = 0) = \prod_{i=1}^{t} q_{\Delta_i}(\Delta_i = 0). \tag{C.216}$$

Proof. By Lemma C.1, we have that for any $t \in \mathbb{N}_0$

$$q(T \geq t) = \sum_{i=t}^{\infty} q(T = i | T \geq i) \prod_{j=1}^{i} q(T \neq j-1 | T \geq j-1) = \prod_{i=1}^{t} q(T \neq i-1 | T \geq i-1). \tag{C.217}$$

The result follows by replacing $q(T = i|T \geq i)$ by $q_{\Delta_{i+1}}(\Delta_{i+1} = 1)$, $q(T \neq j-1|T \geq j-1)$ by $q_{\Delta_j}(\Delta_j = 0)$, and $q(T \neq i-1|T \geq i-1)$ by $q_{\Delta_i}(\Delta_i = 0)$. □

Lemma C.3. *Let $q_T(t)$ and $p_T(t)$ be discrete probability distributions with support $\mathbb{N}_0$. Then for any $k \in \mathbb{N}_0$,*

$$\begin{aligned}
&\mathbb{E}_{t \sim q(T \,|\, T \geq k)} \left[\log \frac{q(T = t \,|\, T \geq k)}{p(T = t \,|\, T \geq k)} \right] \\
&\quad = f(q, p, k) + q(T \neq k \,|\, T \geq k)\, \mathbb{E}_{t \sim q(T \,|\, T \geq k+1)} \left[\log \frac{q(T = t \,|\, T \geq k+1)}{p(T = t \,|\, T \geq k+1)} \right].
\end{aligned} \tag{C.218}$$

Proof. Consider $\mathbb{E}_{t \sim q(T \,|\, T \geq k)} \left[\log \frac{q(T=t \,|\, T \geq k)}{p(T=t \,|\, T \geq k)} \right]$ and note that by the law of total expectation we can rewrite it as

$$\begin{aligned}
&\mathbb{E}_{t \sim q(T \,|\, T \geq k)} \left[\log \frac{q(T = t \,|\, T \geq k)}{p(T = t \,|\, T \geq k)} \right] \\
&= q(T = k \,|\, T \geq k)\, \mathbb{E}_{t \sim q(T \,|\, T = k)} \left[\log \frac{q(T = t \,|\, T \geq k)}{p(T = t \,|\, T \geq k)} \right] \\
&\quad + q(T \neq k \,|\, T \geq k)\, \mathbb{E}_{t \sim q(T \,|\, T \geq k+1)} \left[\log \frac{q(T = t \,|\, T \geq k)}{p(T = t \,|\, T \geq k)} \right] && \text{(C.219)} \\
&= q(T = k \,|\, T \geq k) \log \frac{q(T = k \,|\, T \geq k)}{p(T = k \,|\, T \geq k)} && \text{(C.220)} \\
&\quad + q(T \neq k \,|\, T \geq k)\, \mathbb{E}_{t \sim q(T | T \geq k+1)} \left[\log \frac{q(T = t \,|\, T \geq k)}{p(T = t \,|\, T \geq k)} \right]. && \text{(C.221)}
\end{aligned}$$

For all values of $T \geq k+1$, we have that

$$q(T = t \,|\, T \geq k) = q(T = t \,|\, T \geq k+1) q(T \neq k \,|\, T \geq k) \tag{C.222}$$

$$p(T = t \,|\, T \geq k) = p(T = t \,|\, T \geq k+1) p(T \neq k \,|\, T \geq k) \tag{C.223}$$

and so we can rewrite the expectation in Equation (C.220) as

$$\begin{aligned}
&\mathbb{E}_{t \sim q(T \,|\, T \geq k+1)} \left[\log \frac{q(T = t \,|\, T \geq k)}{p(T = t \,|\, T \geq k)} \right] && \text{(C.224)} \\
&= \mathbb{E}_{t \sim q(T \,|\, T \geq k+1)} \left[\log \frac{q(T = t \,|\, T \geq k)}{p(T = t \,|\, T \geq k)} + \log \frac{q(T \neq k \,|\, T \geq k)}{p(T \neq k \,|\, T \geq k)} \right] && \text{(C.225)} \\
&= \mathbb{E}_{t \sim q(T \,|\, T \geq k+1)} \left[\log \frac{q(T = t \,|\, T \geq k)}{p(T = t \,|\, T \geq k)} \right] + \log \frac{q(T \neq k \,|\, T \geq k)}{p(T \neq k \,|\, T \geq k)} && \text{(C.226)}
\end{aligned}$$

Combining Equation (C.226) with Equation (C.220), we have

$$\begin{aligned}
&\mathbb{E}_{t\sim q(T\,|\,T\geq k)}\left[\log\frac{q(T=t\,|\,T\geq k)}{p(T=t\,|\,T\geq k)}\right] \\
&\quad=\underbrace{q(T=k\,|\,T\geq k)\log\frac{q(T=k\,|\,T\geq k)}{p(T=k\,|\,T\geq k)}+q(T\neq k\,|\,T\geq k)\log\frac{q(T\neq k\,|\,T\geq k)}{p(T\neq k\,|\,T\geq k)}}_{\stackrel{\text{def}}{=}f(q,p,k)} \\
&\qquad+q(T\neq k\,|\,T\geq k)\,\mathbb{E}_{t\sim q(T\,|\,T\geq k+1)}\left[\log\frac{q(T=t\,|\,T\geq k+1)}{p(T=t\,|\,T\geq k+1)}\right],
\end{aligned} \tag{C.227}$$

which concludes the proof. □

Lemma C.4. *Let $q_T(t)$ and $p_T(t)$ be discrete probability distributions with support $\mathbb{N}_0$. Then the KL divergence from q_T to p_T can be written as*

$$\mathbb{D}_{\mathrm{KL}}(q_T\,||\,p_T)=\sum_{t=0}^{\infty}q(T\geq t)f(q_T,p_T,t) \tag{C.228}$$

where $f(q_T,p_T,t)$ is shorthand for

$$f(q_T,p_T,t)=q(T=t\,|\,T\geq t)\log\frac{q(T=t\,|\,T\geq t)}{p(T=t\,|\,T\geq t)}+q(T\neq t\,|\,T\geq t)\log\frac{q(T\neq t\,|\,T\geq t)}{p(T\neq t\,|\,T\geq t)}. \tag{C.229}$$

Proof. Note that $q(T=k)$ denotes the probability that the distribution q assigns to the event $T=k$ and $q(T\geq m)$ denotes the tail probability, that is, $q(T\geq m)=\sum_{t=m}^{\infty}q(T=t)$. We will write $q(T|T\geq m)$ to denote the conditional distribution of q given $T\geq m$, that is, $q(T=k|T\geq m)=\mathbb{1}[k\geq m]q(T=k)/q(T\geq m)$. We will use analogous notation for p.

By the definition of the KL divergence and using the fact that, since the support is bounded from below by $T=0$, $q(T=0)=q(T=0\,|\,T\geq 0)$, we have

$$\mathbb{D}_{\mathrm{KL}}(q_T\parallel p_T)=\mathbb{E}_{t\sim q(T)}\left[\log\frac{q(T=t)}{p(T=t)}\right]=\mathbb{E}_{t\sim q(T\,|\,T\geq 0)}\left[\log\frac{q(T=t\,|\,T\geq 0)}{p(T=t\,|\,T\geq 0)}\right]. \tag{C.230}$$

Using Lemma C.3 with $k = 0, 1, 2, 3, \ldots$, we can expand the above expression to get

$$\mathbb{D}_{\mathrm{KL}}(q_T \parallel p_T) \tag{C.231}$$

$$= f(q_T, p_T, 0) + q(T \neq 0 \,|\, T \geq 0)\, \mathbb{E}_{t \sim q(T \,|\, T \geq 1)} \left[\log \frac{q(T = t \,|\, T \geq 1)}{p(T = t \,|\, T \geq 1)} \right] \tag{C.232}$$

$$\begin{aligned} &= f(q, p, 0) + q(T \neq 0 \,|\, T \geq 1) f(q_T, p_T, 1) \\ &\quad + q(T \neq 0 \,|\, T \geq 0) q(T \neq 1 \,|\, T \geq 1)\, \mathbb{E}_{t \sim q(T \,|\, T \geq 2)} \left[\log \frac{q(T = t \,|\, T \geq 2)}{p(T = t \,|\, T \geq 2)} \right] \end{aligned} \tag{C.233}$$

$$\begin{aligned} &= \underbrace{1}_{= q(T \geq 0)} \cdot f(q, p, 0) \\ &\quad + \underbrace{q(T \neq 0 \,|\, T \geq 0)}_{= q(T \geq 1)} f(q, p, 1) \\ &\quad + \underbrace{q(T \neq 0 \,|\, T \geq 0) q(T \neq 1 \,|\, T \geq 1)}_{= q(T \geq 2)} f(q_T, p_T, 2) \\ &\quad + \underbrace{q(T \neq 0 \,|\, T \geq 0) q(T \neq 1 \,|\, T \geq 1) q(T \neq 2 \,|\, T \geq 2)}_{= q(T \geq 3)} \mathbb{E}_{t \sim q(T \,|\, T \geq 3)} \left[\log \frac{q(T = t \,|\, T \geq 3)}{p(T = t \,|\, T \geq 3)} \right] \end{aligned} \tag{C.234}$$

$$= \sum_{t=0}^{\infty} q(T \geq t) f(q_T, p_T, t), \tag{C.235}$$

where $f(q_T, p_T, t)$ is shorthand for

$$f(q_T, p_T, t) = q(T = t \,|\, T \geq t) \log \frac{q(T = t \,|\, T \geq t)}{p(T = t \,|\, T \geq t)} + q(T \neq t \,|\, T \geq t) \log \frac{q(T \neq t \,|\, T \geq t)}{p(T \neq t \,|\, T \geq t)}, \tag{C.236}$$

and we used the fact that, by Lemma C.1,

$$q(T \geq t) = \prod_{k=1}^{t} q(T \neq k - 1 \,|\, T \geq k - 1). \tag{C.237}$$

This completes the proof. □

Lemma C.5. *Let $q_T(t)$ and $p_T(t)$ be discrete probability distributions with support $\mathbb{N}_0$, let Δ_t be a Bernoulli random variable, with success defined as $T = t$ given that $T \geq t$, and let q_{Δ_t} and p_{Δ_t} be discrete probability distributions over Δ_t for $t \in \mathbb{N}_0 \backslash \{0\}$, so that*

$$q_{\Delta_{t+1}}(\Delta_{t+1} = 0) \stackrel{\text{def}}{=} q(T \neq t \,|\, T \geq t) \qquad q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \stackrel{\text{def}}{=} q(T = t \,|\, T \geq t) \tag{C.238}$$

$$p_{\Delta_{t+1}}(\Delta_{t+1} = 0) \stackrel{\text{def}}{=} p(T \neq t \,|\, T \geq t) \qquad p_{\Delta_{t+1}}(\Delta_{t+1} = 1) \stackrel{\text{def}}{=} p(T = t \,|\, T \geq t). \tag{C.239}$$

Then the KL divergence from q_T to p_T can be written as

$$\mathbb{D}_{\mathrm{KL}}(q_T \,||\, p_T) = \sum_{t=0}^{\infty} \Big(\prod_{k=1}^{t} q_{\Delta_k}(\Delta_k = 0) \Big) \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \,||\, p_{\Delta_{t+1}}). \tag{C.240}$$

Proof. The result follows from Lemma C.4, Equation (C.237), Equation (C.238), and the definition of f.

In detail, from Lemma C.1, and Equation (C.238) we have that

$$q(T \geq t) = \prod_{k=1}^{t} q(T \neq k-1 \,|\, T \geq k-1) = \prod_{k=1}^{t} q_{\Delta_k}(\Delta_k = 0). \tag{C.241}$$

From the definition of $f(q_T, p_T, t)$, we have

$$f(q_T, p_T, t) = q(T = t \,|\, T \geq t) \log \frac{q(T = t \,|\, T \geq t)}{p(T = t \,|\, T \geq t)} + q(T \neq t \,|\, T \geq t) \log \frac{q(T \neq t \,|\, T \geq t)}{p(T \neq t \,|\, T \geq t)} \tag{C.242}$$

$$= q_{\Delta_{t+1}}(\Delta_{t+1} = 0) \log \frac{q_{\Delta_{t+1}}(\Delta_{t+1} = 0)}{p_{\Delta_{t+1}}(\Delta_{t+1} = 0)} + q(\Delta_{t+1} = 1) \log \frac{q_{\Delta_{t+1}}(\Delta_{t+1} = 1)}{p_{\Delta_{t+1}}(\Delta_{t+1} = 1)} \tag{C.243}$$

$$= \mathbb{D}_{\mathrm{KL}}(q_{\Delta_{t+1}} \,\|\, p_{\Delta_{t+1}}). \tag{C.244}$$

Combining Equation (C.241), Equation (C.244), and Equation (C.228) completes the proof. □

C.2 Experimental Details

C.2.1 Environment

Ant. The Ant domain is based on the "Ant-v3" OpenAI Gym (Brockman et al., 2016) environment, with three modifications: the gear ratio is reduced from 150 to 120, the contact force sensors are removed from the state, and there is no termination condition and the episode only terminates after a fixed amount of time. In this environment, the state space is 23-dimensional, consisting of the XYZ coordinate of the center of the torso, the orientation of the ant (in quaternion), and the angle and angular velocity of all 8 joints. The action space is 8-dimensional and corresponds to the torque to apply to each joint. The desired outcome consists of the desired XYZ, orientation, and joint angles at a position that is 5 meters down and to the right of the initial position. This desired pose is shown in Figure 5.2.

Sawyer Push. In this environment, the state and goal space is 4-dimensional and the action space is 2-dimensional. The state and goal consist of the XY position of the end effector (EE) and the XY position of the puck. The object is on a $40\,\text{cm} \times 50\,\text{cm}$ table and starts 20 cm in front of the hand. The goal puck position is fixed to 15 cm forward and 30 cm to the right of the initial hand position, while the goal hand position is 5 cm behind and 20 cm to the right of the initial hand position. The action is the change in position in each XY direction, with a maximum change of 3 cm per direction at each time step. The episode horizon is 100.

Sawyer Window and Faucet. In this environment, the state and goal space is 6-dimensional and the action space is 2-dimensional. The state and goal consist of the XYZ position of the end effector (EE) and the XYZ position of the window or faucet endpoint. The hand is initialized away from the window and faucet. The EE goal XYZ position is set to the initial window or faucet position. The action is the change in position in each XYZ direction. For the window task, the goal position is to close the window, and for the faucet task, the goal position is to rotate the faucet 90 degrees counter-clockwise from above.

Box 2D. In this environment, the state space is a 4×4 grid with a 2×2 box in the middle. The agent is initialized at $(-3.5, -2)$ and the desired outcome is $(3.5, 2)$. The action is the XY velocity of the agent, with wall collisions taken into account and maximum velocity of 0.2 in each direction. To make the environment stochastic, we add Gaussian noise to actions with mean zero and a standard deviation that is 10% of the maximum action magnitude.

Tabular Box 2D (Figure 5.1). We implemented a tabular version of ODAC and applied it to the 2D environment shown in Figure 5.1. We discretize the environment into an 8×8 grid of states. The actions correspond to moving up, down, left, or right. With probability $1 - \epsilon$, this action is taken. If the agent runs into a wall or boundary, the agent stays in its current state. With probability $\epsilon = 0.1$, the commanded action is ignored and a neighboring grid state (including the current state) is uniformly sampled as the next state. The policy and Q-function are represented with look-up tables and randomly initialized. The entropy reward is weighted by 0.01 and the time prior p_T is geometric with parameter 0.5. The dynamics model $p_d^{(0)}$ is initialized to assign uniform probability to each next state for every state and action. In each iteration, we simulate data collection by updating the dynamics model with the running average update $p_d^{(t+1)} = 0.99 p_d^{(t)} + 0.01 p_d$, where p_d is the true dynamics, and update the policy and Q-function according to Equation (5.41) and Equation (5.39), respectively. Figure 5.1 shows that, in contrast to the binary-reward setting, the learned reward provides shaping for the policy, which solves the task within 100 iterations.

C.2.2 Algorithm

Pseudocode for the complete algorithm is shown in Algorithm 2.

C.2.3 Implementation Details

Dynamics model. For the Ant and Sawyer experiments, we train a neural network to output the mean and standard deviation of a Laplace distribution. This distribution is then used to model the distribution over the *difference* between the current state and the next state, which we found to be more reliable than predicting the next state. So, the overall distribution is given by a Laplace distribution with learned mean μ and fixed scale σ computed via

$$p_\psi = \text{Laplace}(\mu = g_\psi(\mathbf{s}, \mathbf{a}) + f(\mathbf{s}), \sigma = 0.00001)$$

where g_ψ is the output of a network and f is a function that maps a state into a goal.

For the 2D Navigation experiment, we use a Gaussian distribution. The dynamics neural network has hidden layers of size $[64, 64]$ with ReLU hidden activations. For the Ant and Sawyer experiments, there is no output activation. For the linear-Gaussian and 2D Navigation experiments, we use a tanh output.

Algorithm 2 Outcome-Driven Actor–Critic

Require: Policy π_θ, Q-function Q_ϕ, dynamics model p_ψ, replay buffer $\mathcal{R}$, and map from state to achieved goal f.
  for $n = 0, \ldots, N-1$ episodes **do**
    Sample initial state $\mathbf{s}_0$ from environment.
    Sample goal $\mathbf{g}$ from environment.
    for $t = 0, \ldots, H-1$ steps **do**
      Get action $\mathbf{a}_t \sim \pi_\theta(\mathbf{s}_t, \mathbf{g})$.
      Get next state $\mathbf{s}_{t+1} \sim p(\cdot \,|\, \mathbf{s}_t, \mathbf{a}_t)$.
      Store $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, \mathbf{g})$ into replay buffer $\mathcal{R}$.
      Sample transition $(\mathbf{s}, \mathbf{a}, \mathbf{s}', \mathbf{g}) \sim \mathcal{R}$.
      Compute reward $r = \log p_\psi(\mathbf{g} \,|\, \mathbf{s}, \mathbf{a}) - \mathbb{D}_{\mathrm{KL}}(q_\Delta(\cdot|\mathbf{s}, \mathbf{a}) \,\|\, p(\Delta))$.
      Compute $q(\Delta_{t+1} = 0|\mathbf{s}, \mathbf{a})$ using Equation (5.35).
      Update Q_ϕ using Equation (5.42) and data $(\mathbf{s}, \mathbf{a}, \mathbf{s}', \mathbf{g}, r)$.
      Update π_θ using Equation (5.43) and data $(\mathbf{s}, \mathbf{a}, \mathbf{g})$.
      Update p_ψ using Equation (5.44) and data $(\mathbf{s}, \mathbf{a}, \mathbf{s}')$.
    end for
    for $t = 0, ..., H-1$ steps **do**
      for $i = 0, ..., k-1$ steps **do**
        Sample future state $\mathbf{s}_{h_i}$, where $t < h_i \leq H-1$.
        Store $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, f(\mathbf{s}_{h_i}))$ into $\mathcal{R}$.
      end for
    end for
  end for

Reward normalization. Because the different experiments have rewards of very different scale, we normalize the rewards by dividing by a running average of the maximum reward magnitude. Specifically, for every reward r in the ith batch of data, we replace the reward with

$$\hat{r} = r/C_i$$

where we update the normalizing coefficient C_i using each batch of reward $\{r_b\}_{b=1}^B$:

$$C_{i+1} \leftarrow (1-\lambda) \times C_i + \lambda \max_{b \in [1,\ldots,B]} |r_b|$$

and C_i is initialized to 1. In our experiments, we use $\lambda = 0.001$.

Target networks. To train our Q-function, we use the technique from Fujimoto et al. (2018) in which we train two separate Q-networks with target networks and take the minimum over the two to compute the bootstrap value. The target networks are updated using a slowly moving average of the parameters after every batch of data:

$$\bar{\phi}_{i+1} = (1-\tau)\bar{\phi}_i + \tau\phi_i.$$

In our experiments, we used $\tau = 0.001$.

Automatic entropy tuning. We use the same technique as in Haarnoja et al. (2018b) to weight the rewards against the policy entropy term. Specifically, we pre-multiply the entropy term in

$$\hat{V}(\mathbf{s}', \mathbf{g}) \approx Q_{\bar{\phi}}(\mathbf{s}', \mathbf{a}', \mathbf{g}) - \log \pi(\mathbf{a}'|\mathbf{s}'; \mathbf{g}),$$

by a parameter α that is updated to ensure that the policy entropy is above a minimum threshold. The parameter α is updated by taking a gradient step on the following function with each batch of data:

$$\mathcal{F}_\alpha(\alpha) = -\alpha \left(\log \pi(\mathbf{a} \,|\, \mathbf{s}, \mathbf{g}) + \mathcal{H}_{\text{target}}\right)$$

where $\mathcal{H}_{\text{target}}$ is the target entropy of the policy. We follow the procedure in Haarnoja et al. (2018b) to choose $\mathcal{H}_{\text{target}}$ and choose $\mathcal{H}_{\text{target}} = -D_{\text{action}}$, where D_{action} is the dimension of the action space.

Exploration policy. Because ODAC is an off-policy algorithm, we are free to use any exploration policy. For the Ant and Sawyer tasks, we simply sample from the current policy. For the 2D Navigation task, at each time step, the policy takes a random action with probability 0.3.

Evaluation policy. For evaluation, we use the mean of the learned policy for selecting actions.

Hyperparameters. Table C.1 lists hyperparameters specific to each environment. Table C.2 lists the hyperparameters that were shared across the experiments.

Table C.1: Environment-specific hyperparameters.

Environment	Horizon	Q-function and policy network hidden units
Box 2D	100	[64, 64]
Ant	100	[400, 300]
Fetch Push	50	[64, 64]
Sawyer Push	100	[400, 300]
Sawyer Window	100	[400, 300]
Sawyer Faucet	100	[400, 300]

Table C.2: General hyperparameters used for all experiments.

Hyperparameter	**Value**
# training batches per environment step	1
batch size	256
discount factor	0.99
policy hidden activation	ReLU
Q-function hidden activation	ReLU
replay buffer size	1 million
hindsight relabeling strategy	future
hindsight relabeling probability	80%
target network update speed τ	0.001
reward scale update speed λ	0.001

C.3 Further Empirical Results

C.3.1 Further Ablation Study Results

We show the full ablation learning curves in Figure C.1. We see that ODAC consistently performs the best, and that ODAC with a fixed model also performs well. However, on a few tasks, and in particular the Fetch Push and Sawyer Faucet tasks, we see that using a fixed q_T hurts the performance, suggesting that our derived formula in Equation (5.35) results in better empirical performance.

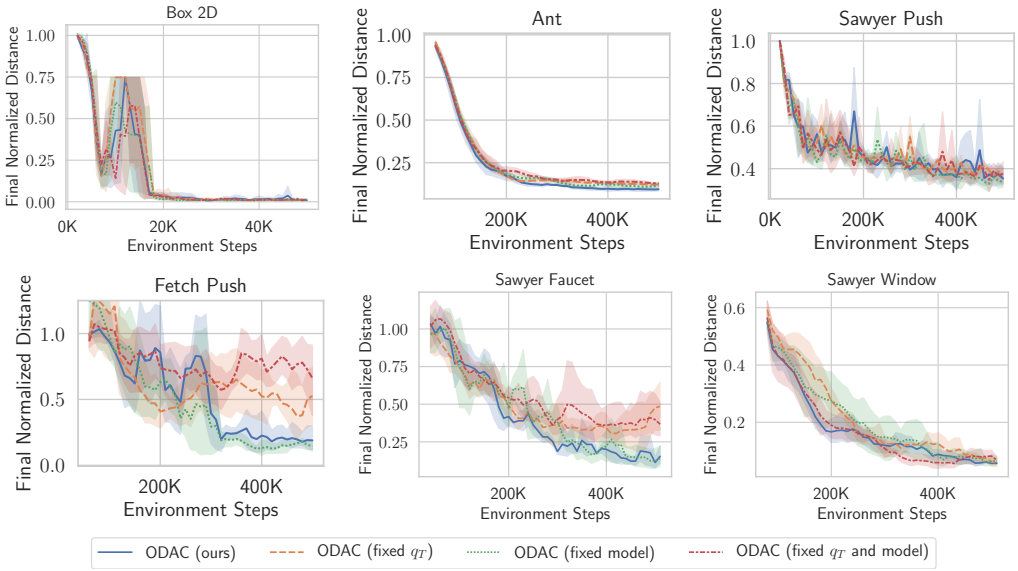


Figure C.1: Ablation results across environments. We see that using our derived q_T optimality equation is important for best performance across all six tasks and that ODAC is not sensitive to the quality of the dynamics model.

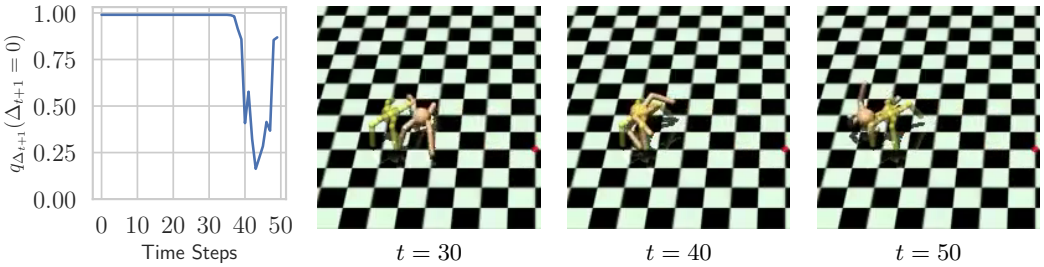


Figure C.2: The inferred $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ versus time during an example trajectory in the Ant environment. As the ant robot falls over, $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ drops in value. We see that the optimal posterior $q^{\star}_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ given in Proposition 5.3 automatically assigns a high likelihood of terminating when this irrecoverable state is first reached, effectively acting as a dynamic discount factor.

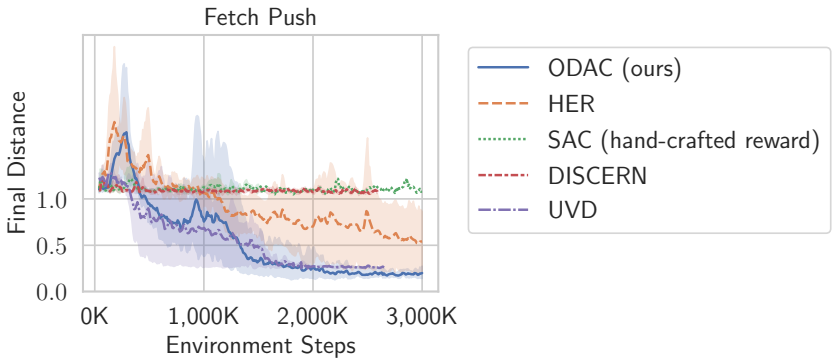


Figure C.3: Comparison of different methods for desired outcomes $\mathbf{g}$ sampled uniformly from the set of admissible states.

C.3.2 Comparisons under Oracle Goal Sampling

For exploration, Andrychowicz et al. (2017) explore the benefits of HER either using a single, fixed goal during exploration (see Section 4.3 of Andrychowicz et al. (2017)) or using oracle goal sampling, that is, during exploration, a new goal is sampled each episode from a uniform distribution over the set of all reachable goals in the environment. As such, oracle goal sampling requires knowledge of the environment to sample reachable goals. For example, in the 2D box experiment (Figure 5.1), points inside the gray block in the center are not reachable goal states, and this additional information must be available when performing oracle goal sampling.

To demonstrate the impact of sampling the desired outcome $\mathbf{g}$ during exploration, we evaluate ODAC and related methods on the Fetch task when using oracle goal sampling. As shown in Figure C.3, the performances of UVD and ODAC are similar and both outperform other methods. These results suggest that UVD depends more heavily on sampling outcomes from the set of desired outcomes than ODAC. The significant decrease in performance when the desired outcome $\mathbf{g}$ is fixed may be due to the fact that uniformly sampling $\mathbf{g}$ implicitly provides a curriculum for learning. For example, in the Box 2D environment, goal states sampled above the box can train the agent to move around the obstacle, making it easier to learn how to reach the other side of the box. Without this guidance, prior methods often "get stuck" on the other side of the box. In contrast, ODAC consistently performs well in this more challenging setting, suggesting that the log-likelihood signal provides good guidance to the policy.

C.3.3 Comparison to Model-Based Planning

ODAC learns a dynamics model but does not use it for planning and instead relies on the derived Bellman updates to obtain a policy. However, a natural question is whether or not the method would benefit from using this model to perform model-based planning, as in Janner et al. (2019). We assess this by comparing ODAC with a

model-based baseline that uses a one-step look-ahead. In particular, we follow the training procedure in Janner et al. (2019) with $k = 1$. To ensure a fair comparison, we use the exact same dynamics model architecture as in ODAC and match the update-to-environment step ratio to be 4-to-1 for both methods.

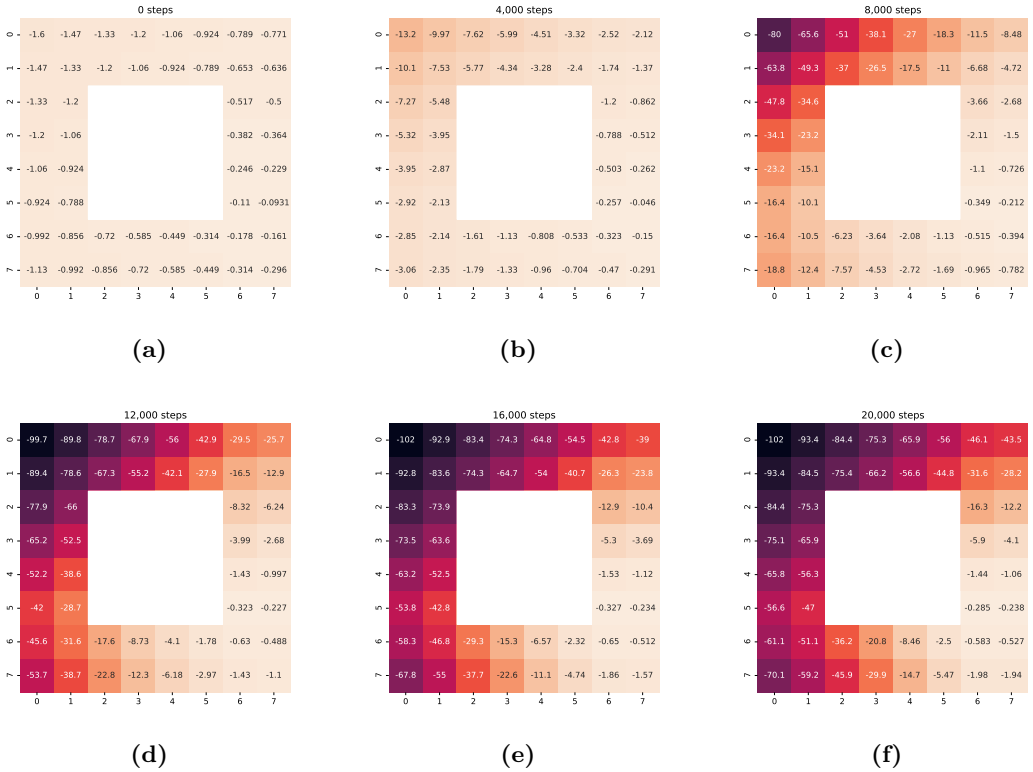
Table C.3 shows the final distance to the goal. Using the same dynamics model, ODAC, which does not use the dynamics model to perform planning and only uses it to compute rewards, outperforms the model-based planning method. While a better model might lead to better performance for the model-based baseline, these results suggest that ODAC is not sensitive to model quality to the same degree as model-based planning methods.

Table C.3: Normalized final distances (lower is better) across four random seeds, multiplied by a factor of 100.

Environment	ODAC (mean ± std. error)	Dyna (mean ± std. error)
Box 2D	0.74 (0.091)	0.87 (0.058)
Ant	33 (27)	102 (0.83)
Sawyer Faucet	14 (6.3)	100 (5)
Fetch Push	12 (3.7)	96 (3.8)
Sawyer Push	58 (8.7)	96 (0.39)
Sawyer Window	4.4 (1.5)	116 (14)

C.3.4 Reward Visualization

We visualize the reward for the Box 2D environment in Figure C.4. We see that over the course of training, the reward function initially flattens out near $\mathbf{g}$, making learning easier by encouraging the policy to focus on moving just out of the top left corner of the environment. Later in training (around 16,000 steps), the policy learns to move out of the top left corner, and we see that the reward changes to have a stronger reward gradient near $\mathbf{g}$. We also note that the rewards are much more negative for being far from $\mathbf{g}$ at the end of training: the top left region changes from having a penalty of -1.6 to -107. Overall, these visualizations show that the reward function automatically changes during training and provides a strong reward signal for different parts of the state space depending on the behavior of the policy.



(g) Learning Curve associated with Reward Visualized

(h) Environment Visualization

Figure C.4: We visualize the rewards over the course of training on a single random seed for the Box 2D environment. To visualize the reward, we discretize the continuous state space and evaluate $r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta)$ for $\mathbf{a} = \vec{0}$ at different states. As shown in Figure C.4h, the desired outcome $\mathbf{g}$ is near the bottom right, and the states in the center are invalid. After 4,000–8,000 environment steps, the reward is more flat near $\mathbf{g}$, and only provides a reward gradient far from $\mathbf{g}$. After 20 thousand environment steps, the reward gradient is much larger again near $\mathbf{g}$, and the penalty for being in the top left corner has changed from -1.6 to -107.