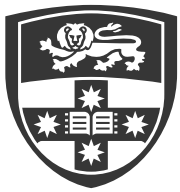
LOUIS CHI-CHUN TIAO

PROBABILISTIC MACHINE LEARNING IN THE AGE OF DEEP LEARNING: NEW PERSPECTIVES FOR GAUSSIAN PROCESSES, BAYESIAN OPTIMISATION AND BEYOND

PROBABILISTIC MACHINE LEARNING IN THE AGE OF DEEP LEARNING: NEW PERSPECTIVES FOR GAUSSIAN PROCESSES, BAYESIAN OPTIMISATION AND BEYOND

LOUIS CHI-CHUN TIAO



THE UNIVERSITY OF SYDNEY

School of Computer Science
Faculty of Engineering
The University of Sydney

A thesis submitted to fulfil requirements for the degree of Doctor of Philosophy

Louis Chi-Chun Tiao, *Probabilistic Machine Learning in the Age of Deep Learning:* New Perspectives for Gaussian Processes, Bayesian Optimisation and Beyond, Doctor of Philosophy (PhD) Thesis © Sep 2023

SUPERVISORS:
Fabio T. Ramos
Edwin V. Bonilla

DECLARATION

This is to certify that to the best of my knowledge, the content of this thesis is my own work. This thesis has not been submitted for any degree or other purposes.

I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

Sydney, Australia, Sep 2023

Louis Chi-Chun Tiao

For my parents

ACKNOWLEDGMENTS

This thesis embodies the culmination of years of research that led me on a profound personal and intellectual journey across several countries, connecting me with many remarkable people who have profoundly shaped my path. The constraints of brevity prevent me from fully expressing my gratitude to everyone in this limited space. I invite you to peruse the unabridged acknowledgments, where I more comprehensively extend my heartfelt appreciation to the collective of advisors, mentors, colleagues, friends, and family whose guidance, support, and motivation were integral throughout this arduous yet fulfilling process:

Within these pages, I'd like to gratefully acknowledge the Australian Government for funding my research through the Research Training Program (RTP) Scholarship Program and the Data61 detachment of the Commonwealth Scientific and Industrial Research Organisation (CSIRO) for contributing supplemental funding.

I'd also like to express my sincere appreciation to my supervisors, Fabio Ramos and Edwin Bonilla. I'm grateful to Fabio for cultivating my taste for high-impact problems and providing the latitude to explore independently while fostering a supportive framework for productivity and excellence. Working with Edwin has been a privilege, not only for his immense scientific acumen and deeply knowledgeable supervision but also for his genuine care and support. Over the years of our collaboration, it has been abundantly clear that Edwin always had my best interest at heart and a sincere desire to see me succeed. I'm deeply thankful for his boundless patience and steadfast faith in me during the moments of struggle and self-doubt that marked the "messy middle" of my journey. I can't imagine I was a joy to work with at all times, but in spite of this, I've never felt Edwin waiver in his belief in me, and this has made all the difference.

On a personal note, I'm grateful to my enduring better half, Jem, for her unending patience, sacrifice, and devotion through it all. She has taken on extra responsibilities so I could fully immerse myself in this body of work, all the while providing me care, laughter, comfort, and joy when I needed it most. In her selfless support, she has brought me quiet solace during storms, deep comfort in dark clouds, and abundant joy every step along the way. Any contributions or successes I have achieved through this PhD are as much a testament to her

efforts as my own. I also owe immense thanks to our sweet canine girl Penelope, whose gentle companionship, joyful antics, and affectionate cuddles have sustained me throughout this endeavor; I appreciate you more than you'll ever know. I could not have reached this milestone without these two precious souls by my side.

I'm grateful to my father, who instilled in me a passion for science and engineering from an early age. Whether we were tinkering with electronics or building models together, he nourished my curiosity and love of making things. I'll always remember with fondness all the times he took out of his demanding schedule as a clinical radiologist to tutor me in maths and physics, which provided me with a strong foundation for all my subsequent pursuits. His continued support over the years made this milestone possible.

I'm grateful to my mother – raising me alone in a new land must have taken tremendous courage. Watching her build a new life taught me resilience and determination. She faced every obstacle with poise and tenacity, showing me how to turn challenges into opportunities. Her tireless support and buoyant encouragement propelled me through my long education, spurring me onward at every turn. Our bond has blossomed into a cherished friendship where intellectual discussion flows freely alongside heartfelt life advice. I have learned extensively from her wisdom, compassion, and incisive intellect. Reaching this milestone would have been inconceivable without her years of selfless sacrifice and visionary guidance that set me firmly on my path.

It is with immeasurable gratitude and love that I dedicate this thesis to my parents.

The greatest enemy of knowledge is not ignorance,
it is the illusion of knowledge.

— Daniel J. Boorstin (often misattributed to Stephen Hawking)

To know that we know what we know,
and to know that we do not know what we do not know,
that is true knowledge.

— Nicolaus Copernicus

Real knowledge is to know the extent of one's ignorance.

— Confucius

ABSTRACT

Advances in artificial intelligence (AI) are rapidly transforming our world, with systems now matching or surpassing human capabilities in areas ranging from game-playing to scientific discovery. Much of this progress traces back to machine learning (ML), particularly deep learning and its ability to uncover meaningful patterns and representations in data. However, true intelligence in AI demands more than raw predictive power; it requires a principled approach to making decisions under uncertainty. This highlights the necessity of probabilistic ML, which offers a systematic framework for reasoning about the unknown in ML models through probability theory and Bayesian inference.

Gaussian processes (GPs) stand out as a quintessential probabilistic model, offering flexibility, data efficiency, and well-calibrated uncertainty estimates. They are integral to many sequential decision-making algorithms, notably Bayesian optimisation (BO), which has emerged as an indispensable tool for optimising expensive and complex black-box objective functions. While considerable efforts have focused on improving GP scalability, performance gaps persist in practice when compared against neural networks (NNs) due, in large, to their lack of representation learning capabilities. This, among other natural deficiencies of GPs, have hampered the capacity of BO to address critical real-world optimisation challenges.

This thesis aims to unlock the potential of deep learning within probabilistic methods and reciprocally lend probabilistic perspectives to deep learning. The key contributions are: (1) Extending orthogonally-decoupled sparse GP approximations to incorporate nonlinear NN activation layers as inter-domain features, mitigating the limitations of prior work and bringing predictive performance closer to NNs while retaining the advantages of GPs. (2) Framing cycle-consistent adversarial networks (CYCLEGANS) for unpaired image-to-image translation as variational inference (VI) in an implicit latent variable model, providing a Bayesian perspective on this powerful class of deep generative models. (3) Introducing a model-agnostic reformulation of BO based on binary classification that eliminates restrictions on the underlying representation of the objective function, enabling the seamless integration of flexible modelling paradigms like deep learning to tackle complex optimisation problems. By enriching the interplay between deep learning and probabilistic ML, this thesis advances the foundations of AI and facilitates the development of more capable and dependable automated decision-making systems.

PUBLICATIONS

Some of the material, including figures, tables, concepts, and ideas, presented in this thesis have previously appeared in the following works (co-)authored during the course of the PhD degree,

[1] Pantelis Elinas, Edwin V Bonilla, and Louis C Tiao. "Variational Inference for Graph Convolutional Networks in the Absence of Graph Data and Adversarial Settings". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 18648–18660. URL: https://bit.ly/elinas2020variational. **(Accepted as *Spotlight* presentation).**

[2] Aaron Klein, Louis C Tiao, Thibaut Lienart, Cedric Archambeau, and Matthias Seeger. "Model-based Asynchronous Hyperparameter and Neural Architecture Search". In: *arXiv preprint arXiv:2003.10865* (2020).

[3] Rafael Oliveira, Louis C Tiao, and Fabio T Ramos. "Batch Bayesian Optimisation via Density-Ratio Estimation with Guarantees". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 29816–29829. URL: https://bit.ly/oliveira2022batch.

[4] Louis C Tiao. "Numerical Methods for Improved Decoupled Sampling of Gaussian Processes". In: *Secondmind Labs Technical Reports* (2021).

[5] Louis C Tiao, Edwin V Bonilla, and Fabio T Ramos. "Cycle-Consistent Adversarial Learning as Approximate Bayesian Inference". In: *ICML 2018 Theoretical Foundations and Applications of Deep Generative Models*. Stockholm, Sweden, July 2018. **(Accepted as *Oral* presentation).**

[6] Louis C Tiao, Aaron Klein, Matthias W Seeger, Edwin V Bonilla, Cedric Archambeau, and Fabio T Ramos. "BORE: Bayesian Optimization by Density-Ratio Estimation". 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. 10289–10300. URL: https://proceedings.mlr.press/v139/tiao21a.html. **(Accepted as *Oral* presentation).**

[7] Louis C. Tiao, Vincent Dutordoir, and Victor Picheny. "Spherical Inducing Features for Orthogonally-Decoupled Gaussian Processes". In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, July 2023, pp. 34143–34160. URL: https://proceedings.mlr.press/v202/tiao23a.html. **(Accepted as *Oral* presentation).**

AUTHORSHIP ATTRIBUTION STATEMENT. The contents of this thesis relate more precisely to the works listed above as follows:

- Chapter 3 corresponds to the work published as [7], which was a continuation of the research initiated while a student researcher at Secondmind Labs in Cambridge, UK. I developed the methodology, including mathematical derivation and analyses, in conjunction with V. Dutordoir. I implemented the core methodology, carried out the experiments, analysed the empirical results, and wrote the drafts of the manuscript. V. Dutordoir provided source code and ongoing technical guidance for numerical calculations relating to the spherical harmonics in high dimensions and the Fourier analysis of the spherical NN activations. In addition, each co-author provided advisory and editorial contributions that helped refine the final manuscript.

- Chapter 4 corresponds to the work published as [5]. I developed the methodology, including mathematical derivation and analyses, in conjunction with E. Bonilla. I implemented the core methodology, carried out the experiments, analysed the empirical results, and wrote the drafts of the manuscript, which were refined through careful editing by E. Bonilla. Each co-author provided advisory and editorial contributions that helped refine the final manuscript.

- Chapter 5 corresponds to the work published as [6]. I developed the methodology with the help of A. Klein, who provided a continuous stream of invaluable suggestions since its early infancy, and M. Seeger, who contributed insightful mathematical derivations and analyses. I implemented the initial prototype of the method variant based on multi-layer perceptron (MLP) classifiers and carried out toy experiments on synthetic problems. A. Klein implemented the remaining variants based on tree-based ensembles and carried out large-scale experiments on a diverse range of challenging problems. Finally, I wrote the drafts of the manuscript, which were refined through careful editing by F. Ramos and C. Archambeau. Each co-author pro-

vided advisory and editorial contributions that helped refine the final manuscript.

- Appendix A corresponds to the ongoing work listed as [4], which was primarily carried out while a student researcher at Secondmind Labs. I developed the methodology at the suggestion of V. Dutordoir and V. Picheny. I implemented the methodology, carried out the experiments, analysed the empirical results, and wrote the drafts of the manuscript. In addition, each co-author provided advisory contributions that helped refine the manuscript.

- The work listed as [2] was jointly led by A. Klein and myself and initiated in my capacity as an Applied Science intern at Amazon Web Services in Berlin, Germany. The core methodology, including its mathematical derivations and analyses, was developed principally by M. Seeger, with contributions from A. Klein and myself. M. Seeger implemented early prototypes of the method, later refined by A. Klein and myself, who also carried out large-scale experiments on a diverse range of challenging problems. Since the conclusion of my internship, A. Klein, T. Lienart, and David Salinas have dedicated considerable effort to expanding the experimental analysis. In addition, each co-author provided advisory and editorial contributions that helped refine the final manuscript. While this work touches upon the central themes of this thesis, specifically in its development of multi-output GPs for asynchronous multi-fidelity BO (in turn for the tuning of deep learning models), this thesis' focus rather lies in devising new deep-learning-based approximation techniques for the GP and BO frameworks themselves. As such, while still relevant, this work is arguably tangential to the core topic and has thus been excluded.

- The work published as [1] was principally led by P. Elinas and E. Bonilla and initiated in my part-time capacity as a research software engineer at CSIRO's Data61 in Sydney, Australia. I had a role in implementing initial prototypes, carrying out toy experiments on synthetic problems, writing early drafts of the manuscript, and helping to refine directions and ideas. This work is relevant to the central themes of this thesis insofar as it develops a VI approach for an unwieldy class of probabilistic models, in particular, deep learning models of graph data that contain discrete hidden variables representing links. However, beyond the formative stages of this project, my contributions to this work were marginal, so it has been excluded from this thesis.

- The work published as [3] is a follow-up to [6], intending to address some of the questions left unresolved from the earlier work. It was principally led by R. Oliveira, and I had a role in implementing early prototypes, carrying out toy experiments on synthetic problems, and helping to refine directions and ideas. Beyond this, my contributions to this work were insignificant, so it has been excluded from this thesis.

In addition to the statements above, in cases where I am not the corresponding author of a published item, permission to include the published material has been granted by the corresponding author.

Sep 2023

Louis Chi-Chun Tiao

As supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Sep 2023

Fabio T. Ramos

CONTENTS

Mathematical Relations

$a \triangleq b$	a is equal to b by definition
$a \overset{D}{=} b$	a is equal to b in distribution
$a \propto b$	a is proportional to b, i. e., $a = \text{const} \cdot b$
$a \approx b$	a is approximately equal to b, i. e., $\|a - b\| < \epsilon$ for small real number $\epsilon > 0$

Numbers, Arrays & Sets

a	A scalar (integer or real)
$\mathbf{a}$	A vector
$\mathbf{A}$	A matrix
$\mathbf{0}_n, \mathbf{0}$	A vector of zeros of length n or implied by context
$\mathbf{I}_n, \mathbf{I}$	Identity matrix with n rows and columns or dimensionality implied by context
$\operatorname{diag} \mathbf{a}$	A square, diagonal matrix with diagonal entries given by $\mathbf{a}$
$\mathbb{N}, \mathbb{Z}$	The set of natural numbers and integers, respectively
$\mathbb{R}, \mathbb{C}$	The set of real and complex numbers, respectively
$\mathbb{R}^d$	The d-dimensional vector space of real numbers

Linear Algebra

$\mathbf{A}^\top, \mathbf{a}^\top$	Transpose of a matrix or vector
$\mathbf{A}^{-1}$	Inverse of square matrix
$\det \mathbf{A}$	Determinant of square matrix
$\operatorname{tr} \mathbf{A}$	Trace of square matrix
$\mathbf{A} \succeq 0$	Matrix $\mathbf{A}$ is positive semidefinite
$\mathbf{A}^{\frac{1}{2}}$	Square root of a matrix, specifically the Cholesky decomposition: a lower-triangular matrix $\mathbf{L}$ that satisfies $\mathbf{L}\mathbf{L}^\top = \mathbf{A}$

Functions & Functional Analysis

$f : \mathcal{X} \to \mathcal{Y}$	A function with domain $\mathcal{X}$ and range $\mathcal{Y}$
$f : \mathbf{x} \mapsto g(\mathbf{x})$	A function that maps x to $g(x)$; i. e., $f(\mathbf{x}) \triangleq g(\mathbf{x})$
$f \circ g$	Composition of functions f and g; $f \circ g : \mathbf{x} \mapsto f(g(\mathbf{x}))$

$\mathcal{O}(\cdot)$	Asymptotic upper bound ("big O"); $f(n) = \mathcal{O}(g(n))$ for $f, g : \mathbb{N} \to \mathbb{N}$ if $f(n)/g(n)$ is bounded as $n \to \infty$
$\mathbb{R}^{\mathcal{X}}$	The space of functions $f : \mathcal{X} \to \mathbb{R}$
$\mathcal{H}_k, \mathcal{H}$	Reproducing kernel Hilbert space associated with kernel k or implied by context
$\langle \cdot, \cdot \rangle_{\mathcal{H}}, \langle \cdot, \cdot \rangle$	Inner product associated with Hilbert space $\mathcal{H}$ or implied by context
$\| \cdot \|, \| \cdot \|_p$	L^2 norm of a vector; L^p norm if subscript p is specified

Calculus

$\frac{\mathrm{d}y}{\mathrm{d}x}$	Total derivative of y with respect to x
$\frac{\partial y}{\partial x}$	Partial derivative of y with respect to x
$\frac{\partial f}{\partial \mathbf{x}}$	Jacobian matrix $\mathbf{J} \in \mathbb{R}^{m \times n}$ of $f : \mathbb{R}^n \to \mathbb{R}^m$
$\int f(\mathbf{x}) \, \mathrm{d}\mathbf{x}$	Definite integral over the entire domain of $\mathbf{x}$
$\int_{\mathcal{X}} f(\mathbf{x}) \, \mathrm{d}\mathbf{x}$	Definite integral with respect to $\mathbf{x}$ over the set $\mathcal{X}$

Probability and Information Theory

$p(\mathbf{x}), q(\mathbf{x})$	A probability density, latter used to emphasise approximation
$\mathbf{x} \sim p(\mathbf{x})$	Random variable $\mathbf{x}$ is distributed according to $p(\mathbf{x})$
$\mathbb{E}_{p(\mathbf{x})}[f(\mathbf{x})], \mathbb{E}[f(\mathbf{x})]$	Expectation of $f(\mathbf{x})$ under $p(\mathbf{x})$ or implied by context
$\mathrm{Cov}(\cdot, \cdot)$	Covariance between random variables
$\mathbb{H}[\cdot]$	Shannon entropy of a random variable
$\mathcal{D}_f[p \parallel q]$	f-divergence between distributions with densities p and q
$\mathcal{D}_{\mathrm{KL}}[p \parallel q], \mathrm{KL}[p \parallel q]$	Kullback-Leibler divergence between distributions with densities p and q
$\mathcal{U}[a, b]$	Uniform distribution with lower and upper bounds a and b
$\mathrm{Bern}(\rho)$	Bernoulli distribution with parameter ρ
$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}), \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate Gaussian distribution (on $\mathbf{x}$) with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$
$\mathcal{GP}(f; m, k)$	Gaussian process; $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ denotes $f(\mathbf{x})$ is a distributed as a Gaussian process with mean function m and covariance function (kernel) k
δ_{ij}	Kronecker delta; $\delta_{ij} = 1$ iff $i = j$ and 0 otherwise
$\delta(x - x_0)$	Dirac delta on x with point mass at x_0

Optimisation

$f^* = \min_{\mathbf{x}} f(\mathbf{x})$	A minimum of function $f(\mathbf{x})$

$\mathbf{x}^* = \arg\min_{\mathbf{x}} f(\mathbf{x})$	A minimiser of function $f(\mathbf{x})$

Special Functions

$\sigma(x)$	Sigmoid function, typically the logistic sigmoid $x \mapsto (1 + \exp(-x))^{-1}$
$\textsc{relu}(x)$	Rectified linear unit activation; positive part of x, i. e., $x \mapsto \max(0, x)$
$\text{Softplus}(x)$	Softplus activation; $x \mapsto \log(1 + \exp(x))$

1 INTRODUCTION

Artificial intelligence (AI) stands poised to be among the most disruptive technologies of our era. The breakneck pace of recent AI advancements has been spearheaded by machine learning (ML), particularly the resurgence of *deep learning*. Deep learning is as old as the first general-purpose electronic computer; with roots tracing back to the 1940s and '50s [169, 219], the revival of deep learning, beginning in the early 2010s, was catalysed by a series of breakthroughs that shattered previously perceived limitations and captivated the collective imagination. These breakthroughs span various domains, including computer vision [84, 133, 211, 217], speech recognition [87, 103], natural language processing [21, 274], protein folding [121], generative art and artificial creativity [86, 104, 208, 215], as well as reinforcement learning for robotics control [147, 175] and achieving superhuman-level gameplay [174, 232].

machine learning
deep learning

Nevertheless, it is crucial to view these developments as means to an ultimate end rather than an end in themselves. Arguably, the true pinnacle of AI's capabilities lies in optimal *decision-making*, whether that entails offering analyses and insights to aid humans in making better decisions or completely automating the decision-making process altogether. Practically any task directed towards a well-defined objective can be boiled down to a cascade of decisions. At a fundamental level, operating a vehicle involves a continuous stream of decisions involving accelerating, braking, and turning. Financial trading revolves around decisions to buy, sell, or hold various assets. Even complex engineering tasks, such as designing an aerofoil, involve a sequence of decisions about adjusting design variables to achieve desirable aerodynamic characteristics.

decision-making systems

Yet, the intricacies of decision-making surpass what any single advancement in deep learning can address. While convolutional neural networks (CNNs) can facilitate object detection tasks in autonomous vehicles, recurrent neural networks (RNNs) can aid in forecasting market dynamics for systematic trading, and physics-informed NNs can assist in predicting aerodynamic effects, it remains the case that no target or quantity of interest can be entirely known or predictable (indeed, if they were, the pursuit of predictive modelling and ML would be superfluous). Instead, predictions often prove unreliable, or at best, *uncertain*, due to the limitations of our knowledge and the complexity and variability inherent in the underlying real-world processes. The impressive power of deep learning models often overshadows their ignorance of the limits of their own knowledge and the extent of

uncertainty, epistemic
uncertainty, aleatoric

uncertainty in their predictions. When these predictions are integrated into a sequential decision-making framework, such uncertainty can amplify, compound, and lead to catastrophic consequences. In the context of aeronautical engineering, this could result in inefficient designs; in quantitative finance, it can lead to devastating capital losses; and in autonomous driving, it can even cost lives.

Bayesian statistics
probabilistic machine learning

PROBABILISTIC MACHINE LEARNING. Grounded in the laws of probability and Bayesian statistics [10, 138], *probabilistic* ML provides a consistent framework for systematically reasoning about the unknown. The probabilistic approach to ML acknowledges that the real world is fraught with uncertainty and embraces this uncertainty as an inherent part of decision-making. Unlike traditional methods, including those of deep learning, it recognises model predictions not as absolute truths that can be represented as single *point estimates* produced from a deterministic mapping, but as full *probability distributions* that capture the potential outcomes of a random variable as it propagates through some underlying data-generating process. In a *probabilistic model*, all quantities are treated as random variables governed by probability distributions – the data are treated as observed variables, which are influenced by some underlying hidden variables, e. g., the model parameters. A prior distribution is used to express reasonable values for these hidden variables and to eliminate implausible ones. The relationship between observed and hidden variables is described using the likelihood, and the process of Bayesian inference amounts to calculating, using basic laws of probability, a posterior distribution over the hidden factors conditioned on the observed data, which can be seen as a refinement of the prior beliefs in light of new evidence. While the posterior distribution can be useful in and of itself, its primary role lies in facilitating subsequent prediction and decision-making by providing full probability distributions over predicted outcomes. This capability allows the decision-maker to assess the range of possible scenarios and their associated probabilities, enabling a more nuanced understanding of uncertainty and risk, which is indispensable in complex, dynamic environments where the repercussions of incorrect decisions can be severe. In essence, probabilistic ML equips autonomous decision-making systems with a probabilistic worldview, enabling them to navigate ambiguity and make sound decisions in the face of imperfect information.

point estimate

random variable
probabilistic model

PROBABILISTIC ML VS. DEEP LEARNING. While deep learning has dominated recent AI advances, probabilistic ML remains as important as ever and continues to offer valuable tools for addressing AI challenges that can not be fully resolved by deep learning alone. Although both approaches can be combined to create hybrid methods that leverage their respective strengths, some defining characteristics have

traditionally set deep learning apart from probabilistic ML. Perhaps most notably, probabilistic ML approaches can achieve remarkable predictive performance even when data is scarce. In contrast, deep learning models tend to be data-intensive by nature, often demanding datasets of a scale proportional to their size (i.e., their parameter count) [106], which has seen explosive growth in recent years [3, 194, 205, 231, 266]. With that being said, inference in many probabilistic models poses computational problems that are difficult to scale. On the other hand, deep learning approaches have excelled in scalability, a key factor contributing to their widespread success. This scalability is bolstered by their compatibility with various speed-enhancing mechanisms such as stochastic optimisation, specialised hardware accelerators (GPUs and TPUs), as well as distributed and/or cloud-based computing infrastructure. To bridge this gap, substantial research effort has been devoted to enabling probabilistic ML to benefit from these advantages through optimisation-based approximations to Bayesian inference [118].

Moreover, as mentioned earlier, these paradigms are by no means mutually exclusive. Indeed, it is often possible to directly extend existing models with a Bayesian treatment of their parameters, adding a layer of probabilistic reasoning to the model, and allowing it to not only make predictions but also estimate the uncertainty associated with those predictions. An excellent example is the Bayesian neural network (BNN), which treats the weights as hidden variables and leverages posterior inference to provide predictions while estimating associated uncertainties, delivering a more robust and principled approach to deep learning [18, 154, 185].

Bayesian neural network

The Bayesian formalism naturally gives rise to many popular methods and paradigms, often in the form of point estimates or other kinds of approximations. The quintessential example of this is found in linear regression, in particular, in ridge and lasso regression [260], which correspond variously to maximum *a posteriori* (MAP) estimates in Bayesian linear regression (BLR) models with prior distributions possessing different sparsity-inducing characteristics [82] – more broadly, mitigations against over-fitting tend to arise organically in Bayesian methods, which is why they are frequently characterised as being fundamentally more robust against over-fitting [286, §5.2]. Likewise, the once *à la mode* support vector machines (SVMs) can be seen as MAP estimates for a class of nonparametric Bayesian models [195], dropout [246] in NNs can be seen as a variational approximation to exact inference in BNNs [74], and unsupervised learning methods such as factor analysis (FA) [242] and principal component analysis (PCA) [198] are instances of a class of latent variable models (LVMs) [8, 261] known as linear-Gaussian factor models [221], to name just a few examples. Time and again, classical approaches have not only benefitted from being viewed through the Bayesian perspective but

maximum a posteriori

have also been enriched and redefined by the depth of insights this framework provides.

1.1 THESIS GOALS

The over-arching goal of this thesis is to continue advancing the integration and cross-pollination between deep learning and probabilistic ML. We aim to further the interplay between these two fields, both by incorporating probabilistic interpretations and uncertainty quantification into popular deep learning frameworks, and by leveraging the representational power of deep NNs to improve established Bayesian methods. This dual-pronged approach provides fresh perspectives and taps the complementary strengths of both paradigms, advancing the foundations of AI and facilitating the development of more capable and dependable decision support frameworks. Ultimately, we strive to unlock the potential of deep learning within high-impact probabilistic ML methodologies, and to lend useful Bayesian perspectives on current deep learning techniques.

GAUSSIAN PROCESS MODELS. Arguably, no family of probabilistic models embodies the ethos of probabilistic ML and illustrates its nuances and parallels with deep learning quite like the Gaussian process (GP). Accordingly, they shall occupy a prominent place in our thesis. In particular, GPs stand out as the ideal choice when dealing with limited data, offer the flexibility to encode prior beliefs through the covariance function, and provide predictive uncertainty estimates with a fine calibration that is second to none. Conversely, they are challenging to scale to large datasets, a limitation that has spurred extensive research and development efforts. Furthermore, in contrast to deep learning models, which are often lauded for their ability to automatically uncover valuable patterns and features in data, GPs have at times been dismissed as unsophisticated smoothing mechanisms [157]. Despite these apparent disparities, GPs are intricately connected to NNs in numerous ways. Among these, one of the most classical and well-known relationships is the convergence of single-layer NNs with randomly initialised weights toward GPs in the infinite-width limit [185]. Similar links have also been identified between GPs and infinitely wide *deep* NNs [143, 166].

Gaussian process

In an effort to elevate the representational capabilities of GPs to a level comparable with deep NNs, deep GPs (DGPs) [49] stack together multiple layers of GPs. Additional efforts to construct efficient sparse GP approximations have leveraged the advantageous properties of computations on the hypersphere [65], which has led to DGP models in which the propagation of posterior predictive means is equivalent to a forward pass through a deep NN [66, 252]. Notably, as a side effect, this model effectively provides uncertainty estimates for deep NN

through its predictive variance. Among the contributions of our thesis is the further development of this framework, integrating cutting-edge techniques [223, 230] to address some of its practical limitations, thereby narrowing the performance gap between GPs and deep NNs.

Probabilistic models, serving a crucial role as decision support tools, routinely aid scientific discovery in fields such as physics and astronomy, guiding advancements in areas of medicine and healthcare encompassing bioinformatics, epidemiology, and medical diagnosis. Beyond that, these models have wide-ranging applications in economics, econometrics, and the social sciences. Moreover, they are indispensable in various engineering disciplines, such as robotics and environmental engineering. Among the many probabilistic models, GPs stand out as a powerful driving force behind a number of important sequential decision-making frameworks, including active learning [108] and reinforcement learning [55], and the broader area of probabilistic numerics at large [95]. Notably, Bayesian optimisation (BO) [20, 75, 228] is one major area that relies heavily on GPs and will feature extensively in our thesis.

Bayesian optimisation

BAYESIAN OPTIMISATION. Bayesian optimisation (BO) is a powerful methodology dedicated to the global optimisation of complex and resource-intensive objective functions. In contrast to classical optimisation methods, BO excels even when dealing with functions that lack strong assumptions or guarantees. These functions may not be convex, possess no gradients, lack a well-defined mathematical form, and observable only indirectly through noisy measurements.

black-box function

At its core, BO is a sequential decision-making algorithm. It relies on observations from past function evaluations to determine the next candidate location for evaluation in pursuit of optimal solutions. BO leverages a probabilistic model, often a GP, to represent its knowledge and beliefs about the unknown function. This model is continuously updated with the acquisition of each new observation, enabling the algorithm to adapt its behaviour and make sound decisions based on the evolving information.

sequential decision-making algorithm

BO effectively manages uncertainty inherent in such sequential decision-making processes by making use of the probabilistic model to the fullest, harnessing the entire predictive distribution, particularly, the predictive uncertainty, to select promising candidate solutions that bring the most value to the optimisation process. This generally consists not merely of those most likely to optimise the objective function (i. e., *exploiting* that which is known), but also those likely to reveal the most knowledge and information about the function itself (i. e., *exploring* that which remains unknown).

exploitation

exploration

This pronounced emphasis on well-calibrated uncertainty distinguishes BO as one of the standout "killer apps" for GPs and a jewel in the crown of probabilistic ML applications. In practice, BO has proven

instrumental across science, engineering, and industry, where efficiency and cost-effectiveness are paramount. Its applications include protein engineering [216, 295], material discovery [227], experimental physics (e. g., experiments involving ultra-cold atoms [282] and free-electron lasers [63]), environmental monitoring (sensor placement) [76, 163], and the design of aerodynamic aerofoils [70, 137], integrated circuits [153, 265], broadband high-efficiency power amplifiers [32], and fast-charging protocols for lithium-ion batteries [4]. Notably, it has played a crucial role in automating the hyperparameter tuning of various ML models [236, 270], especially deep learning models, thus representing yet another way in which probabilistic ML has contributed to the advancement of deep learning.

hyperparameter optimisation

automated machine learning

However, GPs are not universally suitable for all BO problem scenarios. They are most effective when dealing with smooth, stationary functions with homoscedastic noise and a relatively modest input dimensionality. Additionally, GPs are easiest to work with for functions with a single output and purely continuous inputs. While a surprisingly wide array of real-world challenges satisfy these conditions, many high-impact problems, such as *de novo* molecular design, which involves sequential inputs; neural architecture search (NAS), which involves structured inputs with intricate conditional dependencies; and automotive safety engineering, which involve numerous constraints and multiple objectives, clearly fall outside of this scope. This is not to say that GPs cannot be extended to such challenging scenarios. However, such extensions almost always come at a cost. Consequently, it makes sense to appeal to alternative modelling paradigms more naturally suited to specific tasks, e. g., employing random forests (RFs) to handle discrete and structured inputs, or deep NNs for capturing nonstationary behaviour and dealing with multiple objectives. A major contribution of this thesis is the introduction of a new formulation of BO that seamlessly accommodates virtually any modelling paradigm, including deep learning, without any compromise.

1.2 THESIS OVERVIEW

The core contributions of our thesis are summarised as follows:

1. We improve upon the framework for sparse hyperspherical GP approximations that employ nonlinear activations as inter-domain inducing features. This framework serves as a bridge between GPs and NNs, with posterior predictive mean taking the form of single-layer feedforward NNs. Our thesis examines some practical issues associated with this approach and proposes an extension that takes advantage of the orthogonal decoupling of GPs to mitigate these limitations. In particular, we introduce spherical inter-domain features to construct more flexible data-dependent basis functions for both the principal and orthogonal

components of the GP approximation. We demonstrate that incorporating orthogonal inducing variables under this framework not only alleviates these shortcomings but also offers superior scalability compared to alternative strategies.

2. We provide a probabilistic perspective on CYCLEGANS, a cutting-edge deep generative model for style transfer and image-to-image translation. Specifically, we frame the problem of learning cross-domain correspondences without paired data as Bayesian inference in a LVM, in which the goal is to uncover the hidden representations of entities from one domain as entities in another. First, we introduce implicit LVMS, which allow flexible prior specification over latent representations as implicit distributions. Next, we develop a new VI framework that minimises a symmetrised statistical divergence between the variational and true joint distributions. Finally, we show that CYCLEGANS emerge as a closely-related variant of our framework, providing a useful interpretation as a Bayesian approximation.

3. We introduce a model-agnostic formulation of BO based on classification. Building on the established links between class-probability estimation (CPE), density-ratio estimation (DRE), and the improvement-based acquisition functions, we reformulate the acquisition function as a binary classifier over candidate solutions. This approach eliminates the need for an explicit probabilistic model of the objective function and casts aside the limitations of tractability constraints. As a result, our model-agnostic BO approach substantially broadens its applicability across diverse problem scenarios, accommodating flexible and scalable modelling paradigms such as deep learning without necessitating approximations or sacrificing expressive and representational capacity.

Accordingly, our thesis is organised as follows:

- Chapter 2 lays the necessary groundwork for our thesis. We begin by outlining the fundamental principles of probability and Bayesian statistics, which form the basis of probabilistic ML. Additionally, we introduce the widely-adopted method of approximate Bayesian inference known as VI. Our discussion underscores the central role played by statistical divergences, prompting us to delve into a larger family of divergences and motivating our discussion of DRE. With a solid foundation in place, we shift our focus to GPS, providing an introductory overview and highlighting the most commonly-used sparse approximations. Finally, we conclude this background chapter by introducing the basic concepts behind BO.

- Chapter 3 examines orthogonally-decoupled sparse GPs with spherical NN activation features, as summarised in 1 above.
- Chapter 4 examines cycle-consistent adversarial networks from the perspective of approximate Bayesian inference, as summarised in 2 above.
- Chapter 5 examines our model-agnostic approach to BO based on binary classification and DRE, as summarised in 3 above.
- Chapter 6 brings this thesis to a close by reflecting on our main contributions and situating them in the broader landscape of probabilistic methods in ML. Finally, we conclude by presenting our outlook on the avenues for future research and development in this rapidly evolving field.

2 BACKGROUND

2.1 PROBABILISTIC MACHINE LEARNING

Probabilistic models have become pillars of modern ML. They are at the core of powerful frameworks that can uncover hidden structures, learn useful representations, and efficiently utilise them to make accurate predictions or generate realistic samples. Through the formalism of probability theory and Bayesian inference, probabilistic models provide a coherent framework for systematically reasoning about the unknown. Such a framework possesses notable advantages: it can quantify uncertainty in predictions, naturally handle missing data, and avoid over-fitting to spurious patterns. The probabilistic approach to ML is deeply embedded in many of its most impactful applications today.

In a probabilistic model, all quantities are treated as random variables – the data is treated as *observed*, or, *known*, variables, which are assumed to be governed by some underlying *hidden*, *latent*, or, *unknown* variables. Let $\mathcal{D}$ be the set of observed variables and $\mathcal{H}$ the set of hidden variables, with the joint density

variables, observed

variables, latent

joint density

$$p(\mathcal{D}, \mathcal{H}) = p(\mathcal{D} \mid \mathcal{H}) p(\mathcal{H}).$$

Notably, the distribution of the observed variables is assumed to be governed by the hidden variables. In particular, a *prior* density $p(\mathcal{H})$ is placed on the hidden variables $\mathcal{H}$, reflecting the beliefs about its plausible values, and to rule out absurd values that should not be entertained. Its relationship to the observed variables $\mathcal{D}$ is then defined through the *likelihood function*, or, simply, *likelihood*, $p(\mathcal{D} \mid \mathcal{H})$. Note this conditional is sometimes also referred to as the *observational model*. Now, the problem of inference in Bayesian models amounts to computing the *posterior* density $p(\mathcal{H} \mid \mathcal{D})$, the conditional probability of the hidden factors given the observed data. By Bayes' theorem,

prior

likelihood

observational model

posterior

Bayes' theorem

$$p(\mathcal{H} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \mathcal{H}) p(\mathcal{H})}{p(\mathcal{D})}. \tag{2.1}$$

The posterior can be seen as a refinement of the prior beliefs in light of observed data. In the Bayesian learning framework, the posterior can be updated iteratively as more data or new evidence becomes available.

To compute the conditional in Equation (2.1) exactly, one must compute the denominator $p(\mathcal{D})$, often referred to as the model *evidence*, It is also known as the *marginal likelihood*, since it is obtained by

evidence

marginal likelihood

marginalising out the hidden variables from the joint density,

$$p(\mathcal{D}) = \int p(\mathcal{D} \mid \mathcal{H}) p(\mathcal{H}) \, \mathrm{d}\mathcal{H}. \tag{2.2}$$

The posterior distribution in Equation (2.1) may be useful in and of itself, but is most commonly used downstream in a number of ways, e. g., for decision-making, or as the new prior as additional data arrives, or to make predictions on unseen data $\mathcal{D}_*$,

posterior predictive

$$p(\mathcal{D}_* \mid \mathcal{D}) = \int p(\mathcal{D}_* \mid \mathcal{H}) p(\mathcal{H} \mid \mathcal{D}) \, \mathrm{d}\mathcal{H}.$$

Despite its conceptual simplicity, exact Bayesian inference is often fraught with intractabilities. Specifically, computing the evidence integral in Equation (2.2) proves to be a frequent source of difficulties for many model families. This computation can exhibit exponential time complexity, rendering it *computationally* intractable. Even with the advanced hardware available today, an unassuming polynomial time complexity is still considered computationally intractable when dealing with sufficiently large datasets. For example, as of the current writing, algorithms with a cost of $\mathcal{O}(N^3)$ are typically deemed prohibitively slow when N is on the modest order of thousands [99, 277]. Moreover, in many cases, this integral doesn't even have a closed-form expression (e. g., due to non-conjugacy), rendering it *analytically* intractable. Consequently, the accurate and efficient evaluation of the evidence integral stands as a paramount challenge when performing Bayesian inference for the vast array of complex models that dominate modern probabilistic ML.

intractability, computational

intractability, analytical

When it is not feasible to carry out exact inference, one must instead resort to approximate inference techniques. Some dominant forms of approximate inference include the Laplace approximation [155], expectation propagation (EP) [173], sampling-based approaches such as Markov chain Monte Carlo (MCMC) [187], or optimisation-based approaches such as VI [118, 276]. In this thesis, we shall focus on VI, which turns out to be a common thread that weaves together a number of seemingly disaparate research topics.

approximate inference

2.2 VARIATIONAL INFERENCE

The basic idea of variational inference (VI) is to cast inference as an optimisation problem [17]. We first specify a family $\mathcal{Q}$ of densities over the latent variables. Each member $q \in \mathcal{Q}$ is a candidate approximation to the exact posterior $p(\mathcal{H} \mid \mathcal{D})$. We then optimise over this family to find that member that minimises the Kullback–Leibler (KL) divergence to the exact posterior,

Kullback–Leibler divergence

$$q^*(\mathcal{H}) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}\left[q(\mathcal{H}) \parallel p(\mathcal{H} \mid \mathcal{D})\right]. \tag{2.3}$$

Having found the optimal approximate density $q^*(\mathcal{H})$, it can then be used as a subtitute for the exact posterior density. However, a difficulty remains – explicitly spelling out the KL divergence in Equation (2.3) reveals its dependence on $p(\mathcal{D})$, the model evidence from Equation (2.2),

$$\begin{aligned}\mathrm{KL}\left[q(\mathcal{H}) \parallel p(\mathcal{H} \mid \mathcal{D})\right] &\triangleq \mathbb{E}_{q(\mathcal{H})}\left[\log \frac{q(\mathcal{H})}{p(\mathcal{H} \mid \mathcal{D})}\right] = \mathbb{E}_{q(\mathcal{H})}\left[\log \frac{p(\mathcal{D})q(\mathcal{H})}{p(\mathcal{D}, \mathcal{H})}\right] \\ &= \log p(\mathcal{D}) + \mathbb{E}_{q(\mathcal{H})}\left[\log \frac{q(\mathcal{H})}{p(\mathcal{D}, \mathcal{H})}\right] \qquad (2.4)\end{aligned}$$

However, let's not forget that the intractability of the evidence is the *raison d'être* of approximate inference in the first place. Clearly, directly minimising the KL is infeasible, prompting the need to consider an alternative strategy.

2.2.1 *Evidence Lower Bound*

evidence lower bound

This brings us to the well-known evidence lower bound (ELBO) objective, which is defined as

$$\mathrm{ELBO}(q) \triangleq \mathbb{E}_{q(\mathcal{H})}\left[\log \frac{p(\mathcal{D}, \mathcal{H})}{q(\mathcal{H})}\right]. \qquad (2.5)$$

Crucially, as the name suggests, the ELBO is a lower bound on the model evidence. In particular, adding $\mathrm{ELBO}(q)$ to both sides of Equation (2.4), we get

$$\log p(\mathcal{D}) = \mathrm{ELBO}(q) + \mathrm{KL}\left[q(\mathcal{H}) \parallel p(\mathcal{H} \mid \mathcal{D})\right].$$

Hence, the ELBO consists of the negative KL divergence and the log marginal likelihood, which is a constant wrt $q(\mathcal{H})$. Thus seen, maximising the ELBO is equivalent to minimising the KL divergence in Equation (2.3). Moreover, since the KL divergence is nonnegative, $\mathrm{KL}\left[\cdot \parallel \cdot\right] \geq 0$, it further follows that the ELBO is a lower bound on the log marginal likelihood, $\log p(\mathcal{D}) \geq \mathrm{ELBO}(q)$, for any $q \in \mathcal{Q}$. This bound can also be derived using Jensen's inequality, as originally shown by Jordan et al. [118].

We can expand the ELBO as

$$\mathrm{ELBO}(q) = \mathbb{E}_{q(\mathcal{H})}[\log p(\mathcal{D} \mid \mathcal{H})] - \mathrm{KL}\left[q(\mathcal{H}) \parallel p(\mathcal{H})\right]. \qquad (2.6)$$

The first term in Equation (2.6) is commonly referred to as the expected log-likelihood (ELL), while the second term is the negative KL between the approximate posterior $q(\mathcal{H})$ and prior $p(\mathcal{H})$. The ELL term encourages the approximate density to place its mass on configurations of the latent variables that explain the observed data, while the negative KL divergence term encourages densities that resemble the prior. Combined, these terms constitute the ELBO and reflect the

expected log-likelihood

usual balance between the likelihood and prior – and between data fit and regularisation.

Under benign conditions, the solution $q^*(\mathcal{H})$ to the optimisation problem outlined in Equation (2.3) can be derived analytically. An illustrative example of this is found in GPs, a widely used family of models that we shall formally introduce in Section 2.4. Specifically, in the sparse GP regression (SGPR) framework discussed in Section 2.4.2.3, the optimal $q^*(\mathcal{H})$ has a closed-form expression. However, in most cases, $q^*(\mathcal{H})$ is obtained through a hill-climbing optimisation procedure, specifically, gradient ascent, applied to an analytical form of the ELBO. This approach is employed in the more general sparse variational GP (SVGP) framework, presented in Section 2.4.2.1, where the likelihood is not (necessarily) Gaussian. If the likelihood factorises, the use of mini-batch training for stochastic optimisation [105], as explained in Section 2.4.2.2, allows for scaling to massive datasets.

More generally, in other scenarios, such as VI with blackbox likelihoods [209], discrete hidden variables [114, 158], or implicit distributions [110, 267], one or more components of the ELBO may lack analytical tractability and thus necessitate further approximations. Chapter 3 of this thesis focuses on improving inference in the SVGP framework through the use of NN basis functions, Chapter 4 examines a new kind of VI scheme designed to handle implicit distributions, and Appendix A explores the efficient posterior sampling of GPs and their sparse variational approximations.

For a complete resource on the foundations of VI, we refer the interested reader to the review article of Blei, Kucukelbir, and McAuliffe [17], now a contemporary classic.

2.3 STATISTICAL DIVERGENCES AND DENSITY-RATIO ESTIMATION

Statistical divergences quantify the dissimilarity between probability distributions and are essential in probabilistic ML. In the preceding section on variational inference, we saw a prime example of one such divergence, namely, the well-known KL divergence. In fact, the KL divergence is just one of many divergences that belong to a larger family of statistical divergences known as the f-divergences [48, 146], also known as the Ali-Silvey distances [2]. For a convex, lower-semicontinuous function $f : \mathbb{R}_+ \to \mathbb{R}$ satisfying $f(1) = 0$, the f-divergence between two distributions with probability densities $p(\mathbf{x})$ and $q(\mathbf{x})$ is defined as

f-divergence

$$\mathcal{D}_f\left[p(\mathbf{x}) \parallel q(\mathbf{x})\right] \triangleq \mathbb{E}_{q(\mathbf{x})}\left[f\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right)\right]. \tag{2.7}$$

For instance, the familiar KL divergence that appears extensively in VI – more precisely, the *reverse*[1] KL divergence $\mathrm{KL}\left[q \parallel p\right]$ – is obtained as a special case of Equation (2.7) under the setting $f : u \mapsto -\log u$. At the heart of Equation (2.7) is the fraction, or, *ratio*,

$$r(\mathbf{x}) \triangleq \frac{p(\mathbf{x})}{q(\mathbf{x})} \tag{2.8}$$

with density $p(\mathbf{x})$ as the numerator and $q(\mathbf{x})$ as the denominator. This crucial quantity is referred to as the *density-ratio* of $p(\mathbf{x})$ and $q(\mathbf{x})$. The density-ratio is known variously in other parts of the literature as the "likelihood ratio," or the "importance weight". Clearly, when either one or both of the densities are unavailable in analytical form, either due to intractabilities or intentional modelling choices, the f-divergence from Equation (2.7) will also be analytically intractable. Perhaps the most notable case of such intractabilities is in the framework of generative adversarial networks (GANs) [86], in which the underlying goal is to minimise some f-divergence between two distributions where neither admits a tractable density, and one must therefore rely solely on their samples [177, 191].

density-ratio

More broadly, the problem of DRE is concerned with approximating density-ratios when no information is available from distributions p or q other than their samples. The DRE problem is pervasive throughout ML and arises in a impressively diverse range of contexts, e. g., in covariate shift adaptation [15, 250, 268], energy-based models (EBMS) [90, 92, 269], VI [110, 172], likelihood-free inference [64, 257, 267], mutual information estimation [11], bias-correction for generative models [39, 91], and Bayesian experimental design (BED) [129, 130]. Chapter 5 of this thesis demonstrates how DRE arises in the context of BO [241, 259], a close cousin of BED. Furthermore, as alluded to earlier, Chapter 4 discusses a novel VI approach that relies heavily on DRE to deal with implicit distributions.

The most obvious but naïve approach to tackling the DRE problem is to separately estimate the densities $p(\mathbf{x})$ and $q(\mathbf{x})$ using, e. g., kernel density estimation (KDE) [233], and then to use their ratio as an approximation to the unknown true density-ratio. Not surprisingly, this approach suffers from a large host of issues, most of which are well-documented by Sugiyama, Suzuki, and Kanamori [251]. We discuss these at further length in Chapter 5, with an added emphasis on the drawbacks that most impact applications in global optimisation.

Not surprisingly, there is a substantial body of existing works on DRE [251]. Recognising the deficiencies of the naïve KDE approach, a myriad alternatives have since been proposed, including KL importance estimation procedure (KLIEP) [250], kernel mean matching (KMM) [88], unconstrained least-squares importance fitting (ULSIF) [122], and relative ULSIF (RULSIF) [292]. In this thesis, we shall primarily focus

1 the KL divergence is asymmetric

on CPE, introduced in Section 2.3.2, an effective and versatile approach that has found widespread adoption in a diverse range of contexts such as those mentioned above.

2.3.1 *Variational Divergence Estimation*

The problem of estimating statistical divergences and, by extension, density-ratios, using only samples [188, 191] can be effectively tackled by leveraging the framework of convex analysis [213]. Convex analysis is a vast topic in its own right. For a light and intuitive introduction to convex duality (albeit applied in a different context), the reader is envouraged to to consult the self-contained section from the text of Bishop [16, §10.5]. Now, every convex, lower-semicontinuous function f has a convex dual $f^\star$, also known as the Fenchel conjugate [213]. More precisely, function f and its convex dual $f^\star$ are related as follows,

convex dual
Fenchel conjugate

$$f(u) = \max_t \{ut - f^\star(t)\}, \qquad f^\star(t) = \max_u \{ut - f(u)\}. \tag{2.9}$$

The convex dual is *involutory*, meaning that the convex dual of $f^\star$ is simply $f^{\star\star} = f$. Since f is convex, its first derivative f' is strictly nondecreasing. Therefore, we can reparameterise the variational formulation of $f(u)$ from Equation (2.9) by substituting t with $f'(s)$ (for some s in the domain of f'),

$$f(u) = \max_s \left\{ uf'(s) - f^\star(f'(s)) \right\}.$$

Substituting this into the f-divergence from Equation (2.7) and invoking Jensen's inequality gives the lower bound

$$\mathcal{D}_f \left[p(\mathbf{x}) \parallel q(\mathbf{x}) \right] \geq \max_{\boldsymbol{\theta}} \left\{ \mathbb{E}_{p(\mathbf{x})}[f'(r_{\boldsymbol{\theta}}(\mathbf{x}))] - \mathbb{E}_{q(\mathbf{x})}[f^\star(f'(r_{\boldsymbol{\theta}}(\mathbf{x})))] \right\}, \tag{2.10}$$

where $r_{\boldsymbol{\theta}} : \mathcal{X} \to \mathbb{R}_+$ is some mapping with parameters $\boldsymbol{\theta}$. This is a powerful bound with far-reaching implications. Firstly, observe that this lower bound objective does not strictly rely on the densities $p(\mathbf{x})$ and $q(\mathbf{x})$ – to efficiently maximise this objective in practice, e. g., using stochastic gradients with the reparameterisation trick, we need only be able to draw samples from $p(\mathbf{x})$ and $q(\mathbf{x})$. Secondly, some straightforward calculus of variations shows that the bound is tighest when $r_{\boldsymbol{\theta}}(\mathbf{x}) = r(\mathbf{x})$, i. e., when the parameterised mapping is precisely the density-ratio introduced in Equation (2.8). In other words, optimising the objective in Equation (2.10) to obtain a tight lower bound directly goes hand-in-hand with obtaining an accurate estimate of the density-ratio.

2.3.2 *Class-Probability Estimation*

We've just discussed a general framework for simulatenously estimating divergences and addressing the DRE problem. Let's now consider

a prominent special case of this known as density-ratio estimation by class-probability estimation (CPE) [15, 37, 170, 203, 251]. Let $\pi_{\boldsymbol{\theta}}$ be a probabilistic classifier: a mapping $\pi_{\boldsymbol{\theta}} : \mathcal{X} \to [0,1]$ parameterised by $\boldsymbol{\theta}$. Recall the well-known binary cross-entropy (BCE) loss, also known as the log loss, prevalent in binary classification,

binary cross-entropy

$$\mathcal{L}(\boldsymbol{\theta}) \triangleq -\mathbb{E}_{p(\mathbf{x})}[\log \pi_{\boldsymbol{\theta}}(\mathbf{x})] - \mathbb{E}_{q(\mathbf{x})}[\log(1 - \pi_{\boldsymbol{\theta}}(\mathbf{x}))]. \tag{2.11}$$

Interestingly, there is a lower bound on the BCE loss [86] that can be expressed in terms of an f-divergence, namely, the Jensen-Shannon (JS) divergence $\mathcal{D}_{\text{JS}}[p \parallel q]$, which is a symmetrised variant of the KL divergence,

Jensen-Shannon divergence

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \geq -2\left(\mathcal{D}_{\text{JS}}[p(\mathbf{x}) \parallel q(\mathbf{x})] - \log 2\right).$$

To see this, let's first parameterise the classifier as

$$\pi_{\boldsymbol{\theta}}(\mathbf{x}) \triangleq \sigma(\log r_{\boldsymbol{\theta}}(\mathbf{x})), \tag{2.12}$$

where σ denotes the logistic sigmoid function and $r_{\boldsymbol{\theta}}$ is some function parameterised by $\boldsymbol{\theta}$. The intermediate (pre-activation) output $\log r_{\boldsymbol{\theta}}(\mathbf{x})$ is known as the *logits*, or *log-odds*.

logits, log-odds

In the special case of

$$f_{\text{BCE}}(u) \triangleq u \log u - (u+1)\log(u+1) \tag{2.13}$$

in Equation (2.10), we get

$$\begin{aligned}
2\left(\mathcal{D}_{\text{JS}}[p(\mathbf{x}) \parallel q(\mathbf{x})] - \log 2\right) &= \mathcal{D}_{f_{\text{BCE}}}[p(\mathbf{x}) \parallel q(\mathbf{x})] \\
&\geq \max_{\boldsymbol{\theta}} \left\{\mathbb{E}_{p(\mathbf{x})}[\log \sigma(\log r_{\boldsymbol{\theta}}(\mathbf{x}))] + \mathbb{E}_{q(\mathbf{x})}[\log(1 - \sigma(\log r_{\boldsymbol{\theta}}(\mathbf{x})))]\right\} \\
&= \max_{\boldsymbol{\theta}} \{-\mathcal{L}(\boldsymbol{\theta})\} = -\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}),
\end{aligned}$$

and negating both sides gives the desired bound. Like in Equation (2.10), the BCE loss is minimised when $r_{\boldsymbol{\theta}}(\mathbf{x}) = r(\mathbf{x})$, or equivalently when

$$\pi_{\boldsymbol{\theta}}(\mathbf{x}) = \sigma(\log r(\mathbf{x})) = \frac{p(\mathbf{x})}{p(\mathbf{x}) + q(\mathbf{x})},$$

where $r(\mathbf{x})$ is the true density-ratio defined in Equation (2.8). Importantly, this provides a straightforward means of recovering a density-ratio estimator from a probabilistic classifier

$$r_{\boldsymbol{\theta}}(\mathbf{x}) = \exp \sigma^{-1}(\pi_{\boldsymbol{\theta}}(\mathbf{x})) = \frac{\pi_{\boldsymbol{\theta}}(\mathbf{x})}{1 - \pi_{\boldsymbol{\theta}}(\mathbf{x})},$$

and vice versa. Thus, we've obtained a direct way of casting the problem of DRE as the well-studied problem of CPE. Furthermore, this general approach is not restricted only to the BCE loss but extends to any other proper scoring rule [85] that produce well-calibrated probabilistic predictions, such as the hinge loss [218].

proper scoring rule

The CPE approach described here constitutes the predominant approach to DRE. It's not difficult to imagine why, considering the veritable cornucopia of user-friendly, off-the-shelf software frameworks that are available for supervised learning. Notable examples include scikit-learn [199] a versatile library covering a wide range of different paradigms, as well as specialised libraries like XGBoost [34] for decision tree ensembles with extreme gradient-boosting (XGBOOST), and PyTorch [197]/Lightning and TensorFlow [1]/Keras [40] for deep neural networks (DNNS), to name just a few. These frameworks have made it easier than ever to train powerful classifiers, driving the widespread adoption of the CPE approach to tackling the problem of DRE.

TOY 1D EXAMPLE. Consider the following toy example where the densities $\ell(x)$ and $g(x)$ are *known* and given exactly by the following (mixture of) Gaussians,

$$\ell(x) \triangleq 0.3\mathcal{N}(2, 1^2) + 0.7\mathcal{N}(-3, 0.5^2), \qquad \text{and} \qquad g(x) \triangleq \mathcal{N}(0, 2^2),$$

as illustrated by the *solid red* and *blue* lines in Figure 2.1, respectively.

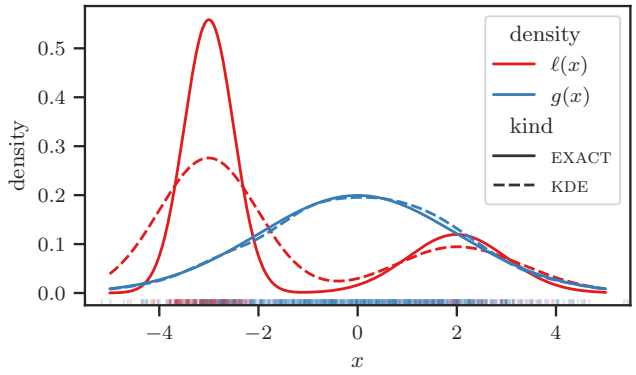
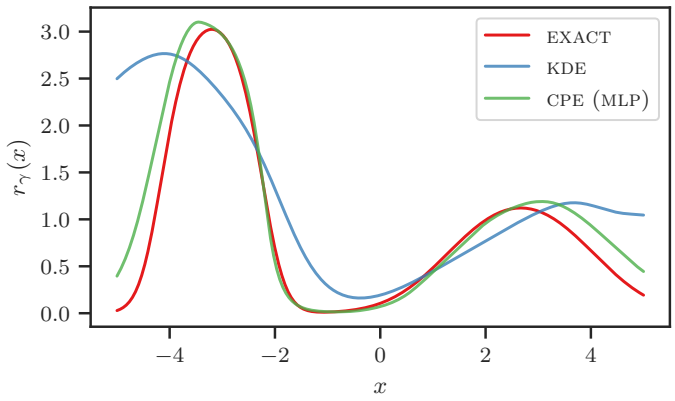


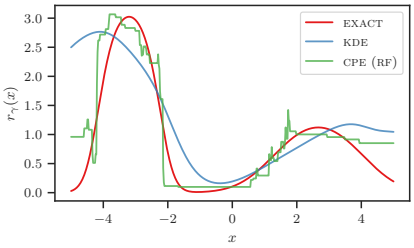
Figure 2.1: Densities $\ell(x)$ and $g(x)$ and their (kernel density) estimates.

We draw a total of $N = 1,000$ samples from these distributions, with a fraction $\gamma = 1/4$ drawn from $\ell(x)$ and the remainder from $g(x)$. These are represented by the vertical markers along the bottom of the x-axis (a so-called "rug plot"). Then, two KDEs, shown with *dashed* lines, are fit on these respective sample sets, with kernel bandwidths selected according to the "normal reference" rule-of-thumb. We see that, for both densities, the modes are recovered well, while for $\ell(x)$, the variances are overestimated in both of its mixture components. As we shall see, this has deleterious effects on the resulting density-ratio estimate.

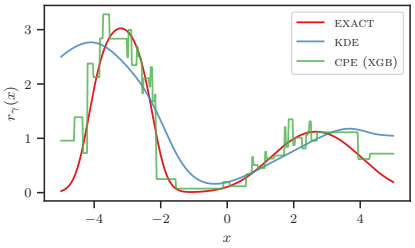
In Figure 2.2a, we represent the true *relative* density-ratio with the *red* line. Note that the relative density-ratio, as we shall see in Section 5.2.1, is a generalisation of the *ordinary* density-ratio we introduced at the beginning of this section. For the purposes of the present discussion,



(a) The relative density-ratio, estimated with an MLP classifier.



(b) The relative density-ratio, estimated with a RF classifier.



(c) The relative density-ratio, estimated with an XGBOOST classifier.

Figure 2.2: Synthetic toy example with (mixtures of) Gaussians.

its precise definition is immaterial as the same analysis applies both to relative and ordinary density-ratios. The estimate resulting from taking the ratio of the KDEs is shown in *blue*, while that of the CPE method described in this section is shown in *green*. In this subfigure, the probabilistic classifier consists of a simple MLP with 3 hidden layers, each with and 32 units and `elu` activations. In Figures 2.2b and 2.2c, we show the same, but with RF and XGBOOST classifiers.

The CPE methods appear, at least visually, to recover the exact density ratios well, whereas the KDE method does so quite poorly. Perhaps the more important quality to focus on, particularly if used in the context of global optimisation as in Chapter 5, is the *mode* of the density-ratio functions. In the case of the KDE method, we can see that this deviates significantly from that of the true density-ratio. In this instance, although KDE fit $g(x)$ well and recovered the modes of $\ell(x)$ accurately, even a slight overestimation of the variance in the latter led to a significant shift in the maximiser of the resulting density-ratio functions.

2.4 GAUSSIAN PROCESSES

We now shift gears and turn our focus to Gaussian processes (GPs), a class of nonparametric Bayesian models that provide a powerful framework for reasoning about unknown functions. GPs are ubiquitous in probabilistic ML [156]. They exhibit remarkable data efficiency, achieving high accuracy even with limited data. Moreoever, they inherently possess mechanisms that help to mitigate over-fitting, and can flexibly encode prior beliefs and assumptions through their covariance function. Last but not least, by virtue of their ability to faithfully capture predictive uncertainty, they form the backbone of many sequential decision-making procedures that require reliable uncertainty estimates to appropriately balance important trade-offs such as that of *exploration* and *exploitation*, for instance, in active learning [108], reinforcement learning [55], Bayesian optimisation [20, 75, 228] (covered in-depth separately in Section 2.5), probabilistic numerics [95], and more.

2.4.1 *Gaussian Process Regression*

More formally, GPs are a flexible class of distributions over functions. A random function $f : \mathcal{X} \to \mathbb{R}$ on some domain $\mathcal{X} \subseteq \mathbb{R}^D$ is distributed according to a GP if, at any finite collection of input locations $\mathbf{X}_* \subseteq \mathcal{X}$, its values $\mathbf{f}_* = f(\mathbf{X}_*)$ follow a Gaussian distribution. A GP is fully determined by its covariance function $k(\mathbf{x}, \mathbf{x}')$ and mean function, which can be assumed without loss of generality to be constant (e. g., zero).

covariance function

Consider a supervised learning problem in which we have a dataset $\{\mathbf{x}_n, y_n\}_{n=1}^N$ consisting of scalar outputs y_n, which are related to $f_n \triangleq f(\mathbf{x}_n)$, the value of some unknown function $f(\cdot)$ at input $\mathbf{x}_n \in \mathcal{X}$, through the likelihood $p(y_n \mid f_n)$. A powerful modelling approach consists of specifying a GP prior on the latent function $f(\cdot)$,

$$f(\mathbf{x}) \sim \mathcal{GP}\left(0, k(\mathbf{x}, \mathbf{x}')\right). \tag{2.14}$$

Let $\mathbf{X}$ denote the inputs, $\mathbf{f}$ the corresponding latent function values, and $\mathbf{y}$ the outputs. In the regression setting, the outputs $\mathbf{y}$ are assumed to be noisy observations of the latent values $\mathbf{f}$, typically related through a Gaussian likelihood

$$p(\mathbf{y} \mid \mathbf{f}, \beta) = \mathcal{N}(\mathbf{y} \mid \mathbf{f}, \beta^{-1}\mathbf{I}), \tag{2.15}$$

for some *precision* $\beta > 0$.

precision

Under this likelihood, the posterior predictive density $p(\mathbf{f}_* \mid \mathbf{y})$ at test inputs is has the closed-form expression

$$p(\mathbf{f}_* \mid \mathbf{y}) = \mathcal{N}\left(\mathbf{K}_{*\mathbf{f}}(\mathbf{K}_{\mathbf{ff}} + \beta^{-1}\mathbf{I})^{-1}\mathbf{y}, \mathbf{K}_{**} - \mathbf{K}_{*\mathbf{f}}(\mathbf{K}_{\mathbf{ff}} + \beta^{-1}\mathbf{I})^{-1}\mathbf{K}_{\mathbf{f}*}\right). \tag{2.16}$$

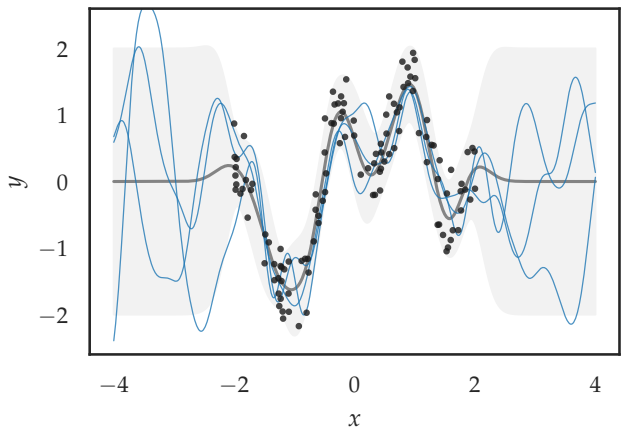


Figure 2.3: Gaussian process (GP) posterior predictive density on the synthetic one-dimensional SNELSON1D dataset [235]; three random functions sampled from this density are indicated by the blue curves.

Clearly, evaluating this density has a time complexity of $\mathcal{O}(N^3)$, which stems from the costs associated with calculating the matrix inverse of $\mathbf{K_{ff}} + \beta^{-1}\mathbf{I}$. Furthermore, for other (i. e., non-Gaussian) likelihoods the closed-form expression for $p(\mathbf{f}_* \mid \mathbf{y})$ is generally unavailable.

2.4.1.1 *Covariance Functions*

The covariance function holds a pivotal role in GP models, as it encapsulates prior beliefs and assumptions about the latent function of interest. It provides a means to encode various characteristics such as periodicity, roughness, and smoothness (or, to be more precise, *orders of differentiability*), etc. Specifically, let us examine the family of *stationary* covariance functions, which are translation invariant in the input space. In other words, $k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}')$ only depends on the difference $\mathbf{x} - \mathbf{x}'$ between the input locations $\mathbf{x}$ and $\mathbf{x}'$. This can be expressed mathematically as

stationarity

$$k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}') = \kappa_{\boldsymbol{\theta}}(\mathbf{x} - \mathbf{x}'),$$

for some function $\kappa_{\boldsymbol{\theta}}$, where $\boldsymbol{\theta}$ consists of some collection of parameters. The use of a stationary covariance function reflects the assumption that the relationship between $f(\mathbf{x})$ and $f(\mathbf{x}')$ is fully characterised by the difference between $\mathbf{x}$ and $\mathbf{x}'$. In particular, consider the squared exponential (SE) kernel, or, the exponentiated quadratic kernel, which can be expressed in terms of function $\kappa_{\boldsymbol{\theta}}$ of the difference $t \triangleq x - x'$,

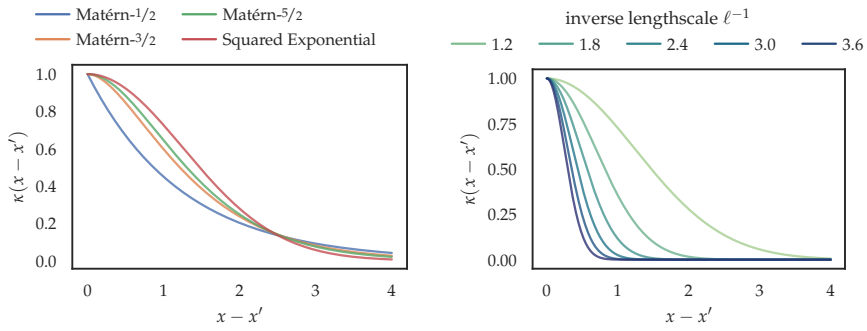
squared exponential kernel

$$\kappa_{\boldsymbol{\theta}}(t) = \sigma_f^2 \exp\left(-\frac{t^2}{2\ell^2}\right), \tag{2.17}$$

where the parameters $\boldsymbol{\theta} \triangleq \{\ell, \sigma_f^2\}$ are made up of the *characteristic lengthscale* ℓ and the variance, or, *amplitude*, σ_f^2. Generalising the

characteristic lengthscale

amplitude



(a) Profile of various covariance functions with characteristic lengthscale $\ell = 5/4$.

(b) The squared exponential (SE) covariance function with varying characteristic lengthscales.

Figure 2.4: Several widely-used stationary covariance functions.

squared exponential (SE) kernel to D dimensions, we have

$$\kappa_{\boldsymbol{\theta}}(\mathbf{t}) = \sigma_f^2 \exp\left(-\frac{1}{2}\mathbf{t}^\top \boldsymbol{\Lambda}^{-1}\mathbf{t}\right), \tag{2.18}$$

for some nonsingular matrix $\boldsymbol{\Lambda}$. The most common and arguably useful choice for $\boldsymbol{\Lambda}$ is the diagonal matrix,

$$\boldsymbol{\Lambda} \triangleq \operatorname{diag}(\ell_1^2, \ldots, \ell_D^2),$$

consisting of the characteristic lengthscales $\ell_1, \ldots, \ell_D$, each associated with an input dimension. Intuitively, each lengthscale dictates how close the input location needs to be (along the associated dimension) for the function values to exhibit high correlation. This effectively implements the functionality known as automatic relevance determination (ARD) [185], because the relevance of an input dimension is inversely proportional to the corresponding lengthscale – that is, an input dimension with a large associated lengthscale will have virtually no influence on the covariance, effectively disregarding its variations during inference [286]. The SE covariance function is infinitely differentiable, implying that the latent function $f(\mathbf{x})$ will have derivatives of all orders. However, assuming such a degree of smoothness is often unreasonable for many applications. For this reason, many practitioners appeal to the Matérn family of covariance functions [247], which were originally named after Matérn [164]. This family of functions offers greater flexibility in modelling various degrees of smoothness, which can be adjusted by specifying a smoothness parameter ν.

automatic relevance determination

Matérn kernel

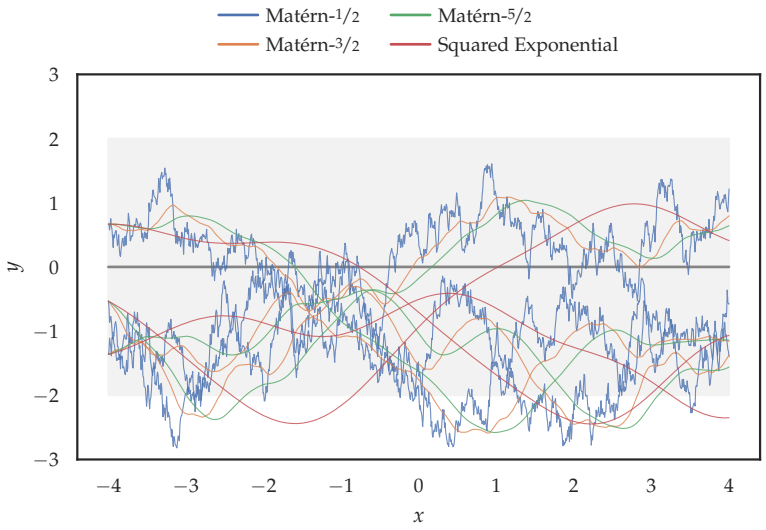


Figure 2.5: Gaussian process (GP) prior samples resulting from different stationary covariance functions with characteristic lengthscale $\ell = 5/4$; three random samples are drawn for each covariance function.

Specifically, in the case when ν is half-integer, i. e., $\nu = \rho + 1/2$ for some nonnegative integer ρ, the Matérn-ν covariance function can be expressed as

$$\begin{aligned}\kappa_{\boldsymbol{\theta}}^{(\nu)}(\mathbf{t}) =& \sigma_f^2 \exp\left(-\sqrt{2\nu\mathbf{t}^\top\mathbf{M}^{-1}\mathbf{t}}\right)\frac{\Gamma(\rho+1)}{\Gamma(2\rho+1)} \\ &\times \sum_{i=0}^{\rho}\frac{(\rho+i)!}{i!(\rho-i)!}\left(\sqrt{8\nu\mathbf{t}^\top\mathbf{M}^{-1}\mathbf{t}}\right)^{\rho-i}.\end{aligned} \tag{2.19}$$

There are a few properties worth noting here. First, the latent function $f(\mathbf{x})$ will have derivatives up to order ρ. This is consistent with the fact that we obtain the SE kernel in the limit as $\nu \to \infty$. The most interesting cases are $\nu = 1/2, 3/2, 5/2$, with the last perhaps being the most widely-used in practice. The choice of $\nu = 5/2$ signifies a prior belief that the latent function $f(\mathbf{x})$ is twice differentiable (since $\rho = 2$), which has been advocated as a helpful assumption, e. g., in the context of global optimisation [236].

2.4.1.2 *Hyperparameter Estimation*

We've already discussed how to obtain the posterior predictive density at test inputs for a given set of hyperparameters, such as $\{\boldsymbol{\theta}, \beta\}$ for noise precision β and kernel parameters $\boldsymbol{\theta}$. As one can imagine from our earlier discussions, these hyperparameters exert a large influence on the behaviour of the GP and its predictions. However, determining the appropriate values for these hyperparameters is often challenging and impractical to do manually. In most cases, when the ideal fully-

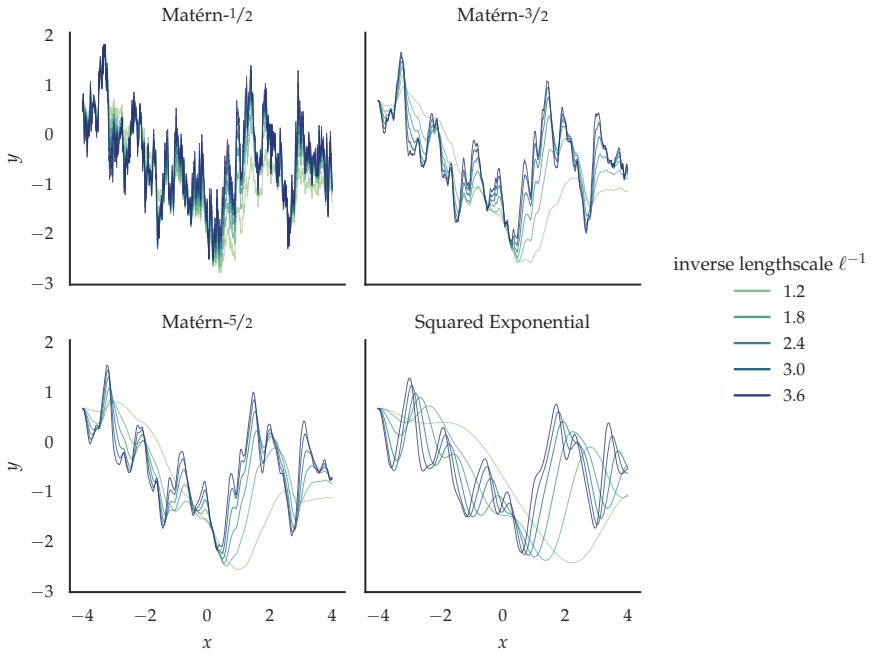


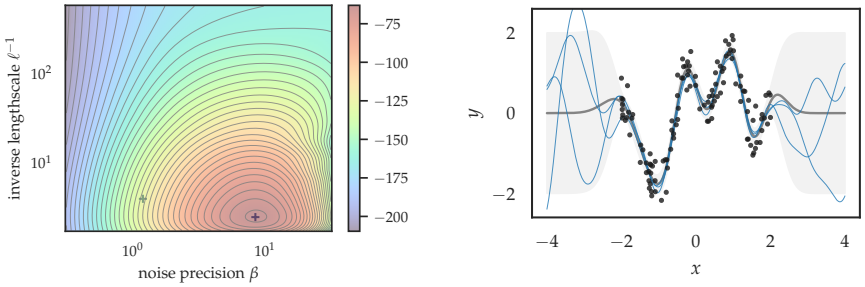
Figure 2.6: Gaussian process (GP) prior samples resulting from different stationary covariance functions with varying lengthscales; one sample is drawn for each combination.

Bayesian treatment of these hyperparemeters proves too unwieldy, it is common practice to adopt a configuration that maximises the marginal likelihood, known as type-II maximum likelihood estimation (MLE). In the regression setting we have been discussing, the marginal likelihood has the closed-form expression

$$\log p(\mathbf{y} \mid \boldsymbol{\theta}, \beta) = \log \int p(\mathbf{y} \mid \mathbf{f}, \beta) p(\mathbf{f} \mid \boldsymbol{\theta}) \, \mathrm{d}\mathbf{f} = \log \mathcal{N}(\mathbf{y} \mid \mathbf{0}, \mathbf{K}_{\mathbf{ff}} + \beta^{-1}\mathbf{I})$$
$$= -\frac{1}{2}\mathbf{y}^\top (\mathbf{K}_{\mathbf{ff}} + \beta^{-1}\mathbf{I})^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}_{\mathbf{ff}} + \beta^{-1}\mathbf{I}| - \frac{N}{2}\log 2\pi. \quad (2.20)$$

The first term is a quadratic in the observations $\mathbf{y}$, which encourages a precise fit to the data. On the other hand, the second term acts a regulariser that discourages overly complex models. Consequently, optimising the hyperparameters wrt to the marginal likelihood automatically strikes a balance between data fit and model complexity, ultimately seeking the simplest model that best explains the data. It is due to this mechanism that GP models are often characterised as being inherently robust against over-fitting. However, it is important to note that the ability to mitigate over-fitting is more accurately attributed to the marginal likelihood, which is essentially what distinguishes the Bayesian inference approach from other approaches based purely on optimisation [286, §5.2].

As with the predictive density in Equation (2.16), the time complexity of evaluating the marginal likelihood is dominated by the $\mathcal{O}(N^3)$ cost of computing the matrix inverse and determinant. Furthermore,



(a) Marginal likelihood of a GP regression model with a SE covariance function and amplitude $\sigma_f = 1$ on the SNELSON1D dataset.

(b) Posterior predictive density with *optimal* hyperparameters.

Figure 2.7: Hyperparameter estimation in a GP regression model and its effects; the two '+' markers correspond to optimal and reasonably-good-but-not-quite-optimal settings of the hyperparameters (ℓ, β). The resulting posterior predictive densities, visualised in Figures 2.3 and 2.7b, respectively, reveal a clear contrast in predictive uncertainty, with the optimal hyperparameters delivering finely tuned confidence intervals.

apart from the case of the GP regression model with Gaussian noise from Equation (2.15), which serves as an exception that proves the rule, the marginal likelihood is generally analytically intractable for arguably the majority of interesting models in probabilistic ML. These two intractabilities have long been recognised as the most significant challenges in establishing the practicality and widespread adoption of GPS.

2.4.2 *Sparse Gaussian Processes*

sparse Gaussian process

A range of sparse GP methods have been developed over the years to mitigate these limitations [46, 204, 226, 234]. Broadly speaking, in sparse GPS, one summarises $f(\cdot)$ succinctly in terms of *inducing variables*, which are values $\mathbf{u} \triangleq f(\mathbf{Z})$ taken at a collection of M locations $\mathbf{Z} = [\mathbf{z}_1 \cdots \mathbf{z}_M]^\top$, where $\mathbf{z}_m \in \mathcal{X}$. Not least among these approaches is SVGP/SGPR, first proposed by [262], which casts sparse GPS within the framework of VI, which we described earlier in Section 2.2. In this section, we examine this framework in detail and discuss some of the extensions for blackbox likelihoods and large-scale inference with mini-batch training [57, 98, 99].

Specifically, the joint distribution of the model augmented by inducing variables $\mathbf{u}$ is $p(\mathbf{f}, \mathbf{u}, \mathbf{y}) = p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f}, \mathbf{u})$, where the joint over $(\mathbf{f}, \mathbf{u})$ factorises as $p(\mathbf{f}, \mathbf{u}) = p(\mathbf{f} \mid \mathbf{u}) p(\mathbf{u})$. The prior $p(\mathbf{u})$ is

$$p(\mathbf{u}) = \mathcal{N}(\mathbf{0}, \mathbf{K}_{\mathbf{uu}}), \tag{2.21}$$

and the conditional $p(\mathbf{f} \mid \mathbf{u})$ is

$$p(\mathbf{f} \mid \mathbf{u}) = \mathcal{N}(\mathbf{f} \mid \mathbf{Q}_{\mathbf{fu}}\mathbf{u}, \mathbf{K}_{\mathbf{ff}} - \mathbf{Q}_{\mathbf{ff}}), \tag{2.22}$$

where $\mathbf{Q}_{\mathbf{ff}} \triangleq \mathbf{Q}_{\mathbf{fu}}\mathbf{K}_{\mathbf{uu}}\mathbf{Q}_{\mathbf{uf}}$ and $\mathbf{Q}_{\mathbf{fu}} \triangleq \mathbf{K}_{\mathbf{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}$. The joint variational distribution is defined as $q(\mathbf{f}, \mathbf{u}) \triangleq q(\mathbf{f} \mid \mathbf{u})q(\mathbf{u})$ where

$$q(\mathbf{u}) \triangleq \mathcal{N}(\mathbf{m}_{\mathbf{u}}, \mathbf{C}_{\mathbf{u}}) \tag{2.23}$$

for variational parameters $\mathbf{m}_{\mathbf{u}} \in \mathbb{R}^M$ and $\mathbf{C}_{\mathbf{u}} \in \mathbb{R}^{M \times M}$ s.t. $\mathbf{C}_{\mathbf{u}} \succeq 0$. Commonly, for convenience, one simply defines $q(\mathbf{f} \mid \mathbf{u}) \triangleq p(\mathbf{f} \mid \mathbf{u})$. At unseen points $\mathbf{f}_* \triangleq f(\mathbf{X}_*)$, integrating out $\mathbf{u}$ leads to the test predictive density

$$q(\mathbf{f}_*) = \mathcal{N}(\mathbf{f}_* \mid \mathbf{Q}_{*\mathbf{u}}\mathbf{m}_{\mathbf{u}}, \mathbf{K}_{**} - \mathbf{Q}_{*\mathbf{u}}(\mathbf{K}_{\mathbf{uu}} - \mathbf{C}_{\mathbf{u}})\mathbf{Q}_{\mathbf{u}*}), \tag{2.24}$$

where parameters $\mathbf{m}_{\mathbf{u}}$ and $\mathbf{C}_{\mathbf{u}}$ are learned by minimising the KL divergence between the approximate and exact posteriors, $\mathrm{KL}\left[q(\mathbf{f}^*, \mathbf{f}, \mathbf{u}) \parallel p(\mathbf{f}^*, \mathbf{f}, \mathbf{u} \mid \mathbf{y})\right]$. Conveniently, since the posteriors factorise as

$$q(\mathbf{f}^*, \mathbf{f}, \mathbf{u}) = p(\mathbf{f}^* \mid \mathbf{f}, \mathbf{u})q(\mathbf{f}, \mathbf{u}),$$

and

$$p(\mathbf{f}^*, \mathbf{f}, \mathbf{u} \mid \mathbf{y}) = p(\mathbf{f}^* \mid \mathbf{f}, \mathbf{u}, \cancel{\mathbf{y}})p(\mathbf{f}, \mathbf{u} \mid \mathbf{y}),$$

the common factor $p(\mathbf{f}^* \mid \mathbf{f}, \mathbf{u})$ cancels each other to simplify the KL,

$$\mathrm{KL}\left[q(\mathbf{f}^*, \mathbf{f}, \mathbf{u}) \parallel p(\mathbf{f}^*, \mathbf{f}, \mathbf{u} \mid \mathbf{y})\right] = \mathrm{KL}\left[q(\mathbf{f}, \mathbf{u}) \parallel p(\mathbf{f}, \mathbf{u} \mid \mathbf{y})\right].$$

Refer to Section 2.A for details. Now, by Bayes' rule, we have

$$\begin{aligned} &\mathrm{KL}\left[q(\mathbf{f}, \mathbf{u}) \parallel p(\mathbf{f}, \mathbf{u} \mid \mathbf{y})\right] && (2.25)\\ &= \iint q(\mathbf{f}, \mathbf{u}) \log \frac{q(\mathbf{f}, \mathbf{u})}{p(\mathbf{f}, \mathbf{u} \mid \mathbf{y})} \, \mathrm{d}\mathbf{f}\mathrm{d}\mathbf{u} \\ &= \log p(\mathbf{y}) - \iint q(\mathbf{f}, \mathbf{u}) \log \frac{p(\mathbf{f}, \mathbf{u}, \mathbf{y})}{q(\mathbf{f}, \mathbf{u})} \, \mathrm{d}\mathbf{f}\mathrm{d}\mathbf{u}. && (2.26) \end{aligned}$$

The astute reader might find this familiar, as it an instance of the general expression we examined in Section 2.2. Indeed, if we define the ELBO as

$$\mathrm{ELBO}(q) \triangleq \iint q(\mathbf{f}, \mathbf{u}) \log \frac{p(\mathbf{f}, \mathbf{u}, \mathbf{y})}{q(\mathbf{f}, \mathbf{u})} \, \mathrm{d}\mathbf{f}\mathrm{d}\mathbf{u},$$

then, upon re-arranging Equation (2.26), we get

$$\log p(\mathbf{y}) = \mathrm{ELBO}(q) + \mathrm{KL}\left[q(\mathbf{f}, \mathbf{u}) \parallel p(\mathbf{f}, \mathbf{u} \mid \mathbf{y})\right].$$

Now, since $p(\mathbf{f}, \mathbf{u}, \mathbf{y})$ factorises as

$$p(\mathbf{f}, \mathbf{u}, \mathbf{y}) = p(\mathbf{y} \mid \mathbf{f}, \cancel{\mathbf{u}})p(\mathbf{f}, \mathbf{u}) = p(\mathbf{y} \mid \mathbf{f})p(\mathbf{f} \mid \mathbf{u})p(\mathbf{u}),$$

we can simplify the ELBO to

$$\begin{aligned}\text{ELBO}(q) &= \iint p(\mathbf{f} \mid \mathbf{u}) q(\mathbf{u}) \log \frac{p(\mathbf{y} \mid \mathbf{f}) \cancel{p(\mathbf{f} \mid \mathbf{u})} p(\mathbf{u})}{\cancel{p(\mathbf{f} \mid \mathbf{u})} q(\mathbf{u})} \, \mathrm{d}\mathbf{f}\mathrm{d}\mathbf{u} \\ &= \int q(\mathbf{u}) \log \frac{F(\mathbf{y}, \mathbf{u}) p(\mathbf{u})}{q(\mathbf{u})} \mathrm{d}\mathbf{u}, \end{aligned} \tag{2.27}$$

where we have defined

$$F(\mathbf{y}, \mathbf{u}) \triangleq \exp\left(\int p(\mathbf{f} \mid \mathbf{u}) \log p(\mathbf{y} \mid \mathbf{f}) \, \mathrm{d}\mathbf{f} \right). \tag{2.28}$$

We can re-arrange the ELBO of Equation (2.27) into the usual composition made up of ELL and KL divergence terms,

$$\text{ELBO}(q) = \int q(\mathbf{u}) \log F(\mathbf{y}, \mathbf{u}) \mathrm{d}\mathbf{u} - \text{KL}\left[q(\mathbf{u}) \parallel p(\mathbf{u}) \right]. \tag{2.29}$$

Interestingly, $\log\left[F(\mathbf{y}, \mathbf{u})\right]$ is a lower bound on the log conditional probability $\log p(\mathbf{y} \mid \mathbf{u})$ – quite simply, by Jensen's inequality, we have

$$\begin{aligned}\log p(\mathbf{y} \mid \mathbf{u}) &= \log \mathbb{E}_{p(\mathbf{f} \mid \mathbf{u})}[p(\mathbf{y} \mid \mathbf{f})] \\ &\geq \mathbb{E}_{p(\mathbf{f} \mid \mathbf{u})}[\log p(\mathbf{y} \mid \mathbf{f})] = \log F(\mathbf{y}, \mathbf{u}).\end{aligned}$$

Refer to the manuscript of Hensman, Matthews, and Ghahramani [99, Equation 1] for a discussion of the role that this "intermediate" lower bound plays in various contexts.

It is worth mentioning that there are some nuanced technical concerns over whether maximising the ELBO in Equation (2.29) truly minimises the KL divergence between the prior and posterior stochastic processes. We shall not delve further into this issue here except to note that these were largely resolved by Matthews et al. [165].

OPTIMAL VARIATIONAL DISTRIBUTION. From the ELBO as expressed in Equation (2.27), it's evident that the maximising variational distribution takes the form $q^*(\mathbf{u}) \propto F(\mathbf{y}, \mathbf{u}) p(\mathbf{u})$,

$$q^*(\mathbf{u}) = \frac{F(\mathbf{y}, \mathbf{u}) p(\mathbf{u})}{\int F(\mathbf{y}, \mathbf{u}) p(\mathbf{u}) \, \mathrm{d}\mathbf{u}}. \tag{2.30}$$

This can also be verified through the use of calculus of variations, as shown in Section 2.B, or by applying Jensen's inequality, but in the opposite direction.

COLLAPSED LOWER BOUND. If we now substitute q^* back into the ELBO, we get the so-called *collapsed* lower bound,

$$\text{ELBO}(q^*) = \log\left(\int p(\mathbf{u}) F(\mathbf{y}, \mathbf{u}) \, \mathrm{d}\mathbf{u} \right). \tag{2.31}$$

This bound is "collapsed" in the sense that it is no longer a function of q (it is already optimal wrt q) but implicitly remains a function of other (hyper)parameters such as the kernel parameters $\boldsymbol{\theta}$ and the inducing input locations $\mathbf{Z}$.

2.4.2.1 *General Likelihoods*

black-box likelihood

When we make no assumptions about the explicit form of the likelihood $p(\mathbf{y} \mid \mathbf{f})$ nor of its structure or behaviour, it is characterised as "black-box". The integral that constitutes the ELL term in Equation (2.29) is generally intractable for black-box likelihoods. However, if we marginalise out $\mathbf{u}$ to rewrite the ELL as

$$\begin{aligned}\int q(\mathbf{u}) \log F(\mathbf{y}, \mathbf{u}) \, \mathrm{d}\mathbf{u} &= \int \left(\int q(\mathbf{u}) p(\mathbf{f} \mid \mathbf{u}) \, \mathrm{d}\mathbf{u} \right) \log p(\mathbf{y} \mid \mathbf{f}) \, \mathrm{d}\mathbf{f} \\ &= \int q(\mathbf{f}) \log p(\mathbf{y} \mid \mathbf{f}) \, \mathrm{d}\mathbf{f},\end{aligned}$$

we can approximate it efficiently using numerical integration methods such as Monte Carlo (MC) estimation or quadrature rules, by virtue of the fact that the marginal $q(\mathbf{f})$ is available in the analytical form of Equation (2.24) and can thus be sampled easily,

$$\int q(\mathbf{f}) \log p(\mathbf{y} \mid \mathbf{f}) \, \mathrm{d}\mathbf{f} \approx \frac{1}{S} \sum_{s=1}^{S} \log p(\mathbf{y} \mid \mathbf{f}^{(s)}), \quad \mathbf{f}^{(s)} \sim q(\mathbf{f})$$

Moreover, because $q(\mathbf{f})$ is Gaussian, we can utilise simple and effective rules like Gauss-Hermite quadrature, described further in Appendix A for a different application.

2.4.2.2 *Factorised Likelihoods (for Scalability)*

Further, suppose the likelihood factorises, i. e., the observations depend point-wise on the latent functions,

$$p(\mathbf{y} \mid \mathbf{f}) = \prod_{n=1}^{N} p(y_n \mid f_n),$$

we then have

$$\int q(\mathbf{f}) \log p(\mathbf{y} \mid \mathbf{f}) \mathrm{d}\mathbf{f} = \sum_{n=1}^{N} \int q(f_n) \log p(y_n \mid f_n) \, \mathrm{d}f_n.$$

Therefore, the ELBO can be written as

$$\text{ELBO}(q) = \sum_{n=1}^{N} \mathbb{E}_{q(f_n)}[\log p(y_n \mid f_n)] - \text{KL}\left[q(\mathbf{u}) \parallel p(\mathbf{u})\right].$$

Importantly, it's clear that this objective is amenable to mini-batch training for stochastic optimisation [98].

2.4.2.3 *Gaussian Likelihood (for Regression)*

Now suppose the problem at hand is regression, for which the likelihood of choice is typically the Gaussian from Equation (2.15). We can show that

$$F(\mathbf{y}, \mathbf{u}) = \mathcal{N}(\mathbf{y} \mid \mathbf{Q}_{\mathbf{fu}}\mathbf{u}, \beta^{-1}\mathbf{I}) \times \exp\left(-\frac{\beta}{2} \operatorname{tr}(\mathbf{K}_{\mathbf{ff}} - \mathbf{Q}_{\mathbf{ff}})\right). \tag{2.32}$$

Refer to Section 2.C for detailed derivations. This framework, first studied in the landmark paper by Titsias [262], is often referred to as sparse GP regression (SGPR).

OPTIMAL VARIATIONAL DISTRIBUTION. Since the likelihood is Gaussian, by Equation (2.32), the maximiser of the ELBO in Equation (2.30) is the product of two exponentiated-quadratic functions of $\mathbf{u}$. When normalised, this becomes

$$q^*(\mathbf{u}) = \mathcal{N}(\mathbf{u} \mid \beta \mathbf{K_{uu}} \mathbf{M}^{-1} \mathbf{K_{uf}} \mathbf{y}, \mathbf{K_{uu}} \mathbf{M}^{-1} \mathbf{K_{uu}}), \tag{2.33}$$

where $\mathbf{M} \triangleq \mathbf{K_{uu}} + \beta \mathbf{K_{fu}} \mathbf{K_{uf}}$. Refer to Section 2.D for details.

COLLAPSED LOWER BOUND. The optimal lower bound wrt q from Equation (2.31) now becomes

$$\text{ELBO}(q^*) = \log \mathcal{N}(\mathbf{y} \mid \mathbf{0}, \mathbf{Q_{ff}} + \beta^{-1}\mathbf{I}) - \frac{\beta}{2} \operatorname{tr}(\mathbf{K_{ff}} - \mathbf{Q_{ff}}).$$

Refer to Section 2.E for further details. It's instructive at this point to compare this with the log marginal likelihood $\log p(\mathbf{y})$ of the exact GP regression setting from Equation (2.20). We readily see that $\log p(\mathbf{y}) = \text{ELBO}(q^*)$ when $\mathbf{Q_{ff}} = \mathbf{K_{ff}}$. Furthermore, evaluating $\log p(\mathbf{y})$ has a computational complexity of $\mathcal{O}(N^3)$, whereas calculating the ELBO has a complexity of $\mathcal{O}(NM^2 + M^3)$.

TEST PREDICTIVE DISTRIBUTION. Finally, we can obtain the posterior predictive density at test inputs $\mathbf{X}_*$ by substituting the mean and covariance from Equation (2.33) into $\mathbf{m_u}$ and $\mathbf{C_u}$ from Equation (2.24),

$$q(\mathbf{f}_*) = \mathcal{N}\left(\mathbf{f}_* \mid \beta \mathbf{K}_{*\mathbf{u}} \mathbf{M}^{-1} \mathbf{K_{uf}} \mathbf{y}, \mathbf{K}_{**} - \mathbf{K}_{*\mathbf{u}} (\mathbf{K_{uu}}^{-1} - \mathbf{M}^{-1}) \mathbf{K}_{\mathbf{u}*}\right). \tag{2.34}$$

All told, we see that SGPR has time complexity $\mathcal{O}(M^3)$ at prediction time and $\mathcal{O}(NM^2 + M^3)$ during training, with a space complexity of $\mathcal{O}(NM + M^2)$, which offers a substantial speedup over exact inference when $M \ll N$.

Finding a comprehensive, self-contained resource that provides derivations of the equations summarised in this section is surprisingly difficult. Consequently, the pieces necessary to construct the contents of this section and its derivations are collected variously from the unpublished technical report of Titsias [263], the technical notes of Bui and Turner, the paper of Hensman, Matthews, and Ghahramani [99], as well as the PhD theses of Bui [23], Matthews [168], and Van der Wilk [272].

For the newcomer to Bayesian statistics, it is instructive to derive these independently, as they invoke nearly all the essential tools of the trade, such as identities relating to conditioning, marginalisation, and affine transformations of Gaussians, the Woodbury matrix identity,

Jensen's inequality, calculus of variations, "completing the square", including less standard ones such as the "inner-product as outer-product-trace" identity. We reiterate only a few of these here, as most can be found in the well-known texts of Deisenroth, Faisal, and Ong [56, p. MML], Murphy [181, MLaPP], Bishop [16, PRML], and Williams and Rasmussen [286, GPML].

Deriving the quantities from this section not only provides an ideal exercise regimen for hands-on practice with these vital tools, taking the journey from exact GP regression to SGPR, SVGP, and its stochastic variant offers a prime example of a model family that effectively spans the spectrum of exactness and approximation often present in Bayesian modelling and VI that we previously alluded to in Section 2.2. This progression leads us from an exact posterior to an closed-form optimal variational posterior, followed by a variational posterior optimised wrt an exact deterministic ELBO and, ultimately, to one optimised wrt a stochastic ELBO.

In this section, we have examined sparse GPs through the lens of VI. This framework, which we and others have referred to as SVGP, also goes by the name of the variational free energy (VFE) framework, owing to the ELBO's interpretation from the perspective of statistical thermodynamics. The framework known as *stochastic* variational GP [98] shares the same acronym as SVGP, but specifically pertains to the scalable mini-batch variant of the VFE framework. Other prominent sparse GP methods, such as the deterministic training conditional (DTC) [226] and the fully independent training conditional (FITC) [234], are beyond the scope of this thesis. Nonetheless, the topic of their connection to VFE is fascinating, and we direct the interested reader to the manuscript by Bui, Yan, and Turner [22] and the thesis of Bui [23] for a unifying framework under the umbrella of EP. For further insights and practical implications of their connections, we recommend the manuscript by Bauer, Wilk, and Rasmussen [9] and the thesis of Van der Wilk [272].

2.4.3 *Random Fourier Features*

In the previous section, we examined a kind of GP approximation that effectively approximates the GP *posterior* predictive density. Now let's examine a different approximation – one that approximates the covariance function itself, and, therefore, the *prior*.

Bayesian linear model

Consider the Bayesian linear regression (BLR) model with weights $\mathbf{w} \in \mathbb{R}^L$,

$$f(\mathbf{x}) = \sum_{i=1}^{L} w_i \phi_i(\mathbf{x}) = \boldsymbol{\phi}^\top(\mathbf{x})\mathbf{w}, \tag{2.35}$$

basis functions

for some set of L *basis functions*, or, *features*, $\boldsymbol{\phi}(\mathbf{x}) = [\phi_1(\mathbf{x}) \dots \phi_L(\mathbf{x})]^\top \in \mathbb{R}^L$. As before, in Equation (2.15), the observed targets y are assumed

to be function values corrupted by additive noise ε, which are further assumed to be iid Gaussian with zero mean and precision $\beta > 0$,

$$y = f(\mathbf{x}) + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \beta^{-1}). \tag{2.36}$$

This implies the likelihood $p(\mathbf{y} \mid \mathbf{w}) = \mathcal{N}(\mathbf{y} \mid \mathbf{\Phi}\mathbf{w}, \beta^{-1}\mathbf{I})$, where $\mathbf{\Phi} \triangleq \boldsymbol{\phi}(\mathbf{X}) \in \mathbb{R}^{N \times L}$. Suppose we have a Gaussian prior over the weights $p(\mathbf{w}) = \mathcal{N}(\mathbf{w} \mid \mathbf{0}, \mathbf{\Sigma}_\mathbf{w})$. When f is evaluated at a finite collection of T locations $\mathbf{X}_*$, the vector $\mathbf{f}_* = f(\mathbf{X}_*) \in \mathbb{R}^T$ follows the Gaussian distribution $\mathcal{N}(\mathbf{f}_* \mid \mathbf{0}, \mathbf{\Phi}_* \mathbf{\Sigma}_\mathbf{w} \mathbf{\Phi}_*^\top)$ where $\mathbf{\Phi}_* \triangleq \boldsymbol{\phi}(\mathbf{X}_*) \in \mathbb{R}^{T \times L}$. In other words f is by definition a GP with the covariance function $k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^\top \mathbf{\Sigma}_\mathbf{w} \boldsymbol{\phi}(\mathbf{x}')$. This is known as the weight-space perspective of GPs [286]. Sampling random functions $f(\cdot)$ from the prior amounts to sampling $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_\mathbf{w})$. Therefore, if $\mathbf{\Sigma}_\mathbf{w}$ is diagonal, as is often the case in practice, $f(\cdot)$ can be sampled cheaply at a cost of $\mathcal{O}(L)$. Additionally, for a given realisation of $\mathbf{w}$, the corresponding sample $f(\mathbf{x})$ is a deterministic function – importantly, one that is differentiable wrt $\mathbf{x}$. Consequently, the weight-space approximation is relied upon in Thompson sampling [258] to address sequential decision-making problems that require balancing exploration and exploitation, as we will discuss further in Section 2.5.2.4.

weight-space approximation

Now, the posterior weight density is

$$p(\mathbf{w} \mid \mathbf{y}) = \mathcal{N}\left(\beta(\mathbf{\Sigma}_\mathbf{w}^{-1} + \beta\mathbf{\Phi}^\top\mathbf{\Phi})^{-1}\mathbf{\Phi}^\top\mathbf{y}, (\mathbf{\Sigma}_\mathbf{w}^{-1} + \beta\mathbf{\Phi}^\top\mathbf{\Phi})^{-1}\right) \tag{2.37}$$

Assuming $\mathbf{\Sigma}_\mathbf{w} = \mathbf{I}$, the covariance function is becomes $k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\phi}(\mathbf{x}')$ and the posterior density simplifies to

$$p(\mathbf{w} \mid \mathbf{y}) = \mathcal{N}\left((\mathbf{\Phi}^\top\mathbf{\Phi} + \beta^{-1}\mathbf{I})^{-1}\mathbf{\Phi}^\top\mathbf{y}, \beta^{-1}(\mathbf{\Phi}^\top\mathbf{\Phi} + \beta^{-1}\mathbf{I})^{-1}\right).$$

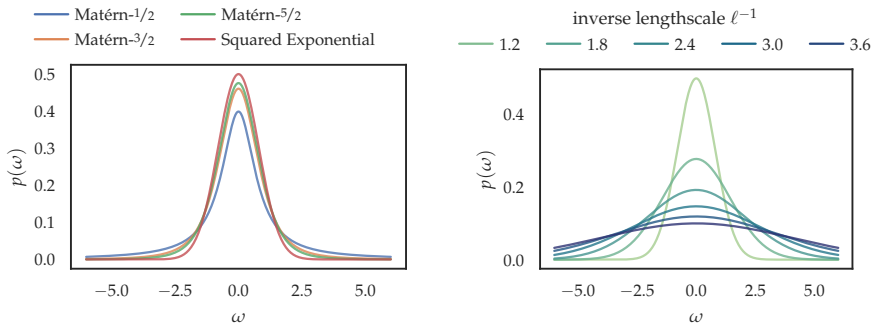
Thus seen, the computational complexity of evaluating this density is dominated by the cost associated with inverting the matrix $\mathbf{\Phi}^\top\mathbf{\Phi} + \beta^{-1}\mathbf{I}$. Through judicious application of the Woodbury matrix identity, this cost is $\mathcal{O}(\min\{L, N\}^3)$.

Now, by the *kernel trick*, a kernel k can be seen as an inner product in a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ equipped with a feature map $\varphi : \mathcal{X} \to \mathcal{H}$. For separable $\mathcal{H}$, we can approximate this inner product as

kernel trick

$$k(\mathbf{x}, \mathbf{x}') = \langle \varphi(\mathbf{x}), \varphi(\mathbf{x}') \rangle_\mathcal{H} \approx \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\phi}(\mathbf{x}'), \tag{2.38}$$

for some finite-dimensional feature map $\boldsymbol{\phi} : \mathcal{X} \to \mathbb{R}^L$. In particular, let us focus on the stationary covariance functions, which possess properties that can be leveraged to construct efficient approximations. Extensions beyond stationary covariance functions are possible through the application of Mercer's theorem and the Karhunen–Loève expansion [30, 73]. In Chapter 3, we discuss an example of this in the spherical harmonics for *zonal* covariance functions.



(a) Spectral density of various covariance functions with characteristic lengthscale $\ell = 5/4$.

(b) Spectral density of the squared exponential (SE) covariance function with varying characteristic lengthscales.

Figure 2.8: Spectral densities of the stationary covariance functions from Section 2.4.1.1.

Kernel	$\kappa(\mathbf{t})$	$p(\boldsymbol{\omega})$
SE	$\exp\left(-\frac{1}{2}\mathbf{t}^\top\mathbf{M}^{-1}\mathbf{t}\right)$	$\mathcal{N}\left(\mathbf{0}, \mathbf{M}^{-1}\right)$
Matérn-3/2	$\left(1+\sqrt{3\mathbf{t}^\top\mathbf{M}^{-1}\mathbf{t}}\right)\exp\left(-\sqrt{3\mathbf{t}^\top\mathbf{M}^{-1}\mathbf{t}}\right)$	$t_3\left(\mathbf{0}, \mathbf{M}^{-1}\right)$
Matérn-5/2	$\left(1+\sqrt{5\mathbf{t}^\top\mathbf{M}^{-1}\mathbf{t}}+\frac{5}{3}\mathbf{t}^\top\mathbf{M}^{-1}\mathbf{t}\right)\exp\left(-\sqrt{5\mathbf{t}^\top\mathbf{M}^{-1}\mathbf{t}}\right)$	$t_5\left(\mathbf{0}, \mathbf{M}^{-1}\right)$

Table 2.1: Fourier transform pairs of stationary covariance function $\kappa(\mathbf{t})$ and their spectral density $p(\boldsymbol{\omega})$, with $\mathbf{t} \triangleq \mathbf{x} - \mathbf{x}'$ and $\mathbf{M} \triangleq \operatorname{diag}(\ell_1^2, \ldots, \ell_D^2)$.

Theorem 2.4.1 (Bochner's theorem)**.** *A continuous, translation invariant kernel $k(\mathbf{x}, \mathbf{x}') = \kappa(\mathbf{x} - \mathbf{x}')$ is positive definite if and only if it is the Fourier transform of a nonnegative, finite measure μ,*

$$\kappa(\mathbf{x} - \mathbf{x}') = \int e^{-i\boldsymbol{\omega}^\top(\mathbf{x}-\mathbf{x}')}\,\mathrm{d}\mu(\boldsymbol{\omega}).$$

spectral density

If measure μ has a density $p(\boldsymbol{\omega})$, it is referred to as the *spectral density*, or, *power spectrum*, associated with kernel k. We have the following Fourier transform pair,

$$\kappa(\mathbf{t}) = \int p(\boldsymbol{\omega}) e^{-i\boldsymbol{\omega}^\top\mathbf{t}}\,\mathrm{d}\boldsymbol{\omega}, \qquad \text{and} \qquad p(\boldsymbol{\omega}) = \frac{1}{2\pi}\int \kappa(\mathbf{t}) e^{i\boldsymbol{\omega}^\top\mathbf{t}}\,\mathrm{d}\mathbf{t}. \tag{2.39}$$

For example, for the 1D SE kernel from Equation (2.17), we can calculate its corresponding spectral density using Equation (2.39) to obtain

$$p(\omega) = \mathcal{N}\left(\omega \,|\, 0, \ell^{-2}\right). \tag{2.40}$$

Refer to Section 2.F for details. More generally, for the D-dimensional SE kernel from Equation (2.18), we have

$$p(\boldsymbol{\omega}) = \mathcal{N}\left(\mathbf{0}, \mathbf{M}^{-1}\right),$$

and, for the Matérn-ν kernel from Equation (2.19), we have

$$p(\boldsymbol{\omega}) = t_{2\nu}\left(\mathbf{0}, \mathbf{M}^{-1}\right),$$

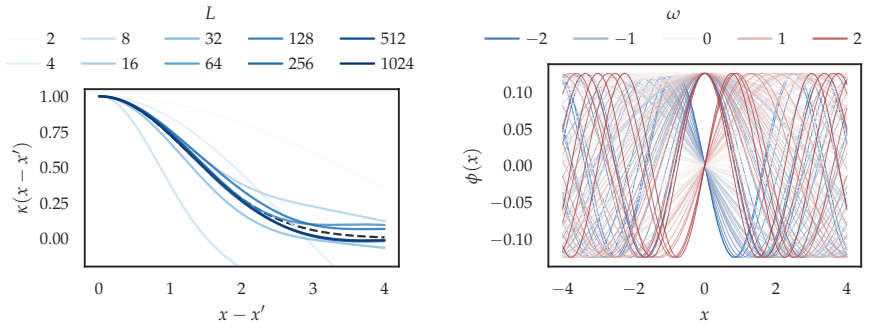
where $t_{2\nu}$ denotes the Student's t-distribution with 2ν degrees of freedom. See Table 2.1 for a summary of popular stationary kernels and their spectral densities. Now, assuming $p(\boldsymbol{\omega})$ is *even symmetric*, $\kappa(\mathbf{t})$ from Equation (2.39) is real-valued and simplifies further to the Fourier *cosine* transform,

$$\begin{aligned}\kappa(\mathbf{x} - \mathbf{x}') &= \int p(\boldsymbol{\omega}) \cos\left(\boldsymbol{\omega}^\top(\mathbf{x} - \mathbf{x}')\right) \mathrm{d}\boldsymbol{\omega} \\ &= \mathbb{E}_{p(\boldsymbol{\omega})}[\cos\left(\boldsymbol{\omega}^\top(\mathbf{x} - \mathbf{x}')\right)]. \end{aligned} \tag{2.41}$$

Let $\psi_{\boldsymbol{\omega}} : \mathbb{R}^D \to \mathbb{R}^2$ denote the projection in some random direction $\boldsymbol{\omega} \sim p(\boldsymbol{\omega})$, mapped to the unit circle,

$$\psi_{\boldsymbol{\omega}}(\mathbf{x}) \triangleq \begin{bmatrix}\cos \boldsymbol{\omega}^\top \mathbf{x} \\ \sin \boldsymbol{\omega}^\top \mathbf{x}\end{bmatrix}. \tag{2.42}$$

Using elementary trigonometric identities, we can show that the inner



(a) Covariance function approximations $\boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\phi}(\mathbf{x}') \approx k(\mathbf{x}, \mathbf{x}')$ for $L = 2^i$ and increasing values of $i = 1, \ldots, 10$.

(b) An example realisation of the basis functions $\boldsymbol{\phi} : \mathcal{X} \to \mathbb{R}^L$ for $L = 2^6$.

Figure 2.9: A example random Fourier features (RFF) decomposition of the SE covariance function with characteristic lengthscale $\ell = 5/4$. The exact values of the covariance function are indicated by the dashed black line.

product of $\psi_{\boldsymbol{\omega}}$ evaluated at inputs $\mathbf{x}$ and $\mathbf{x}'$ is

$$\psi_{\boldsymbol{\omega}}(\mathbf{x})^\top \psi_{\boldsymbol{\omega}}(\mathbf{x}') = \cos\left(\boldsymbol{\omega}^\top(\mathbf{x} - \mathbf{x}')\right). \tag{2.43}$$

Refer to Section 2.G for details. Finally, by Equation (2.41), we recover the kernel k by taking the expectation of Equation (2.43) on both sides,

$$\begin{aligned}\mathbb{E}_{p(\boldsymbol{\omega})}[\psi_{\boldsymbol{\omega}}(\mathbf{x})^\top \psi_{\boldsymbol{\omega}}(\mathbf{x}')] &= \mathbb{E}_{p(\boldsymbol{\omega})}[\cos\left(\boldsymbol{\omega}^\top(\mathbf{x} - \mathbf{x}')\right)] \\ &= k(\mathbf{x}, \mathbf{x}'). \end{aligned} \tag{2.44}$$

This shows that the inner product of Equation (2.43) is an unbiased estimator of $k(\mathbf{x}, \mathbf{x}')$. In other words, evaluating the kernel amounts to computing the expectation in the LHS of Equation (2.44). Hence, in order to approximate the kernel, we can leverage techniques of numerical integration [51] to construct a set of basis functions, or, features, $\boldsymbol{\phi} : \mathcal{X} \to \mathbb{R}^L$, such that

$$\boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\phi}(\mathbf{x}') = \sum_{i=1}^{L} \phi_i(\mathbf{x})^\top \phi_i(\mathbf{x}') \approx \mathbb{E}_{p(\boldsymbol{\omega})}[\psi_{\boldsymbol{\omega}}(\mathbf{x})^\top \psi_{\boldsymbol{\omega}}(\mathbf{x}')] = k(\mathbf{x}, \mathbf{x}').$$

Fourier feature decomposition

We refer to this as the *Fourier feature decomposition*. Perhaps the most well-known example of this is the (award-winning) random Fourier features (RFF) decomposition of Rahimi and Recht [206], in which $\phi_i : \mathbf{x} \mapsto \sqrt{2/L}\cos(\boldsymbol{\omega}^{(i)} \cdot \mathbf{x} + b^{(i)})$ for $\boldsymbol{\omega}^{(i)} \sim p(\boldsymbol{\omega})$ and $b^{(i)} \sim \mathcal{U}[0, 2\pi]$. This feature decomposition is based on the relatively straightforward application of MC estimation in combination with a few trigonometric identities.

In Appendix A, we provide a detailed derivation of the random Fourier features (RFF) decomposition, in addition to alternative feature decompositions based on various numerical integration schemes. For further details on the weight-space approximation and generalisations beyond stationary covariance functions, the interested reader may refer to the manuscript of Wilson et al. [291] upon which our treatment of this topic is based.

2.5 BAYESIAN OPTIMISATION

Bayesian optimisation (BO) is a powerful framework for efficiently locating the global optima of expensive black-box functions [20, 75, 228]. It can be seen as a sequential algorithm for decision-making amidst the uncertainties inherent in the problem of global optimisation.

global optimisation

Formally, for a real-valued blackbox function $f : \mathcal{X} \to \mathbb{R}$, the goal of global optimisation is to locate an input $\mathbf{x} \in \mathcal{X}$ at which it is minimised,

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}).$$

Throughout our presentation, we shall focus on the *minimisation* problem without loss of generality, as any maximisation problem can be translated into a minimisation problem, and vice versa, simply by negating the function of interest. In contrast with classical mathematical optimisation, which frequently rely upon a number of simplifying assumptions, BO is particularly well-equipped to address problems with the following general properties:

OPAQUE. The functions are largely inscrutable, lacking a well-defined functional form or useful closed-form expression (hence, characterised as "black boxes"). Additionally, these functions do not

Algorithm 1: A generic sequential decision-making procedure for optimisation.

Input: blackbox function $f : \mathcal{X} \to \mathbb{R}$, initial dataset $\mathcal{D}_0$.
repeat
  $\mathbf{x}_N \leftarrow \textsc{policy}(\mathcal{D}_{N-1})$ // suggest next candidate location
  $y_N \leftarrow \textsc{evaluate}(\mathbf{x}_N)$ // evaluate f at the suggested location
  $\mathcal{D}_N \leftarrow \mathcal{D}_{N-1} \cup \{(\mathbf{x}_N, y_N)\}$ // update dataset
  $N \leftarrow N + 1$
until *termination condition satisfied*

provide helpful "hints" or "clues" typically exploited by traditional optimisation methods, such as first-order gradients, let alone higher-order derivatives. Lastly, the function is assumed to be nonconvex, which is to say that a local optimum is not automatically considered a globally optimal solution.

EXPENSIVE. The functions are assumed to be costly to evaluate. Since evaluations require substantial resources like time and money, the function cannot be trivially optimised by exhaustive evaluation.

IMPRECISE. The mechanism by which the function is evaluated is assumed to be imperfect, involving randomness, low-fidelity simulation, or indirect observations through noisy measurements.

Simply stated, BO only requires a way to obtain noisy observations of an objective function at suggested locations. It should go without saying that these characteristics are not *preconditions* for BO, but rather represent the complex problem scenarios where BO demonstrates its strength and versatility.

sequential decision-making

Every optimisation procedure boils down to making a series of decisions. In each iteration, we are tasked with deciding which candidate location is the most promising to evaluate next. These decisions must be made in the face of uncertainty, as we cannot know the outcome of an evaluation beforehand, even with access to past observations. Further, the sequential nature of the optimisation process exacerbates the impact of this uncertainty. Any sound optimisation framework must be equipped manage this uncertainty. In light of these considerations, it is helpful to approach BO from the perspective of *Bayesian decision theory* [12, 54], which views it as a principled framework that provides a systematic approach to decision-making under uncertainty tailored for global optimisation. Thus, our remaining treatment of this topic will follow the decision-theoretic introduction provided by Garnett [75].

Bayesian decision theory

The procedure in Algorithm 1 formalises a generic approach to global optimisation. The procedure is initialised with a dataset $\mathcal{D}_0$, which typically consists of a small handful of existing observations

made at randomly-selected locations. For notational simplicity, suppose $\mathcal{D}_0 = \emptyset$. Then, in iteration N, the dataset consists of past observations $\mathcal{D}_N = \{\mathbf{x}_n, y_n\}_{n=1}^N$, where $y_n = f(\mathbf{x}_n) + \varepsilon$ for some additive noise ε. In other words, output y_n is the (inexact) function value at input $\mathbf{x}_n$, assumed to be corrupted by some noise, typically Gaussian distributed $\varepsilon \sim \mathcal{N}(0, \beta^{-1})$, with some precision $\beta > 0$. This effectively leads to the observation model previously introduced in Equations (2.15) and (2.36).

state
optimisation policy
action
outcome

Now, the observed dataset $\mathcal{D}_N$ (which can be viewed as the *state*) is mapped, through an *optimisation policy*, to the candidate location $\mathbf{x}$ to be evaluated next (which can be viewed as an *action*). This location $\mathbf{x}$ is in turn mapped, through evaluation of the blackbox function, to a corresponding value y (which can be viewed as the *outcome*). Finally, the state is updated by appending the new observation $(\mathbf{x}, y)$ to the dataset, and the process is repeated until the termination criteria are met.

The optimisation policy varies along two principal axes. They are either: (1) *deterministic* or *stochastic*, and (2) *adaptive* or *non-adaptive*. Non-adaptive policies disregard the data, exemplified by methods such as grid search and random search [13], which are in turn representative of deterministic and stochastic policies, respectively. On the other hand, BO methods are driven by adaptive optimisation policies that leverage past data to make informed future decisions.

surrogate model
utility function
acquisition function

Accordingly, a hallmark of BO methods is that they maintain a probabilistic model known as the *surrogate model*, which encapsulates our knowledge and beliefs about the unknown function. These beliefs are continuously updated as new data is acquired, allowing the algorithm to adapt its behaviour to make optimal decisions based on the evolving information. In addition to the surrogate model, often a *utility function* $U(y)$ is specified to encode our preferences for the kinds of observations that are considered useful. These preferences are connected to the posterior beliefs, through the surrogate model's posterior predictive density $p(y \mid \mathbf{x}, \mathcal{D}_N)$, to form the *acquisition function* $\alpha(\mathbf{x}; \mathcal{D}_N)$, which serves as a criterion or score for candidate locations, indicating the benefit they bring to the optimisation procedure. Ultimately, the optimisation policy produces the maximiser of the acquisition function,

$$\textsc{policy} : \mathcal{D}_N \mapsto \arg\max_{\mathbf{x} \in \mathcal{X}} \alpha(\mathbf{x}; \mathcal{D}_N).$$

The reason that this approach works at all (namely, optimising a function by optimising yet another function) is that the acquisition function is designed to be more manageable than the unknown function $f(\mathbf{x})$. Specifically, the acquisition function is usually relatively inexpensive to evaluate, possesses closed-form expressions, and offers analytically tractable gradients. As a result, they can be optimised efficiently using conventional, readily available mathematical optimisation methods.

All acquisition functions try to negotiate between the opposing forces of exploration and exploitation. In the context of optimisation, exploitation is the tendency to favour locations where the function value is expected to be low (assuming the goal is minimisation), while exploration is the tendency to favour locations where there is a high degree of uncertainty concerning the function value, enabling the acquisition of more data to improve the model and make more informed decisions in the future. The key to an effective optimisation approach lies in striking a balance within the acquisition function, ensuring that neither force overpowers the other.

In the remainder of this section, we provide an overview of the key components we have introduced, namely, the surrogate model and acquisition function. In particular, we examine the main considerations for their design and discuss several proven approaches.

Before moving on, a quick word on notation: throughout the earlier chapters we have used $p(f_* \mid \mathbf{y})$ to denote the posterior predictive density. This is itself a shorthand for $p(f_* \mid \mathbf{x}_*, \mathbf{X}, \mathbf{y})$, which, considering that $\mathcal{D}_N$ is another way to denote $(\mathbf{X}, \mathbf{y})$, is not too dissimilar to the $p(y \mid \mathbf{x}, \mathcal{D}_N)$ notation used here. In the present context, the asterisks are no longer required for the purpose of distinguishing unseen test points as the observations are instead disambiguated by indexed subscripts (i. e., $\mathbf{x}_n, y_n$).

2.5.1 *Surrogate Models*

From the high-level description of BO we have presented above, it shouldn't be difficult to appreciate the importance of having a consistent framework for systematically reasoning about unknown functions. Therefore, it is not surprising that GPs have emerged as the predominant model family in BO. Indeed, BO is often regarded as the "killer application" for GPs.

GPs possess several compelling characteristics that make them an ideal choice as surrogate models for BO. First and foremost, GP models offer reliable and well-calibrated predictive uncertainty estimates, which has proven to be of crucial importance in practice [228]. Second, to be specific, the GP regression model with Gaussian noise (i. e., the "textbook" version described in Section 2.4) stands out as a rare example of a highly-flexible model that retains its analytical tractability. Notably, both the posterior predictive density and the marginal likelihood can be computed analytically – see Equations (2.16) and (2.20). This tractability is crucial, as eliminating the need for approximate inference implies not having to compromise on the quality and accuracy of uncertainty quantification and hyperparameter estimation for Bayesian model selection. Despite their favorable tractability properties, GPs remain highly expressive, incorporating only a limited set of assumptions related to smoothness, stationarity, and characteristic

lengthscales. These assumptions are generally mild and do not impose significant restrictions in most problems. On the contrary, such assumptions often prove beneficial in many real-world optimisation problems.

With this being said, it's always possible to find counterexamples of problem scenarios in BO for which GPs are ill-suited. When the unknown function is believed to exhibit nonstationary behaviour, augmenting a stationary covariance function by warping the inputs through a nonlinear mapping can create a more expressive nonstationary covariance function. Notable examples of such warping functions include using cdfs that are flexible yet succinctly parameterised [238] or employing deep neural networks (DNNs) [28, 288]. Similarly, when the measurement error is believed to be heteroscedastic, extensions can applied to the observation model [89, 159]. In more complex scenarios involving discrete (ordered and unordered) inputs [77], sequential inputs [178], or structured inputs with conditional dependencies [116], it can be challenging to devise useful covariance functions. It goes without saying that even the most promising approaches introduce a significant footprint to the framework, not least in terms of computational overhead or additional parameters to contend with. Moreover, none of this makes mention of the fact that there is no straightforward workaround for the more fundamental limitation of exact GP regression, which has a computational cost that scales cubically with the number of observations. This limitation precludes running BO for extended horizons on problems that require numerous evaluations to reach a global optimum. While the sparse GP approximations described in Section 2.4.2 can be readily applied, it is essential to allocate the inducing points properly [179]. Neglecting this careful allocation often leads to impractical solutions with degraded performance due to poorly calibrated uncertainty estimates [228].

nonstationarity
input warping

heteroscedasticity

If resorting to approximations becomes inevitable, it stands to reason that leveraging alternative estimators that are explicitly designed to address these specific problem scenarios could potentially provide greater advantages. For example, when dealing with functions involving discrete or structured inputs or high-dimensionalities, ensembles of decision tree regressors such as extreme gradient-boosting (XGBOOST) [34] and random forests (RFS) [19] offer attractive alternatives. In particular, RFS underpin the popular sequential model-based algorithm configuration (SMAC) method [111]. In a similar vein, the tree-structured Parzen estimator (TPE) method [14], on which we expand further in Chapter 5, has also enjoyed considerable success. These approaches can handle complex input structures and have proven effective in various applications, particularly in hyperparameter optimisation (HPO) for automated machine learning (AUTOML).

Similarly, for modelling nonstationarity, capturing nonlinear behaviour, or handling multi-output functions in settings like multi-

task [255], multi-fidelity [124], or multi-objective [101] optimisation, BNNs provide an attractive choice [200, 237, 243, 281]. The prominent approaches are Bayesian to varying extents. For instance, Snoek et al. [237] consider a Bayesian treatment of only the final layer of weights in a posthoc manner, effectively leading to the BLR model described in Section 2.4.3 with neural network (NN) basis functions. In contrast, Springenberg et al. [243] adopt a more thoroughly Bayesian approach that encompasses all the NN weights, and utilise sampling-based inference, specifically, stochastic gradient Hamiltonian Monte Carlo (SGHMC) [33], to approximate the posterior predictive density. Recent efforts to enhance the performance of BNNs in BO have focused on leveraging the latest advancements in Bayesian deep learning [131, 145].

Thus seen, ensuring tractability of the posterior predictive density often necessitates making compromises in the form of simplifications and crude approximations. Unfortunately, these compromises can often inhibit the expressive power and the range of benefits offered by these alternative surrogate model families. Consequently, there is no model family that can perfectly address all problem scenarios and provide an ideal solution without incurring some trade-offs.

In Chapter 5, we explore an alternative paradigm for BO that circumvents the need for an explicit model of the unknown function, instead focusing on directly approximating the acquisition function. This reframing effectively sidesteps the tractability requirements and opens the door to powerful model families that would otherwise render the predictive density unwieldy or simply intractable to compute.

2.5.2 *Acquisition Functions*

Almost without exception, acquisition functions rely on the predictive density to represent posterior beliefs about the unknown function in order to score the potential benefit of a candidate location. In certain cases, this score incorporates a preference for outcomes specified through a *utility function*. This thesis is primarily concerned with acquisition functions of this nature, so-called the *improvement-based* acquisition functions, such as the well-established probability of improvement (PI) [117] and expected improvement (EI) [176]. Despite the emergence of numerous new and sophisticated acquisition functions like knowledge gradient (KG) [225], entropy search (ES) [96], predictive ES (PES) [102], and their variants [279], the improvement-based acquisition functions remain widely used. Such functions can generally be expressed as an *expectation* of the *utility* function,

improvement-based acquisition function

expected utility

$$\alpha(\mathbf{x}; \mathcal{D}_N, \tau) \triangleq \mathbb{E}_{p(y \mid \mathbf{x}, \mathcal{D}_N)}[U(y; \tau)], \tag{2.45}$$

where τ denotes a parameter representing some threshold and $U(y; \tau)$ denotes a utility function that typically depends on the difference be-

tween τ and y (i. e., the "improvement"). By convention, τ is set to the *incumbent*, the lowest function value observed so far, $\tau = \min_n y_n$ [290].

2.5.2.1 *Probability of Improvement*

In the classical PI acquisition function [117], the utility function simply indicates whether y improves upon some threshold τ,

$$U_{\text{PI}}(y, \tau) \triangleq \mathbb{I}(\tau - y > 0). \tag{2.46}$$

Suppose the posterior predictive density takes the form of a Gaussian

$$p(y \mid \mathbf{x}, \mathcal{D}_N) = \mathcal{N}(y \mid \mu(\mathbf{x}), \sigma^2(\mathbf{x})). \tag{2.47}$$

Then, Equation (2.45) leads to

$$\alpha_{\text{PI}}(\mathbf{x}; \mathcal{D}_N, \tau) = p(y \leq \tau \mid \mathbf{x}, \mathcal{D}_N) = \Psi(Z_\tau(\mathbf{x})), \tag{2.48}$$

where

$$Z_\tau(\mathbf{x}) \triangleq \frac{\tau - \mu(\mathbf{x})}{\sigma(\mathbf{x})},$$

and Ψ denotes the cdf of the standard normal distribution

2.5.2.2 *Expected Improvement (EI)*

In EI [176], the utility function quantifies the nonnegative amount by which y improves upon threshold τ,

$$U_{\text{EI}}(y, \tau) \triangleq \max(\tau - y, 0). \tag{2.49}$$

This is known as the *improvement* utility function. When the predictive density is the Gaussian from Equation (2.47), the expectation from Equation (2.45) is of the improvement utility function (hence the name), and evaluates to

$$\alpha_{\text{EI}}(\mathbf{x}; \mathcal{D}_N, \tau) = \sigma(\mathbf{x}) \cdot \left[Z_\tau(\mathbf{x}) \cdot \Psi(Z_\tau(\mathbf{x})) + \psi(Z_\tau(\mathbf{x})) \right], \tag{2.50}$$

where ψ denotes the pdf of the standard normal distribution.

In Figure 2.10, we plot the EI/PI criteria as functions of the posterior predictive mean $\mu(\mathbf{x})$ and variance $\sigma^2(\mathbf{x})$. We see that the value to which PI assigns $\mathbf{x}$ depends primarily on whether the predictive mean $\mu(\mathbf{x})$ exceeds the threshold, in this example $\tau = 0$, and less so on the predictive variance $\sigma^2(\mathbf{x})$. Furthermore, particularly when the predictive variance is close to zero, the function is essentially piecewise constant with a discontinuity at τ. In other words, and as can be expected from simply looking at its analytical expression alone, PI either rewards a high or low value depending on whether or not $\mu(\mathbf{x})$ exceeds the threshold, but is indifferent to the amount by which it does. In practice, this can lead to the optimisation procedure getting stuck in local optima and inadequately exploring the search space [75]. In

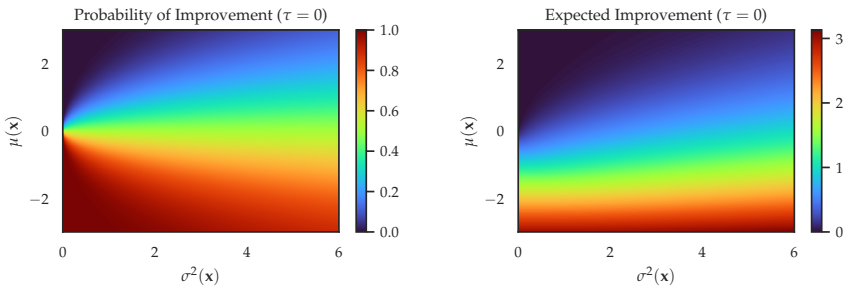


Figure 2.10: Values of improvement-based acquisition functions plotted in terms of the posterior predictive mean $\mu(\mathbf{x})$ and variance $\sigma^2(\mathbf{x})$. PI (*left*) heavily favours exploitation while EI (*right*) strikes a slightly better balance between exploitation and exploration.

contrast, the EI criterion does take into account the amount by which a candidate location in expectation improves upon the threshold. Furthermore, broadly speaking, for any given fixed value of $\mu(\mathbf{x})$, the reward assigned by EI increases as the uncertainty, or, more precisely, the variance $\sigma^2(\mathbf{x})$, increases. Thus seen, EI is less prone than PI to exploit too aggressively to its own detriment.

While the exact expressions of Equations (2.48) and (2.50) are both easy to evaluate and optimise, the conditions necessary to satisfy Equation (2.47) can often come at the expense of flexibility and expressiveness. In Chapter 5, we will consider an altogether different way to express PI/EI themselves.

2.5.2.3 *Upper/Lower Confidence Bound*

The upper confidence bound (UCB) [244] function is another popular criterion. UCB has its roots in the multi-armed bandits literature [135] and come with favorable theoretical properties and provable regret bounds. To maintain consistency with our running context of function *minimisation*, we shall discuss the LCB. Like PI/EI the LCB criterion can also be expressed in terms of the predictive mean and variance μ and σ^2,

$$\alpha_{\text{LCB}}(\mathbf{x}; \mathcal{D}_N, \lambda) \triangleq -\mu(\mathbf{x}) + \sqrt{\lambda} \cdot \sigma(\mathbf{x}),$$

where, similar to τ in the improvement-based criteria, λ is a parameter that controls the tendency to explore. Interestingly, UCB/LCB cannot be expressed in terms of the expected utility from Equation (2.45). UCB/LCB is known as an *optimistic* acquisition function, since, by design, it behaves optimistically in the presence of uncertainty. Indeed, from Figure 2.11, we readily see that it assigns greater value to locations $\mathbf{x}$ where the level of uncertainty, or, more precisely, the predictive variance $\sigma^2(\mathbf{x})$, is high.

optimistic acquisition function

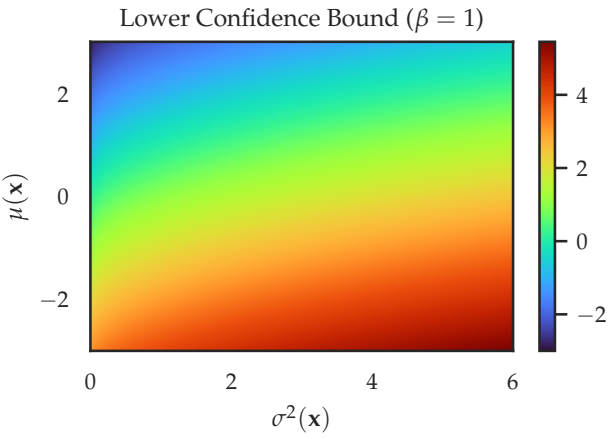


Figure 2.11: Values of the lower confidence bound (LCB) criterion with $\lambda = 1$ plotted in terms of the posterior predictive mean $\mu(\mathbf{x})$ and variance $\sigma^2(\mathbf{x})$. LCB is said to favour exploration, since it behaves optimistically in the face of uncertainty – a higher value is assigned to regions where the variance $\sigma^2(\mathbf{x})$ is large.

2.5.2.4 *Thompson Sampling*

Thompson sampling, a widely-used optimisation policy in BO, was adapted for continuous optimisation from a policy originally proposed for the multi-armed bandit problem almost a century ago [258].

Unlike the acquisition functions discussed earlier, which represented adaptive, *deterministic* policies, Thompson sampling is an adaptive, *stochastic* policy. Like previous acquisition functions, it still depends on the posterior predictive distribution, but does not explicitly involve the predictive mean and variance. Instead, Thompson sampling involves realisations of the unknown objective function randomly sampled from the predictive distribution itself,

$$\alpha_{\mathrm{TS}}(\mathbf{x}; \mathcal{D}_N) \triangleq f(\mathbf{x}), \qquad f \sim p(f_* \mid \mathbf{x}_*, \mathbf{X}, \mathbf{y}).$$

In other words, while the improvement-based polices from Sections 2.5.2.1 and 2.5.2.2 select the best candidate solution *in expectation* by averaging over the objective functions, Thompson sampling determines the best candidate solution according to a *randomly sampled* objective function. Thus seen, this approach balances exploration and exploitation by sampling observations proportional to their probability of optimality, effectively encouraging exploitation, while the stochasticity inherent in random sampling ensures exploration [75].

In practice, sampling random functions from a GP that can be evaluated at arbitrary points, let alone efficiently optimised, poses a considerable challenge. Consequently, a dominant approach adopts the weight-space perspective of GPs, leveraging its spectral decomposition to obtain a posterior weight density. Weights $\mathbf{w}$ can be sampled efficiently from this posterior and used to construct random functions

$f(\mathbf{x}) \triangleq \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x})$ that are (approximately) equal in distribution to GP posterior samples, yet are easy to manipulate and optimise [102, 228, 289, 291]. In Appendix A, we explore the use of various numerical integration methods to further improve the computational efficiency of sampling random functions from GP posteriors.

2.6 SUMMARY

This chapter laid the essential groundwork for our thesis by introducing fundamental concepts in probabilistic modelling, Bayesian statistics, and variational inference. We highlighted the role of statistical divergences and density-ratio estimation in approximate inference, establishing a foundation for advanced topics in probabilistic ML. Our discussion also included Gaussian processes and their sparse approximations based on VI, concluding with the basic concepts behind Bayesian optimisation.

Our discussion of Gaussian processes and variational inference set the stage for our subsequent exploration of orthogonally-decoupled sparse Gaussian processes with spherical neural network activation features. This forms the focus of Chapter 3, representing a unique integration neural networks with Gaussian processes. Similarly, our examination of variational inference, the variational estimation of f-divergences, and density-ratio estimation, laid the groundwork for a new derivation of CYCLEGANS from the perspective of approximate Bayesian inference, which we examine in Chapter 4. Lastly, the basic concepts of density-ratio estimation and Bayesian optimisation introduced here forms the basis for our model-agnostic approach to Bayesian optimisation based on binary classification, which we discuss in Chapter 5.

In summary, this chapter provides the the necessary foundation for the advanced methodologies described in the subsequent chapters, bridging fundamental principles with new perspectives in probabilistic ML.

ADDENDUM

2.A KL DIVERGENCE SIMPLIFICATION

The KL divergence simplifies as follows:

$$\begin{aligned}
&\mathrm{KL}\left[q(\mathbf{f}^*, \mathbf{f}, \mathbf{u}) \parallel p(\mathbf{f}^*, \mathbf{f}, \mathbf{u} \mid \mathbf{y})\right] \\
&\quad = \iiint p(\mathbf{f}^* \mid \mathbf{f}, \mathbf{u}) q(\mathbf{f}, \mathbf{u}) \log \frac{\cancel{p(\mathbf{f}^* \mid \mathbf{f}, \mathbf{u})} q(\mathbf{f}, \mathbf{u})}{\cancel{p(\mathbf{f}^* \mid \mathbf{f}, \mathbf{u})} p(\mathbf{f}, \mathbf{u} \mid \mathbf{y})} \, \mathrm{d}\mathbf{f}^* \mathrm{d}\mathbf{f} \mathrm{d}\mathbf{u} \\
&\quad = \iint q(\mathbf{f}, \mathbf{u}) \log \frac{q(\mathbf{f}, \mathbf{u})}{p(\mathbf{f}, \mathbf{u} \mid \mathbf{y})} \, \mathrm{d}\mathbf{f} \mathrm{d}\mathbf{u} = \mathrm{KL}\left[q(\mathbf{f}, \mathbf{u}) \parallel p(\mathbf{f}, \mathbf{u} \mid \mathbf{y})\right].
\end{aligned}$$

2.B OPTIMAL VARIATIONAL DISTRIBUTION FOR GENERAL LIKELIHOODS

We have

$$\begin{aligned}
\mathrm{ELBO}(q) &= \iint p(\mathbf{f} \mid \mathbf{u}) q(\mathbf{u}) \log p(\mathbf{y} \mid \mathbf{f}) \mathrm{d}\mathbf{f} \mathrm{d}\mathbf{u} + \iint p(\mathbf{f} \mid \mathbf{u}) q(\mathbf{u}) \log \frac{p(\mathbf{u})}{q(\mathbf{u})} \mathrm{d}\mathbf{f} \mathrm{d}\mathbf{u} \\
&= \int q(\mathbf{u}) \left(\int p(\mathbf{f} \mid \mathbf{u}) \log p(\mathbf{y} \mid \mathbf{f}) \mathrm{d}\mathbf{f} \right) \mathrm{d}\mathbf{u} + \int q(\mathbf{u}) \log \frac{p(\mathbf{u})}{q(\mathbf{u})} \mathrm{d}\mathbf{u} \\
&= \int q(\mathbf{u}) \log F(\mathbf{y}, \mathbf{u}) \mathrm{d}\mathbf{u} + \int q(\mathbf{u}) \log \frac{p(\mathbf{u})}{q(\mathbf{u})} \mathrm{d}\mathbf{u} \\
&= \int q(\mathbf{u}) \log \frac{F(\mathbf{y}, \mathbf{u}) p(\mathbf{u})}{q(\mathbf{u})} \mathrm{d}\mathbf{u}.
\end{aligned}$$

Taking the functional derivative of the ELBO wrt to $q(\mathbf{u})$, we get

$$\begin{aligned}
\frac{\partial}{\partial q(\mathbf{u})} \mathrm{ELBO}(q) &= \frac{\partial}{\partial q(\mathbf{u})} \left(\int \log \frac{F(\mathbf{y}, \mathbf{u}) p(\mathbf{u})}{q(\mathbf{u})} q(\mathbf{u}) \, \mathrm{d}\mathbf{u} \right) \\
&= \int \frac{\partial}{\partial q(\mathbf{u})} \left(\log \frac{F(\mathbf{y}, \mathbf{u}) p(\mathbf{u})}{q(\mathbf{u})} q(\mathbf{u}) \right) \mathrm{d}\mathbf{u} \\
&= \int \log \frac{F(\mathbf{y}, \mathbf{u}) p(\mathbf{u})}{q(\mathbf{u})} \left(\frac{\partial}{\partial q(\mathbf{u})} q(\mathbf{u}) \right) + \\
&\qquad q(\mathbf{u}) \left(\frac{\partial}{\partial q(\mathbf{u})} \log \frac{F(\mathbf{y}, \mathbf{u}) p(\mathbf{u})}{q(\mathbf{u})} \right) \mathrm{d}\mathbf{u} \\
&= \int \log \frac{F(\mathbf{y}, \mathbf{u}) p(\mathbf{u})}{q(\mathbf{u})} + q(\mathbf{u}) \left(-\frac{1}{q(\mathbf{u})} \right) \mathrm{d}\mathbf{u} \\
&= \int \log F(\mathbf{y}, \mathbf{u}) + \log p(\mathbf{u}) - \log q(\mathbf{u}) - 1 \, \mathrm{d}\mathbf{u}.
\end{aligned}$$

Setting this expression to zero, we obtain

$$\begin{aligned}
&\log q^*(\mathbf{u}) = \log F(\mathbf{y}, \mathbf{u}) + \log p(\mathbf{u}) - 1 \\
\Rightarrow \qquad &q^*(\mathbf{u}) \propto F(\mathbf{y}, \mathbf{u}) p(\mathbf{u}).
\end{aligned}$$

2.C INTERMEDIATE LOWER BOUND FOR GAUSSIAN LIKELIHOODS

To carry out this derivation, we will need to recall the following two straightforward identities. First, we can express the inner product between two vectors as the trace of their outer product,

$$\mathbf{a}^\top \mathbf{b} = \operatorname{tr}(\mathbf{a}\mathbf{b}^\top).$$

Second, we have the following relationship between the covariance matrix $\operatorname{Cov}[\mathbf{a}]$ and the auto-correlation matrix $\mathbb{E}[\mathbf{a}\mathbf{a}^\top]$,

$$\begin{aligned} & \operatorname{Cov}[\mathbf{a}] = \mathbb{E}[\mathbf{a}\mathbf{a}^\top] - \mathbb{E}[\mathbf{a}]\,\mathbb{E}[\mathbf{a}]^\top \\ \Leftrightarrow \quad & \mathbb{E}[\mathbf{a}\mathbf{a}^\top] = \operatorname{Cov}[\mathbf{a}] + \mathbb{E}[\mathbf{a}]\,\mathbb{E}[\mathbf{a}]^\top \end{aligned}$$

Additionally, let's denote the mean and covariance of the prior conditional $p(\mathbf{f} \mid \mathbf{u})$ in Equation (2.22) as

$$\mathbf{b} \triangleq \mathbf{Q}_{\mathbf{fu}}\mathbf{u}, \qquad \text{and} \qquad \mathbf{S}_{\mathbf{ff}} \triangleq \mathbf{K}_{\mathbf{ff}} - \mathbf{Q}_{\mathbf{ff}},$$

respectively. Together, these allow us to write

$$\begin{aligned} \log F(\mathbf{y}, \mathbf{u}) &= \int \log \mathcal{N}(\mathbf{y}|\mathbf{f}, \beta^{-1}\mathbf{I}) \mathcal{N}(\mathbf{f} \mid \mathbf{b}, \mathbf{S}_{\mathbf{ff}})\, \mathrm{d}\mathbf{f} \\ &= -\frac{\beta}{2} \int (\mathbf{y}-\mathbf{f})^\top (\mathbf{y}-\mathbf{f}) \mathcal{N}(\mathbf{f} \mid \mathbf{b}, \mathbf{S}_{\mathbf{ff}})\, \mathrm{d}\mathbf{f} - \frac{N}{2}\log(2\pi\beta^{-1}) \\ &= -\frac{\beta}{2} \int \operatorname{tr}\left(\mathbf{y}\mathbf{y}^\top - 2\mathbf{y}\mathbf{f}^\top + \mathbf{f}\mathbf{f}^\top\right) \mathcal{N}(\mathbf{f} \mid \mathbf{b}, \mathbf{S}_{\mathbf{ff}})\, \mathrm{d}\mathbf{f} - \frac{N}{2}\log(2\pi\beta^{-1}) \\ &= -\frac{\beta}{2} \operatorname{tr}\left(\mathbf{y}\mathbf{y}^\top - 2\mathbf{y}\mathbf{b}^\top + \mathbf{S}_{\mathbf{ff}} + \mathbf{b}\mathbf{b}^\top\right) - \frac{N}{2}\log(2\pi\beta^{-1}) \\ &= -\frac{\beta}{2}(\mathbf{y}-\mathbf{b})^\top(\mathbf{y}-\mathbf{b}) - \frac{N}{2}\log(2\pi\beta^{-1}) - \frac{\beta}{2}\operatorname{tr}(\mathbf{S}_{\mathbf{ff}}) \\ &= \log \mathcal{N}(\mathbf{y} \mid \mathbf{b}, \beta^{-1}\mathbf{I}) - \frac{\beta}{2}\operatorname{tr}(\mathbf{S}_{\mathbf{ff}}). \end{aligned}$$

Therefore, we have

$$F(\mathbf{y}, \mathbf{u}) = \mathcal{N}(\mathbf{y} \mid \mathbf{b}, \beta^{-1}\mathbf{I}) \times \exp\left(-\frac{\beta}{2}\operatorname{tr}(\mathbf{S}_{\mathbf{ff}})\right). \tag{2.51}$$

as required.

2.D OPTIMAL VARIATIONAL DISTRIBUTION FOR GAUSSIAN LIKELIHOODS

Firstly, the optimal variational distribution can be found in closed-form as

$$\begin{aligned} q^*(\mathbf{u}) &\propto F(\mathbf{y}, \mathbf{u}) p(\mathbf{u}) \\ &\propto \mathcal{N}(\mathbf{y} \mid \mathbf{Q}_{\mathbf{fu}}\mathbf{u}, \beta^{-1}\mathbf{I}) \mathcal{N}(\mathbf{u} \mid \mathbf{0}, \mathbf{K}_{\mathbf{uu}}) \\ &\propto \exp\left(-\frac{\beta}{2}(\mathbf{y}-\mathbf{Q}_{\mathbf{fu}}\mathbf{u})^\top(\mathbf{y}-\mathbf{Q}_{\mathbf{fu}}\mathbf{u}) - \frac{1}{2}\mathbf{u}^\top \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{u}\right) \\ &\propto \exp\left(-\frac{1}{2}\left(\mathbf{u}^\top \boldsymbol{\Lambda}\mathbf{u} - 2\beta(\mathbf{Q}_{\mathbf{uf}}\mathbf{y})^\top \mathbf{u}\right)\right), \end{aligned}$$

where

$$\mathbf{\Lambda} \triangleq \mathbf{K}_{\mathbf{uu}}^{-1} + \beta \mathbf{Q}_{\mathbf{uf}} \mathbf{Q}_{\mathbf{fu}} = \mathbf{K}_{\mathbf{uu}}^{-1} (\mathbf{K}_{\mathbf{uu}} + \beta \mathbf{K}_{\mathbf{uf}} \mathbf{K}_{\mathbf{fu}}) \mathbf{K}_{\mathbf{uu}}^{-1}.$$

By completing the square, we get

$$\begin{aligned} q^*(\mathbf{u}) &\propto \exp\left(-\frac{1}{2}(\mathbf{u} - \beta \mathbf{\Lambda}^{-1} \mathbf{Q}_{\mathbf{uf}} \mathbf{y})^\top \mathbf{\Lambda} (\mathbf{u} - \beta \mathbf{\Lambda}^{-1} \mathbf{Q}_{\mathbf{uf}} \mathbf{y})\right) \\ &\propto \mathcal{N}(\mathbf{u} \,|\, \beta \mathbf{\Lambda}^{-1} \mathbf{Q}_{\mathbf{uf}} \mathbf{y}, \mathbf{\Lambda}^{-1}). \end{aligned}$$

If we define

$$\mathbf{M} \triangleq \mathbf{K}_{\mathbf{uu}} + \beta \mathbf{K}_{\mathbf{uf}} \mathbf{K}_{\mathbf{fu}}$$

so that

$$\mathbf{\Lambda} = \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{M} \mathbf{K}_{\mathbf{uu}}^{-1},$$

we finally get

$$q^*(\mathbf{u}) = \mathcal{N}(\mathbf{u} \,|\, \beta \mathbf{K}_{\mathbf{uu}} \mathbf{M}^{-1} \mathbf{K}_{\mathbf{uf}} \mathbf{y}, \mathbf{K}_{\mathbf{uu}} \mathbf{M}^{-1} \mathbf{K}_{\mathbf{uu}}),$$

as required.

2.E COLLAPSED LOWER BOUND FOR GAUSSIAN LIKELIHOODS

We have

$$\begin{aligned} \text{ELBO}(q^*) &= \log\left(\int p(\mathbf{u}) F(\mathbf{y}, \mathbf{u}) \, \mathrm{d}\mathbf{u}\right) \\ &= \log\left[\exp\left(-\frac{\beta}{2} \operatorname{tr}(\mathbf{S}_{\mathbf{ff}})\right) \int \mathcal{N}(\mathbf{y} \,|\, \mathbf{Q}_{\mathbf{fu}} \mathbf{u}, \beta^{-1} \mathbf{I}) p(\mathbf{u}) \, \mathrm{d}\mathbf{u}\right] \\ &= \log \int \mathcal{N}(\mathbf{y} \,|\, \mathbf{Q}_{\mathbf{fu}} \mathbf{u}, \beta^{-1} \mathbf{I}) \mathcal{N}(\mathbf{u} \,|\, \mathbf{0}, \mathbf{K}_{\mathbf{uu}}) \, \mathrm{d}\mathbf{u} - \frac{\beta}{2} \operatorname{tr}(\mathbf{S}_{\mathbf{ff}}) \\ &= \log \mathcal{N}(\mathbf{y} \,|\, \mathbf{0}, \beta^{-1} \mathbf{I} + \mathbf{Q}_{\mathbf{fu}} \mathbf{K}_{\mathbf{uu}} \mathbf{Q}_{\mathbf{uf}}) - \frac{\beta}{2} \operatorname{tr}(\mathbf{S}_{\mathbf{ff}}) \\ &= \log \mathcal{N}(\mathbf{y} \,|\, \mathbf{0}, \mathbf{Q}_{\mathbf{ff}} + \beta^{-1} \mathbf{I}) - \frac{\beta}{2} \operatorname{tr}(\mathbf{S}_{\mathbf{ff}}). \end{aligned}$$

2.F SPECTRAL DENSITY OF THE SQUARED EXPONENTIAL KERNEL

We calculate the spectral density for the SE kernel in 1D from Equation (2.17). Using Equation (2.39), we have

$$\begin{aligned} p(\omega) &= \frac{1}{2\pi} \int k(t, 0) e^{i\omega t} \, \mathrm{d}t \\ &= \frac{\ell}{\sqrt{2\pi}} \int \mathcal{N}(t \,|\, 0, \ell^2) e^{i\omega t} \, \mathrm{d}t \\ &= \frac{\ell}{\sqrt{2\pi}} \exp\left(-\frac{1}{2} \ell^2 \omega^2\right) = \mathcal{N}\left(\omega \,|\, 0, \ell^{-2}\right), \end{aligned}$$

as required.

2.G COSINE DIFFERENCE AS INNER PRODUCT

Firstly, recall the *angle sum-and-difference* trigonometric identity,

$$\cos \alpha \pm \beta = \cos \alpha \cos \beta \mp \sin \alpha \sin \beta. \tag{2.52}$$

Taking the inner product of ψ_ω evaluated at inputs $\mathbf{x}$ and $\mathbf{x}'$, we obtain

$$\begin{aligned} \psi_\omega(\mathbf{x})^\top \psi_\omega(\mathbf{x}') &= \cos\left(\boldsymbol{\omega}^\top \mathbf{x}\right) \cos\left(\boldsymbol{\omega}^\top \mathbf{x}'\right) + \sin\left(\boldsymbol{\omega}^\top \mathbf{x}\right) \sin\left(\boldsymbol{\omega}^\top \mathbf{x}'\right) \\ &= \cos\left(\boldsymbol{\omega}^\top (\mathbf{x} - \mathbf{x}')\right), \end{aligned} \tag{2.53}$$

as required.

3

ORTHOGONALLY-DECOUPLED SPARSE GAUSSIAN PROCESSES WITH SPHERICAL NEURAL NETWORK ACTIVATION FEATURES

PREFACE

This chapter is derived from work previously published as:

Louis C Tiao et al. "BORE: Bayesian Optimization by Density-Ratio Estimation". 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. 10289–10300. URL: https://proceedings.mlr.press/v139/tiao21a.html. **(Accepted as *Oral* presentation).**

3.1 INTRODUCTION

Gaussian processes (GPs) provide a versatile and robust framework for modelling unknown functions, offering data efficiency, flexible encoding of prior beliefs, and reliable uncertainty estimation. Their broad application in sequential decision-making makes them invaluable in diverse fields of ML and optimisation.

In spite of their many advantages, GPs are often compared unfavourably to deep NNs for their poor scalability to large datasets, and their inability to capture rich hierarchies of abstract representations [28, 192, 288]. While GPs are the infinite-width limit of NNs and therefore, in theory, have infinitely more basis functions [185], these basis functions are static and fully determined by the covariance function [156]. This makes it difficult for GPs to flexibly adapt to complex and structured data from which it is beneficial for the basis functions to learn and encode useful representations.

Considerable research effort has been devoted to sparse approximations for GPs [46, 204, 226, 234]. Not least of these is sparse variational GPs (SVGPs) [98, 99, 262], which we examined in Section 2.4. Such advances have not only improved the scalability of GPs, but also unlocked more flexibility in model specification. In particular, the use of *inter-domain* inducing variables in SVGP [140] effectively equips the GP approximation with data-dependent basis functions. Recent works have exploited this to construct a new family of SVGP models in which the basis functions correspond to activations of a feed-forward NN [66, 252]. By stacking multiple layers to form a DGP [49], the propagation of the predictive distribution accurately resembles a forward-pass through a deep NN.

In this chapter, we show that while this approach results in a posterior predictive with a more expressive mean, the variance estimate is typically less accurate and tends to be over-dispersed. Additionally, we examine some practical challenges associated with this method, such as limitations on the use of certain popular kernel and NN activation choices. To address these issues, we propose an extension that aims to mitigate these limitations. Specifically, when viewed from the function-space perspective, the posterior predictive of SVGP depends on a single set of basis functions that is determined by only a finite collection of inducing variables. Recent advances introduce an orthogonal set of basis functions as a means of capturing additional variations remaining from the standard basis [36, 223, 230]. We extend this framework by introducing inter-domain variables to construct more flexible data-dependent basis functions for both the standard and orthogonal components. In particular, we show that incorporating NN activation inducing functions under this framework is an effective way to ameliorate the aforementioned shortcomings. Our experiments on numerous benchmark datasets demonstrate that this extension leads to improvements in predictive performance against comparable alternatives.

3.2 INTER-DOMAIN INDUCING FEATURES

Recall from Equation (2.24) that the test predictive density at unseen points $\mathbf{f}_* \triangleq f(\mathbf{X}_*)$ is

$$q(\mathbf{f}_*) = \mathcal{N}\left(\mathbf{f}_* \,|\, \mathbf{Q}_{*\mathbf{u}}\mathbf{m}_\mathbf{u}, \mathbf{K}_{**} - \mathbf{Q}_{*\mathbf{u}}(\mathbf{K}_{\mathbf{uu}} - \mathbf{C}_\mathbf{u})\mathbf{Q}_{\mathbf{u}*}\right), \tag{3.1}$$

where parameters $\mathbf{m}_\mathbf{u}$ and $\mathbf{C}_\mathbf{u}$ are free parameters. In the RKHS associated with k, this predictive density has a dual representation in which the mean and covariance share the same basis determined by $\mathbf{u}$ [36, 223]. More specifically, the basis function is effectively the vector-valued function $\mathbf{k}_\mathbf{u} : \mathcal{X} \to \mathbb{R}^M$ whose m-th component is defined as

$$[\mathbf{k}_\mathbf{u}(\mathbf{x})]_m \triangleq \mathrm{Cov}\left(f(\mathbf{x}), u_m\right). \tag{3.2}$$

In the standard definition of inducing points as presented in Section 2.4.2,

$$[\mathbf{k}_\mathbf{u}(\mathbf{x})]_m = k(\mathbf{z}_m, \mathbf{x}),$$

therefore, the basis function is solely determined by the covariance function k and the local influence of pseudo-input $\mathbf{z}_m$.

inter-domain inducing features

Inter-domain inducing features are a generalisation of standard inducing variables in which each variable u_m is defined through the transformation of $f(\cdot)$ by

$$u_m \triangleq L_m[f].$$

for some linear operator $L_m : \mathbb{R}^{\mathcal{X}} \to \mathbb{R}$. A particularly useful operator is the integral transform,

$$L_m[f] \triangleq \int_{\mathcal{X}} f(\mathbf{x})\phi_m(\mathbf{x})\,\mathrm{d}\mathbf{x},$$

which was originally employed by Lázaro-Gredilla and Figueiras-Vidal [140]. Refer to the manuscript of Wilk et al. [283] for a more thorough and contemporary treatment. A closely related form is the scalar projection of f onto some ϕ_m in the RKHS $\mathcal{H}$ of k,

$$L_m[f] \triangleq \langle f, \phi_m \rangle_{\mathcal{H}}, \tag{3.3}$$

which leads to

$$[\mathbf{k}_{\mathbf{u}}(\mathbf{x})]_m = \phi_m(\mathbf{x})$$

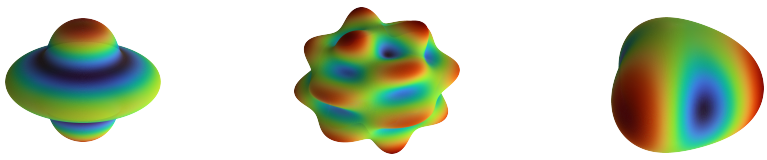
by the reproducing property of the RKHS. This, in effect, equips the GP approximation with adaptive basis functions ϕ_m that are not solely determined by a fixed kernel, and suitable choices can lead to sparser representations and considerable computational benefits [25, 65, 97, 253].

reproducing property

3.2.1 *Spherical Harmonics Inducing Features*

An instance of inter-domain features in the form of Equation (3.3) are the variational Fourier features (VFFS) [97], in which ϕ_m form an orthogonal basis of trigonometric functions. This formulation offers significant computational advantages but scales poorly beyond a small handful of dimensions. To address this, Dutordoir, Durrande, and Hensman [65] propose a generalisation of VFFS using the spherical harmonics for ϕ_m, which can be viewed as a multi-dimensional extension of the Fourier basis.

spherical harmonics



(a) Zonal: $Y_{4,0}$ (b) Tesseral: $Y_{8,3}$ (c) Sectorial: $Y_{3,3}$

Figure 3.1: A visual representation of three different *surface harmonics of the first kind*. This set of examples originates from the monograph of Efthimiou and Frye [67].

The construction relies on the Mercer's decomposition of *zonal* kernels, which can be seen as the analog of stationary kernels in Euclidean spaces, but for hyperspheres. They can be expressed as

Mercer's decomposition

zonal kernels

$k(\mathbf{x}, \mathbf{x}') = \kappa\left(\tilde{\mathbf{x}}^\top \tilde{\mathbf{x}}'\right)$ for some *shape function* $\kappa : [-1, 1] \to \mathbb{R}$, where $\tilde{\boldsymbol{\eta}} \triangleq \frac{\boldsymbol{\eta}}{\|\boldsymbol{\eta}\|} \in \mathbb{S}^{d-1}$ for any $\boldsymbol{\eta} \in \mathbb{R}^d$. Loosely speaking, just as stationary kernels are determined by the *distance* between inputs, zonal kernels depend only on the *angle* between inputs.

The spherical harmonics form an orthonormal basis on $L_2(\mathbb{S}^{d-1})$ consisting of the eigenfunctions of the kernel operator $\mathcal{K}$,

$$\mathcal{K} Y_{\ell,j} = a_\ell Y_{\ell,j},$$

where $Y_{\ell,j}$ is the spherical harmonic of level ℓ and order j, and a_ℓ is the corresponding eigenvalue, or, Fourier coefficient.

A visualisation is shown in Figure 3.1. The spherical harmonics are tricky to visualise not only in higher dimensions, but even in three dimensions, because they are, in general, complex-valued. Here we show, in three dimensions, several *surface harmonics of the first kind*, which correspond to the real or imaginary parts of the spherical harmonics, depending on the value of j. The surface harmonics can be further divided into the categories of *zonal* which are of the form $Y_{\ell,0}$, *tesseral* $Y_{\ell,j}$ for $j \neq 0$, and *sectorial* $Y_{\ell,\ell}$. Each of the surface harmonics shown here is a representative example of its respective category.

Conveniently, by the Funk-Hecke theorem, the Fourier coefficient a_ℓ amounts to the one-dimensional integral

$$a_\ell = \frac{\Omega_d}{C_\ell^{(\alpha)}(1)} \int_{-1}^{1} \kappa(t) C_\ell^{(\alpha)}(t) (1 - t^2)^{\frac{d-3}{2}} \, \mathrm{d}t,$$

where $C_\ell^{(\alpha)}$ is the Gegenbauer polynomial of degree ℓ and $\alpha \triangleq (d-1)/2$. Now, the number $J(d, \ell)$ of spherical harmonics that exist at a given level ℓ is determined by the multiplicity of eigenvalue a_ℓ.

Thus, $\kappa(t)$ can be represented by

$$\kappa(t) = \|\boldsymbol{\xi}\| \|\boldsymbol{\xi}'\| \sum_{\ell=0}^{\infty} \sum_{j=1}^{J(d,\ell)} a_\ell Y_{\ell,j}(\tilde{\boldsymbol{\xi}}) Y_{\ell,j}(\tilde{\boldsymbol{\xi}}'), \tag{3.4}$$

where $t \triangleq \tilde{\boldsymbol{\xi}}^\top \tilde{\boldsymbol{\xi}}'$ for $\boldsymbol{\xi}, \boldsymbol{\xi}' \in \mathbb{R}^d$. We refer the reader to the manuscript of Dutordoir et al. [66, Appendix B] for a concise summary of spherical harmonics in multiple dimensions.

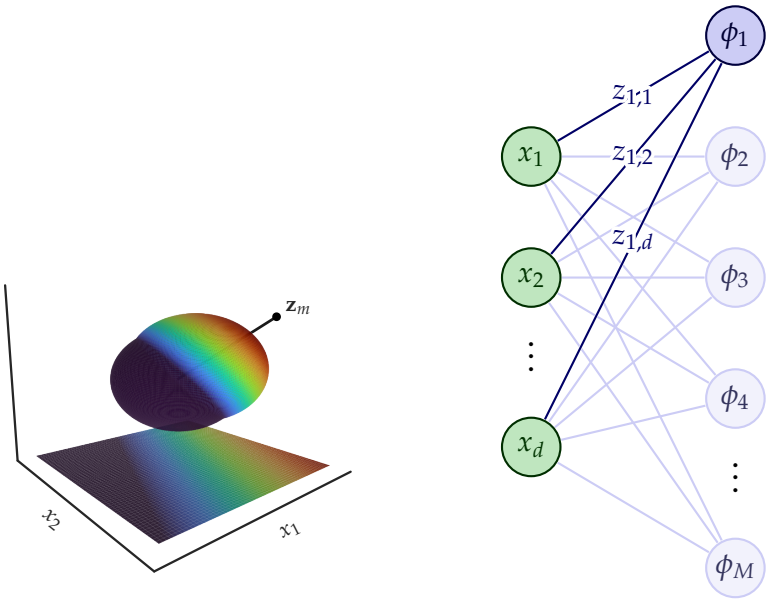
Importantly, Equation (3.4) directly yields a Mercer decomposition for zonal kernels. In particular, let λ_ℓ denote the Fourier coefficients associated with kernel k. This gives rise to the inter-domain features $\phi_m \triangleq Y_{\ell,j}$, where m indexes the pairs (ℓ, j). Crucially, because the spherical harmonics constitute an orthogonal system, this leads to a diagonal covariance

$$[\mathbf{K}_{\mathbf{uu}}]_{mm'} \triangleq \mathrm{Cov}\left(u_m, u_{m'}\right) = \lambda_m^{-1} \delta_{mm'},$$

where $\lambda_m \triangleq \lambda_\ell$ and δ denotes the Kronecker delta.

3.2.2 *Spherical Neural Network Inducing Features*

The recent works of Dutordoir et al. [66] and Sun, Shi, and Grosse [252] aim to construct inter-domain features ϕ_m such that $\mathbf{k}_{\mathbf{u}}(\mathbf{x})$ in Equation (3.2) corresponds to a hidden layer in a feed-forward NN: $\sigma(\boldsymbol{\beta}\mathbf{x})$, for some $\boldsymbol{\beta} \in \mathbb{R}^{M\times d}$ and activation σ such as the SOFTPLUS or the rectified linear unit (RELU) function.



(a) An example RELU-activated hidden unit $H_m : \mathcal{X} \to \mathbb{R}$ visualised on the unit sphere in 3D and projected onto a plane.

(b) Architectural diagram of the basis functions $\mathbf{k}_{\mathbf{u}}(\mathbf{x})$, which corresponds to the hidden layer of a feedforward NN when each $\phi_m \triangleq H_m$ represents a hidden unit.

Figure 3.2: Basis functions $\mathbf{k}_{\mathbf{u}} : \mathcal{X} \to \mathbb{R}^M$ as hidden layers of a feedforward NN.

In particular, let $H_m : \mathcal{X} \to \mathbb{R}$ denote the output of the m-th hidden unit. Additionally, let us project this function onto the unit hypersphere,

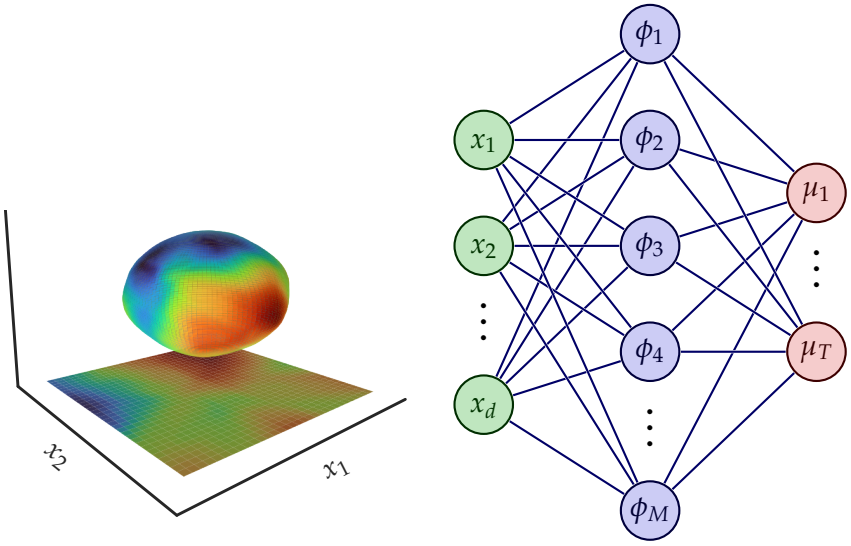
$$H_m(\mathbf{x}) \triangleq \|\mathbf{z}_m\|\|\mathbf{x}\| \cdot \sigma\left(\frac{\mathbf{z}_m^\top \mathbf{x}}{\|\mathbf{z}_m\|\|\mathbf{x}\|}\right). \tag{3.5}$$

See Figure 3.2 for a visualisation of this function. Now, since this function is itself zonal, it can be represented in terms of the spherical harmonics as in Equation (3.4). Let ς_ℓ denote its associated Fourier coefficient. Thus, the inter-domain features can be defined as $\phi_m \triangleq H_m$, which leads to the covariance

$$[\mathbf{K}_{\mathbf{uu}}]_{mm'} = \sum_{\substack{\ell=0:\\ \lambda_\ell \neq 0}}^{\infty} \frac{\varsigma_\ell^2}{\lambda_\ell} \frac{\ell+\alpha}{\alpha} C_\ell^{(\alpha)}\left(\frac{\mathbf{z}_m^\top \mathbf{z}_{m'}}{\|\mathbf{z}_m\|\|\mathbf{z}_{m'}\|}\right), \tag{3.6}$$

where λ_ℓ denotes the Fourier coefficients associated with kernel k.

We refer to this construction as the ACTIVATED SVGP. Notably, the posterior predictive mean of the ACTIVATED SVGP is equivalent to the output of a single-layer feedforward NN, as illustrated in Figure 3.3. Through this perspective, one can reason by analogy that the posterior predictive variance serves as a measure of uncertainty in the predictions of the NN. The ACTIVATED SVGP has been shown to produce competitive results, especially when multiple layers are composed to form a DGP [49]. In this configuration, the propagation of the predictive means closely emulates a forward-pass through a deep NN.



(a) The predictive mean of an ACTIVATED SVGP model with RELU activation features, visualised on the unit sphere in 3D and projected onto a plane.

(b) Architectural diagram of the predictive means of a (multi-output) ACTIVATED SVGP model.

Figure 3.3: The predictive mean of an ACTIVATED SVGP model corresponds to a single-layer feedforward NN.

Despite these favorable properties, ACTIVATED SVGPS have several limitations when it comes to their use with common covariance functions. Before elaborating on them in Section 5.3, we discuss the orthogonally-decoupled GP framework on which our proposed extension relies.

3.3 ORTHOGONALLY DECOUPLED INDUCING POINTS

Recent work has improved the efficiency of sparse GP methods through the structured decoupling of inducing variables [36, 223, 230]. This not only enables the use of more variables at a reduced computational expense but also allows for more flexibility in modelling the predictive mean and covariance independently. We focus on the general framework of Shi, Titsias, and Mnih [230] under which its predecessors can be subsumed.

In particular, let the random function $f(\mathbf{x})$ from Equation (2.14) be decomposed into the sum of two independent GPs,

$$f(\mathbf{x}) = g(\mathbf{x}) + h(\mathbf{x}),$$

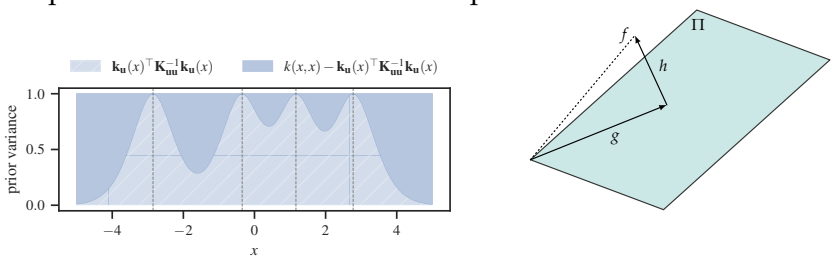
where

$$g(\mathbf{x}) \sim \mathcal{GP}(0, \mathbf{k}_\mathbf{u}^\top(\mathbf{x})\mathbf{K}_\mathbf{uu}^{-1}\mathbf{k}_\mathbf{u}(\mathbf{x}')), \quad h(\mathbf{x}) \sim \mathcal{GP}(0, s(\mathbf{x}, \mathbf{x}'))$$

and let the covariance function $s(\mathbf{x}, \mathbf{x}')$ be defined according to the Schur complement of $\mathbf{K}_\mathbf{uu}$,

$$s(\mathbf{x}, \mathbf{x}') \triangleq k(\mathbf{x}, \mathbf{x}') - \mathbf{k}_\mathbf{u}^\top(\mathbf{x})\mathbf{K}_\mathbf{uu}^{-1}\mathbf{k}_\mathbf{u}(\mathbf{x}'),$$

where $\mathbf{k}_\mathbf{u}$ is defined in Equation (3.2). Intuitively, one can view g as the projection of f onto $\mathbf{u}$, and $h \perp g$, i.e. h is *orthogonal* to g [97] in the statistical sense of linear independence [214]. See Figure 3.4 for an illustration of the priors of $g(\mathbf{x})$ and $h(\mathbf{x})$ and a geometric interpretation in terms of vector subspaces.



(a) Prior variance decomposed. The prior variance of $f(\mathbf{x})$ is $k(\mathbf{x}, \mathbf{x}) = \alpha$ for kernel amplitude $\alpha = 1$, which can be decomposed as the sum of the prior variances of $g(\mathbf{x})$ and $h(\mathbf{x})$. Vertical dashed lines indicate the location of inducing inputs $\mathbf{z}_m$ for $m = 1, \ldots, 4$. At these locations, the variance of $g(\mathbf{x})$ is one while that of $h(\mathbf{x})$ is zero.

(b) Orthogonal decomposition of function f wrt the hyperplane $\Pi \triangleq \{\boldsymbol{\alpha}^\top\mathbf{k}_\mathbf{u}(\cdot); \boldsymbol{\alpha} \in \mathbb{R}^M\}$; function g is the orthogonal projection of f onto Π and function h is the residual component perpendicular to Π.

Figure 3.4: Function $f(\mathbf{x})$ decomposed as the sum of two independent GPs.

Let $\mathbf{h}$ be the values of h at observed inputs $\mathbf{X}$, i.e. $\mathbf{h} \triangleq h(\mathbf{X})$. Then we have

$$p(\mathbf{h}) = \mathcal{N}(\mathbf{h} \,|\, \mathbf{0}, \mathbf{S}_\mathbf{ff}),$$

where $\mathbf{S}_\mathbf{ff} \triangleq \mathbf{K}_\mathbf{ff} - \mathbf{Q}_\mathbf{ff}$. This allows one to reparameterise $\mathbf{f} \sim p(\mathbf{f} \,|\, \mathbf{u})$ from Equation (2.22), for a given $\mathbf{u}$, as

$$\mathbf{f} = \mathbf{Q}_\mathbf{fu}\mathbf{u} + \mathbf{h}, \qquad \mathbf{h} \sim p(\mathbf{h}). \tag{3.7}$$

The model's joint distribution can now be written as

$$p(\mathbf{y}, \mathbf{h}, \mathbf{u}) = p(\mathbf{y} \,|\, \mathbf{h}, \mathbf{u})p(\mathbf{h})p(\mathbf{u}),$$

where the likelihood is now

$$p(\mathbf{y} \,|\, \mathbf{h}, \mathbf{u}) = \mathcal{N}(\mathbf{y} \,|\, \mathbf{Q}_\mathbf{fu}\mathbf{u} + \mathbf{h}, \beta^{-1}\mathbf{I}).$$

	W	X	Z
$f(\cdot)$	$\mathbf{v}$	$\mathbf{f}$	$\mathbf{u}$
$h(\cdot)$	$\mathbf{v}'$	$\mathbf{h}$	-

Table 3.1: Summary of notation: relationships between input locations and output variables.

Next, *orthogonal* inducing variables $\mathbf{v}$, which represent the values of f at a collection of K orthogonal inducing locations $\mathbf{W} \triangleq [\mathbf{w}_1 \cdots \mathbf{w}_K]^\top$, are introduced. Similarly, inducing variables $\mathbf{v}'$ represent the values of h at $\mathbf{W}$. The reader may find it helpful to refer to Table 3.1 for a summary of the relationships between the input locations and the output variables defined thus far.

Now, by definition, $\mathbf{v}$ is linearly dependent on $\mathbf{v}'$

$$\mathbf{v} = \mathbf{Q}_{\mathbf{vu}}\mathbf{u} + \mathbf{v}', \tag{3.8}$$

where $\mathbf{Q}_{\mathbf{vu}} \triangleq \mathbf{K}_{\mathbf{vu}}\mathbf{K}_{\mathbf{uu}}^{-1}$, which is analogous to the relationship between $\mathbf{f}$ and $\mathbf{h}$ in Equation (3.7). Therefore, one need only be concerned with the treatment of $\mathbf{v}'$. The joint distribution of the model augmented by the variables $\mathbf{v}'$ now becomes

$$p(\mathbf{y}, \mathbf{h}, \mathbf{u}, \mathbf{v}') = p(\mathbf{y} \,|\, \mathbf{h}, \mathbf{u})p(\mathbf{u})p(\mathbf{h}, \mathbf{v}'),$$

where $p(\mathbf{h}, \mathbf{v}') = p(\mathbf{h} \,|\, \mathbf{v}')p(\mathbf{v}')$ for $p(\mathbf{v}') = \mathcal{N}(\mathbf{0}, \mathbf{S}_{\mathbf{vv}})$ and $p(\mathbf{h} \,|\, \mathbf{v}') = \mathcal{N}(\mathbf{h} \,|\, \mathbf{S}_{\mathbf{fv}}\mathbf{S}_{\mathbf{vv}}^{-1}\mathbf{v}', \mathbf{S}_{\mathbf{ff}} - \mathbf{S}_{\mathbf{fv}}\mathbf{S}_{\mathbf{vv}}^{-1}\mathbf{S}_{\mathbf{vf}})$, with

$$\mathbf{S}_{\mathbf{vf}} \triangleq \mathbf{K}_{\mathbf{vf}} - \mathbf{Q}_{\mathbf{vf}}, \quad \mathbf{Q}_{\mathbf{vf}} \triangleq \mathbf{Q}_{\mathbf{vu}}\mathbf{K}_{\mathbf{uu}}\mathbf{Q}_{\mathbf{uf}}, \tag{3.9}$$

$$\mathbf{S}_{\mathbf{vv}} \triangleq \mathbf{K}_{\mathbf{vv}} - \mathbf{Q}_{\mathbf{vv}}, \quad \mathbf{Q}_{\mathbf{vv}} \triangleq \mathbf{Q}_{\mathbf{vu}}\mathbf{K}_{\mathbf{uu}}\mathbf{Q}_{\mathbf{uv}}. \tag{3.10}$$

Let the variational distribution now be $q(\mathbf{h}, \mathbf{u}, \mathbf{v}') = p(\mathbf{h} \,|\, \mathbf{v}')q(\mathbf{u}, \mathbf{v}')$, where $q(\mathbf{u}, \mathbf{v}') \triangleq q(\mathbf{u})q(\mathbf{v}')$ and $q(\mathbf{v}') \triangleq \mathcal{N}(\mathbf{m}_{\mathbf{v}}, \mathbf{C}_{\mathbf{v}})$ for variational parameters $\mathbf{m}_{\mathbf{v}} \in \mathbb{R}^K$ and $\mathbf{C}_{\mathbf{v}} \in \mathbb{R}^{K \times K}$ s. t. $\mathbf{C}_{\mathbf{v}} \succeq 0$. This gives the test predictive density $q(\mathbf{f}_*) = \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_{**})$, where

$$\boldsymbol{\mu}_* \triangleq \mathbf{Q}_{*\mathbf{u}}\mathbf{m}_{\mathbf{u}} + \mathbf{S}_{*\mathbf{v}}\mathbf{S}_{\mathbf{vv}}^{-1}\mathbf{m}_{\mathbf{v}}, \tag{3.11}$$

$$\begin{aligned}\boldsymbol{\Sigma}_{**} \triangleq \mathbf{K}_{**} &+ \mathbf{Q}_{*\mathbf{u}}(\mathbf{C}_{\mathbf{u}} - \mathbf{K}_{\mathbf{uu}})\mathbf{Q}_{\mathbf{u}*} \\ &+ \mathbf{S}_{*\mathbf{v}}\mathbf{S}_{\mathbf{vv}}^{-1}(\mathbf{C}_{\mathbf{v}} - \mathbf{S}_{\mathbf{vv}})\mathbf{S}_{\mathbf{vv}}^{-1}\mathbf{S}_{\mathbf{v}*}.\end{aligned} \tag{3.12}$$

Thus seen, prediction incurs a cost of $\mathcal{O}(M^3 + K^3)$ in this framework.

Like the so-called ODVGP framework of Salimbeni et al. [223], when seen from the dual RKHS perspective, the predictive mean can be decomposed into a component that shares the same standard basis as the covariance, in addition to another component that is *orthogonal* to the standard basis. However, this framework extends ODVGP further by also decomposing the predictive covariance into parts corresponding to the standard and orthogonal bases. Accordingly, setting $\mathbf{C}_{\mathbf{v}} = \mathbf{S}_{\mathbf{vv}}$ recovers the ODVGP framework, and further setting $\mathbf{m}_{\mathbf{v}} = \mathbf{0}$ recovers the standard SVGP framework.

COVARIANCE STRUCTURE. Now, unlike $q(\mathbf{u}, \mathbf{v}')$, which factorizes according to the mean-field assumption, $q(\mathbf{u}, \mathbf{v})$ has a full covariance structure by virtue of the relationship described in Equation (3.8). Specifically, we have $q(\mathbf{u}, \mathbf{v}) = \mathcal{N}(\mathbf{m}_{\mathbf{uv}}, \mathbf{C}_{\mathbf{uv}})$, where

$$\mathbf{m}_{\mathbf{uv}} \triangleq \begin{bmatrix} \mathbf{m}_{\mathbf{u}} \\ \mathbf{Q}_{\mathbf{vu}}\mathbf{m}_{\mathbf{u}} + \mathbf{m}_{\mathbf{v}} \end{bmatrix},$$

and

$$\mathbf{C}_{\mathbf{uv}} \triangleq \begin{bmatrix} \mathbf{C}_{\mathbf{u}} & \mathbf{C}_{\mathbf{u}}\mathbf{Q}_{\mathbf{uv}} \\ \mathbf{Q}_{\mathbf{vu}}\mathbf{C}_{\mathbf{u}} & \mathbf{C}_{\mathbf{v}} + \mathbf{Q}_{\mathbf{vu}}\mathbf{C}_{\mathbf{u}}\mathbf{Q}_{\mathbf{uv}} \end{bmatrix}.$$

3.4 METHODOLOGY

We begin this section by outlining some of the limitations of ACTIVATED SVGPS that preclude the use of numerous kernels and inducing features, not the least of which being popular choices of kernels such as the SE kernel and the Matérn family of kernels, combined with NN inducing features with RELU activations.

The root cause of these issues can be seen in Figure 3.5, where the Fourier coefficients of various combinations of kernels and activation features are visualized. Specifically, for each combination, we compare the (root of the) kernel coefficients $\sqrt{\lambda_\ell}$ against the feature coefficients ς_ℓ at increasing levels $\ell = 1, \ldots, 35$. The posterior predictives that result from fitting ACTIVATED SVGP models with these combinations are shown in Figure 3.6. We consider the Matérn-$5/2$ kernel as our running example, but the analysis extends to all stationary kernels.

SPECTRA MISMATCH. For the Matérn kernel (left column of panes in Figures 3.5 and 3.6), we see that there are multiple levels ℓ at which the feature coefficients are zero while the corresponding kernel coefficients are nonzero. Such discrepancies in the spectra yields a poor Nyström approximation $\mathbf{Q}_{\mathbf{ff}}$ that fails to fully capture the prior covariance $\mathbf{K}_{\mathbf{ff}}$ induced by the kernel, which subsequently leads to the overestimation of the predictive variance and therefore a suboptimal ELBO. In contrast, the Arccos kernel does not suffer from this pathology.

RKHS INNER PRODUCT. The RKHS inner product associated with zonal kernels in general is a series consisting of ratios of Fourier coefficients. Since the RELU feature coefficients (top row of panes in Figures 3.5 and 3.6) decay at the same rate as the square root of the kernel coefficients, this results in a divergent series which in turn renders the RKHS inner product indeterminate. In contrast, the feature coefficients of the comparatively smoother SOFTPLUS activation (bottom row of panes in Figures 3.5 and 3.6) decay at a much faster rate, and thus yields a well-defined RKHS inner product. For the reasons outlined

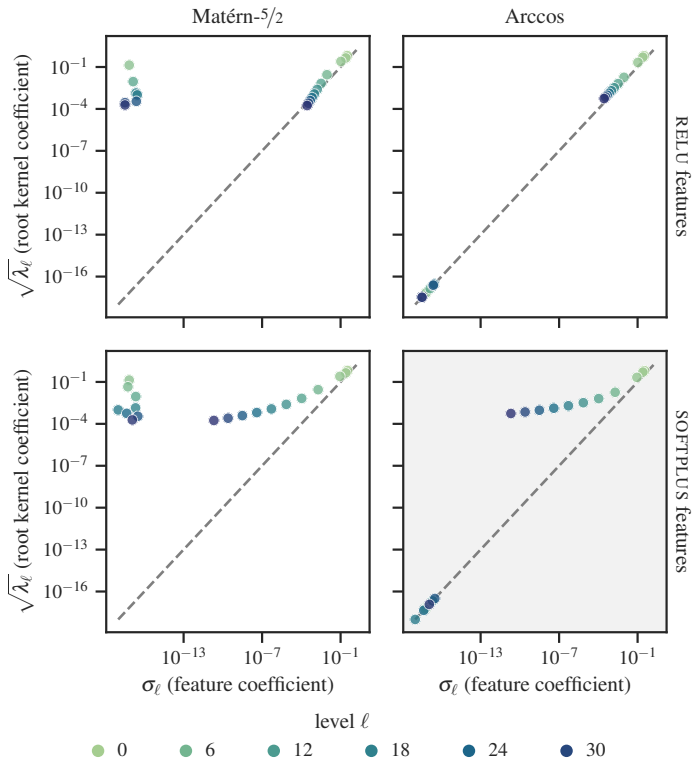


Figure 3.5: Comparison of the Fourier coefficients of various kernels and activation features for increasing levels $\ell = 1, \ldots, 35$.

above, the work of Dutordoir et al. [66] restricted its scope to the use of the Arccos kernel in conjunction with the SOFTPLUS activation (pane highlighted in gray in Figure 3.5).

TRUNCATION ERROR. Lastly, as expected, the truncation of the series in Equation (3.6) at some finite number L of spherical harmonic levels often leads to overly smooth predictive response surfaces and overestimation of the variance.

Spherical Features for Orthogonally-Decoupled GPs

We propose extending the orthogonally-decoupled GP framework (Section 3.3) to use inter-domain inducing features. Accordingly, let $u_m \triangleq \langle f, \phi_m \rangle_{\mathcal{H}}$ and $v_k \triangleq \langle f, \psi_k \rangle_{\mathcal{H}}$ for some arbitrary choices of $\phi_m, \psi_k \in \mathcal{H}$. This generalizes the framework of Shi, Titsias, and Mnih [230] since, by the reproducing property, setting $\phi_m : \mathbf{x} \mapsto k(\mathbf{z}_m, \mathbf{x})$ and $\psi_k : \mathbf{x} \mapsto k(\mathbf{w}_k, \mathbf{x})$ leads to standard inducing points, $u_m = f(\mathbf{z}_m), v_k = f(\mathbf{w}_k)$. In particular, we define $\phi_m \triangleq H_m$, the m-th unit of the spherical activation layer (Equation (3.5)) described in Section 3.2.2, and $\psi_k(\mathbf{x}) \triangleq k(\mathbf{w}_k, \mathbf{x})$. The posterior predictive of the model described in Section 3.3, summarized by Equations (3.11) and (3.12), is fully determined by

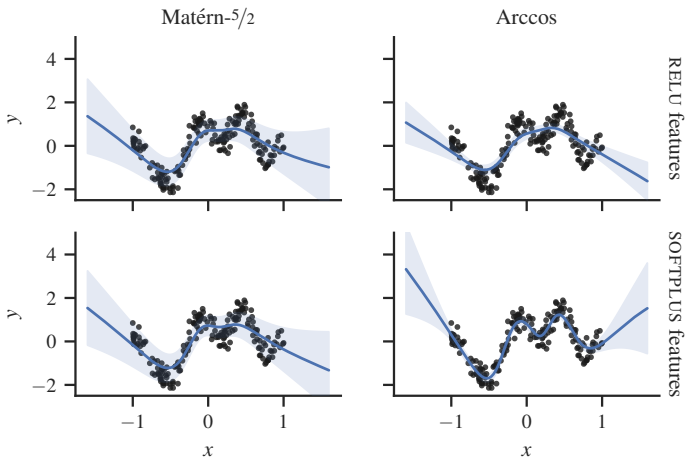


Figure 3.6: Posterior predictives of ACTIVATED SVGP models various kernels and activation features and $L = 16$ levels.

the covariances $\mathbf{K_{ff}}, \mathbf{K_{uf}}, \mathbf{K_{vf}}, \mathbf{K_{uu}}, \mathbf{K_{vu}}$ and $\mathbf{K_{vv}}$. Recall that $[\mathbf{K_{uf}}]_{mn} = [\mathbf{k_u}(\mathbf{x}_n)]_m$ and $\mathbf{K_{uu}}$ is precisely as expressed in Equation (3.6). We have

$$
\begin{aligned}
[\mathbf{K_{vf}}]_{kn} &\triangleq \mathrm{Cov}\left(v_k, f(\mathbf{x}_n)\right) = k(\mathbf{w}_k, \mathbf{x}_n), \\
[\mathbf{K_{vu}}]_{km} &\triangleq \mathrm{Cov}\left(v_k, u_m\right) = \phi_m(\mathbf{w}_k), \\
[\mathbf{K_{vv}}]_{kk'} &\triangleq \mathrm{Cov}\left(v_k, v_{k'}\right) = k(\mathbf{w}_k, \mathbf{w}_{k'}).
\end{aligned}
$$

Note that the cross-covariance $\mathbf{K_{vu}}$ between $\mathbf{u}$ and $\mathbf{v}$ can be interpreted as the forward-pass of the orthogonal pseudo-input $\mathbf{w}_k$ through the NN activation H_m. Crucially, these terms constitute the orthogonal basis and provide additional degrees of flexibility, through free parameters $\mathbf{W}$, that can compensate for errors remaining from the original basis – in both the predictive mean and variance. Suffice it to say, this is not the only possible choice but is one that possesses a number of appealing properties.

As discussed in Section 3.3, the addition of K inducing variables incurs a cost of $\mathcal{O}(M^3 + K^3)$. More precisely: suppose the exact cost is $C \cdot (M^3 + K^3)$ operations for some constant C wrt M, K. Further, suppose $K = B \cdot M$ for some $B > 0$. Then there are a total of $(B+1) \cdot M$ inducing variables (orthogonal or otherwise) and the cost becomes $(B^3 + 1)C \cdot M^3$. By comparison, incorporating the same number of inducing variables in SVGP costs $(B+1)^3 C \cdot M^3$. That is, this approach leads to a $(B^3 + 1)$-fold increase in the constant rather than a $(B+1)^3$-fold increase. Concretely, this means that doubling the number of inducing variables doubles the constant in this approach, but leads to an *eight-fold* increase in SVGP. While such a difference vanishes asymptotically for large M and K, it still has a considerable impact for modest sizes ($M, K < 1,000$) that are feasible in practice. Thus seen, incorporating an orthogonal basis spanned by K inducing variables

is a more cost-effective strategy for improving ACTIVATED SVGP than increasing M or the truncation level L.

3.5 RELATED WORK

Efforts to establish the connection between GPs and NNs have been ongoing for decades. Notably, as first identified by Neal [185], the behavior of a single-layer NN converges to that of a GP as its width grows to infinity. This phenomenon extends to various activations [156, 284], not least the RELU activation [38], which pervades modern deep learning. It also applies in the reverse direction, i. e., one can derive new NN activations from a given GP prior [171]. More recently, several works have identified similar parallels between GPs and DNNs [143, 166] and broadly, networks of various architectures [293]. Despite their close relationship, finite NNs consistently surpass their GP counterparts in practice [78, 190]. Although GPs have stood out for offering precise uncertainty calibration and being amenable to Bayesian inference without the need to resort to approximations, they are less scalable by nature and have limited representational capacity. This limitation stems from the fixed nature of their effective basis functions, as we alluded to earlier in this chapter. In short, taken to the infinite-width limit, the basis functions become unable to flexibly adapt to the inputs [156]. Bayesian neural networks (BNNS) present a compelling middle ground, combining uncertainty estimation with the representation learning of *finite* deep neural networks through *approximate* Bayesian inference [18, 74, 154, 185]. Deep GPs (DGPS) [49] serve as a complementary approach to DNNs. By utilising GPs layers in place of weighted affine layers, DGPS achieve superior uncertainty estimation by virtue of their infinite width. Nevertheless, in practice, DGPS can be cumbersome to optimise and scale.

Inter-domain inducing features [140] provide a different approach to defining inducing variables. Unlike traditional inducing points, inducing features employ a linear transformation of the latent function, resulting in basis functions that are not purely predetermined by the kernel but have the flexibility to adapt according to the inputs. Notably, this approach can provide substantial computational gains and the ability to specify more expressive GP approximations. For instance, the variational Fourier features [97] effectively lead to basis functions consisting of the Fourier basis. This design leads to a block-diagonal structure in the covariance matrix, substantially improving computational efficiency. However, this approach fails to scale gracefully beyond a few dimensions. The spherical harmonic features [65] can be seen as an extension of this approach. They not only facilitates inference in higher dimensions but also results in a diagonal covariance, slashing inference costs to a linear relationship with the number of inducing variables. Further developments build on inducing features

in the spherical domain to represent NNs on the sphere in terms of its spherical harmonics expansion [66, 252]. Rather than computational gains, it provides yet another compelling middle ground between GPs and NNs by tapping the benefits of both. In particular, under this framework, the predictive mean emulates the forward pass of a single-layer feedforward NN, enabling superior representation learning through adaptive basis functions. Simultaneously, the posterior variance directly provides uncertainty estimation without the need to marginalise layer weights. Nonetheless, as discussed in Section 3.4, this framework has a tendency to underestimate the predictive uncertainty.

Concurrent with these developments, recent efforts have focused on improving the flexibility and efficiency of sparse GP methods through the structured decoupling of inducing variables. This not only allows the use of a greater number of inducing variables with less computational demand, but also offers greater independence and flexibility in representing the predictive means and covariances. More specifically, when viewed from the dual RKHS perspective, the predictive mean and variance in SVGP [262] share a common set of basis functions. Cheng and Boots [36] propose a novel parameterisation that allows the predictive mean to employ its own distinct set of basis functions. However, without appropriate constraints on these additional basis functions, this model leads to a poorly-conditioned, nonconvex optimization problem, making it unwieldy to train [94]. Building on this decoupled parameterisation, Salimbeni et al. [223] propose the ODVGP framework, in which the predictive mean similarly employs a distinct set of basis functions. Unlike the approach of Cheng and Boots, the predictive mean maintains a dependency on the principal basis, which is shared with the predictive covariance. Furthermore, any function that can be represented by the additional basis in ODVGP is, by design, *orthogonal* to the span of the principal basis. This results in a better-conditioned optimisation problem that is suitable for natural gradient methods, and substantially enhances the model's flexibility by virtue of its ability to capture remaining variations that the principal basis fails to account for. More recently, Shi, Titsias, and Mnih [230] derive a more general framework from a probabilistic modelling perspective by augmenting the SVGP model with additional inducing variables. This leads to a predictive density that not only encompasses that of the ODVGP as a special case, but also allows for a more flexible calibration of the predictive uncertainty. It achieves this by allowing the predictive *covariance* to be decoupled into the principal and orthogonal bases, enabling more precise tuning of uncertainty estimates. Overall, this not only makes more efficient use of inducing variables but also enhances predictive accuracy and uncertainty estimation.

3.6 EXPERIMENTS

We describe the experiments conducted to empirically validate our approach. The open-source implementation of our method can be found on GitHub at: ltiao/spherical-orthogonal-gaussian-processes. Further information concerning the experimental set-up and various implementation details can be found in Section 3.A.

3.6.1 *Synthetic 1D Dataset*

We highlight some notable properties of our method on the one-dimensional dataset of Snelson and Ghahramani [235].

First we fit ACTIVATED SVGP models with different combinations of kernels and activation features using $L = 8$ truncation levels. The resulting posterior predictives are shown in Figure 3.7. More specifically, in Figure 3.7a, we see that none of the model fits are particularly tight due in part to truncation errors, since we are using relatively few levels. This is especially true of the Matérn kernel (left column of panes), which results in a posterior that is not only too smooth but also clearly suffering from an overestimation of the variance. A conceptually straightforward way to improve performance is to increase the truncation level. Accordingly, Figure 3.6 (introduced earlier in Section 5.3) showed results from effectively the exact same set-up, but with twice the number of levels ($L = 16$). With this increase, we see a clear improvement in the Arccos-SOFTPLUS case, but no discernible difference in the other combinations. Notably, the overestimation of the variances in the Matérn kernel persists. By comparison, Figure 3.7b shows results from using $L = 8$ truncation levels, but with the addition of $K = 8$ orthogonal inducing variables. Remarkably, incorporating just a handful of these variables produces substantial improvements, not least for the Matérn kernel.

Figure 3.8 offers a deeper insight into the underlying mechanisms that contribute to these improvements. Here we plot the predictive variance (Equation (3.12)) in terms of its constituent parts. In Figure 3.8a, we see that the variance estimate with Matérn kernels is heavily distorted by large spurious contributions in the $\mathbf{K_{ff}} - \mathbf{Q_{ff}}$ term (*dark blue solid line*), which is caused by the pathology described in Section 5.3. On the other hand, in Figure 3.8b, such spurious contributions also appear, but are offset by the subtractive term $\mathbf{S_{fv}}\mathbf{S_{vv}}^{-1}\mathbf{S_{vf}}$ (*dark orange dashed line*). This term constitutes the orthogonal basis, and provides added flexibility that is effective at nullifying errors introduced by the original basis.

Each of the three variations discussed above are repeated 5 times, and some quantitative results are summarized in Figure 3.9. Specifically, we report the ELBO and the *throughput*, i.e. the average number of optimisation iterations completed per second. The ACTIVATED SVGP

with $L = 8$ truncation levels, as seen in Figures 3.7a and 3.8a, is represented by the *blue circular markers*. The model resulting from doubling the number of levels $L = 16$, as seen in Figure 3.6, is represented by the *orange circular markers*. As discussed, this leads to an improvement in the Arccos-SOFTPLUS case, but to modest or no improvements otherwise. However, we can now see that this has come at a significant computational expense, as the throughput has reduced by roughly half. On the other hand, the model resulting from retaining the same truncation level but incorporate an orthogonal basis consisting of $K = 8$ variables, as seen in Figures 3.7b and 3.8b, is represnted by the *blue cross markers*. This can be seen to have roughly the same footprint as doubling the truncation level, but leads to a considerably improved model fit, especially in cases involving the Matérn kernel (the only exception is in the Arccos-SOFTPLUS case, where doubling the truncation level retains a slight advantage). All told, incorporating an orthogonal basis has roughly the same cost as doubling the truncation level but leads to significantly better performance improvements.

3.6.2 *Regression on UCI Repository Datasets*

We evaluate our method on a number of well-studied regression problems from the UCI repository of datasets [61]. In particular, we consider the YACHT, CONCRETE, ENERGY, KIN8NM and POWER datasets. Additional results on the larger datasets from this collection can be found in Section 3.B.2.

We fit variations of SVGP with the Arccos, Matérn, and SE kernels, and (a) standard inducing points, and inter-domain inducing features based on (b) RELU- and (c) SOFTPLUS-activated inducing features. For each of these variants, we consider three combinations of base and orthogonal inducing variables: (i-ii) 128 and 256 base inducing variables (and no orthogonal inducing variables), and (iii) 128 base inducing variables with 128 orthogonal inducing variables. The activation features are truncated at $L = 6$ levels. Our proposed method is represented by the combinations consisting of RELU- and SOFTPLUS-activated features with orthogonal inducing variables (b-c,iii). The remaining combinations, against which we benchmark, correspond to the original SVGP (a,i-ii) [262], SOLVEGP (a,iii) [230], and ACTIVATED SVGP (b-c,i-ii) [66].

To quantitatively assess performance, we report the test root-mean-square error (RMSE) and negative log predictive density (NLPD), shown in Figures 3.10 and 3.11, respectively. Unless otherwise stated, for each method and problem, we perform random sub-sampling validation by aggregating results from 5 repetitions across 10% held-out test sets. Within the training set, the inputs and outputs are standardized, i. e. scaled to have zero mean and unit variance and subsequently restored to the original scale at test time.

We observe that, irrespective of the choice of kernel, when using activation features, whether RELU- or SOFTPLUS-activated, augmenting the model with orthogonal bases significantly improves performance, notably even more so than doubling the number of base inducing variables. This can readily be seen across all datasets on both the NLPD and RMSE metrics. Further, with the Arccos kernel, it outperforms its counterparts based on standard inducing points across most datasets (the exception being the POWER dataset). With the Matérn and SE kernels, it achieves results comparable to its standard inducing points counterparts in most datasets.

3.6.3 *Large-scale Regression on Airline Delays Dataset*

Finally, we consider a large-scale regression dataset concerning U.S. commercial airline delays in 2008. The task is to forecast the duration of delays in reaching the destination of a given flight, utilising information such as the route distance, airtime, scheduled month, day of the week, and other relevant factors, as well as characteristics of the aircraft such as its age (number of years since deployment). The complete dataset encompasses 5,929,413 flights, of which we randomly select 1M observations without replacement to form a subset that is more manageable but still considerable in scale. Results on a reduced 100K subset can be found in Section 3.B.1.

To quantitatively assess performance, we report the test RMSE and NLPD evaluated on a 1/3 held-out test set. The results are shown in the top and bottom rows of Figure 3.12, respectively. Within the training set, the inputs and outputs are standardized, i. e. scaled to have zero mean and unit variance and subsequently restored to the original scale at test time.

Given the immense volume of data at hand, we are compelled to utilise mini-batch training for stochastic optimisation [98]. To this end, we use the Adam optimizer [126] with its typical default settings (learning rate 1×10^{-3}, $\beta_1 = 0.9, \beta_2 = 0.999$). Our batch size is set to 5,000, and we train the models for a total of 1,200 epochs.

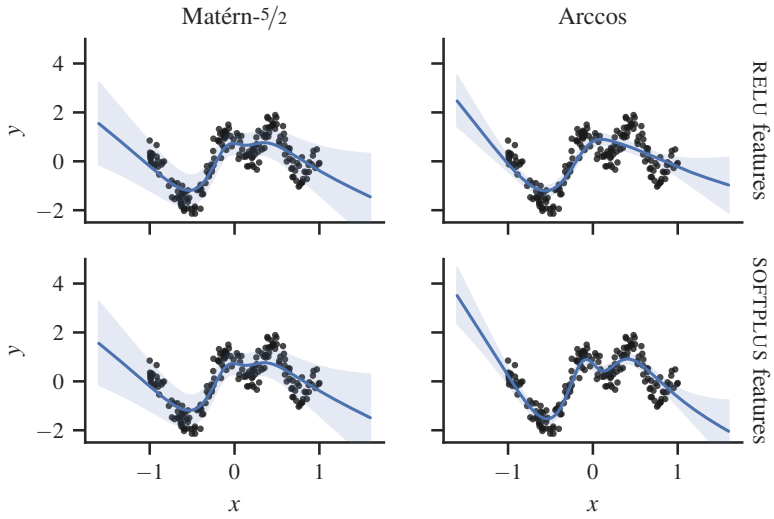
We fit variations of SVGP with the Arccos kernel and (a) standard inducing points and (b) inter-domain inducing features based on SOFTPLUS-activated inducing features. For each of these variants, we consider three combinations of base and orthogonal inducing variables: (i-ii) 500 and 1,000 base inducing variables (and no orthogonal inducing variables), and (iii) 500 base inducing variables with 500 orthogonal inducing variables. The activation features are truncated at $L = 6$ levels. Our proposed method is represented by the combination consisting of SOFTPLUS-activated features with orthogonal inducing variables (b,iii). The remaining combinations, against which we benchmark, correspond to the mini-batch SVGP (a,i-ii) [98], SOLVEGP (a,iii) [230], and ACTIVATED SVGP (b,i-ii) [66].

The outcomes are as expected when employing standard inducing points (left). In particular, doubling the number of base inducing points from 500 to 1,000 demonstrates significant improvements. Furthermore, by using 500 base inducing points alongside 500 orthogonal inducing points, we achieve comparable performance to having 1,000 base inducing points, while enjoying improved computationally efficiency. In contrast, when examinining the ACTIVATED SVGP model with SOFTPLUS features (right), it's apparent that it underperforms compared to the original SVGP counterparts. Furthermore, doubling the number of inducing features from 500 to 1,000 has virtually no effect. However, by incorporating orthogonal bases into the ACTIVATED SVGP model with 500 features following our proposed approach, we witness substantial improvements and achieve comparable performance to its standard inducing points counterparts.

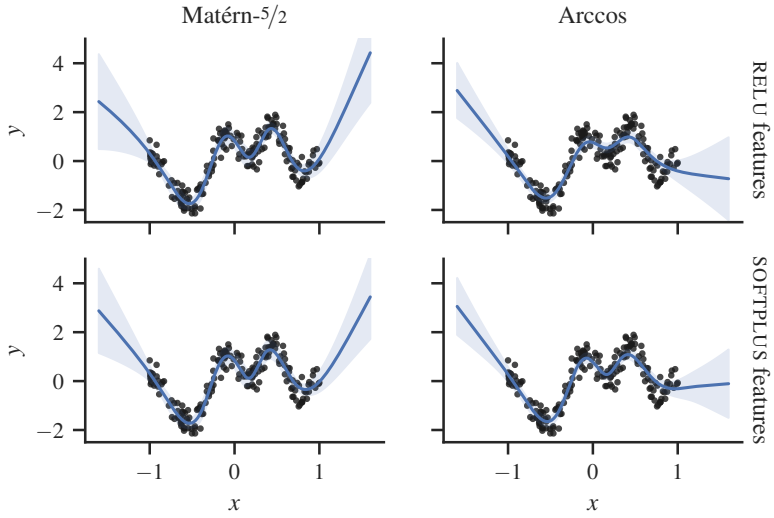
3.7 SUMMARY

We considered the use of inter-domain inducing features in the orthogonally-decoupled SVGP framework, specifically, the spherical activation features, and showed that this alleviates some of the practical issues and shortcomings associated with the ACTIVATED SVGP model. We demonstrated the effectiveness of this approach by conducting empirical evaluations on several problems, and showed that this leads to enhanced predictive performance over more computationally demanding alternatives such as increasing the truncation levels or the number of inducing variables.

Future work will explore alternative designs of inter-domain inducing features to construct new standard and orthogonal bases that provide additional complementary benefits.

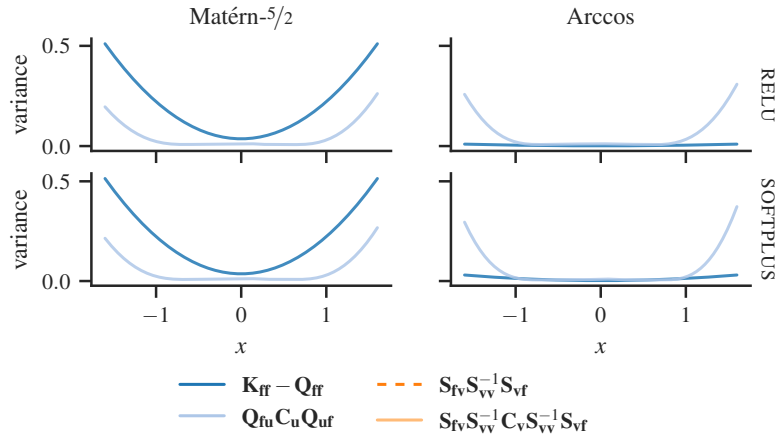


(a) Inducing activation features with $L = 8$ levels.

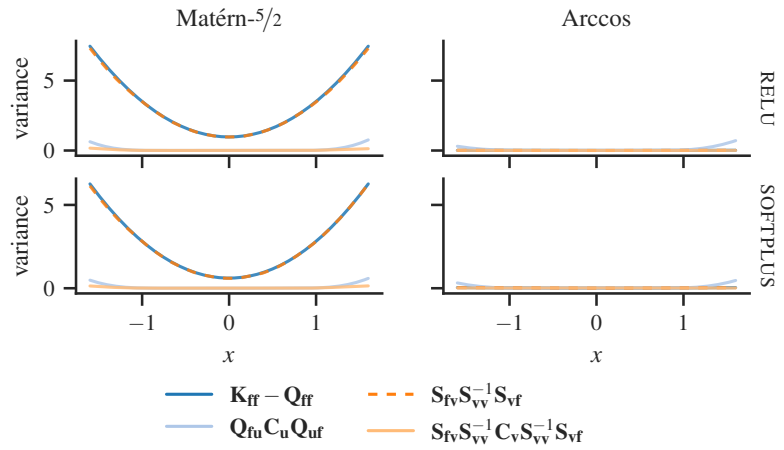


(b) Inducing activation features with $L = 8$ levels and $K = 8$ orthogonal bases (*our method*).

Figure 3.7: Posterior predictives of ACTIVATED SVGP with various kernels and activation features on the 1D Snelson dataset; *black circular markers* represent the observations; *blue solid lines* and *shaded regions* denote the mean and the ± 2 standard deviations, resp.



(a) Inducing activation features with $L = 8$ levels.



(b) Inducing activation features with $L = 8$ levels and $K = 8$ orthogonal bases (*our method*).

Figure 3.8: Decomposition of the posterior predictive variances of SVGP with various kernels and activation features on the 1D SNELSON dataset (see Figure 3.7) into its constituent terms; the additive terms that constitute the predictive variance are indicated by *solid* lines, while the subtractive terms are indicated by *dashed* lines; terms that constitute the predictive variance of the original SVGP model [262] have a *blue* hue, while additional terms introduced by the orthogonally-decoupled model [230] have an *orange* hue.

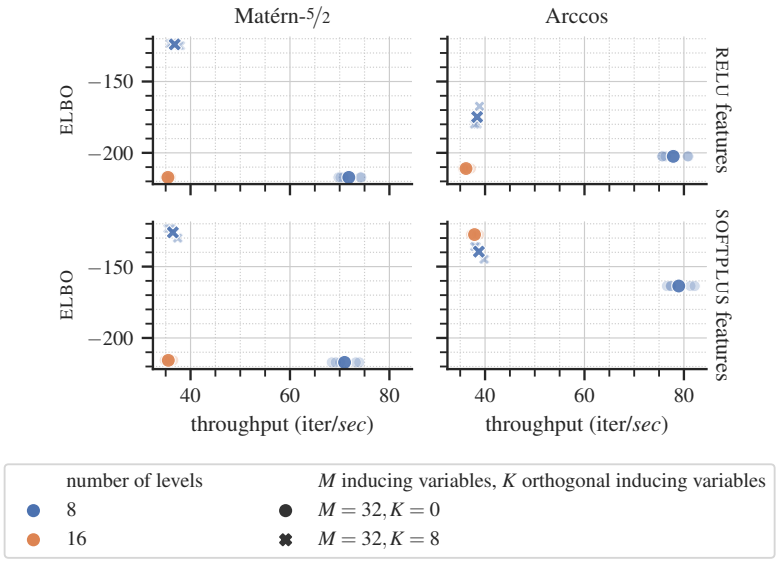


Figure 3.9: The ELBO and throughput (of model fitting) for various kernels and activation features and the configurations visualized in Figure 3.6, Figures 3.7a and 3.8a, and Figures 3.7b and 3.8b; markers with low opacity represent the individual runs, while markers with high opacity represent the mean of each group.

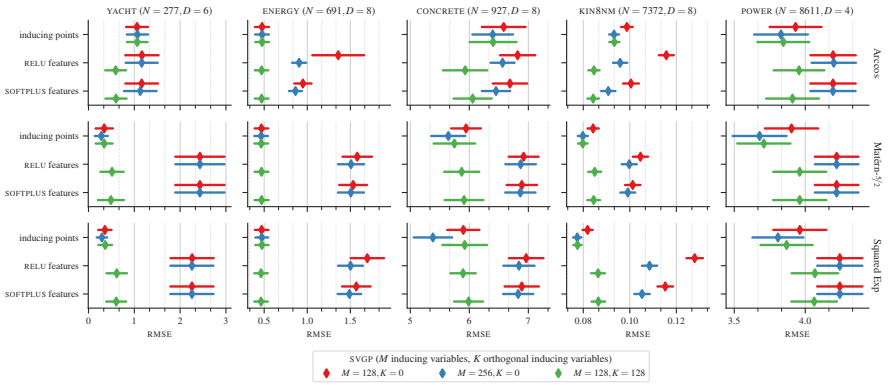


Figure 3.10: Test RMSE on regression problems from the UCI repository of datasets for various kernels and activation features. Along the rows labeled *"inducing points"*, the red and blue markers (♦, ♦) represent the original SVGP model [262], while the green markers (♦) represent SOLVEGP [230]. Along the remaining rows, the red and blue markers (♦, ♦) represent the ACTIVATED SVGP [66], while the green markers (♦) represent our proposed approach.

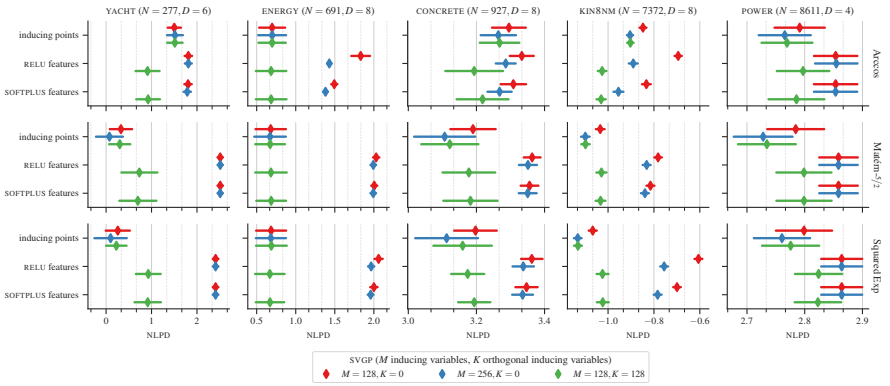


Figure 3.11: Test NLPD on regression problems from the UCI repository of datasets for various kernels and activation features. Along the rows labeled *"inducing points"*, the red and blue markers (♦, ♦) represent the original SVGP model [262], while the green markers (♦) represent SOLVEGP [230]. Along the remaining rows, the red and blue markers (♦, ♦) represent the ACTIVATED SVGP [66], while the green markers (♦) represent our proposed approach.

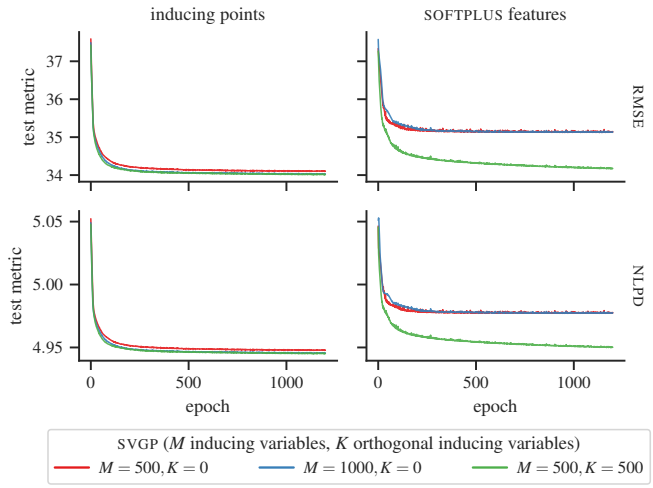


Figure 3.12: Test metrics, RMSE and NLPD, on the large-scale 2008 U.S. airline delays dataset using the *Arccos* kernel with standard inducing points and SOFTPLUS-activated features. Along the column labeled "inducing points", the red and blue lines (— and —) represent the mini-batch SVGP [98], while the green line (—) represents SOLVEGP [230]. Along the column labeled "SOFTPLUS features", the red and blue lines (— and —) represent the ACTIVATED SVGP [66], while the green line (—) represents our proposed approach.

ADDENDUM

3.A EXPERIMENTAL SET-UP AND IMPLEMENTATION DETAILS

3.A.1 *Hardware*

All experiments were carried out on a consumer-grade laptop computer with an Intel Core™ i7-11800H (8 Cores) @ 4.6GHz Processor, 16GB Memory, and a NVIDIA GeForce RTX™ 3070 Laptop (Mobile/Max-Q) Graphics Card.

3.A.2 *Software*

Our method is implemented by extending functionality from the GPFlow software library [167]. The code will be released as open-source software upon publication. Additional software dependencies upon which our implementation relies, either directly or indirectly, are enumerated in Table 3.A.1.

Table 3.A.1: Key software dependencies.

Method	Software Library	URL (`github.com/*`)
SVGP [262]	GPFlow	`GPflow/GPflow`
ODVGP [223]	-	`hughsalimbeni/orth_decoupled_var_gps`
SOLVEGP [230]	-	`thjashin/solvegp`
VISH [65]	Spherical Harmonics	`vdutor/SphericalHarmonics`
ACTIVATED SVGP [66]	-	`vdutor/ActivatedDeepGPs`
-	Bayesian Benchmarks	`hughsalimbeni/bayesian_benchmarks`

3.A.3 *Hyperparameters*

We adopt sensible defaults across all problems and datasets; no hand-tuning is applied to any specific one. The choices of the hyperparameters and other relevant dependencies are summarized as follows:

OPTIMISATION. We use the L-BFGS optimizer [26, 301] with the default settings from `scipy.optimize` [275].

LIKELIHOOD. The Gaussian likelihood variance is initialized to 1.0 across all experiments.

KERNEL PARAMETER INITIALISATION. All stationary kernels are initialized with unit lengthscale and amplitude.

VARIATIONAL PARAMETER INITIALISATION. The variational distributions $q(\mathbf{u})$, $q(\mathbf{v}')$ are initialized with zero mean and identity covariance $\mathbf{m} = \mathbf{0}, \mathbf{C} = \mathbf{I}$.

WHITENED PARAMETERISATION. We do *not* use the whitened parameterisation (as used, for example, by Hensman et al. [100] and Murray and Adams [182]) in either $q(\mathbf{u})$ or $q(\mathbf{v}')$.

INDUCING POINT INITIALISATION. We make our best effort to ensure a fair comparison against baselines involving standard inducing points. To this end, we adopt the best practice of first optimising the variational parameters, not least the inducing input locations $\mathbf{Z}$ (and $\mathbf{W}$ where applicable), before jointly optimising all of the free parameters. This initialisation phase is done for up to 100 iterations of the L-BFGS algorithm.

3.B ADDITIONAL RESULTS

3.B.1 *Regression on Airline Delays Dataset*

We repeat the experiment outlined in Section 3.6.3, focusing on a reduced subset of the 2008 U.S. airline delays dataset that consists of 100K randomly selected observations. Unlike the previous experimental set-up, the parameters are optimised for a total of 1,000 epochs. Additionally, we report aggregated results from 5 repetitions across $1/3$ held-out test sets. The results are shown in Figure 3.B.1.

3.B.2 *Extra UCI Repository Datasets*

Results on a few larger regression datasets from the UCI repository can be found in Figure 3.B.2. In this analysis, we adopted the same combination of activation features and sparse GP models as described in Section 3.6.2. However, in contrast to Section 3.6.2, we restrict our focus to the Arccos kernel.

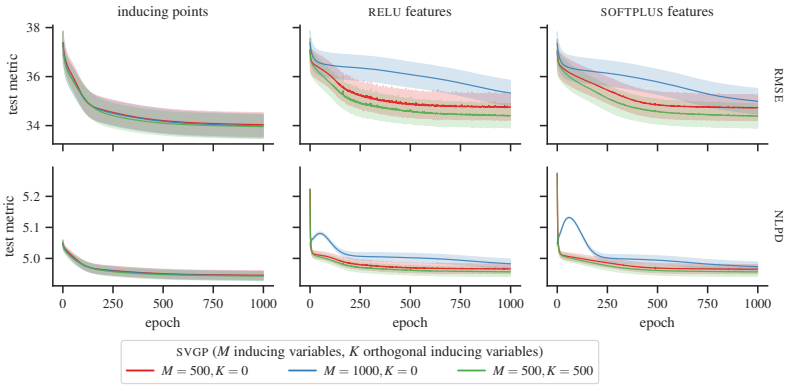


Figure 3.B.1: Test metrics, RMSE and NLPD, aggregated across 5 random subsampling test splits on a 100K subset of the 2008 U.S. airline delays dataset. Results are shown for models using the *Arccos* kernel with standard inducing points and various activation features. Along the column labeled "inducing points", the red and blue lines (— and —) represent the mini-batch SVGP [98], while the green line (—) represents SOLVEGP [230]. Along the column labeled "SOFTPLUS features", the red and blue lines (— and —) represent the ACTIVATED SVGP [66], while the green line (—) represents our proposed approach.

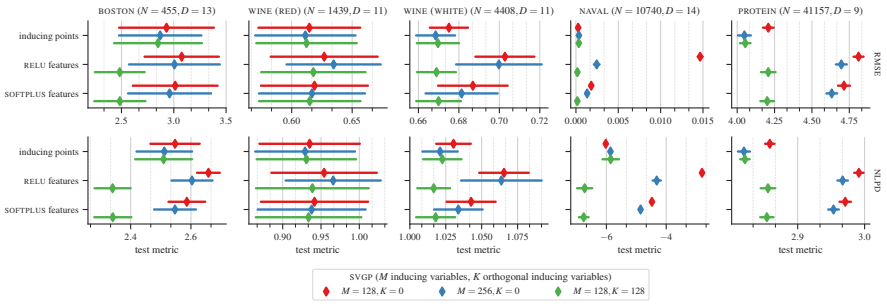


Figure 3.B.2: Test metrics, RMSE and NLPD, on an extra set of larger regression problems from the UCI dataset repository for the *Arccos* kernel and various activation features. Along the rows labeled "*inducing points*", the red and blue markers (♦, ♦) represent the original SVGP model [262], while the green markers (♦) represent SOLVEGP [230]. Along the remaining rows, the red and blue markers (♦, ♦) represent the ACTIVATED SVGP [66], while the green markers (♦) represent our proposed approach.

4

CYCLE-CONSISTENT GENERATIVE ADVERSARIAL NETWORKS AS A BAYESIAN APPROXIMATION

PREFACE

This chapter is derived from work previously published as:

Louis C Tiao, Edwin V Bonilla, and Fabio T Ramos. "Cycle-Consistent Adversarial Learning as Approximate Bayesian Inference". In: *ICML 2018 Theoretical Foundations and Applications of Deep Generative Models*. Stockholm, Sweden, July 2018. **(Accepted as *Oral* presentation).**

4.1 INTRODUCTION

Learning correspondences between entities from different domains is an important and challenging problem in ML, especially in the *absence of paired data*. Consider for example the task of image-to-image translation where we want to learn a mapping from an image in a source domain, such as a photograph of a natural scene, to a corresponding image in a target domain, such as the realisation of such a scene in an 1860s celebrated artist's signature impressionistic style. The shortage of ground-truth pairings from the source domain to the target domain renders standard supervised approaches infeasible, thus motivating the need for unsupervised learning.

Within unsupervised approaches, a number of recently proposed CYCLEGAN methods have achieved remarkable success in addressing this problem [125, 302]. As their name suggests, these approaches are based upon two heuristics: (i) adversarial learning and (ii) cycle consistency. The former, adversarial learning [86], allows images in the source domain to be translated to output images that, to an auxiliary discriminator, are indistinguishable from images in the target domain, thereby matching their distributions. However, while distribution matching is necessary, it is insufficient to guarantee one-to-one mappings between the images, as the problem is heavily under-constrained. Briefly stated, the cycle-consistency is the constraint that an image mapped to a target domain should be *representable* in the original domain. It is this constraint that significantly shrinks the space of possible solutions.

Beyond the empirical risk minimisation framework motivated intuitively by the two principles mentioned above, the original CYCLEGAN formulation lacks any further theoretical justification. Besides providing sound quantification of uncertainty, a LVM allows us to disentangle our modelling assumptions from the inference machinery used to

reason about the model variables. Interpreting standard methods from a Bayesian perspective has contributed significantly to the understanding of these methods and to the development of new approaches [74, 261].

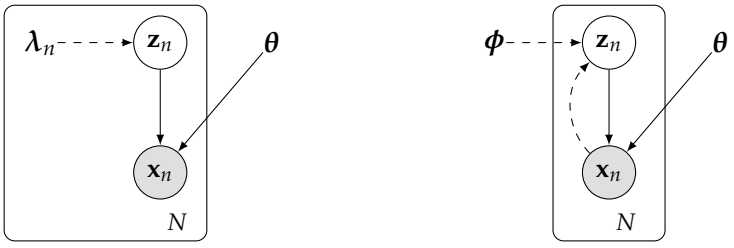
In this chapter, we introduce *implicit* LVMs, where the prior over hidden representations can be specified flexibly as an *implicit* distribution. We develop a VI algorithm for this model based on minimisation of the *symmetric* KL divergence between a *variational joint* and the exact joint distribution, in contrast to traditional reverse KL minimisation, which notoriously underestimate posterior's / exact distribution's support. Lastly, we demonstrate that the state-of-the-art CYCLEGANS as proposed contemporaneously by Kim et al. [125] and Zhu et al. [302] can be derived as a special case within our proposed VI framework, thus establishing its connection to approximate Bayesian inference methods.

4.2 IMPLICIT LATENT VARIABLE MODELS

LVMs are an indispensable tool for uncovering the hidden representations of observed data. In a LVM, an observation $\mathbf{x}$ is assumed governed by its underlying hidden variable $\mathbf{z}$, which is drawn from a prior $p(\mathbf{z})$ and related to $\mathbf{x}$ through the likelihood $p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})$. Accordingly, the joint density of $\mathbf{x}$ and $\mathbf{z}$ is given by

$$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z}). \tag{4.1}$$

Given data distribution $q^*(\mathbf{x})$ and a finite collection $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$ of observations $\mathbf{x}_n \sim q^*(\mathbf{x})$, and the set of corresponding latent variables $\mathbf{Z} = \{\mathbf{z}_n\}_{n=1}^N$, the joint over all variables factorises as, $p_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{Z}) = \prod_{n=1}^N p_{\boldsymbol{\theta}}(\mathbf{x}_n, \mathbf{z}_n)$.



(a) Without amortised inference, each local latent variable is governed by its own local variational parameters.

(b) With amortized inference, we condition on observed variables and employ a single set of global variational parameters.

Figure 4.1: Graphical representation of the *generative model* (solid) and the *recognition model* (dashed).

The graphical representation of implicit LVMs is depicted in Figure 4.1. Instead of approximating the exact posterior $p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{x}_n)$ for each $\mathbf{x}_n$, using a separate variational distribution $q(\mathbf{z}; \boldsymbol{\lambda}_n)$ with local

variational parameters $\boldsymbol{\lambda}_n$, we condition on $\mathbf{x}$ and optimise a single set of variational parameters $\boldsymbol{\phi}$ across all $\mathbf{x} \sim q^*(\mathbf{x})$. Accordingly, the variational distribution is denoted $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) \triangleq q(\mathbf{z} \mid \mathbf{x}; \boldsymbol{\phi})$.

4.2.1 *Prescribed Likelihood*

We specify the likelihood through a mapping $\mathcal{F}_{\boldsymbol{\theta}}$ that takes as input random noise $\boldsymbol{\xi}$ and latent variable $\mathbf{z}$,

$$\mathbf{x} \sim p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}) \quad \Leftrightarrow \quad \mathbf{x} = \mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\xi}; \mathbf{z}), \quad \boldsymbol{\xi} \sim p(\boldsymbol{\xi}). \tag{4.2}$$

We shall restrict our attention to *prescribed* likelihoods, where evaluation of their density is tractable. This requires that $\mathcal{F}_{\boldsymbol{\theta}}(\,\cdot\,; \mathbf{z})$ be a diffeomorphism wrt $\boldsymbol{\xi}$ and density $p(\boldsymbol{\xi})$ be tractable. For example, when $\mathcal{F}_{\boldsymbol{\theta}}(\,\cdot\,; \mathbf{z})$ is a location-scale transform of noise $\boldsymbol{\xi}$ and $p(\boldsymbol{\xi})$ is Gaussian, we recover Gaussian observation models.

Our model specification is sufficiently general for encapsulating a broad range of familiar latent variable models, even when we make simplifying assumptions on the mapping $\mathcal{F}_{\boldsymbol{\theta}}(\,\cdot\,; \mathbf{z})$. In particular, consider the special case where the mapping is an affine transformation of the noise vector $\boldsymbol{\xi}$,

$$\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\xi}; \mathbf{z}) \triangleq \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}) + \boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\mathbf{z})^{\frac{1}{2}} \boldsymbol{\xi}, \quad \boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

for functions $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ parameterised by $\boldsymbol{\theta}$ that take $\mathbf{z}$ as input. To simplify matters further, assume $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ is constant wrt to its input, i. e. $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\mathbf{z}) = \boldsymbol{\Psi}$ for all $\mathbf{z}$. The likelihood can then be written explicitly as

$$p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}), \boldsymbol{\Psi}).$$

FACTOR ANALYSIS & PROBABILISTIC PCA. In the case where the mean function $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ is an affine transformation of $\mathbf{z}$,

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}) \triangleq \mathbf{W}\mathbf{z} + \mathbf{b},$$

and the covariance matrix is diagonal $\boldsymbol{\Psi} = \operatorname{diag}(\psi_1^2, \ldots, \psi_D^2)$, we recover FA [8]. Furthermore, when the covariance matrix is isotropic $\boldsymbol{\Psi} = \psi^2 \mathbf{I}$, we recover *probabilistic principal component analysis* (PPCA) [261]. In this example, the parameters $\boldsymbol{\theta}$ consist of the factor loading matrix $\mathbf{W}$, the bias vector $\mathbf{b}$ and the covariance matrix $\boldsymbol{\Psi}$.

DEEP AND NONLINEAR LATENT VARIABLE MODELS. By introducing nonlinearities to the mean function, we are able to recover nonlinear factor analysis [139], nonlinear Gaussian sigmoid belief networks [72], and other more sophisticated variants of deep latent variable models. When the mapping is defined by a MLP, we can recover simple instances of a variational autoencoder (VAE) with a Gaussian probabilistic decoder [127, 212].

4.2.2 *Implicit Prior*

In LVMs, the prior typically specified as a prescribed distribution, e. g. a factorised Gaussian centered at zero. Oftentimes, however, the practitioner possesses prior knowledge that simply cannot be embodied within a prescribed distribution. To address this limitation, we introduce *implicit* LVMs, wherein the prior over latent variables is specified as an implicit distribution $p^*(\mathbf{z})$, given only by a finite collection $\mathbf{Z}^* = \{\mathbf{z}^*_m\}_{m=1}^M$ of its samples,

$$\mathbf{z}^*_m \sim p^*(\mathbf{z}). \tag{4.3}$$

This formulation offers the utmost degree of flexibility in the treatment of prior information, the difficulties of which have hindered the application of Bayesian statistics since the time of Laplace [115].

EXAMPLE: UNPAIRED IMAGE-TO-IMAGE TRANSLATION . Suppose we have collections of images $\mathbf{X}$ and $\mathbf{Z}^*$, which are assumed to be draws from the data distribution $q^*(\mathbf{x})$ and implicit prior distribution $p^*(\mathbf{z})$, respectively. For example, these might be photographs of natural landscapes and the paintings of Van Gogh. The goal of unpaired image-to-image translation is to learn the correspondence between variables $\mathbf{x}$ and $\mathbf{z}$ by capturing the underlying generative process specified by mapping $\mathcal{F}_\theta$. This defines the likelihood $p_\theta(\mathbf{x} \mid \mathbf{z})$ –a conditional density of $\mathbf{x}$ given $\mathbf{z}$. Continuing with the above example, the problem amounts to learning parameters $\boldsymbol{\theta}$ of the mapping such that this conditional yields photorealistic renderings of scenes portrayed in Van Gogh's paintings. Furthermore, the resulting posterior on the latent representation $p_\theta(\mathbf{z} \mid \mathbf{x})$ – a conditional density of $\mathbf{z}$ given $\mathbf{x}$ – should produce renderings of landscape scenery in Van Gogh's iconic style.

4.3 VARIATIONAL INFERENCE

In this section, we describe the first component of our bipartite VI framework. In traditional VI, one specifies a family $\mathcal{Q}$ of densities over the latent variables and seeks the member $q \in \mathcal{Q}$ closest in KL divergence to the exact posterior $p_\theta(\mathbf{z} \mid \mathbf{x})$ [17, 119, 276].

4.3.1 *Prescribed Variational Posterior*

We begin by describing the variational family $q \in \mathcal{Q}$. We adopt the common practice of *amortising* inference using an inference network [83]. Namely, instead of approximating the exact posterior $p_\theta(\mathbf{z} \mid \mathbf{x}_n)$ for each $\mathbf{x}_n$, using a separate variational distribution $q(\mathbf{z}; \boldsymbol{\lambda}_n)$ with local variational parameters $\boldsymbol{\lambda}_n$, we condition on $\mathbf{x}$ and optimise a single set of variational parameters $\boldsymbol{\phi}$ across all $\mathbf{x} \sim q^*(\mathbf{x})$.

The variational distribution $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$ is specified through an inverse mapping $\mathcal{G}_{\boldsymbol{\phi}}$ that takes as input random noise $\boldsymbol{\epsilon}$ and observed variable $\mathbf{x}$,

$$\mathbf{z} \sim q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) \quad \Leftrightarrow \quad \mathbf{z} = \mathcal{G}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}; \mathbf{x}), \quad \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon}). \tag{4.4}$$

Just as mapping $\mathcal{F}_{\theta}$ underpins the generative model, mapping $\mathcal{G}_{\boldsymbol{\phi}}$ underpins the *recognition model* [52]. As with the likelihood, *we restrict our attention to prescribed variational distributions.*

As depicted in Figure 4.1b, the dependency relationship between the variational parameters and the latent variables mirrors that of the model parameters and observed variables. This symmetry is crucial to the derivation of CYCLEGAN later in Section 4.5.3.2.

4.3.2 *Reverse KL Variational Objective*

Minimising the reverse KL between the exact and variational posterior is equivalent to maximising the ELBO, or minimising its *negative*, defined as

$$\begin{aligned} \mathcal{L}_{\text{NELBO}}(\boldsymbol{\theta}, \boldsymbol{\phi}) \triangleq \mathbb{E}_{q^*(\mathbf{x}) q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})}[-\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})] \\ + \mathbb{E}_{q^*(\mathbf{x})} \text{KL}\left[q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) \parallel p^*(\mathbf{z})\right]. \end{aligned} \tag{4.5}$$

The first term of the ELBO is the (negative) ELL, defined as

$$\mathcal{L}_{\text{NELL}}(\boldsymbol{\theta}, \boldsymbol{\phi}) \triangleq \mathbb{E}_{q^*(\mathbf{x}) q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})}[-\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})]. \tag{4.6}$$

It is easy to perform stochastic gradient-based optimisation of this term by applying the reparameterisation trick [127, 212],

$$\mathcal{L}_{\text{NELL}}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{q^*(\mathbf{x}) p(\boldsymbol{\epsilon})}[-\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathcal{G}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}; \mathbf{x}))]. \tag{4.7}$$

However, the second term – the KL divergence between $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$ and implicit prior $p^*(\mathbf{z})$ – is not so straightforward. In particular, the KL divergence can be expressed as

$$\text{KL}\left[q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) \parallel p^*(\mathbf{z})\right] \triangleq \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})}[\log r^*(\mathbf{z}; \mathbf{x})], \tag{4.8}$$

where $r^*(\mathbf{z}; \mathbf{x})$ is defined as the ratio of densities,

$$r^*(\mathbf{z}; \mathbf{x}) \triangleq \frac{q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})}{p^*(\mathbf{z})}. \tag{4.9}$$

The dependence on this density ratio is problematic since the prior $p^*(\mathbf{z})$ is implicit and cannot be evaluated directly. To overcome this, we resort to methods for approximating f-divergences between implicit distributions, which are inextricably tied to DRE [177, 251].

4.3.3 *Approximate Divergence Minimisation*

Although we are primarily interested in estimating the KL divergence of Equation (4.8), we give a generalised treatment that is applicable to all f-divergences [2, 47]. We denote a generic member of the family of f-divergences between distributions p and q as $\mathcal{D}_f\left[p \parallel q\right] \triangleq \mathbb{E}_p[f(q/p)]$, for some convex lower-semicontinuous function $f : \mathbb{R}_+ \to \mathbb{R}$.

Leveraging results from convex analysis, Nguyen, Wainwright, and Jordan [188] devise a variational lower bound that estimates an f-divergence through samples when either or both of the densities are unavailable. Nowozin, Cseke, and Tomioka [191] extend this framework to derive GAN objectives that minimise arbitrary f-divergences. These results underpin our methodology, and we restate a variant of it here for completeness.

Theorem 4.3.1 (Nguyen, Wainwright, and Jordan [188])**.** *Let $f^\star$ be the convex dual of f and $\mathcal{R}$ a class of functions with codomains equivalent to the domain of f'. We have the following lower bound on the f-divergence between distributions $p(\mathbf{u})$ and $q(\mathbf{u})$,*

$$\begin{aligned}\mathcal{D}_f\left[p(\mathbf{u}) \parallel q(\mathbf{u})\right] \geq \max_{\hat{r}\in\mathcal{R}}\{&\mathbb{E}_{q(\mathbf{u})}[f'(\hat{r}(\mathbf{u}))] \\ &- \mathbb{E}_{p(\mathbf{u})}[f^\star(f'(\hat{r}(\mathbf{u})))]\},\end{aligned}$$

where equality is attained when $\hat{r}(\mathbf{u})$ is exactly the true density ratio $\hat{r}(\mathbf{u}) = {}^{q(\mathbf{u})}/_{p(\mathbf{u})}$.

Applying Theorem 4.3.1 to $p^*(\mathbf{z})$ and $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}_n)$ for a given $\mathbf{x}_n$, and optimising over a class of functions indexed by parameters $\boldsymbol{\omega}_n$, we obtain the following lower bound on their divergence,

$$\mathcal{D}_f\left[p^*(\mathbf{z}) \parallel q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}_n)\right] \geq \max_{\boldsymbol{\omega}_n}\left\{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}\mid\mathbf{x}_n)}[f'(r_{\boldsymbol{\omega}_n}(\mathbf{z}))] - \mathbb{E}_{p^*(\mathbf{z})}[f^\star(f'(r_{\boldsymbol{\omega}_n}(\mathbf{z})))]\right\}.$$

While this provides a way to estimate any f-divergence between implicit prior $p^*(\mathbf{z})$ and variational distribution $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}_n)$ with only samples, it also requires us to optimise a separate density ratio estimator with parameters $\boldsymbol{\omega}_n$ for each observed $\mathbf{x}_n$. Instead, as with the posterior approximation, we also amortise the density ratio estimator by conditioning on $\mathbf{x}$ and optimising a single set of parameters $\boldsymbol{\alpha}$ across all $\mathbf{x} \sim q^*(\mathbf{x})$. Accordingly, the estimator becomes $r_{\boldsymbol{\alpha}}(\mathbf{z}; \mathbf{x})$, taking also $\mathbf{x}$ as input. We now maximise an instance of the following generalised objective,

$$\begin{aligned}\mathcal{L}_f^{\text{latent}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi}) \triangleq \mathbb{E}_{q^*(\mathbf{x})q_{\boldsymbol{\phi}}(\mathbf{z}\mid\mathbf{x})}&[f'(r_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x}))] \\ &- \mathbb{E}_{q^*(\mathbf{x})p^*(\mathbf{z})}[f^\star(f'(r_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x})))].\end{aligned} \tag{4.10}$$

Corollary 4.3.2. *We have the lower bound,*

$$\mathbb{E}_{q^*(\mathbf{x})}\mathcal{D}_f\left[p^*(\mathbf{z}) \parallel q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})\right] \geq \max_{\boldsymbol{\alpha}} \mathcal{L}_f^{\text{latent}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi}), \tag{4.11}$$

with equality at $r_{\boldsymbol{\alpha}}(\mathbf{z}; \mathbf{x}) = r^*(\mathbf{z}; \mathbf{x})$.

DENSITY RATIO ESTIMATION OBJECTIVE. We write $\mathcal{L}_f^{\text{latent}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi})$ to denote the DRE objective, wherein $\boldsymbol{\phi}$ is fixed, while $\boldsymbol{\alpha}$ is a free parameter that varies as this objective is *maximised*, thus tightening the bound of Equation (4.11) and the estimate of the density ratio $r_{\boldsymbol{\alpha}}(\mathbf{z}; \mathbf{x})$.

DIVERGENCE MINIMISATION LOSS. Conversely, the *divergence minimisation (*DM*)* loss, denoted as $\mathcal{L}_f^{\text{latent}}(\boldsymbol{\phi} \mid \boldsymbol{\alpha})$, is *minimised* wrt $\boldsymbol{\phi}$ while $\boldsymbol{\alpha}$ remains fixed, thus approximately minimising the f-divergence. In theory, this should be symmetric to the DRE objective, $\mathcal{L}_f^{\text{latent}}(\boldsymbol{\phi} \mid \boldsymbol{\alpha}) \triangleq \mathcal{L}_f^{\text{latent}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi})$. However, alternative settings are often used in practice to alleviate the problem of vanishing gradients, as we shall see in Section 4.5.

By applying Theorem 4.3.2 for the setting $f_{\text{KL}}(u) \triangleq u \log u$, we instantiate a lower bound on the KL divergence of Equation (4.8) in the following objective,

$$\mathcal{L}_{\text{KL}}^{\text{latent}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi}) \triangleq \mathbb{E}_{q^*(\mathbf{x}) q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})}[\log r_{\boldsymbol{\alpha}}(\mathbf{z}; \mathbf{x})] - \mathbb{E}_{q^*(\mathbf{x}) p^*(\mathbf{z})}[r_{\boldsymbol{\alpha}}(\mathbf{z}; \mathbf{x}) - 1]. \tag{4.12}$$

As we discuss in Section 4.A, maximisation of the objective in Equation (4.12) is closely related to the KLIEP [250].

Now, we define the DM loss symmetrically to the DRE objective in Equation (4.12) – terms not involving $\boldsymbol{\phi}$ are omitted,

$$\begin{aligned} \mathcal{L}_{\text{KL}}^{\text{latent}}(\boldsymbol{\phi} \mid \boldsymbol{\alpha}) &\triangleq \mathbb{E}_{q^*(\mathbf{x}) q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})}[\log r_{\boldsymbol{\alpha}}(\mathbf{z}; \mathbf{x})] \\ &\approx \mathbb{E}_{q^*(\mathbf{x})} \text{KL}\left[q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) \parallel p^*(\mathbf{z})\right]. \end{aligned} \tag{4.13}$$

Combined with the ELL, this estimate of the KL divergence yields an approximation to the ELBO where all terms are tractable. These objectives are summarised in the bi-level optimisation problem below,

$$\max_{\boldsymbol{\alpha}} \quad \mathcal{L}_{\text{KL}}^{\text{latent}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi}), \tag{4.14a}$$

$$\min_{\boldsymbol{\phi}, \boldsymbol{\theta}} \quad \mathcal{L}_{\text{KL}}^{\text{latent}}(\boldsymbol{\phi} \mid \boldsymbol{\alpha}) + \mathcal{L}_{\text{NELL}}(\boldsymbol{\theta}, \boldsymbol{\phi}), \tag{4.14b}$$

thus concluding the reverse KL minimisation component of our VI framework.

4.4 SYMMETRIC JOINT-MATCHING VARIATIONAL INFERENCE

We now complete the remaining component of our VI framework. In the previous section, we gave an extension to classical VI, which is fundamentally concerned with approximating the exact posterior. Now, let us instead consider directly approximating the *exact joint* $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$ through a *variational joint* $q_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z})$.

4.4.1 *Variational Joint*

Recall that $q^*(\mathbf{x})$ denotes the empirical data distribution. We define a variational approximation to the exact joint distribution of Equation (4.1) as

$$q_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z}) \triangleq q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) q^*(\mathbf{x}). \tag{4.15}$$

We approximate the exact joint by seeking a variational joint closest in *symmetric* KL divergence, $\mathrm{KL}_{\mathrm{SYM}}\left[p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) \parallel q_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z})\right]$, where

$$\mathrm{KL}_{\mathrm{SYM}}\left[p \parallel q\right] \triangleq \mathrm{KL}\left[p \parallel q\right] + \mathrm{KL}\left[q \parallel p\right]. \tag{4.16}$$

We first look at the reverse KL divergence ($\mathrm{KL}\left[q \parallel p\right]$) term. When expanded, we see that it is equivalent to the negative ELBO up to additive constants,

$$\begin{aligned} \mathrm{KL}\left[q_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z}) \parallel p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})\right] &\triangleq \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z})}\left[\log q_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z}) - \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})\right] && (4.17) \\ &= \mathcal{L}_{\mathrm{NELBO}}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \mathbb{H}[q^*(\mathbf{x})], && (4.18) \end{aligned}$$

where $\mathbb{H}[q^*(\mathbf{x})] \triangleq \mathbb{E}_{q^*(\mathbf{x})}[-\log q^*(\mathbf{x})]$ is the entropy of $q^*(\mathbf{x})$, a constant wrt parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. Hence, minimising the KL divergence of Equation (4.17) can be reduced to minimising $\mathcal{L}_{\mathrm{NELBO}}(\boldsymbol{\theta}, \boldsymbol{\phi})$ of Equation (4.5), without modification.

4.4.2 *Forward KL Variational Objective*

As for the forward KL divergence ($\mathrm{KL}\left[p \parallel q\right]$) term, we have a similar expansion,

$$\begin{aligned} &\mathrm{KL}\left[p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) \parallel q_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z})\right] && (4.19) \\ &\quad \triangleq \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z})\right] && (4.20) \\ &\quad = \mathbb{E}_{p^*(\mathbf{z}) p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})}[\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z})] - \mathbb{H}[p^*(\mathbf{z})]. && (4.21) \end{aligned}$$

In analogy with the ELBO, we introduce a new variational objective that is minimised when the forward KL divergence of Equation (4.19) is minimised. First we define the recognition model analog to the marginal likelihood – the *marginal posterior*, or *aggregated posterior*, given by $q_\phi(\mathbf{z}) \triangleq \int q_\phi(\mathbf{z} \mid \mathbf{x}) q^*(\mathbf{x}) d\mathbf{x}$. It can be approximated by the *aggregate posterior lower bound (APLBO)*. For consistency, we give its *negative*, written as

$$\begin{aligned} \mathcal{L}_{\mathrm{NAPLBO}}(\boldsymbol{\theta}, \boldsymbol{\phi}) \triangleq\ & \mathbb{E}_{p^*(\mathbf{z}) p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})}[-\log q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})] \\ & + \mathbb{E}_{p^*(\mathbf{z})} \mathrm{KL}\left[p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}) \parallel q^*(\mathbf{x})\right]. \end{aligned} \tag{4.22}$$

Furthermore, minimising the KL divergence of Equation (4.19) can be reduced to minimising $\mathcal{L}_{\mathrm{NAPLBO}}(\boldsymbol{\theta}, \boldsymbol{\phi})$,

$$\mathrm{KL}\left[p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) \parallel q_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z})\right] = \mathcal{L}_{\mathrm{NAPLBO}}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \mathbb{H}[p^*(\mathbf{z})].$$

The first term of the negative APLBO is the (negative) *expected log posterior (*ELP*)*, defined as

$$\mathcal{L}_{\text{NELP}}(\boldsymbol{\theta}, \boldsymbol{\phi}) \triangleq \mathbb{E}_{p^*(\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{x}\,|\,\mathbf{z})}[-\log q_{\boldsymbol{\phi}}(\mathbf{z}\,|\,\mathbf{x})]. \tag{4.23}$$

We emphasise a key advantage of having considered the KL between the joint distributions instead of between the *posteriors*. Computing the *forward* KL divergence between the exact and approximate *posterior* distribution is problematic, since it requires evaluating expectations over the exact posterior $p_{\boldsymbol{\theta}}(\mathbf{z}\,|\,\mathbf{x})$, the intractability of which is the reason we appealed to approximate inference in the first place.

In contrast, the forward KL divergence between the exact and approximate *joint* poses no such difficulties – we are able to sidestep the dependency on the exact posterior by expanding it into the form of Equation (4.21). Furthermore, as with the ELBO, we can perform stochastic gradient-based optimisation of the ELP term by applying the same reparameterisation trick as in Equation (4.7).

Now, the KL divergence term of the APLBO in Equation (4.22) can also be expressed as the expected logarithm of a density ratio $r^*(\mathbf{x};\mathbf{z}) \triangleq p_{\boldsymbol{\theta}}(\mathbf{x}\,|\,\mathbf{z})/q^*(\mathbf{x})$ that involves an intractable density $q^*(\mathbf{x})$ – the empirical data distribution. To overcome this, we adopt the same approach as outlined in Section 4.3.3. Namely, we apply Theorem 4.3.1 to $q^*(\mathbf{x})$ and $p_{\boldsymbol{\theta}}(\mathbf{x}\,|\,\mathbf{z}^*)$, and fit an amortised density ratio estimator $r_{\boldsymbol{\beta}}(\mathbf{x};\mathbf{z})$ to $r^*(\mathbf{x};\mathbf{z})$ by maximising an instance of the generalised objective,

$$\begin{aligned}\mathcal{L}_f^{\text{observed}}(\boldsymbol{\beta}\,|\,\boldsymbol{\theta}) \triangleq\ & \mathbb{E}_{p^*(\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{x}\,|\,\mathbf{z})}[f'(r_{\boldsymbol{\beta}}(\mathbf{x};\mathbf{z}))] \\ & - \mathbb{E}_{p^*(\mathbf{z})q^*(\mathbf{x})}[f^\star(f'(r_{\boldsymbol{\beta}}(\mathbf{x};\mathbf{z})))].\end{aligned} \tag{4.24}$$

Corollary 4.4.1. *We have the lower bound,*

$$\mathbb{E}_{p^*(\mathbf{z})}\mathcal{D}_f\left[q^*(\mathbf{x}) \parallel p_{\boldsymbol{\theta}}(\mathbf{x}\,|\,\mathbf{z})\right] \geq \max_{\boldsymbol{\beta}} \mathcal{L}_f^{\text{observed}}(\boldsymbol{\beta}\,|\,\boldsymbol{\theta}), \tag{4.25}$$

with equality at $r_{\boldsymbol{\beta}}(\mathbf{x};\mathbf{z}) = r^*(\mathbf{x};\mathbf{z})$.

By applying Theorem 4.4.1 with the previously defined $f_{\text{KL}}(u)$, we obtain lower bound objective $\mathcal{L}_{\text{KL}}^{\text{observed}}(\boldsymbol{\beta}\,|\,\boldsymbol{\theta})$ on the KL divergence term in Equation (4.22), and a corresponding DM loss $\mathcal{L}_{\text{KL}}^{\text{observed}}(\boldsymbol{\theta}\,|\,\boldsymbol{\beta})$, analogous to the definitions of $\mathcal{L}_{\text{KL}}^{\text{latent}}(\boldsymbol{\alpha}\,|\,\boldsymbol{\phi})$ and $\mathcal{L}_{\text{KL}}^{\text{latent}}(\boldsymbol{\phi}\,|\,\boldsymbol{\alpha})$ in Equations (4.12) and (4.13), respectively. See Table 4.B.3 for a summary of explicit definitions.

Hence, in addition to the bi-level optimisation problems of Equation (4.14) we have,

$$\max_{\boldsymbol{\beta}} \quad \mathcal{L}_{\text{KL}}^{\text{observed}}(\boldsymbol{\beta}\,|\,\boldsymbol{\theta}), \tag{4.26a}$$

$$\min_{\boldsymbol{\phi},\boldsymbol{\theta}} \quad \mathcal{L}_{\text{KL}}^{\text{observed}}(\boldsymbol{\theta}\,|\,\boldsymbol{\beta}) + \mathcal{L}_{\text{NELP}}(\boldsymbol{\theta}, \boldsymbol{\phi}). \tag{4.26b}$$

As shown, the minimisations in Equations (4.14b) and (4.26b) corresponds to minimisation of the symmetric KL over the joints $\text{KL}_{\text{SYM}}\left[p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z}) \parallel q_{\boldsymbol{\phi}}(\mathbf{x},\mathbf{z})\right]$,

while the maximisations in Equations (4.14a) and (4.26a) approximates the divergences, or more precisely, the density ratios involving implicit distributions.

4.5 CYCLEGAN AS A SPECIAL CASE

In this section, we demonstrate that CYCLEGAN methods [125, 302] can be instantiated under our proposed VI framework.

4.5.1 *Basic CycleGAN Framework*

To address the problem of unpaired image-to-image translation as described in Section 4.2.2, the CYCLEGAN model learns two mappings $\boldsymbol{\mu}_{\boldsymbol{\theta}} : \mathbf{z} \mapsto \mathbf{x}$ and $\mathbf{m}_{\boldsymbol{\phi}} : \mathbf{x} \mapsto \mathbf{z}$ by optimising two complementary classes of objectives.

DISTRIBUTION MATCHING. The first are the adversarial objectives, which help match the output of mapping $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ to the empirical distribution $q^*(\mathbf{x})$, and the output of $\mathbf{m}_{\boldsymbol{\phi}}$ to $p^*(\mathbf{z})$. In particular, for mapping $\mathbf{m}_{\boldsymbol{\phi}}$, this involves introducing a discriminator $\mathbf{D}_{\boldsymbol{\alpha}}$ and the saddle-point adversarial objective,

$$\begin{aligned}\ell_{\text{GAN}}^{\text{reverse}}(\boldsymbol{\alpha} \,|\, \boldsymbol{\phi}) \triangleq \mathbb{E}_{p^*(\mathbf{z})}[\log \mathbf{D}_{\boldsymbol{\alpha}}(\mathbf{z})] \\ + \mathbb{E}_{q^*(\mathbf{x})}[\log(1 - \mathbf{D}_{\boldsymbol{\alpha}}(\mathbf{m}_{\boldsymbol{\phi}}(\mathbf{x})))],\end{aligned} \tag{4.27}$$

while minimising it wrt parameters $\boldsymbol{\phi}$. This encourages $\mathbf{m}_{\boldsymbol{\phi}}$ to produce realistic outputs $\mathbf{z} = \mathbf{m}_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{x} \sim q^*(\mathbf{x})$ which, to the discriminator $\mathbf{D}_{\boldsymbol{\alpha}}$, are "indistinguishable" from $\mathbf{z}^* \sim p^*(\mathbf{z})$. A similar adversarial objective is defined for mapping $\boldsymbol{\mu}_{\boldsymbol{\theta}}$,

$$\ell_{\text{GAN}}^{\text{forward}}(\boldsymbol{\beta} \,|\, \boldsymbol{\theta}) \triangleq \mathbb{E}_{p^*(\mathbf{x})}[\log \mathbf{D}_{\boldsymbol{\beta}}(\mathbf{x})] + \mathbb{E}_{p^*(\mathbf{z})}[\log(1 - \mathbf{D}_{\boldsymbol{\beta}}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z})))]. \tag{4.28}$$

CYCLE-CONSISTENCY. Next are the cycle-consistency losses, which enforce tight correspondence between domains by ensuring that reconstruction $\mathbf{x}' = \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{m}_{\boldsymbol{\phi}}(\mathbf{x}))$ is close to the input $\mathbf{x}$, and likewise for $\mathbf{m}_{\boldsymbol{\phi}}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}))$. This is achieved by minimising a reconstruction loss,

$$\ell_{\text{CONST}}^{\text{reverse}}(\boldsymbol{\theta}, \boldsymbol{\phi}) \triangleq \mathbb{E}_{q^*(\mathbf{x})}[\|\mathbf{x} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{m}_{\boldsymbol{\phi}}(\mathbf{x}))\|_\rho^\rho], \tag{4.29}$$

where $\|\cdot\|_\rho$ denotes the ℓ_ρ-norm. A similar loss $\ell_{\text{CONST}}^{\text{forward}}(\boldsymbol{\theta}, \boldsymbol{\phi})$ is defined for the reconstruction of $\mathbf{z}$,

$$\ell_{\text{CONST}}^{\text{forward}}(\boldsymbol{\theta}, \boldsymbol{\phi}) \triangleq \mathbb{E}_{p^*(\mathbf{z})}[\|\mathbf{z} - \mathbf{m}_{\boldsymbol{\phi}}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}))\|_\rho^\rho]. \tag{4.30}$$

These objectives are summarised in the following set of optimisation problems,

$$\max_{\alpha} \ell_{\text{GAN}}^{\text{reverse}}(\alpha \mid \phi), \quad \max_{\beta} \ell_{\text{GAN}}^{\text{forward}}(\beta \mid \theta), \tag{4.31a}$$

$$\min_{\phi,\theta} \ell_{\text{GAN}}^{\text{reverse}}(\phi \mid \alpha) + \ell_{\text{CONST}}^{\text{reverse}}(\theta, \phi), \tag{4.31b}$$

$$\min_{\phi,\theta} \ell_{\text{GAN}}^{\text{forward}}(\theta \mid \beta) + \ell_{\text{CONST}}^{\text{forward}}(\theta, \phi). \tag{4.31c}$$

We now highlight the correspondences between these objectives and those of our proposed VI framework, as summarised in the optimisation problems of Equations (4.14) and (4.26).

4.5.2 *Cycle-consistency as Conditional Entropy Maximisation*

We now demonstrate that minimising the cycle-consistency losses corresponds to maximising the expected log likelihood and variational posterior of Equations (4.6) and (4.23). This can be shown by instantiating specific classes of $p_\theta(\mathbf{x} \mid \mathbf{z})$ and $q_\phi(\mathbf{z} \mid \mathbf{x})$ that recover $\ell_{\text{CONST}}^{\text{reverse}}(\theta, \phi)$ and $\ell_{\text{CONST}}^{\text{forward}}(\theta, \phi)$ from $\mathcal{L}_{\text{NELL}}(\theta, \phi)$ and $\mathcal{L}_{\text{NELP}}(\theta, \phi)$, respectively.

Proposition 4.5.1. *Consider a typical case where the likelihood and the posterior approximation are both Gaussians,*

$$p_\theta(\mathbf{x} \mid \mathbf{z}) \triangleq \mathcal{N}(\mathbf{x} \mid \mu_\theta(\mathbf{z}), \tau^{-1}\mathbf{I}), \qquad q_\phi(\mathbf{z} \mid \mathbf{x}) \triangleq \mathcal{N}(\mathbf{z} \mid \mathbf{m}_\phi(\mathbf{x}), t^{-1}\mathbf{I}).$$

In the limit as the posterior precision t tends to ∞, $\mathcal{L}_{\text{NELL}}(\theta, \phi)$ approaches $\ell_{\text{CONST}}^{\text{reverse}}(\theta, \phi)$ for $\rho = 2$, up to constants[1]. More precisely,

$$\mathcal{L}_{\text{NELL}}(\theta, \phi) \to \frac{\tau}{2}\mathcal{L}_{\text{CONST}}^{\text{reverse}}(\theta, \phi) + \text{const}, \quad \text{as } t \to \infty$$

Similarly, we have,

$$\mathcal{L}_{\text{NELP}}(\theta, \phi) \to \frac{t}{2}\mathcal{L}_{\text{CONST}}^{\text{forward}}(\theta, \phi) + \text{const}, \quad \text{as } \tau \to \infty$$

Proof. First, note the generative mappings underlying the given Gaussian likelihood and approximate posterior are

$$\begin{aligned} & \mathbf{z} \sim p_\theta(\mathbf{x} \mid \mathbf{z}) \triangleq \mathcal{N}(\mathbf{x} \mid \mu_\theta(\mathbf{z}), \tau^{-1}\mathbf{I}), \\ \Leftrightarrow \quad & \mathbf{z} = \mathcal{F}_\theta(\xi; \mathbf{z}) \triangleq \mu_\theta(\mathbf{z}) + \tau^{-\frac{1}{2}}\xi, \qquad \xi \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \end{aligned}$$

and,

$$\begin{aligned} & \mathbf{x} \sim q_\phi(\mathbf{z} \mid \mathbf{x}) \triangleq \mathcal{N}(\mathbf{z} \mid \mathbf{m}_\phi(\mathbf{x}), t^{-1}\mathbf{I}), \\ \Leftrightarrow \quad & \mathbf{x} = \mathcal{G}_\phi(\epsilon; \mathbf{x}) \triangleq \mathbf{m}_\phi(\mathbf{x}) + t^{-\frac{1}{2}}\epsilon, \qquad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \end{aligned}$$

1 we obtain the same result for the case $\rho = 1$ by instead setting both the likelihood and approximate posterior to be Laplace distributions.

respectively. Thus, expanding out $\mathcal{L}_{\text{NELL}}(\boldsymbol{\theta}, \boldsymbol{\phi})$, we have

$$\begin{aligned}
&\mathcal{L}_{\text{NELL}}(\boldsymbol{\theta}, \boldsymbol{\phi}) \\
&\quad = \mathbb{E}_{q^*(\mathbf{x})q_{\boldsymbol{\phi}}(\mathbf{z}\,|\,\mathbf{x})}[-\log p_{\boldsymbol{\theta}}(\mathbf{x}\,|\,\mathbf{z})] \\
&\quad = \mathbb{E}_{q^*(\mathbf{x})p(\boldsymbol{\epsilon})}[-\log p_{\boldsymbol{\theta}}(\mathbf{x}\,|\,\mathcal{G}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}; \mathbf{x}))] \\
&\quad = \frac{\tau}{2}\mathbb{E}_{q^*(\mathbf{x})p(\boldsymbol{\epsilon})}[\|\mathbf{x} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathcal{G}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}; \mathbf{x}))\|_2^2] + \frac{D}{2}\log\frac{2\pi}{\tau} \\
&\quad = \frac{\tau}{2}\mathbb{E}_{q^*(\mathbf{x})p(\boldsymbol{\epsilon})}[\|\mathbf{x} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{m}_{\boldsymbol{\phi}}(\mathbf{x}) + t^{-\frac{1}{2}}\boldsymbol{\epsilon})\|_2^2] + \text{const} \\
&\quad \to \frac{\tau}{2}\mathbb{E}_{q^*(\mathbf{x})}[\|\mathbf{x} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{m}_{\boldsymbol{\phi}}(\mathbf{x}))\|_2^2] + \text{const}, \quad \text{as } t \to \infty \\
&\quad = \frac{\tau}{2}\mathcal{L}^{\text{reverse}}_{\text{CONST}}(\boldsymbol{\theta}, \boldsymbol{\phi}) + \text{const}.
\end{aligned}$$

A similar analysis can be carried out for $\mathcal{L}_{\text{NELP}}(\boldsymbol{\theta}, \boldsymbol{\phi})$ and its deterministic counterpart $\ell^{\text{forward}}_{\text{CONST}}(\boldsymbol{\theta}, \boldsymbol{\phi})$. □

Hence, the cycle-consistency losses can be seen as special cases of the ELL and ELP with *degenerate* conditional distributions. Furthermore, this sheds new light on the roles of the cycle-consistency losses. For example, similar to the ELL, the reverse consistency loss encourages the conditional $q_{\boldsymbol{\phi}}(\mathbf{z}\,|\,\mathbf{x})$ to place its mass on configurations of latent variables that can explain, or in this case, *represent* the data well.

4.5.3 *Distribution Matching as Approximate Divergence Minimisation*

We now discuss how the adversarial objectives $\ell^{\text{reverse}}_{\text{GAN}}(\boldsymbol{\alpha}\,|\,\boldsymbol{\phi})$ and $\ell^{\text{forward}}_{\text{GAN}}(\boldsymbol{\beta}\,|\,\boldsymbol{\theta})$ relate to the KL variational lower bounds of our framework, $\mathcal{L}^{\text{latent}}_{\text{KL}}(\boldsymbol{\alpha}\,|\,\boldsymbol{\phi})$ and $\mathcal{L}^{\text{observed}}_{\text{KL}}(\boldsymbol{\beta}\,|\,\boldsymbol{\theta})$, respectively. To reduce clutter, we restrict our discussion to the reverse objective $\ell^{\text{reverse}}_{\text{GAN}}(\boldsymbol{\alpha}\,|\,\boldsymbol{\phi})$, as the same reasoning readily applies to the forward $\ell^{\text{forward}}_{\text{GAN}}(\boldsymbol{\beta}\,|\,\boldsymbol{\theta})$.

4.5.3.1 *As Density Ratio Estimation by Probabilistic Classification*

Firstly, the connections between GANs, divergence minimisation and DRE are well-established [177, 191, 251]. Although $\ell^{\text{reverse}}_{\text{GAN}}(\boldsymbol{\alpha}\,|\,\boldsymbol{\phi})$ is a scoring rule for probabilistic classification [85], one can readily show that it can also be subsumed as an instance of the generalised variational lower bound $\mathcal{L}^{\text{latent}}_{f}(\boldsymbol{\alpha}\,|\,\boldsymbol{\phi})$. Furthermore, similar to $\mathcal{L}^{\text{latent}}_{\text{KL}}(\boldsymbol{\alpha}\,|\,\boldsymbol{\phi})$, maximising $\ell^{\text{reverse}}_{\text{GAN}}(\boldsymbol{\alpha}\,|\,\boldsymbol{\phi})$ corresponds estimating the intractable density ratio $r^*(\mathbf{z}; \mathbf{x})$ of Equation (4.9).

Lemma 4.5.2. *By setting* $f_{\text{GAN}}(u) = u\log u - (u+1)\log(u+1)$ *in the generalised objective* $\mathcal{L}^{\text{latent}}_{f}(\boldsymbol{\alpha}\,|\,\boldsymbol{\phi})$ *of Equation* (4.10)*, we instantiate the objective*

$$\begin{aligned}
\mathcal{L}^{\text{reverse}}_{\text{GAN}}(\boldsymbol{\alpha}\,|\,\boldsymbol{\phi}) \triangleq\ & \mathbb{E}_{q^*(\mathbf{x})p^*(\mathbf{z})}[\log \mathcal{D}_{\boldsymbol{\alpha}}(\mathbf{z}; \mathbf{x})] \\
& + \mathbb{E}_{q^*(\mathbf{x})q_{\boldsymbol{\phi}}(\mathbf{z}\,|\,\mathbf{x})}[\log(1 - \mathcal{D}_{\boldsymbol{\alpha}}(\mathbf{z}; \mathbf{x}))],
\end{aligned} \tag{4.32}$$

where $\mathcal{D}_{\boldsymbol{\alpha}}(\mathbf{z}; \mathbf{x}) \triangleq 1 - \sigma(\log r_{\boldsymbol{\alpha}}(\mathbf{z}; \mathbf{x}))$*, and* σ *is the logistic sigmoid function.*

Proof. To instantiate $\mathcal{L}^{\text{reverse}}_{\text{GAN}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi})$ of Equation (4.32), it suffices to show that $-f^*_{\text{GAN}}(f'_{\text{GAN}}(r_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x}))) = \log \mathcal{D}_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x})$ and $f'_{\text{GAN}}(r_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x})) = \log(1 - \mathcal{D}_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x}))$, where $\mathcal{D}_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x}) \triangleq 1 - \sigma(\log r_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x}))$. First we compute the first derivative f'_{GAN} and the convex dual f^*_{GAN} of f_{GAN}, which involve straightforward calculations,

$$f'_{\text{GAN}}(u) = \log \sigma(\log u), \quad f^*_{\text{GAN}}(t) = -\log(1 - \exp t).$$

Thus, the composition $(f^*_{\text{GAN}} \circ f'_{\text{GAN}}) : u \mapsto f^*_{\text{GAN}}(f'_{\text{GAN}}(u))$ can be simplified as

$$f^*_{\text{GAN}}(f'_{\text{GAN}}(u)) = -\log(1 - \exp f'_{\text{GAN}}(u)) = -\log(1 - \sigma(\log u)).$$

Applying f'_{GAN} and $f^*_{\text{GAN}} \circ f'_{\text{GAN}}$ to $r_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x})$, we have

$$f'_{\text{GAN}}(r_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x})) = \log \sigma(\log r_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x})) = \log(1 - \mathcal{D}_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x})),$$

and

$$f^*_{\text{GAN}}(f'_{\text{GAN}}(r_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x}))) = -\log(1 - \sigma(\log r_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x}))) = -\log \mathcal{D}_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x}),$$

respectively, as required. □

Lemma 4.5.3. *By specifying a discriminator $\mathcal{D}_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x}) = \mathbf{D}_{\boldsymbol{\alpha}}(\mathbf{z})$ that ignores auxiliary input $\mathbf{x}$, and mapping $\mathcal{G}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon};\mathbf{x}) = \mathbf{m}_{\boldsymbol{\phi}}(\mathbf{x})$ that ignores noise input $\boldsymbol{\epsilon}$, $\mathcal{L}^{\text{reverse}}_{\text{GAN}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi})$ reduces to $\ell^{\text{reverse}}_{\text{GAN}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi})$.*

Proof. Through reparameterisation of $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$, we have

$$\begin{aligned}\mathcal{L}^{\text{reverse}}_{\text{GAN}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi}) = \mathbb{E}_{q^*(\mathbf{x})p^*(\mathbf{z})}[\log \mathcal{D}_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x})] \\ + \mathbb{E}_{q^*(\mathbf{x})p(\boldsymbol{\epsilon})}[\log(1 - \mathcal{D}_{\boldsymbol{\alpha}}(\mathcal{G}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon};\mathbf{x});\mathbf{x}))].\end{aligned}$$

By specifying a discriminator $\mathcal{D}_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x}) = \mathbf{D}_{\boldsymbol{\alpha}}(\mathbf{z})$ that ignores auxiliary input $\mathbf{x}$, and mapping $\mathcal{G}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon};\mathbf{x}) = \mathbf{m}_{\boldsymbol{\phi}}(\mathbf{x})$ that ignores noise input $\boldsymbol{\epsilon}$, this reduces to

$$\begin{aligned}\mathcal{L}^{\text{reverse}}_{\text{GAN}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi}) &= \mathbb{E}_{p^*(\mathbf{z})}[\log \mathbf{D}_{\boldsymbol{\alpha}}(\mathbf{z})] + \mathbb{E}_{q^*(\mathbf{x})}[\log(1 - \mathbf{D}_{\boldsymbol{\alpha}}(\mathbf{m}_{\boldsymbol{\phi}}(\mathbf{x})))] \\ &= \ell^{\text{reverse}}_{\text{GAN}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi}),\end{aligned}$$

as required. □

Proposition 4.5.4. *The reverse adversarial objective $\ell^{\text{reverse}}_{\text{GAN}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi})$ can be subsumed as an instance of the generalised variational lower bound $\mathcal{L}^{\text{latent}}_{f}(\boldsymbol{\alpha} \mid \boldsymbol{\phi})$.*

Theorem 4.5.4 follows directly from Theorems 4.5.2 and 4.5.3.

Now, by Theorem 4.3.2, objective $\mathcal{L}^{\text{reverse}}_{\text{GAN}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi})$ is maximised exactly when $r_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x}) = r^*(\mathbf{z};\mathbf{x})$. Hence, we can interpret $\mathcal{L}^{\text{reverse}}_{\text{GAN}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi})$ as an objective for density-ratio estimation based on probabilistic classification, while $\mathcal{L}^{\text{latent}}_{\text{KL}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi})$ is an objective based on KLIEP.

Now, the default choice of DM loss is $\mathcal{L}^{\text{reverse}}_{\text{GAN}_\text{A}}(\boldsymbol{\phi} \mid \boldsymbol{\alpha}) \triangleq \mathcal{L}^{\text{reverse}}_{\text{GAN}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi})$. Omitting terms not involving $\boldsymbol{\phi}$, this is given by

$$\mathcal{L}^{\text{reverse}}_{\text{GAN}_\text{A}}(\boldsymbol{\phi} \mid \boldsymbol{\alpha}) \triangleq \mathbb{E}_{q^*(\mathbf{x})q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})}[\log(1 - \mathcal{D}_{\boldsymbol{\alpha}}(\mathbf{z}; \mathbf{x}))]. \tag{4.33}$$

Unlike $\mathcal{L}^{\text{latent}}_{\text{KL}}(\boldsymbol{\phi} \mid \boldsymbol{\alpha})$, minimising $\mathcal{L}^{\text{reverse}}_{\text{GAN}_\text{A}}(\boldsymbol{\phi} \mid \boldsymbol{\alpha})$ does not minimise the KL divergence of Equation (4.8). Hence, the minimisation problem of Equation (4.31b) does not correspond to that of Equation (4.14b), and so does not maximise the ELBO, or any known VI objective.

4.5.3.2 *Recovering KL Through Alternative Divergence Minimisation Losses*

Although the default choice of DM loss does not yield a tight correspondence to VI, the existing CYCLEGAN frameworks – and indeed most GAN-based approaches – arbitrarily select an alternative DM loss that avoids vanishing gradients, and work well in practice. Hence, one need only choose an alternative that *does* correspond to minimising the KL divergence of Equation (4.8).

Firstly, of the CYCLEGAN methods, Kim et al. [125] adopt the widely-used DM loss originally suggested by Goodfellow et al. [86],

$$\mathcal{L}^{\text{reverse}}_{\text{GAN}_\text{B}}(\boldsymbol{\phi} \mid \boldsymbol{\alpha}) \triangleq \mathbb{E}_{q^*(\mathbf{x})q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})}[-\log \mathcal{D}_{\boldsymbol{\alpha}}(\mathbf{z}; \mathbf{x})], \tag{4.34}$$

while [302] optimise the Least-Squares GAN (LSGAN) objectives of [162].

Consider the *combination* of losses $\mathcal{L}^{\text{reverse}}_{\text{GAN}_\text{A}}(\boldsymbol{\phi} \mid \boldsymbol{\alpha})$ and $\mathcal{L}^{\text{reverse}}_{\text{GAN}_\text{B}}(\boldsymbol{\phi} \mid \boldsymbol{\alpha})$,

$$\begin{aligned}\mathcal{L}^{\text{reverse}}_{\text{GAN}_\text{C}}(\boldsymbol{\phi} \mid \boldsymbol{\alpha}) &\triangleq \mathcal{L}^{\text{reverse}}_{\text{GAN}_\text{A}}(\boldsymbol{\phi} \mid \boldsymbol{\alpha}) + \mathcal{L}^{\text{reverse}}_{\text{GAN}_\text{B}}(\boldsymbol{\phi} \mid \boldsymbol{\alpha}) \\ &= \mathbb{E}_{q^*(\mathbf{x})q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})}\left[-\log \frac{\mathcal{D}_{\boldsymbol{\alpha}}(\mathbf{z}; \mathbf{x})}{1 - \mathcal{D}_{\boldsymbol{\alpha}}(\mathbf{z}; \mathbf{x})}\right].\end{aligned} \tag{4.35}$$

Proposition 4.5.5. *We have* $\mathcal{L}^{\text{reverse}}_{\text{GAN}_\text{C}}(\boldsymbol{\phi} \mid \boldsymbol{\alpha}) = \mathcal{L}^{\text{latent}}_{\text{KL}}(\boldsymbol{\phi} \mid \boldsymbol{\alpha})$.

Theorem 4.5.5 was originally noted by Sønderby et al. [239] and is shown below.

Proof. Expanding out $\mathcal{L}^{\text{reverse}}_{\text{GAN}_\text{C}}(\boldsymbol{\phi} \mid \boldsymbol{\alpha})$, we have

$$\begin{aligned}\mathcal{L}^{\text{reverse}}_{\text{GAN}_\text{C}}(\boldsymbol{\phi} \mid \boldsymbol{\alpha}) &= \mathbb{E}_{q^*(\mathbf{x})q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})}\left[-\log \frac{\mathcal{D}_{\boldsymbol{\alpha}}(\mathbf{z}; \mathbf{x})}{1 - \mathcal{D}_{\boldsymbol{\alpha}}(\mathbf{z}; \mathbf{x})}\right] \\ &= \mathbb{E}_{q^*(\mathbf{x})q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})}\left[\log \frac{\sigma(\log r_{\boldsymbol{\alpha}}(\mathbf{z}; \mathbf{x}))}{1 - \sigma(\log r_{\boldsymbol{\alpha}}(\mathbf{z}; \mathbf{x}))}\right] \\ &= \mathbb{E}_{q^*(\mathbf{x})q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})}[\log r_{\boldsymbol{\alpha}}(\mathbf{z}; \mathbf{x})] \triangleq \mathcal{L}^{\text{latent}}_{\text{KL}}(\boldsymbol{\phi} \mid \boldsymbol{\alpha}).\end{aligned}$$

Hence, $\mathcal{L}^{\text{reverse}}_{\text{GAN}_\text{C}}(\boldsymbol{\phi} \mid \boldsymbol{\alpha}) = \mathcal{L}^{\text{latent}}_{\text{KL}}(\boldsymbol{\phi} \mid \boldsymbol{\alpha})$ as required. □

Thus, for the setting of the DM loss $\mathcal{L}^{\text{reverse}}_{\text{GAN}_\text{C}}(\boldsymbol{\phi} \mid \boldsymbol{\alpha})$, the minimisation problem of Equation (4.31b) corresponds to that of Equation (4.14b), and thus maximises the ELBO. This is equivalent to fitting the density ratio estimator $r_{\boldsymbol{\alpha}}(\mathbf{z}; \mathbf{x})$ by maximising the objective $\mathcal{L}^{\text{reverse}}_{\text{GAN}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi})$

instead of $\mathcal{L}_{\text{KL}}^{\text{latent}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi})$, and plugging it back into $\mathcal{L}_{\text{KL}}^{\text{latent}}(\boldsymbol{\phi} \mid \boldsymbol{\alpha})$ to approximately minimise the KL divergence of Equation (4.8). Such an approach is prevalent among existing implicit VI methods [110, 172, 202, 267]

SUMMARY OF THEORETICAL CONNECTIONS We have that the cycle-consistency losses are a specific instance of the ELL and ELP, while the adversarial objectives are a specific instance of the variational lower bound for divergence estimation, the maximisation of which can be seen as density ratio estimation by probabilistic classification. By explicitly setting the corresponding divergence minimisation loss such that it leads to minimisation of the required KL divergence terms in the ELBO and APLBO, we subsume the CYCLEGAN model under our proposed VI framework. See Section 4.B for a succinct summary of the relationships.

4.6 RELATED WORK

The work presented in this chapter is closely related to prior efforts to extend the scope of VI to implicit distributions. A recurring theme throughout this line of work is approximation of the ELBO by exploiting the formal connection between density-ratio estimation and GANS [177, 271]. The major axis of variation lies in the choice of the target density ratio being estimated, as dictated by the problem setting. Makhzani et al. [160] and Mescheder, Nowozin, and Geiger [172, AVB] estimate the density ratio $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) / p(\mathbf{z})$ in order to allow for expressive sample-based posterior approximations $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$. This corresponds to the reverse KL minimisation component of our approach, in which we accomodate implicit priors $p(\mathbf{z})$.

Similar to BIGAN [62] and adversarially learned inference (ALI) [59], Tran, Ranganath, and Blei [267, LFVI] match a variational joint to an exact joint distribution by estimating the density ratio $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) / q_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z})$ and use it to approximately minimise the KL divergence. Although this formulation sidesteps the requirement of having *any* tractable densities, their focus is on inference for models with intractable likelihoods $p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})$ and on incorporating the implicit posteriors *à la* AVB. In constrast, in our framework, the joint's intractability instead stems from the implicit prior $p^*(\mathbf{z})$. While we also approximate the exact joint, we do so by minimising a *symmetric* KL divergence. Furthermore, since both $p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})$ and $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$ are prescribed, we evaluate them explicitly as part of our loss functions and estimate a different set of density ratios. This closely resembles the approach of Pu et al. [202], which also minimises the symmetric KL divergence between the joints. However, the focus of their method is not on implicit distributions and thus specifies a different set of losses than ours – one that requires solving more complicated density ratio estimation problems. More

importantly, their method does not yield a tight correspondence to CYCLEGAN models.

A consequence of solely minimising the forward KL (as in frameworks like ALI) rather then minimising the symmetric KL (as in our framework) is *non-identifiability*. This issue has been addressed by Li et al. [144] who proposed a conditional-entropy regulariser to ALI's objective. Although Li et al. [144] examine the link between their method and CYCLEGAN, unlike our work, the relationship is not made explicit in a mathematically precise manner. Additionally, we derive our full objective from the perspective of approximate Bayesian inference. More recently, Chen et al. [31] also highlight issues associated with ALI concerning the quality of data generated from the inferred latent variables. They propose a symmetric VAE that simultaneously inherits the realistic image generation capabilities of adversarial approaches while overcoming the asymmetry limitations of the forward KL divergence inherent in standard VAEs. Additionally, unlike our approach, they do not provide an explicit relationship with CYCLEGANS.

Finally, similar to InfoGAN [35] and VEEGAN [245], the forward KL minimisation component of our method also optimises a model of the latent variables, which is reminiscent of the wake-sleep algorithm for training Helmholtz machines [52]. This is discussed further by Hu et al. [109], who provide a comprehensive treatment of the links between the work mentioned in this section and importantly, the symmetric perspective of *generation* and *recognition* that underpins our approach.

4.7 EXPERIMENTS

SYNTHETIC DATA. First we consider the problem of reducing the dimensionality of the MNIST dataset to a 2D latent space, wherein the prior distribution on the latent representations is specified by its samples (shown in Figure 4.2a). This "banana-shaped distribution" is a commonly used testbed for adaptive MCMC methods [93, 264]. Its samples can be generated by drawing from a bivariate Gaussian with unit variances and correlation $\rho = 0.95$, and transforming them through mapping $H(z_1, z_2) \triangleq [z_1, z_2 - z_1^2 - 1]^T$. While the density of this distribution can be computed, it is withheld from our algorithm and used only in the VAE baseline, which does not permit implicit distributions.

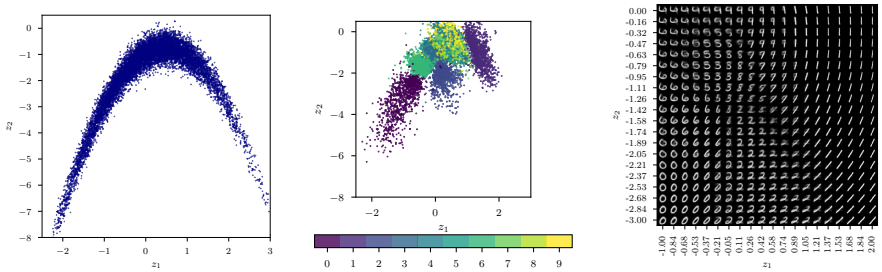
Qualitative and quantitative results are given Figure 4.2 and Table 4.1, which demonstrate superior performance to competing approaches. Observe that instances of the various digit classes are disentangled in this latent space, while still closely matching the shape of the prior distribution, despite having only access to its samples. The resulting manifold of reconstructions is depicted in Figure 4.2c.

In Table 4.1, we report the mean-squared error (MSE) on the reconstructions of observations from the held-out test set and benchmark

Table 4.1: Mean-squared errors of reconstructions.

METHOD	MSE z	MSE x
SJMVI (OURS)	**0.17**	**0.04**
VAE [127]	0.88	**0.04**
AVB [172]	0.29	**0.04**

against VAE / AVB. Also, for the joint approximation to properly match the support of the exact joint, the latent codes should also be representable by its corresponding observation. Hence, we also report the MSE between samples from the prior and their reconstructions. While we find no improvements on reconstruction quality of observations, our method significantly outperforms others in reconstructing latent codes, suggesting our method has greater capacity to faithfully approximate the exact joint.



(a) 10k samples from implicit prior $p^*(\mathbf{z})$.

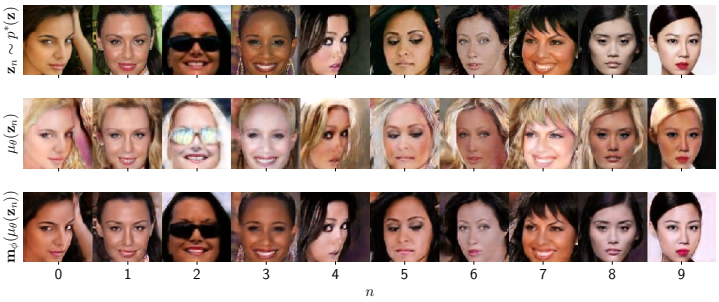
(b) Mean of $q_{\phi}(\mathbf{z} \mid \mathbf{x})$ for every $\mathbf{x}$ from held-out test set of size 10k, colored by digit class.

(c) Mean of $p_{\theta}(\mathbf{x} \mid \mathbf{z})$ for 20×20 values of $\mathbf{z}$ along a uniform grid.

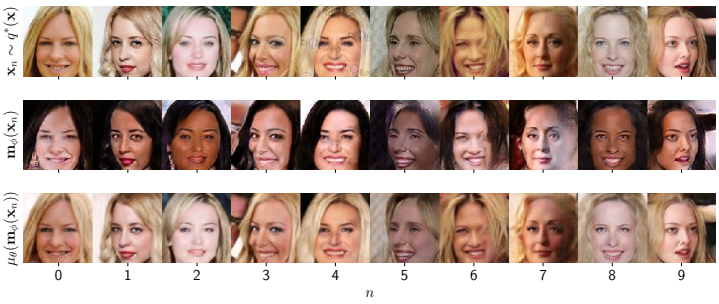
Figure 4.2: Visualisation of 2D latent space and the corresponding observed space manifold.Instances of the various digit classes are disentangled in this latent space, while still closely matching the shape of the prior distribution, despite having only access to its samples .

IMAGE-TO-IMAGE TRANSLATION. We apply our method to the task of transferring features between images of faces on the CelebA dataset [152]. We consider the case where one feature differs between domains. In particular, distributions $q^*(\mathbf{x})$ and $q^*(\mathbf{z})$ are specified by images of women with blond and black hair, respectively. We specify both $p_{\theta}(\mathbf{x} \mid \mathbf{z})$ and $q_{\phi}(\mathbf{z} \mid \mathbf{x})$ as a Laplace distribution, with fixed variance, and mean functions $\boldsymbol{\mu}_{\theta}(\mathbf{z})$ and $\mathbf{m}_{\phi}(\mathbf{x})$ defined by neural networks. Their architectures, as well as those of discriminators $\mathcal{D}_{\alpha}(\mathbf{z}; \mathbf{x})$ and $\mathcal{D}_{\beta}(\mathbf{x}; \mathbf{z})$ are defined in the same way as in [302]. In Figure 4.3, we show the outputs of the mean functions on samples from a hold-out test set, after training for 10 epochs. From Figure 4.3b, we see that

given datapoints $\mathbf{x}$ (top) we're able to learn a posterior over latent representations $\mathbf{z}$ in the other domain (mean is shown in middle). Furthermore, these latent representations are configured so as to maximise the likelihood of observing the original data, as evident from the reconstructions (bottom). Refer to appendix Figure 4.4 for qualitative results produced by the CYCLEGAN baseline approach [125, 302].



(a) Samples from the prior (top), mean of the likelihood (middle), and the mean reconstruction (bottom).

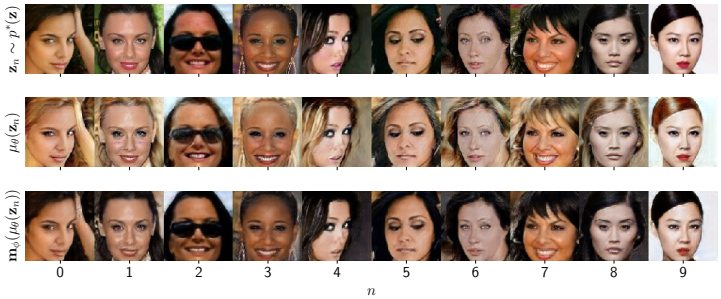


(b) Samples from the data (top), mean of the posterior (middle), and the mean reconstruction (bottom).

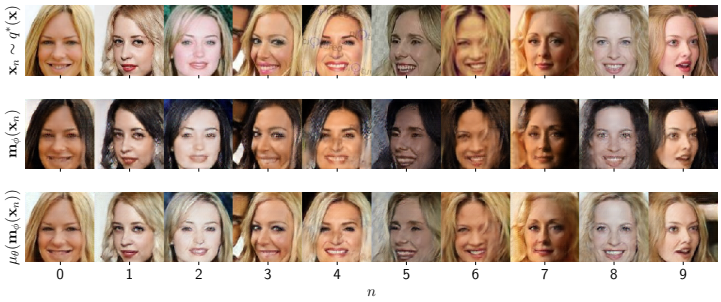
Figure 4.3: Image-to-image translation (blond to black hair) on CelebA dataset [152] performed by our proposed approach.

4.8 SUMMARY

In this chapter, we've provided a theoretical treatment of the link between CYCLEGANS and approximate Bayesian inference. In short, samples from the two domains correspond respectively to those drawn from the data and implicit prior distribution in a implicit LVM (ILVM). Parameter learning in CYCLEGANS corresponds to approximate inference in this ILVM under our proposed VI framework. The forward and reverse mappings in CYCLEGANS arise naturally in the generative and recognition models, while the cycle-consistency constraints correspond to their log probabilities, and the adversarial losses are approximations to an f-divergence. By lifting the limitations of prescribed prior distributions in favour of arbitrarily flexible implicit distributions, we



(a) Samples from the prior (top), output of mapping (middle), and the reconstruction (bottom).



(b) Samples from the data (top), output of mapping (middle), and the reconstruction (bottom).

Figure 4.4: Image-to-image translation (blond to black hair) on CelebA dataset [152] performed by the baseline CYCLEGAN method.

can discover different perspectives on existing learning methods and provide more flexible approaches to probabilistic modelling.

4.A RELATION TO KL IMPORTANCE ESTIMATION PROCEDURE (KLIEP)

We now discuss the connections to KLIEP [250]. Consider the same problem setting as in Section 4.3.3 where we wish to use a parameterised function r_{α} to estimate the exact density ratio,

$$r_{\alpha}(\mathbf{z};\mathbf{x}) \approx r^*(\mathbf{z};\mathbf{x}) \triangleq \frac{q_{\phi}(\mathbf{z}\,|\,\mathbf{x})}{p^*(\mathbf{z})}.$$

We can view $r_{\alpha}(\mathbf{z};\mathbf{x})$ as the correction factor required for $p^*(\mathbf{z})$ to match $q_{\phi}(\mathbf{z}\,|\,\mathbf{x})$. This gives rise to an estimator of $q_{\phi}(\mathbf{z}\,|\,\mathbf{x})$,

$$q_{\alpha}(\mathbf{z}\,|\,\mathbf{x}) \triangleq r_{\alpha}(\mathbf{z};\mathbf{x})p^*(\mathbf{z}) \approx q_{\phi}(\mathbf{z}\,|\,\mathbf{x}).$$

Although in our specific problem setting, the density $q_{\phi}(\mathbf{z}\,|\,\mathbf{x})$ is tractable, we nonetheless fit an auxiliary model $q_{\alpha}(\mathbf{z}\,|\,\mathbf{x})$ to it as a means of fitting the underlying density ratio estimator $r_{\alpha}(\mathbf{z};\mathbf{x})$.

In particular, consider *minimising* the KL divergence between $q_{\phi}(\mathbf{z}\,|\,\mathbf{x})$ and $q_{\alpha}(\mathbf{z}\,|\,\mathbf{x})$ with respect to α,

$$\begin{aligned}
&\mathbb{E}_{q^*(\mathbf{x})}\mathrm{KL}\left[q_{\phi}(\mathbf{z}\,|\,\mathbf{x}) \parallel q_{\alpha}(\mathbf{z}\,|\,\mathbf{x})\right] \\
&\quad\triangleq \mathbb{E}_{q^*(\mathbf{x})}\mathbb{E}_{q_{\phi}(\mathbf{z}\,|\,\mathbf{x})}\left[\log \frac{q_{\phi}(\mathbf{z}\,|\,\mathbf{x})}{q_{\alpha}(\mathbf{z}\,|\,\mathbf{x})}\right], \\
&\quad= \mathbb{E}_{q^*(\mathbf{x})}\mathbb{E}_{q_{\phi}(\mathbf{z}\,|\,\mathbf{x})}\left[\log \frac{q_{\phi}(\mathbf{z}\,|\,\mathbf{x})}{p^*(\mathbf{z})r_{\alpha}(\mathbf{z};\mathbf{x})}\right], \\
&\quad= -\mathbb{E}_{q^*(\mathbf{x})}\mathbb{E}_{q_{\phi}(\mathbf{z}\,|\,\mathbf{x})}[\log r_{\alpha}(\mathbf{z};\mathbf{x})] + \text{const.}
\end{aligned}$$

Hence, this is equivalent to *maximising*

$$\mathbb{E}_{q^*(\mathbf{x})}\mathbb{E}_{q_{\phi}(\mathbf{z}\,|\,\mathbf{x})}[\log r_{\alpha}(\mathbf{z};\mathbf{x})].$$

Now, for the conditional $q_{\alpha}(\mathbf{z}\,|\,\mathbf{x})$ to be a probability density function, its integral must sum to one,

$$\int q_{\alpha}(\mathbf{z}\,|\,\mathbf{x})q^*(\mathbf{x})d\mathbf{x}d\mathbf{z} = 1.$$

Rewriting this integral, we have the constraint

$$\begin{aligned}
\int q_{\alpha}(\mathbf{z}\,|\,\mathbf{x})q^*(\mathbf{x})d\mathbf{x}d\mathbf{z} &= \int r_{\alpha}(\mathbf{z};\mathbf{x})p^*(\mathbf{z})q^*(\mathbf{x})d\mathbf{x}d\mathbf{z} \\
&= \mathbb{E}_{q^*(\mathbf{x})p^*(\mathbf{z})}[r_{\alpha}(\mathbf{z};\mathbf{x})] = 1.
\end{aligned}$$

Table 4.B.1: Relevant latent and observed space f-divergences instantiated for particular settings of f.

		Reverse KL	GAN
	$f(u)$	$u \log u$	$u \log u - (u+1)\log(u+1)$
Latent	$\mathbb{E}_{q^*(\mathbf{x})} \mathcal{D}_f \left[p^*(\mathbf{z}) \parallel q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) \right]$	$\mathbb{E}_{q^*(\mathbf{x})} \mathrm{KL} \left[q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) \parallel p^*(\mathbf{z}) \right]$	$2 \cdot \mathbb{E}_{q^*(\mathbf{x})} \mathcal{D}_{\mathrm{JS}} \left[p^*(\mathbf{z}) \parallel q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) \right] - \log 4$
Observed	$\mathbb{E}_{p^*(\mathbf{z})} \mathcal{D}_f \left[q^*(\mathbf{x}) \parallel p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}) \right]$	$\mathbb{E}_{p^*(\mathbf{z})} \mathrm{KL} \left[p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}) \parallel q^*(\mathbf{x}) \right]$	$2 \cdot \mathbb{E}_{p^*(\mathbf{z})} \mathcal{D}_{\mathrm{JS}} \left[q^*(\mathbf{x}) \parallel p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}) \right] - \log 4$

Combined, we have the following constrained optimisation problem,

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \mathbb{E}_{q^*(\mathbf{x})} \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})} [\log r_{\boldsymbol{\alpha}}(\mathbf{z}; \mathbf{x})] \\ \text{subject to} \quad & \mathbb{E}_{q^*(\mathbf{x}) p^*(\mathbf{z})} [r_{\boldsymbol{\alpha}}(\mathbf{z}; \mathbf{x}) - 1] = 0. \end{aligned}$$

Through the method of Lagrange multipliers, this can be cast as an unconstrained optimisation problem with objective,

$$\begin{aligned} \mathcal{L}^{\text{latent}}_{\text{KLIEP}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi}) \triangleq \mathbb{E}_{q^*(\mathbf{x})} \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})} [\log r_{\boldsymbol{\alpha}}(\mathbf{z}; \mathbf{x})] \\ - \lambda \mathbb{E}_{q^*(\mathbf{x}) p^*(\mathbf{z})} [r_{\boldsymbol{\alpha}}(\mathbf{z}; \mathbf{x}) - 1], \end{aligned}$$

where λ is the Lagrange multiplier. For $\lambda = 1$, $\mathcal{L}^{\text{latent}}_{\text{KLIEP}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi})$ trivially reduces to $\mathcal{L}^{\text{latent}}_{\text{KL}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi})$.

4.B SUMMARY OF DEFINITIONS

In this section, we summarise the definitions of the losses defined in the proposed VI framework of Sections 4.3 and 4.4, and underscore the relationships to their respective counterparts in the CYCLEGAN framework of Section 4.5.

Table 4.B.1 summarises the settings of convex function $f : \mathbb{R}_+ \to \mathbb{R}$ that recover the reverse KL divergence terms within the ELBO and APLBO, and the JS divergence (up to constants) that GANs are known to minimise.

Table 4.B.2 gives the calculations of the terms necessary to explicitly write down instances of the generalised variational lower bound for particular convex functions f – namely the convex dual $f^\star$, the first derivative f' and the composition $f^\star \circ f'$.

Table 4.B.3 gives instances of the variational lower bound that approximate the latent and observed space KL divergences within the ELBO and APLBO, respectively. Additionally, it gives generalised *stochastic* formulations of the GAN objectives in the CYCLEGAN framework, while Table 4.B.4 lists their *deterministic* counterpart.

Lastly, Table 4.B.5 gives forward and reverse cycle-consistency constraints in the CYCLEGAN framework, and the specific class of Gaussian

Table 4.B.2: Calculations for convex functions.

	Reverse KL	GAN
$f(u)$	$u \log u$	$u \log u - (u+1)\log(u+1)$
$f^\star(t)$	$\exp(t-1)$	$-\log(1-\exp t)$
$f'(u)$	$1+\log u$	$\log \sigma(\log u)$
$f^\star(f'(u))$	u	$-\log(1-\sigma(\log u))$

Table 4.B.3: Instances of variational lower bounds on the relevant latent and observed space f-divergences.

		Reverse KL	GAN
	$f(u)$	$u \log u$	$u \log u - (u+1)\log(u+1)$
Latent	$\mathcal{L}_f^{\text{latent}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi}) \triangleq \mathbb{E}_{q^*(\mathbf{x})q_{\boldsymbol{\phi}}(\mathbf{z}\mid\mathbf{x})}[f'(r_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x}))] - \mathbb{E}_{q^*(\mathbf{x})p^*(\mathbf{z})}[f^\star(f'(r_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x})))]$	$\mathcal{L}_{\text{KL}}^{\text{latent}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi}) \triangleq \mathbb{E}_{q^*(\mathbf{x})q_{\boldsymbol{\phi}}(\mathbf{z}\mid\mathbf{x})}[\log r_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x})] - \mathbb{E}_{q^*(\mathbf{x})p^*(\mathbf{z})}[r_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x}) - 1]$	$\mathcal{L}_{\text{GAN}}^{\text{reverse}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi}) \triangleq \mathbb{E}_{q^*(\mathbf{x})q_{\boldsymbol{\phi}}(\mathbf{z}\mid\mathbf{x})}[\log \sigma(\log r_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x}))] + \mathbb{E}_{q^*(\mathbf{x})p^*(\mathbf{z})}[\log(1-\sigma(\log r_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x})))]$
Observed	$\mathcal{L}_f^{\text{observed}}(\boldsymbol{\beta} \mid \boldsymbol{\theta}) \triangleq \mathbb{E}_{p^*(\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{x}\mid\mathbf{z})}[f'(r_{\boldsymbol{\beta}}(\mathbf{x};\mathbf{z}))] - \mathbb{E}_{p^*(\mathbf{z})q^*(\mathbf{x})}[f^\star(f'(r_{\boldsymbol{\beta}}(\mathbf{x};\mathbf{z})))]$	$\mathcal{L}_{\text{KL}}^{\text{observed}}(\boldsymbol{\beta} \mid \boldsymbol{\theta}) \triangleq \mathbb{E}_{p^*(\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{x}\mid\mathbf{z})}[\log r_{\boldsymbol{\beta}}(\mathbf{x};\mathbf{z})] - \mathbb{E}_{p^*(\mathbf{z})q^*(\mathbf{x})}[r_{\boldsymbol{\beta}}(\mathbf{x};\mathbf{z}) - 1]$	$\mathcal{L}_{\text{GAN}}^{\text{forward}}(\boldsymbol{\beta} \mid \boldsymbol{\theta}) \triangleq \mathbb{E}_{p^*(\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{x}\mid\mathbf{z})}[\log \sigma(\log r_{\boldsymbol{\beta}}(\mathbf{x};\mathbf{z}))] + \mathbb{E}_{p^*(\mathbf{z})q^*(\mathbf{x})}[\log(1-\sigma(\log r_{\boldsymbol{\beta}}(\mathbf{x};\mathbf{z})))]$

likelihoods and posteriors that instantiates these constraints (in the limit).

Table 4.B.4: General stochastic GAN objectives and their deterministic counterparts.

	Stochastic	Deterministic
REVERSE	$\mathcal{L}_{\text{GAN}}^{\text{reverse}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi}) \triangleq \mathbb{E}_{q^*(\mathbf{x})p^*(\mathbf{z})}[\log \mathcal{D}_{\boldsymbol{\alpha}}(\mathbf{z};\mathbf{x})] + \mathbb{E}_{q^*(\mathbf{x})p(\boldsymbol{\epsilon})}[\log(1-\mathcal{D}_{\boldsymbol{\alpha}}(\mathcal{G}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon};\mathbf{x});\mathbf{x}))]$	$\ell_{\text{GAN}}^{\text{reverse}}(\boldsymbol{\alpha} \mid \boldsymbol{\phi}) \triangleq \mathbb{E}_{p^*(\mathbf{z})}[\log \mathbf{D}_{\boldsymbol{\alpha}}(\mathbf{z})] + \mathbb{E}_{q^*(\mathbf{x})}[\log(1-\mathbf{D}_{\boldsymbol{\alpha}}(\mathbf{m}_{\boldsymbol{\phi}}(\mathbf{x})))]$
FORWARD	$\mathcal{L}_{\text{GAN}}^{\text{forward}}(\boldsymbol{\beta} \mid \boldsymbol{\theta}) \triangleq \mathbb{E}_{p^*(\mathbf{z})q^*(\mathbf{x})}[\log \mathcal{D}_{\boldsymbol{\beta}}(\mathbf{x};\mathbf{z})] + \mathbb{E}_{p^*(\mathbf{z})p(\boldsymbol{\xi})}[\log(1-\mathcal{D}_{\boldsymbol{\beta}}(\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{\xi};\mathbf{z});\mathbf{z}))]$	$\ell_{\text{GAN}}^{\text{forward}}(\boldsymbol{\beta} \mid \boldsymbol{\theta}) \triangleq \mathbb{E}_{q^*(\mathbf{x})}[\log \mathbf{D}_{\boldsymbol{\beta}}(\mathbf{x})] + \mathbb{E}_{p^*(\mathbf{z})}[\log(1-\mathbf{D}_{\boldsymbol{\beta}}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z})))]$

Table 4.B.5: Negative expected log conditionals and the cycle-consistency constraints.

Gaussian		Degenerate	
$p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})$	$q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$	$p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})$	$q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$
$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}), \tau^{-1}\mathbf{I})$	$\mathcal{N}(\mathbf{z} \mid \mathbf{m}_{\boldsymbol{\phi}}(\mathbf{x}), t^{-1}\mathbf{I})$	$\delta(\mathbf{x}-\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}))$	$\delta(\mathbf{z}-\mathbf{m}_{\boldsymbol{\phi}}(\mathbf{x}))$
$\mathcal{L}_{\text{NELL}}(\boldsymbol{\theta},\boldsymbol{\phi}) \triangleq \frac{\tau}{2}\mathbb{E}_{q^*(\mathbf{x})p(\boldsymbol{\epsilon})}[\lVert\mathbf{x}-\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{m}_{\boldsymbol{\phi}}(\mathbf{x})+t^{-\frac{1}{2}}\boldsymbol{\epsilon})\rVert_2^2] + \frac{D}{2}\log\frac{2\pi}{\tau}$		$\ell_{\text{CONST}}^{\text{reverse}}(\boldsymbol{\theta},\boldsymbol{\phi}) \triangleq \mathbb{E}_{q^*(\mathbf{x})}[\lVert\mathbf{x}-\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{m}_{\boldsymbol{\phi}}(\mathbf{x}))\rVert_2^2]$	
$\mathcal{L}_{\text{NELP}}(\boldsymbol{\theta},\boldsymbol{\phi}) \triangleq \frac{t}{2}\mathbb{E}_{p^*(\mathbf{z})p(\boldsymbol{\xi})}[\lVert\mathbf{z}-\mathbf{m}_{\boldsymbol{\phi}}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z})+\tau^{-\frac{1}{2}}\boldsymbol{\xi})\rVert_2^2] + \frac{K}{2}\log\frac{2\pi}{t}$		$\ell_{\text{CONST}}^{\text{forward}}(\boldsymbol{\theta},\boldsymbol{\phi}) \triangleq \mathbb{E}_{p^*(\mathbf{z})}[\lVert\mathbf{z}-\mathbf{m}_{\boldsymbol{\phi}}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}))\rVert_2^2]$	

5

BAYESIAN OPTIMISATION BY CLASSIFICATION WITH DEEP LEARNING AND BEYOND

PREFACE

This chapter is derived from work previously published as:

Louis C. Tiao, Vincent Dutordoir, and Victor Picheny. "Spherical Inducing Features for Orthogonally-Decoupled Gaussian Processes". In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause et al. Vol. 202. Proceedings of Machine Learning Research. PMLR, July 2023, pp. 34143–34160. URL: https://proceedings.mlr.press/v202/tiao23a.html. **(Accepted as *Oral* presentation).**

Additional context, analysis, and discussion has been included to address the intricacies of how density-ratios relate to the probability of improvement (PI) and expected improvement (EI), and to examine the extension to this work known as likelihood-free BO (LFBO) [241].

5.1 INTRODUCTION

We introduced Bayesian optimisation (BO) in Section 2.5 as a highly effective approach for the global optimisation of expensive blackbox functions [20, 228]. In particular, we saw how BO proposes candidate solutions according to an *acquisition function* that encodes a degree of balance between exploration and exploitation. At the heart of BO lies a probabilistic surrogate model from which the acquisition function is derived.

Among the many acquisition functions that have been devised, the *improvement-based* ones, such as the PI and EI [117, 176] have remained prevalent due in large to their effectiveness despite their relative simplicity. Notably, while acquisition functions are generally challenging to compute, let alone optimise [290], PI/EI offers a closed-form expression when the posterior predictive density of the model follows a Gaussian distribution. However, while this condition makes these acquisition functions easier to work with, it can also preclude the use of richer families of models, as one must ensure analytical tractability of the predictive, often at the expense of expressiveness, or otherwise by resorting to sampling-based approximations [7].

By virtue of its flexibility, desirable conjugacy properties, and ability to produce well-calibrated predictive uncertainty, GP regression [286] is a widely-used probabilistic model in BO. To extend GP-based BO to problems with discrete variables [77], structures with conditional

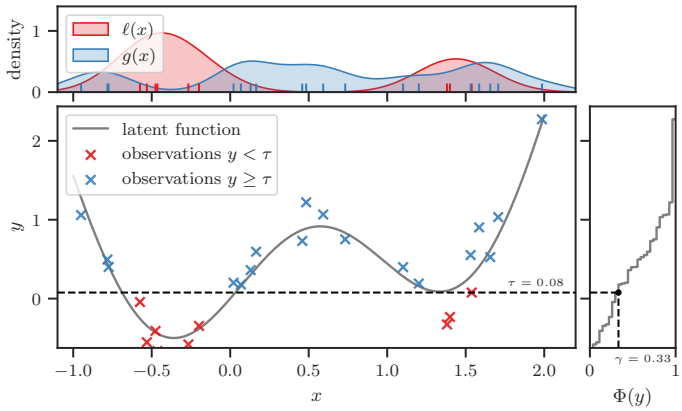


Figure 5.1: Optimising a synthetic function $f(x) = \sin(3x) + x^2 - 0.7x$ with observation noise $\varepsilon \sim \mathcal{N}(0, 0.2^2)$. In the main pane, the noise-free function is represented by the solid gray curve, and $N = 27$ noisy observations are represented by the crosses '$\times$'. Observations with output y in the top-performing $\gamma = 1/3$ proportion are shown in *red*; otherwise, they are shown in *blue*. Their corresponding densities, $\ell(x)$ and $g(x)$, respectively, are shown in the top pane. Bayesian optimisation by density-ratio estimation (BORE) exploits the correspondence between the PI acquisition function and the *ratio* of densities $\ell(x)/g(x)$.

dependencies [116], or to capture nonstationary phenomenon [238], it is common to apply simple modifications to the covariance function, as this can often be done without compromising the tractability of the predictive. Suffice it to say, certain estimators, such as decision trees in the case of discrete variables, are naturally better equipped to deal with these scenarios. Indeed, to scale BO to problem settings that produce vast numbers of observations, such as in transfer learning [255], existing approaches have resorted to alternative model families like random forests (RFS) [111] and BNNS [200, 237, 243]. However, these are often bound by constraints and simplifying assumptions, or must rely on Monte Carlo (MC) methods that make the acquisition function more cumbersome to evaluate and optimise.

Recognising that the surrogate model primarily serves as a means to construct the acquisition function, we shift the usual focus away from the model and toward the acquisition function itself. To this end, we seek an alternative formulation of the acquisition function, specifically, one that potentially opens the door to more powerful estimators for which the predictive density would otherwise be unwieldy or simply intractable to compute. In particular, Bergstra et al. [14] demonstrate that the PI function[1] can be expressed as the *relative* ratio between two densities [292]. To estimate this ratio, they propose a method known as

1 In fact, they make the stronger claim that this holds true for EI, but this assertion could be considered the outcome of spurious mathematical reasoning [75] – we elaborate on these intricacies in Section 5.2.2.2.

the TPE, which naturally handles discrete and tree-structured inputs, and scales linearly with the number of observations. However, in spite of its many advantages, TPE is not without deficiencies.

In the work summarised in this chapter, we make the following contributions: (i) We revisit the TPE approach from first principles and identify its shortcomings in tackling the general DRE problem (Section 5.2). (ii) We propose a simple yet powerful alternative that casts the computation of PI as probabilistic classification (Section 5.3). This approach is built on the aforementioned link between PI and the relative density-ratio, and the correspondence between DRE and CPE. As such, it retains the strengths of the TPE method while mitigating many of its weaknesses. Perhaps most significantly, it enables one to leverage virtually any state-of-the-art classification method available. In Section 5.5, we demonstrate through extensive experiments that our approach competes well with these methods on a diverse range of problems.

5.2 OPTIMISATION POLICIES AND DENSITY-RATIO ESTIMATION

5.2.1 *Relative Density-Ratio*

We introduced the *ordinary* density-ratio earlier in Section 2.3. Now let us generalise this to what is commonly known as the *relative* density-ratio [292]. Namely, for a given pair of densities $\ell(\mathbf{x})$ and $g(\mathbf{x})$, their *γ-relative density-ratio* is defined as

$$r_\gamma(\mathbf{x}) \triangleq \frac{\ell(\mathbf{x})}{\gamma\ell(\mathbf{x}) + (1-\gamma)g(\mathbf{x})}, \tag{5.1}$$

where $\gamma\ell(\mathbf{x}) + (1-\gamma)g(\mathbf{x})$ denotes the *γ-mixture density* with mixing proportion $0 \leq \gamma < 1$. Note that for $\gamma = 0$, we recover the ordinary density-ratio, which we denote $r_0(\mathbf{x}) \triangleq \ell(\mathbf{x})/g(\mathbf{x})$. Further, observe that the relative ratio is related to the ordinary ratio, $r_\gamma(\mathbf{x}) = h_\gamma(r_0(\mathbf{x}))$, where

$$h_\gamma(u) \triangleq \Big(\gamma + u^{-1}(1-\gamma)\Big)^{-1}$$

for $u > 0$.

5.2.2 *Improvement-based Acquisition Functions*

We now discuss how the improvement-based acquisition functions introduced in Section 2.5.2 relate to the ratio in Equation (5.1). First, let the threshold τ be the γ-th quantile of the observed y values, $\tau \triangleq \Phi^{-1}(\gamma)$ where $\gamma = \Phi(\tau) \triangleq p(y \leq \tau; \mathcal{D}_N)$. Thereafter, let the pair of densities be defined as $\ell(\mathbf{x}) \triangleq p(\mathbf{x} \,|\, y \leq \tau; \mathcal{D}_N)$ and $g(\mathbf{x}) \triangleq p(\mathbf{x} \,|\, y > \tau; \mathcal{D}_N)$. An illustrated example of this is shown in Figure 5.1.

5.2.2.1 *Probability of Improvement as a Density-Ratio*

Recall from Section 2.5.2.1 that the PI criterion can be expressed as $\alpha_{\text{PI}}(\mathbf{x}; \mathcal{D}_N, \tau) = p(y \leq \tau \,|\, \mathbf{x}, \mathcal{D}_N)$. By Bayes' rule, we have

$$\alpha_{\text{PI}}(\mathbf{x}; \mathcal{D}_N, \tau) = \frac{p(\mathbf{x} \,|\, y \leq \tau; \mathcal{D}_N) p(y \leq \tau \,|\, \mathcal{D}_N)}{p(\mathbf{x} \,|\, \mathcal{D}_N)}.$$

By definition, the numerator is simply

$$p(\mathbf{x} \,|\, y \leq \tau; \mathcal{D}_N) p(y \leq \tau \,|\, \mathcal{D}_N) = \gamma \cdot \ell(\mathbf{x}),$$

while, similarly, the denominator is

$$\begin{aligned} p(\mathbf{x} \,|\, \mathcal{D}_N) &= \int_{-\infty}^{\infty} p(\mathbf{x} \,|\, y, \mathcal{D}_N) p(y \,|\, \mathcal{D}_N) \, \mathrm{d}y \\ &= \ell(\mathbf{x}) \int_{-\infty}^{\tau} p(y \,|\, \mathcal{D}_N) \, \mathrm{d}y + g(\mathbf{x}) \int_{\tau}^{\infty} p(y \,|\, \mathcal{D}_N) \, \mathrm{d}y \\ &= \gamma \ell(\mathbf{x}) + (1 - \gamma) g(\mathbf{x}). \end{aligned} \tag{5.2}$$

Hence, the PI function can be expressed as the relative density-ratio, up to a constant factor γ,

$$\alpha_{\text{PI}}\left(\mathbf{x}; \mathcal{D}_N, \Phi^{-1}(\gamma)\right) \propto r_\gamma(\mathbf{x}). \tag{5.3}$$

Crucially, this reduces the problem of maximising PI to that of maximising the relative density-ratio,

$$\mathbf{x}_{N+1} = \arg\max_{\mathbf{x} \in \mathcal{X}} \alpha_{\text{PI}}\left(\mathbf{x}; \mathcal{D}_N, \Phi^{-1}(\gamma)\right) = \arg\max_{\mathbf{x} \in \mathcal{X}} r_\gamma(\mathbf{x}). \tag{5.4}$$

To estimate the unknown relative density-ratio, one can appeal to a wide variety of approaches from the DRE literature [251]. We broadly refer to this strategy as Bayesian optimisation by density-ratio estimation (BORE).

5.2.2.2 *Expected Improvement as a Density-Ratio?*

Bergstra et al. [14] assert that, under certain additional assumptions, the EI function can similarly be expressed as the relative density-ratio up to some constant factor. It goes without saying that this directly contradicts the results we have just presented, since clearly PI and EI are by definition not equivalent.

This particular issue has sparked recent discussions, and we analyse the arguments here. We proceed by reproducing the original derivations of Bergstra et al. [14]. Recall from Equation (2.45) that the EI function is defined as the expectation of the improvement utility

function $U_{\text{EI}}(y, \tau)$ over the posterior predictive density $p(y \mid \mathbf{x}, \mathcal{D}_N)$. Expanding this out, we have

$$\begin{aligned}
\alpha_{\text{EI}}(\mathbf{x}; \mathcal{D}_N, \tau) &\triangleq \mathbb{E}_{p(y \mid \mathbf{x}, \mathcal{D}_N)}[U_{\text{EI}}(y, \tau)] \\
&= \int_{-\infty}^{\infty} U_{\text{EI}}(y, \tau) p(y \mid \mathbf{x}, \mathcal{D}_N) \, \mathrm{d}y \\
&= \int_{-\infty}^{\tau} (\tau - y) p(y \mid \mathbf{x}, \mathcal{D}_N) \, \mathrm{d}y \\
&= \frac{1}{p(\mathbf{x} \mid \mathcal{D}_N)} \int_{-\infty}^{\tau} (\tau - y) p(\mathbf{x} \mid y, \mathcal{D}_N) p(y \mid \mathcal{D}_N) \, \mathrm{d}y.
\end{aligned}$$

We've already simplified the denominator $p(\mathbf{x} \mid \mathcal{D}_N)$ in Equation (5.2), and the numerator simplifies to

$$\begin{aligned}
&\int_{-\infty}^{\tau} (\tau - y) p(\mathbf{x} \mid y, \mathcal{D}_N) p(y \mid \mathcal{D}_N) \, \mathrm{d}y \\
&\quad \approx \ell(\mathbf{x}) \int_{-\infty}^{\tau} (\tau - y) p(y \mid \mathcal{D}_N) \, \mathrm{d}y \\
&\quad = \ell(\mathbf{x}) \left(\tau \int_{-\infty}^{\tau} p(y \mid \mathcal{D}_N) \, \mathrm{d}y - \int_{-\infty}^{\tau} y p(y \mid \mathcal{D}_N) \, \mathrm{d}y \right) \\
&\quad = K \cdot \ell(\mathbf{x}),
\end{aligned} \tag{5.5}$$

where

$$K \triangleq \gamma\tau - \int_{-\infty}^{\tau} y p(y \mid \mathcal{D}_N) \, \mathrm{d}y.$$

In contrast with the original derivation, there is not a strict equality in Equation (5.5) because, in general, $p(\mathbf{x} \mid y, \mathcal{D}_N) \neq p(\mathbf{x} | y \leq \tau; \mathcal{D}_N) = \ell(\mathbf{x})$. That is to say, the conditional $p(\mathbf{x} \mid y, \mathcal{D}_N)$ is not constant wrt to y. While Garnett [75] perceives this as a "minor mathematical error" on the part of Bergstra et al. [14], it may also be intepreted as a strong simplifying modelling assumption. Specifically, the assumption states that $p(\mathbf{x} \mid y, \mathcal{D}_N)$ is piecewise constant where $p(\mathbf{x} \mid y, \mathcal{D}_N) = \ell(\mathbf{x})$ for $y \leq \tau$. This approximation is not unreasonable, especially when τ is in close proximity to the global minimum y^*.

The interested reader is referred to the issues thread on the public GitHub repository associated with the BO textbook by Garnett [75] for further discussion. The discourse is further extended by Song and Ermon [240] who subsequently proposed an alternative method [241] that encompasses, in a stricter sense, both PI/EI, and, more generally, any acquisition function that assumes the form of the expected utility in Equation (2.45). The approach is named likelihood-free BO (LFBO) by virtue of its ability to sidestep the cumbersome calculations that such acquisition functions often entail. As we shall see in Section 5.3.3, LFBO is similar to BORE in spirit, but distinct in a few mathematical particulars.

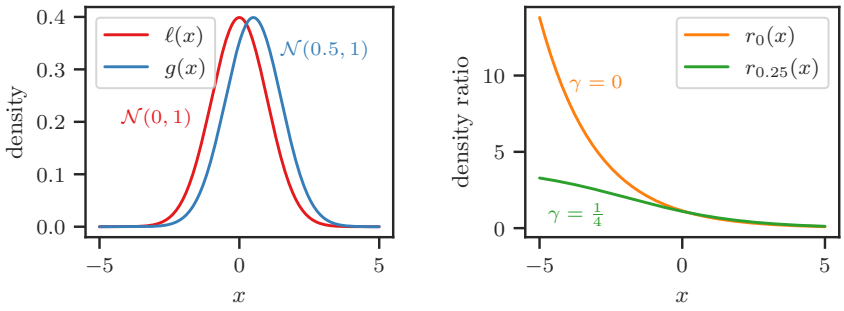


Figure 5.2: Gaussian densities (left) and their γ-relative density-ratios (right), which diverges when $\gamma = 0$ and converges to 4 when $\gamma = 1/4$.

5.2.3 *Tree-structured Parzen Estimator*

The tree-structured Parzen estimator (TPE) [14] is an instance of the BORE framework that seeks to solve the optimisation problem of Equation (5.4) by taking the following approach:

1. Since $r_\gamma(\mathbf{x}) = h_\gamma(r_0(\mathbf{x}))$ where h_γ is strictly non-decreasing, focus instead on maximising[2] $r_0(\mathbf{x})$,

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{X}} r_0(\mathbf{x}).$$

2. Estimate the ordinary density-ratio $r_0(\mathbf{x})$ by separately estimating its constituent numerator $\ell(\mathbf{x})$ and denominator $g(\mathbf{x})$, using a tree-based variant of KDE [233].

It is not hard to see why TPE might be favorable compared to methods based on GP regression – one now incurs an $\mathcal{O}(N)$ computational cost as opposed to the $\mathcal{O}(N^3)$ cost of GP posterior inference. Furthermore, it is equipped to deal with tree-structured, mixed continuous, ordered, and unordered discrete inputs. In spite of its advantages, TPE is not without shortcomings.

5.2.4 *Potential Pitfalls*

The shortcomings of this approach are already well-documented in the DRE literature [251]. Nonetheless, we reiterate here a select few that are particularly detrimental in the context of global optimisation. Namely, the first major drawback of TPE lies within step 1:

SINGULARITIES. Relying on the ordinary density-ratio can result in numerical instabilities since it is unbounded – often diverging to

2 $r_0(\mathbf{x})$ denotes $\gamma = 0$ solely in $r_\gamma(\mathbf{x})$ of Equation (5.1) – it does *not* signify threshold $\tau \triangleq \Phi^{-1}(0)$, which would lead to density $\ell(\mathbf{x})$ containing no mass. We address this subtlety in Section 5.A.

infinity, even in simple toy scenarios (see Figure 5.2 for a simple example). In contrast, the γ-relative density-ratio is always bounded above by γ^{-1} when $\gamma > 0$ [292]. The other potential problems of TPE lie within step 2:

VAPNIK'S PRINCIPLE. Conceptually, independently estimating the densities is actually a more cumbersome approach that violates Vapnik's principle – namely, that when solving a problem of interest, one should refrain from solving a more general problem as an intermediate step [273]. In this instance, *density* estimation is a more general problem that is arguably more difficult than *density-ratio* estimation [123].

KERNEL BANDWIDTH. KDE depends crucially on the selection of an appropriate kernel bandwidth, which is notoriously difficult [196, 229]. Furthermore, even with an optimal selection of a single fixed bandwidth, it cannot simultaneously adapt to low- and high-density regions [256].

ERROR SENSITIVITY. These difficulties are exacerbated by the fact that one is required to select *two* bandwidths, whereby the optimal bandwidth for one individual density is not necessarily appropriate for estimating the density-*ratio* – indeed, it may even have deleterious effects. This also makes the approach unforgiving to misspecification of the respective estimators, particularly in that of the denominator $g(\mathbf{x})$, which has a disproportionately large influence on the resulting density-ratio.

CURSE OF DIMENSIONALITY. For these reasons and more, KDE often falls short in high-dimensional regimes. In contrast, direct DRE methods have consistently been shown to scale better with dimensionality [250].

OPTIMISATION. Ultimately, we care not only about *estimating* the density-ratio, but also *optimising* it wrt to inputs for the purpose of candidate suggestion. Being nondifferentiable, the ratio of TPEs is cumbersome to optimise.

5.3 BAYESIAN OPTIMISATION BY PROBABILISTIC CLASSIFICATION

We propose a different approach to BORE, importantly, one that circumvents the issues of TPE, by seeking to *directly* estimate the unknown ratio $r_\gamma(\mathbf{x})$.

As we alluded to in Section 2.3, there exists a multitude of direct DRE methods. Here, we focus on the conceptually simple and widely-used

method based n class-probability estimation (CPE) [15, 37, 170, 203, 251], which we first introduced in Section 2.3.2. In this section, we extend the analysis to the more general case of the *relative* density-ratio, and to settings in which the classification problem is *unbalanced*.

First, let $\pi(\mathbf{x}) = p(z = 1 \mid \mathbf{x})$ denote the *class-posterior probability*, where z is the binary class label

$$z \triangleq \begin{cases} 1 & \text{if } y \leq \tau, \\ 0 & \text{if } y > \tau. \end{cases}$$

By definition, we have $\ell(\mathbf{x}) = p(\mathbf{x} \mid z = 1)$ and $g(\mathbf{x}) = p(\mathbf{x} \mid z = 0)$. We plug these into Equation (5.1) and apply Bayes' rule, letting the $p(\mathbf{x})$ terms cancel each other out to give

$$r_\gamma(\mathbf{x}) = \frac{p(z = 1 \mid \mathbf{x})}{p(z = 1)} \left(\gamma \cdot \frac{p(z = 1 \mid \mathbf{x})}{p(z = 1)} + (1 - \gamma) \cdot \frac{p(z = 0 \mid \mathbf{x})}{p(z = 0)} \right)^{-1} \tag{5.6}$$

Since, by definition, $p(z = 1) = \gamma$, Equation (5.6) simplifies to

$$r_\gamma(\mathbf{x}) = \gamma^{-1} \pi(\mathbf{x}). \tag{5.7}$$

Refer to Section 5.B for derivations. Thus, Equation (5.7) establishes the link between the class-posterior probability and the relative density-ratio. In particular, the latter is equivalent to the former up to constant factor γ^{-1}.

The astute reader will recognise from Equation (5.3) that, in fact,

$$\alpha_{\text{PI}} \left(\mathbf{x}; \mathcal{D}_N, \Phi^{-1}(\gamma) \right) = \gamma \cdot r_\gamma(\mathbf{x}) = \pi(\mathbf{x}).$$

Therefore, maximising the PI criterion amounts to maximising the class-posterior probability $\pi(\mathbf{x})$, which we can estimate using a probabilistic classifier – a function $\pi_{\boldsymbol{\theta}} : \mathcal{X} \to [0, 1]$ parameterised by $\boldsymbol{\theta}$. To recover the true class-posterior probability, we minimise a proper scoring rule [85] such as the log loss

$$\hat{\mathcal{L}}(\boldsymbol{\theta}) \triangleq -\frac{1}{N} \left(\sum_{n=1}^{N} z_n \log \pi_{\boldsymbol{\theta}}(\mathbf{x}_n) \sum_{n=1}^{N} + (1 - z_n) \log \left(1 - \pi_{\boldsymbol{\theta}}(\mathbf{x}_n)\right) \right). \tag{5.8}$$

Thereafter, we can use $\pi_{\boldsymbol{\theta}}(\mathbf{x})$ as a proxy to the PI criterion,

$$\pi_{\boldsymbol{\theta}}(\mathbf{x}) \approx \alpha_{\text{PI}} \left(\mathbf{x}; \mathcal{D}_N, \Phi^{-1}(\gamma) \right) \tag{5.9}$$

where the approximation is tight at $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \hat{\mathcal{L}}(\boldsymbol{\theta})$. Note $\hat{\mathcal{L}}$ is an unbiased estimate of the log loss $\mathcal{L}$ that first appeared in Equation (5.13). Refer to Section 5.C for details.

Hence, in the so-called BO loop (summarised in Algorithm 2), we alternately optimise (i) the classifier parameters $\boldsymbol{\theta}$ wrt to the log loss (to improve the approximation of Equation (5.9); Line 6), and (ii) the

Algorithm 2: Bayesian optimisation by density-ratio estimation (BORE).

Input: blackbox $f : \mathcal{X} \to \mathbb{R}$, proportion $\gamma \in (0, 1)$, probabilistic classifier $\pi_{\boldsymbol{\theta}} : \mathcal{X} \to [0, 1]$.

1 **while** *under budget* **do**
2 $\quad \tau \leftarrow \Phi^{-1}(\gamma)$ // compute γ-th quantile of $\{y_n\}_{n=1}^N$
3 $\quad z_n \leftarrow \mathbb{I}[y_n \leq \tau]$ for $n = 1, \dots, N$ // assign labels
4 $\quad \tilde{\mathcal{D}}_N \leftarrow \{(\mathbf{x}_n, z_n)\}_{n=1}^N$ // construct auxiliary dataset
5 $\quad$ /* update classifier by optimising parameters $\boldsymbol{\theta}$ wrt log loss */
6 $\quad \boldsymbol{\theta}^* \leftarrow \arg\min_{\boldsymbol{\theta}} \hat{\mathcal{L}}(\boldsymbol{\theta})$ // depends on $\tilde{\mathcal{D}}_N$, see Equation (5.8)
7 $\quad$ /* suggest candidate by optimising input $\mathbf{x}$ wrt classifier */
8 $\quad \mathbf{x}_N \leftarrow \arg\max_{\mathbf{x} \in \mathcal{X}} \pi_{\boldsymbol{\theta}^*}(\mathbf{x})$ // see Equation (5.9)
9 $\quad y_N \leftarrow f(\mathbf{x}_N)$ // evaluate blackbox function
10 $\quad \mathcal{D}_N \leftarrow \mathcal{D}_{N-1} \cup \{(\mathbf{x}_N, y_N)\}$ // update dataset
11 $\quad N \leftarrow N + 1$
12 **end**

classifier input $\mathbf{x}$ wrt to its output (to suggest the next candidate to evaluate; Line 8).

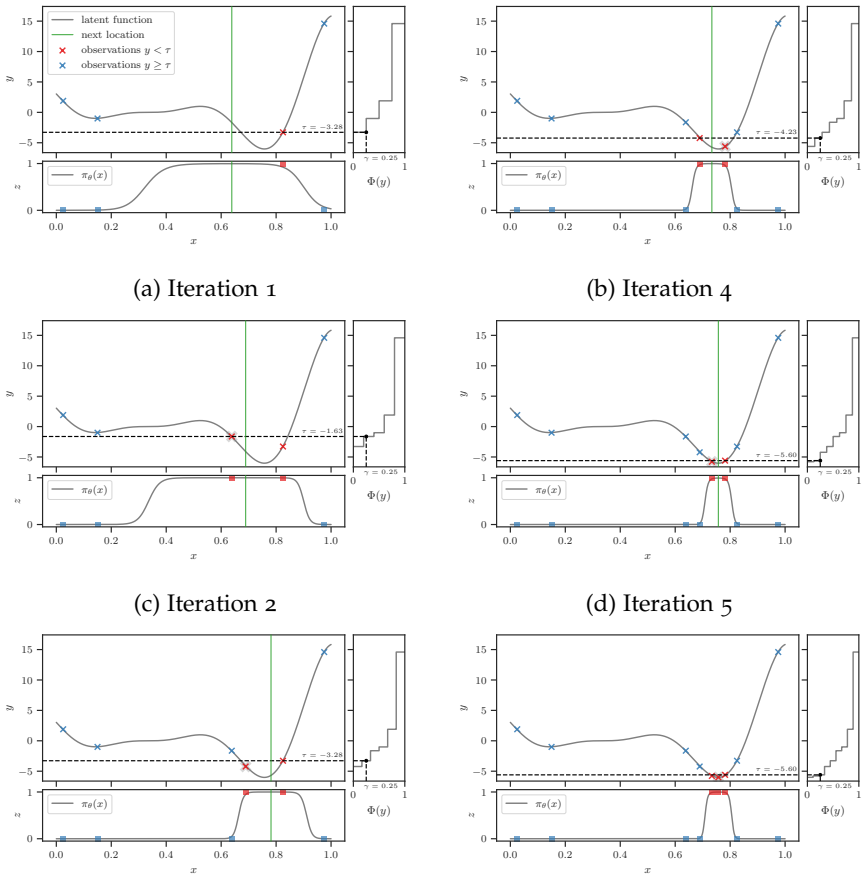
In traditional GP-based PI, Line 8 typically consists of maximising the PI criterion expressed in the form of Equation (2.48), while Line 6 consists of optimising the GP hyperparameters wrt the marginal likelihood. By analogy with our approach, the parameterised function $\pi_{\boldsymbol{\theta}}(\mathbf{x})$ is *itself* an approximation to the PI criterion to be maximised directly, while the approximation is tightened through by optimising the classifier parameters wrt the log loss. In short, we have reduced the problem of computing PI to that of learning a probabilistic classifier, thereby unlocking a broad range of estimators beyond those so far used in BO. Importantly, this enables one to employ virtually any state-of-the-art classification method available and to parameterise the classifier using arbitrarily expressive approximators that potentially have the capacity to deal with non-linear, non-stationary, and heteroscedastic phenomena frequently encountered in practice.

TOY 1D EXAMPLE. To illustrate, in Figure 5.3, we animate Algorithm 2 step by step on a synthetic problem for a half dozen iterations. Specifically, we minimise the FORRESTER function

$$f(x) \triangleq (6x - 2)^2 \sin(12x - 4),$$

in the domain $x \in [0, 1]$ with observation noise $\varepsilon \sim \mathcal{N}(0, 0.05^2)$. The algorithm is started with 4 random initial designs. Each subfigure depicts the state after Lines 6 and 8 – namely, after *updating* and *maximising* the classifier, respectively. In every subfigure, the main pane depicts the noise-free function, represented by the *solid gray* curve, and the set of observations, represented by *crosses* '×'. The

location that was evaluated in the previous iteration is highlighted with a *gray outline*. The right pane shows the empirical cdf (ECDF) of the observed y values. The *vertical dashed black line* in this pane is located at $\gamma = \frac{1}{4}$. The *horizontal dashed black line* is located at τ, the value of y such that $\Phi(y) = \frac{1}{4}$, i. e., $\tau = \Phi^{-1}\left(\frac{1}{4}\right)$. The instances below this horizontal line are assigned binary label $z = 1$, while those above are assigned $z = 0$. This is visualised in the bottom pane, alongside the probabilistic classifier $\pi_\theta(\mathbf{x})$, represented by the *solid gray* curve. Finally, the maximiser of the classifier is represented by the *vertical solid green* line – this denotes the location to be evaluated in the next iteration.



(a) Iteration 1

(b) Iteration 4

(c) Iteration 2

(d) Iteration 5

(e) Iteration 3

(f) Iteration 6

Figure 5.3: Step-by-step animation of Algorithm 2 on the FORRESTER synthetic problem.

5.3.1 *Choice of Proportion γ*

The proportion $\gamma \in (0, 1)$ influences the explore-exploit trade-off. Intuitively, a smaller setting of γ encourages exploitation and leads to fewer modes and sharper peaks in the acquisition function. To see

this, consider that there are by definition fewer candidate inputs $\mathbf{x}$ for which its corresponding output y can be expected to improve over the first quartile ($\gamma = 1/4$) of the observed output values than, say, the third quartile ($\gamma = 3/4$). That being said, given that the class balance rate is by definition γ, a value too close to 0 may lead to instabilities in classifier learning. A potential strategy to combat this is to begin with a perfect balance ($\gamma = 1/2$) and then to decay γ as optimisation progresses.

In this work, we keep γ fixed throughout optimisation, which, on the other hand, has the benefit of providing guarantees about how the classification task evolves. In particular, in each iteration, after having observed a new evaluation, we are guaranteed that the binary label of *at most* one existing instance can flip. This property can be exploited to make classifier learning of Line 6 more efficient. More specifically, assuming the proportion γ is fixed across iterations, then, in each iteration, we are guaranteed the following changes:

1. a new input and its corresponding output $(\mathbf{x}_N, y_N)$ will be added to the dataset, thus
2. creating a shift in the rankings and, by extension, quantiles of the observed y values, in turn
3. leading to the binary label of *at most* one instance to flip.

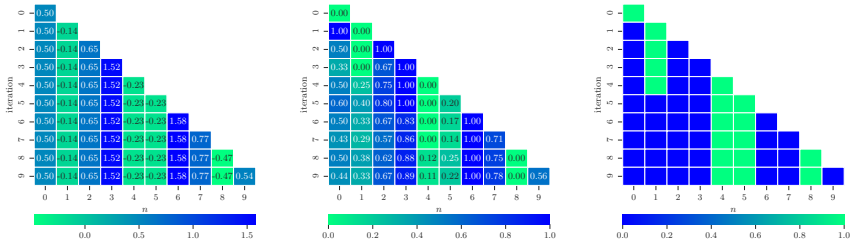
Therefore, between consecutive iterations, changes to the classification dataset are fairly incremental. One can leverage this to make classifier training more efficient, especially in families of classifiers for which re-training entirely from scratch in each iteration is superfluous and wasteful. See Figure 5.4 for an illustrative example, in which the task is to optimise a contrived, synthetic "noise-only" function $f(x) = 0$ with observation noise $\varepsilon \sim \mathcal{N}(0, 1)$, and the proportion is set to $\gamma = 1/4$.

Some viable strategies for reducing per-iteration classier learning overhead may include speeding up convergence by (i) *importance sampling* (e. g., re-weighting new samples and those for which the label have flipped), (ii) *early-stopping* (stop training early if either the loss or accuracy have not changed for some number of epochs) and (iii) *annealing* (decaying the number of epochs or batch-wise training steps as optimisation progresses).

5.3.2 *Choice of Probabilistic Classifier*

We examine a few variations of BORE that differ in the choice of classifier and discuss their strengths and weaknesses across different global optimisation problem settings.

MULTI-LAYER PERCEPTRONS. We propose BORE-MLP, a variant based on MLPs. This choice is appealing not only for (i) its flexibility



(a) Continuous targets y_n (b) Empirical distribution $\Phi(y_n)$ (c) Binary labels z_n

Figure 5.4: Optimising a synthetic "noise-only" function. As we iterate through the BO loop from top to bottom, the array of targets grows from left to right. In each iteration, the size of the array increases by one, resulting in a re-shuffling of the rankings and, by extension, quantiles. This in turn leads to the label for *at most* one instance to flip. Hence, between consecutive iterations, changes to the classification dataset are fairly incremental. This property can be exploited to make classifier training more efficient in each iteration.

and universal approximation guarantees [107] but because (ii) one can easily adopt stochastic gradient descent (SGD) methods to scale up its parameter learning [142], and (iii) it is differentiable end-to-end, thus enabling the use of quasi-Newton methods such as L-BFGS [150] for candidate suggestion. Lastly, since SGD is online by nature, (iv) it is feasible to adapt weights from previous iterations instead of training from scratch. A notable weakness is that MLPs can be over-parameterised and therefore considerably data-hungry.

TREE-BASED ENSEMBLES. We consider two further variants: BORE-RF and BORE-XGB, both based on ensembles of decision trees – namely, random forest (RF) [19] and XGBOOST [34], respectively. These variants are attractive since they inherit from decision trees the ability to (i) deal with discrete and conditional inputs by design, (ii) work well in high-dimensions, and (iii) are scalable and easily parallelizable. Further, (iv) online extensions of RFs [222] may be applied to avoid training from scratch. A caveat is that, since their response surfaces are discontinuous and nondifferentiable, decision trees are difficult to maximise. Therefore, we appeal to random search and evolutionary strategies for candidate suggestion. Further details and a comparison of various approaches is included in Section 5.5.5.2.

In theory, for the approximation of Equation (5.9) to be tight, the classifier is required to produce well-calibrated probabilities [170]. A potential drawback of the BORE-RF variant is that RFs are generally not trained by minimising a proper scoring rule. As such, additional techniques may be necessary to improve calibration [189].

GAUSSIAN PROCESSES. The last variant we consider is BORE-GP, based on a GP classifier (GPC) [285]. Like the GP regression model, GPC offers (i) a high degree of flexibility, at least on smooth functions up to moderate dimensionalities, and (ii) well-calibrated uncertainty estimates (useful for marginalising out the hyperparameters from the acquisition function, as we discuss in Section 5.G). On the other hand, GPC not only loses one of the foremost appeals of GP regression, namely, analytical tractability of the predictive, but it is also not necessarily better equipped to deal with the more problematic settings we have discuseed (discrete variables, high-dimensionalities, etc..), and its scalability is contingent on the choice of inference approximation utilised.

5.3.3 *Likelihood-Free BO by Weighted Classification*

We give a brief overview of the LFBO framework of Song and Ermon [240]. Recall from Section 2.3.1 that every convex, lower-semicontinuous function[3] f can be represented in terms of its convex dual $f^\star$,

$$f(u) = \max_s \left\{ uf'(s) - f^\star(f'(s)) \right\}$$

For any function $\alpha : \mathcal{X} \to \mathbb{R}$, we can leverage this variational representation to obtain a lower bound on expectations of the form $\mathbb{E}[f(\alpha(\mathbf{x}))]$,

$$\begin{aligned}
\mathbb{E}_{p(\mathbf{x})}\left[f(\alpha(\mathbf{x}))\right] &= \mathbb{E}_{p(\mathbf{x})}\left[\max_s \left\{\alpha(\mathbf{x})f'(s) - f^\star(f'(s))\right\}\right] \\
&\geq \max_s \mathbb{E}_{p(\mathbf{x})}\left[\alpha(\mathbf{x})f'(s) - f^\star(f'(s))\right] \\
&\geq \max_{S:\mathcal{X}\to\mathbb{R}} \mathbb{E}_{p(\mathbf{x})}\left[\alpha(\mathbf{x})f'(S(\mathbf{x})) - f^\star(f'(S(\mathbf{x})))\right]. \quad (5.10)
\end{aligned}$$

Using the convexity of f, it's easy to show that the maximiser of Equation (5.10) is

$$\begin{aligned}
S^* &\triangleq \arg\max_{S:\mathcal{X}\to\mathbb{R}} \mathbb{E}_{p(\mathbf{x})}\left[\alpha(\mathbf{x})f'(S(\mathbf{x})) - f^\star(f'(S(\mathbf{x})))\right] \\
&= \alpha
\end{aligned}$$

Let's now optimise Equation (5.10) over a family of functions parameterised by $\boldsymbol{\theta}$,

$$\boldsymbol{\theta}^* \triangleq \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{p(\mathbf{x})}\left[\alpha(\mathbf{x})f'(S_{\boldsymbol{\theta}}(\mathbf{x})) - f^\star(f'(S_{\boldsymbol{\theta}}(\mathbf{x})))\right].$$

This gives an approximation for function $\alpha(\mathbf{x})$,

$$S_{\boldsymbol{\theta}}(\mathbf{x}) \approx \alpha(\mathbf{x}),$$

3 we are overloading the notation f here, which has been used earlier in this chapter to denote the unknown blackbox function we're seeking to minimising

that is tight at $\boldsymbol{\theta} = \boldsymbol{\theta}^*$. Suppose now the function of interest is the expected utility $\alpha(\mathbf{x}; \mathcal{D}_N, \tau) \triangleq \mathbb{E}_{p(y \mid \mathbf{x}, \mathcal{D}_N)}[U(y; \tau)]$ from Equation (2.45). Then we have

$$\begin{aligned}
\boldsymbol{\theta}^* &\triangleq \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{p(\mathbf{x})} \left[\mathbb{E}_{p(y \mid \mathbf{x}, \mathcal{D}_N)}[U(y; \tau)] f'(S_{\boldsymbol{\theta}}(\mathbf{x})) - f^\star(f'(S_{\boldsymbol{\theta}}(\mathbf{x}))) \right] \\
&= \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{p(\mathbf{x}, y)} \left[U(y; \tau) f'(S_{\boldsymbol{\theta}}(\mathbf{x})) - f^\star(f'(S_{\boldsymbol{\theta}}(\mathbf{x}))) \right] \\
&= \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{p(\mathbf{x}, y)} \left[U(y; \tau) \log \pi_{\boldsymbol{\theta}}(\mathbf{x}) + \log(1 - \pi_{\boldsymbol{\theta}}(\mathbf{x})) \right] \qquad (5.11)
\end{aligned}$$

where we obtain Equation (5.11) by setting $\pi_{\boldsymbol{\theta}}(\mathbf{x}) \triangleq \sigma(\log S_{\boldsymbol{\theta}}(\mathbf{x}))$ and $f(u) \triangleq u \log u - (u + 1) \log (u + 1)$ from Equations (2.12) and (2.13). We can view Equation (5.11) as a weighted objective function for binary classification where the utility $U(y; \tau)$ acts as the nonnegative weight and $\pi_{\boldsymbol{\theta}}(\mathbf{x})$ is the probabilistic classifier, as in Section 5.3. Finally, we obtain an approximation to the acquisition function,

$$S_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{\pi_{\boldsymbol{\theta}}(\mathbf{x})}{1 - \pi_{\boldsymbol{\theta}}(\mathbf{x})} \approx \alpha(\mathbf{x}; \tau).$$

Like BORE, this circumvents the need to explicitly solve the integral in Equation (2.45) and thus places no restrictions on the form of $p(y \mid \mathbf{x}, \mathcal{D}_N)$. Furthermore, this approach can be applied to any utility function $U(y; \tau)$, not only PI, but EI, and beyond.

5.4 RELATED WORK

The literature on BO is vast and ever-expanding [20, 71, 75, 228]. Some specific threads pertinent to our work are those that consider alternative modelling paradigms to GPs, e. g., using NNs to obtain greater flexibility and scalability, as in BANANAS [281], ABLR [200], BOHAMIANN [243], and DNGO [237], and using tree ensembles such as RFs to handle discrete and conditional variables, as in SMAC [111]. To negotiate the tractability of the predictive, these methods must either make simplifications or resort to approximations. In contrast, by seeking to directly approximate the acquisition function, BORE is unencumbered by such constraints. Refer to Section 5.G for an expanded discussion.

Beyond the classical improvement-based PI [134] and EI functions [117], a multitude of acquisition functions has been devised, notably, the UCB [244], KG [225], ES [96], PES [102], and max-value ES (MES) [279]. Nonetheless, the improvement-based criteria remain ubiquitous in large because they are conceptually simple, easy to evaluate and optimise, and consistently performs well in practice. As we examined in Section 5.3.3, a variant of our model-agnostic approach to BO known as LFBO [240] accomodates both of the improvement-based criteria, and, more broadly, any acquisition function that can be expressed as the expected utility in Equation (2.45).

Density-ratio estimation is a well-established area with an extensive literature [251]. In light of the drawbacks of the KDE approach discussing in Section 5.2.4, myriad alternatives have been proposed, including KLIEP [250], KMM [88], ULSIF [122], and RULSIF [292]. In this work, we restrict our focus on CPE, which stands out for its effectiveness and versatility, demonstrated by its widespread use in various applications including covariate shift adaptation [15, 250, 268], EBMS [90, 92, 269], GANS [86, 191], likelihood-free inference [64, 257, 267], and beyond. Particularly relevant among these is its applications in BED, a close relative of BO, in which it is similarly used to approximate the expected utility function [129, 130].

5.5 EXPERIMENTS

We describe the experiments conducted to empirically evaluate our method. To this end, we consider a variety of problems, ranging from automated machine learning (AUTOML), robotic arm control, to racing line optimisation.

We provide comparisons against a comprehensive selection of state-of-the-art baselines. Namely, across all problems, we consider random search (RS) [13], GP-BO (using EI with $\gamma = 0$) [117], TPE [14], and SMAC [111]. We also consider evolutionary strategies: differential evolution (DE) [249] for problems with continuous domains, and regularised evolution (RE) [210] for those with discrete domains. Further information about these baselines and the source code for their implementations are included in Section 5.D.

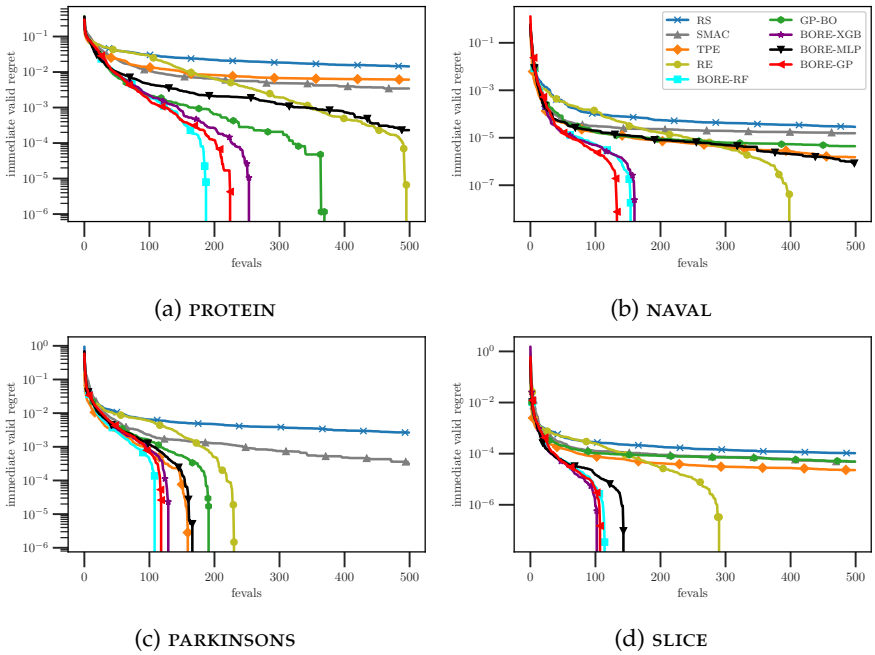
To quantitatively assess performance we report the *immediate regret* (in benchmarks for which the exact global minimum is known), defined as the absolute error between the global minimum and the lowest function value attained thus far. Unless otherwise stated we report, for each benchmark and method, results aggregated across 100 replicated runs.

We set $\gamma = 1/3$ across all variants and benchmarks. For candidate suggestion in the tree-based variants, we use RS with a function evaluation limit of 500 for problems with discrete domains, and DE with a limit of 2,000 for those with continuous domains. Our open-source implementation is available on GitHub at ltiao/bore. Further details concerning the experimental set-up and the implementation of each variant are included in Section 5.E.

5.5.1 *Neural Network Tuning (HPOBench)*

First, we consider the problem of training a two-layer feed-forward NN for regression. Specifically, a NN is trained for 100 epochs with the ADAM optimiser [126], and the objective is the validation MSE. The hyperparameters are the *initial learning rate*, *learning rate schedule*, *batch*

size, along with the layer-specific *widths*, *activations*, and *dropout rates*. We consider four datasets: PROTEIN, NAVAL, PARKINSONS, and SLICE, and utilise HPOBench [128] which tabulates, for each dataset, the MSEs resulting from all possible (62,208) configurations. Additional details are included in Section 5.F.1, and the results are shown in Figure 5.5. We see across all datasets that the BORE-RF and -XGB variants consis-

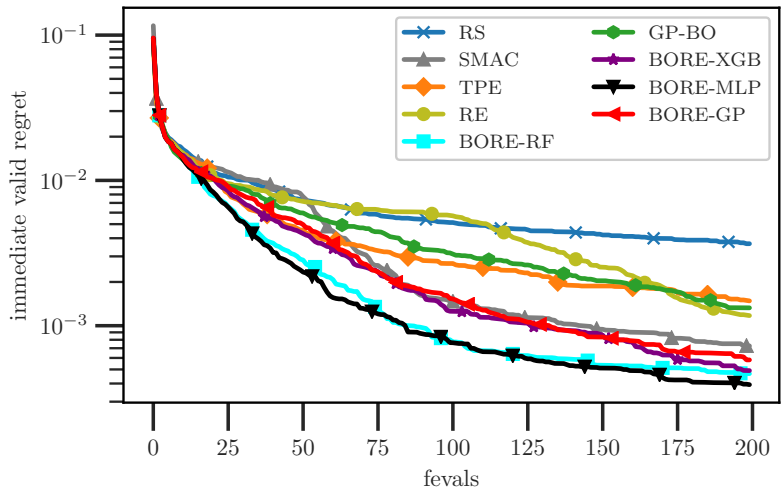


(a) PROTEIN (b) NAVAL

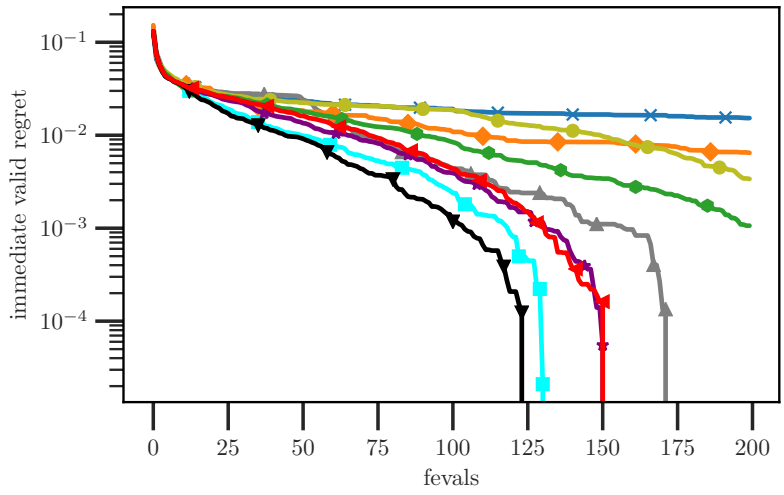
(c) PARKINSONS (d) SLICE

Figure 5.5: Immediate regret over function evaluations on the HPOBench neural network tuning problems ($D = 9$).

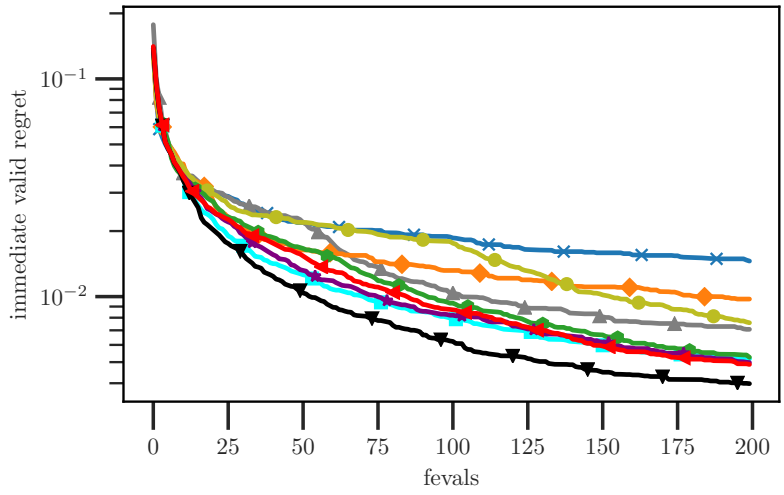
tently outperform all other baselines, converging rapidly toward the global minimum after 1-2 hundred evaluations – in some cases, earlier than any other baseline by over two hundred evaluations. Notably, with the exception being BORE-MLP on the PARKINSONS dataset, all BORE variants outperform TPE, in many cases by a sizable margin.



(a) CIFAR-10



(b) CIFAR-100



(c) ImageNet-16

Figure 5.6: Immediate regret over function evaluations on the NASBench201 NAS problems ($D = 6$).

5.5.2 *Neural Architecture Search (NASBench201)*

Next, we consider a NAS problem, namely, that of designing a neural cell. A cell is represented by a directed acyclic graph (DAG) with 4 nodes, and the task is to assign an *operation* to each of the 6 possible arcs from a set of five operations. We utilise NASBench201 [60], which tabulates precomputed results from all possible $5^6 = 15,625$ combinations for each of the three datasets: CIFAR-10, CIFAR-100 [132], and ImageNet-16 [43]. Additional details are included in Section 5.F.2, and the results are shown in Figure 5.6. We find across all datasets that the BORE variants consistently achieve the lowest final regret among all baselines. Not only that, the BORE variants, in particular BORE-MLP, maintains the lowest regret at anytime (i. e. at any optimisation iteration), followed by BORE-RF, then BORE-XGB/-GP. In this problem, the inputs are purely categorical, whereas in the previous problem they are a mix of categorical and ordinal. For the BORE-MLP variant, categorical inputs are one-hot encoded, while ordinal inputs are handled by simply rounding to their nearest integer index. The latter is known to have shortcomings [77], and might explain why BORE-MLP is the most effective variant in this problem but the least effective in the previous one.

5.5.3 *Robot Arm Pushing*

We consider the 14D control problem first studied by Wang and Jegelka [279]. The problem is concerned with tuning the controllers of robot hands to push objects to some desired locations. Specifically, there are two robots, each tasked with manipulating an object. For each robot, the control parameters include the *location* and *orientation* of its hands, the *moving direction*, *pushing speed*, and *duration*. Due to the prohibitively large number of function evaluations ($\sim$10,000) required to achieve reasonable performance, we omit all GP-based methods from our comparisons on this benchmark. Further, we reduce the number of replicated runs of each method to 50. Additional details are included in Section 5.F.3, and the results are shown in Figure 5.7. We see that BORE-XGB attains the highest reward, followed by BORE-RF and TPE (which attain roughly the same performance), and then BORE-MLP.

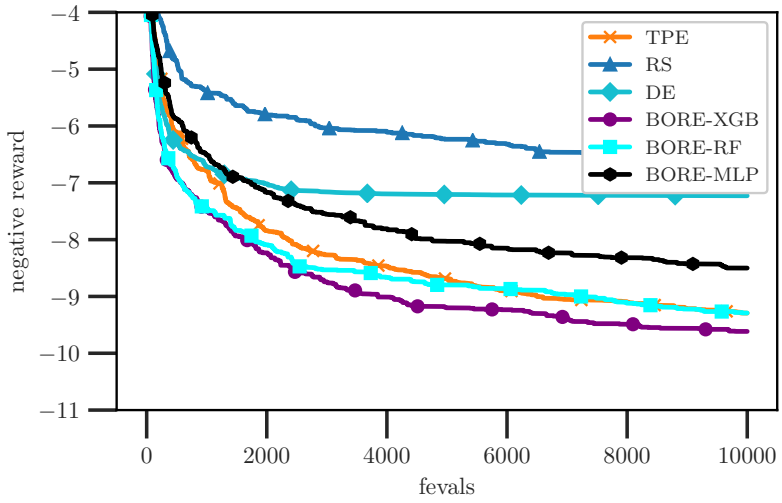
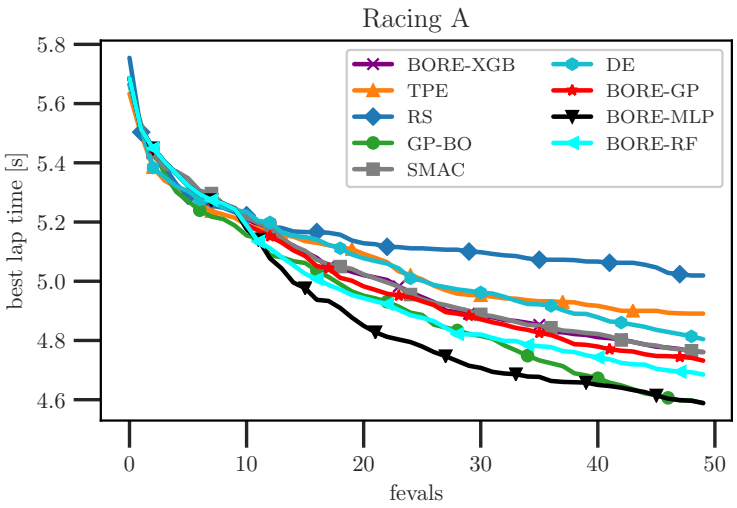
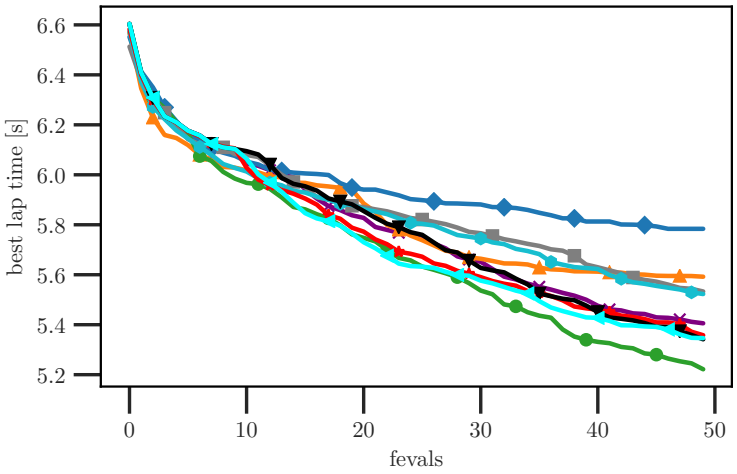


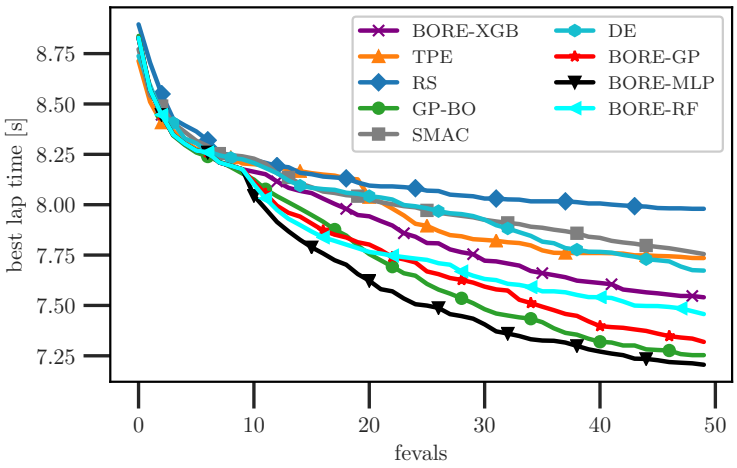
Figure 5.7: Negative reward over function evaluations on the Robot Pushing task ($D = 14$).



(a) UC BERKELEY ($D = 12$)



(b) ETH ZÜRICH A ($D = 20$)



(c) ETH ZÜRICH B ($D = 21$)

Figure 5.8: Best lap times (in seconds) over function evaluations in the racing line optimisation problem on various racetracks.

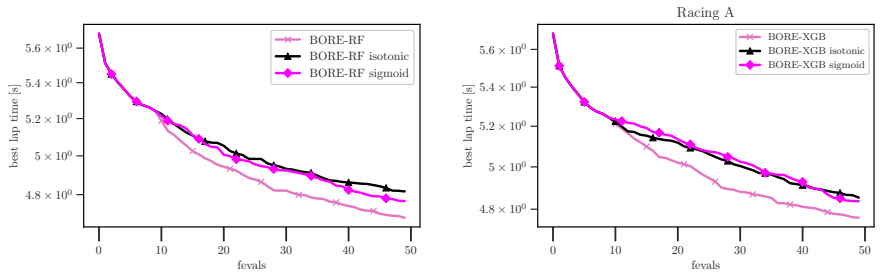
5.5.4 *Racing Line Optimisation.*

We consider the problem of computing the optimal racing line for a given track and vehicle with known dynamics. We adopt the set-up of Jain and Morari [113], who consider the dynamics of miniature scale cars traversing the tracks at UC BERKELEY and ETH ZÜRICH. The racing line is a trajectory determined by D waypoints placed along the length of the track, where the ith waypoint deviates from the centerline of the track by $x_i \in \left[-\frac{W}{2}, \frac{W}{2}\right]$ for some track width W. The task is to minimise the lap time $f(\mathbf{x})$, the minimum time required to traverse the trajectory parameterised by $\mathbf{x} = [x_1 \cdots x_D]^\top$. Additional details are included in Section 5.F.4, and the results are shown in Figure 5.8. First, we see that the BORE variants consistently outperform all baselines except for GP-BO. This is to be expected since the function is continuous, smooth, and has ∼20 dimensions or less. Nonetheless, we find that the BORE-MLP variant performs as well as, or marginally better than, GP-BO on two tracks. In particular, on the UC BERKELEY track, we see that BORE-MLP achieves the best lap times for the first ∼40 evaluations, and is caught up to by GP-BO in the final 10. On ETH ZÜRICH track B, BORE-MLP consistently maintains a narrow lead.

5.5.5 *Ablation Studies*

5.5.5.1 *Effects of calibration*

As discussed in Section 5.3.2, calibrating the classifier may have a profound effect on the tree-based variants of BORE, namely, BORE-RF and BORE-XGB. We consider two popular approaches [189], namely, Platt scaling [201] and isotonic regression [299, 300]. The results shown



(a) Calibrated RF classifiers.

(b) Calibrated XGBOOST classifiers.

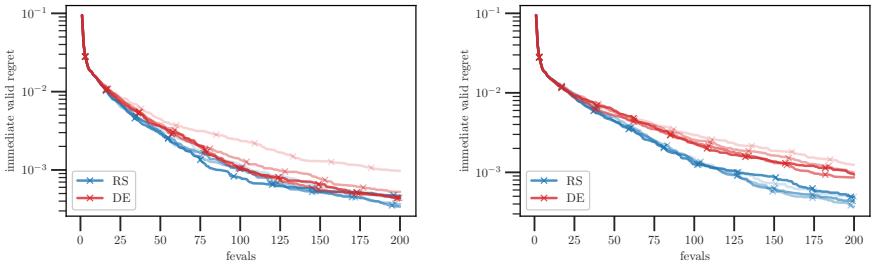
Figure 5.9: Effects of calibrating classifiers in the BORE-RF and BORE-XGB variants. Results of racing line optimisation on the UC BERKELEY track.

in Figure 5.9 suggest that applying these calibration techniques may actually have deleterious effects. However, this can also be adequately explained by overfitting due to insufficient calibration samples. In this particular problem, only a small number of function evaluations are required for convergence to the global minimum, so this produces

a small dataset with which to calibrate the classifier. In the case of isotonic regression, typically ∼1,000 samples are required. Therefore, it remains inconclusive whether calibration may in fact carry benefits, particularly in problem settings that produce large amounts of data.

5.5.5.2 *Effects of different strategies to maximise the acquisition function*

We examine different strategies for maximising the acquisition function (i. e. the classifier) in the tree-based variants of BORE, namely, BORE-RF and BORE-XGB. Decision trees are difficult to maximise since their response surfaces are discontinuous and nondifferentiable. Hence, we consider the following methods: RS and DE. For each method, we further consider different evaluation budgets, i. e., limits on the number of evaluations of the acquisition function. Specifically, we consider the limits 50, 100, 200, and 500.



(a) BORE-RF on CIFAR-10 (b) BORE-XGB on CIFAR-10

Figure 5.10: A comparison of various acquisition optimisation strategies on the NASBench201 problem.

In Figures 5.10a and 5.10b, we show the results of BORE-RF and BORE-XGB, respectively, on the CIFAR-10 dataset of the NASBench201 benchmark, as described in Section 5.F.2. Each curve represents the mean across 100 repeated runs. The opacity is proportional to the function evaluation limit, with the most transparent having the lowest limit and the most opaque having the highest limit. We find that RS appears to outperform DE by a narrow margin. Additionally, for DE, a higher evaluation limit appears to be somewhat beneficial, while the opposite holds true for RS.

5.6 DISCUSSION

In this section, we discuss the limitations of our method and suggest potential approaches to address them. We also discuss the significant outcomes of this work at the time of writing and their potential impact.

EXPLORATION. Similar to the TPE method, BORE generally has a tendency to favour exploitation over exploration. In the case of TPE, the maximiser of the acquisition function $\ell(\mathbf{x})/g(\mathbf{x})$ will be located at the

mode of $\ell(\mathbf{x})$, which has mass concentrated around inputs for which its output value is within the smallest proportion γ of all observed output values (i. e. inputs with label $z = 1$). Recall the classical formulation of EI from Equation (2.50) in which the explore-exploit trade-off is explicitly encoded in mathematical terms. Assuming we had access to its global optimum, then by design the solution is a candidate that strikes a good balance between exploration and exploitation. Indeed, by virtue of having lower predictive uncertainty, previously evaluated candidates will tend to have lower acquisition values, which helps to encourage exploration. In contrast, for TPE and BORE, the previously evaluated candidates labeled $z = 1$ will tend to retain high acquisition values. Therefore, in the worst-case scenario, the global optimum of the acquisition function may become stuck at some local optimum of the blackbox function, or a point within some neighborhood thereof. In practice, implementations of TPE avoid this scenario by introducing stochasticity in the acquisition optimisation, e. g., by randomly sampling from $\ell(\mathbf{x})$ and suggesting the sample that maximises $\ell(\mathbf{x})/g(\mathbf{x})$. We surmise that BORE was able to avoid such pathological cases in our experiments due in part to the sources of randomness inherent to the acquisition optimisation method of choice.

A further detail to note is that the labels z do not remain static throughout optimisation. In other words, the classification dataset is different for each new iteration. Recall that, by construction, only a fraction γ of the observations can have positive labels $z = 1$. With each iteration, observing a new value of y leads to a change in the threshold τ. Since only a fraction γ of observations can lie below this threshold, the labels of existing observations must accordingly flip intermittently throughout optimisation. Thus, as the probabilistic classifier $\pi_\theta(\mathbf{x})$ adapts to these updates, the regions in which it outputs high probabilities will also shift accordingly. Consequently, the classifier response surface will either become multimodal (leading to exploration) or become narrower and more sharply-peaked in the same region (leading to exploitation).

Although not discussed in this thesis, the behavior described above can make simple ϵ-greedy strategies particularly effective at stimulating exploration. Follow-up work by Oliveira, Tiao, and Ramos [193] has considered batch extensions using Stein variational gradient descent (SVGD) [151], which encourages greater diversity in the query batch and provides theoretical guarantees.

HYPERPARAMETER ESTIMATION. Firstly, a noteworthy consequence of seeking to directly approximate PI under its alternative formulation is that the classifier parameters θ in BORE can be interpreted as *hyperparameters* (in the same way that the parameters of the GP kernel are hyperparameters), a deterministic treatment of which based on point estimates can often be viable. For example, in the BORE-MLP

variant, $\boldsymbol{\theta}$ consists of the layer weights, which we are able to estimate using type-II maximum likelihood. In contrast, to utilise NNs in traditional BO, generally the layer weights $\boldsymbol{\omega}$ are parameters that must first be marginalised out in order to compute the predictive $p_{\boldsymbol{\theta}}(y \mid \mathbf{x}, \mathcal{D}_N) = \int p_{\boldsymbol{\theta}}(y \mid \mathbf{x}, \boldsymbol{\omega}) p_{\boldsymbol{\theta}}(\boldsymbol{\omega} \mid \mathcal{D}_N) \, \mathrm{d}\boldsymbol{\omega}$, while the hyperparameters $\boldsymbol{\theta}$, consisting of e. g., the prior and likelihood precisions, may optionally be marginalised out as well (though usually point estimates suffice). Refer to Section 5.G for an expanded discussion on this distinction. As with the GP hyperparameters in GP-BO, in order to encourage exploration, it may be beneficial to consider placing a prior on $\boldsymbol{\theta}$ and marginalising out its uncertainty [236]. Further, compared against GP-BO, a potential downside of BORE is that there may be vastly more *meta-hyperparameters* settings from which to choose. Whereas in GP-BO these might consist of, e. g., the choice of kernel and its isotropy, there are potentially many more possibilities in BORE. In BORE-MLP, this may consist of, e. g., layer depth, widths, activations, etc. – the tuning of which is often the reason one appeals to BO in the first place. While we obtained remarkable results with the proposed variants without needing to deviate from the sensible defaults, in general, for further improvements in calibration and sample diversity, it may be beneficial to consider marginalising out even the meta-hyperparameters [280].

Impact to Date

We discuss the impactful outcomes of this work to date. First, we examine the real-world uses of BORE. Despite the emergence of various players in the field, toolkits such as hyperopt and Optuna still remain the most widely used for HPO, especially in the domain of AUTOML. These libraries rely foremost on the TPE method as their default search algorithm. Indeed, in many settings, e. g., those of high-dimensionality, TPE often outperforms other paradigms such as evolutionary strategies or traditional GP-based BO. Having addressed several of the most profound shortcomings of TPE, BORE has proved, in turn, to consistently outperform TPE. This was not only demonstrated in this chapter but has been independently observed in subsequent works. Thus seen, BORE stands poised to be an ideal candidate to replace TPE as the leading method for hyperparameter search. In fact, BORE has already been adopted as one of the primary search algorithms in SyneTune [224], a rapidly growing open-source framework for HPO developed by Amazon Research.

Second, we highlight BORE as a new research avenue. Less than a year since its initial publication, it has garnered recognition and is set to be featured in the upcoming textbook on Bayesian optimisation by Garnett [75], scheduled to be published later this year. Furthermore, prominent research labs have already embarked on extending BORE's capabilities, such as extensions to multiple objectives [53] and

generalisations to other acquisition functions [241], as we discussed in Section 5.3.3.

5.7 SUMMARY

We have presented a novel methodology for BO based on the relationship between improvement-based acquisition functions and probabilistic classifier. This observation is made through the well-known link between CPE and DRE, and the lesser-known insight that PI can be expressed as a relative density-ratio between two unknown distributions.

We discussed important ways in which TPE, an early attempt to exploit the latter link, falls short. Further, we demonstrated that our CPE-based approach to BORE, in particular, our variants based on the MLP, RF, XGBOOST, and GP classifiers, consistently outperform TPE, and compete well against the state-of-the-art derivative-free global optimisation methods.

Overall, the simplicity and effectiveness of BORE make it a promising approach for black-box optimisation, and its high degree of extensibility provides numerous exciting avenues for future work.

5.A RELATIVE DENSITY-RATIO: UNABRIDGED NOTATION

In Section 5.2, for notational simplicity, we had excluded the dependencies of ℓ, g and r_γ on τ. Let us now define these densities more explicitly as

$$\ell(\mathbf{x};\tau) \triangleq p\left(\mathbf{x} \,|\, y \leq \tau, \mathcal{D}_N\right), \qquad \text{and} \qquad g(\mathbf{x};\tau) \triangleq p\left(\mathbf{x} \,|\, y > \tau, \mathcal{D}_N\right),$$

and accordingly, the γ-relative density-ratio from Equation (5.1) as

$$r(\mathbf{x};\gamma,\tau) = \frac{\ell(\mathbf{x};\tau)}{\gamma\ell(\mathbf{x};\tau) + (1-\gamma)g(\mathbf{x};\tau)}.$$

Recall from Equation (5.3) that

$$\alpha_{\text{PI}}\left(\mathbf{x};\mathcal{D}_N,\Phi^{-1}(\gamma)\right) \propto r\left(\mathbf{x};\gamma,\Phi^{-1}(\gamma)\right). \tag{5.12}$$

In step 1, Bergstra et al. [14] resort to optimising $r(\mathbf{x};0,\Phi^{-1}(\gamma))$, which is justified by the fact that

$$r(\mathbf{x};\gamma,\Phi^{-1}(\gamma)) = h_\gamma\left[r\left(\mathbf{x};0,\Phi^{-1}(\gamma)\right)\right],$$

for strictly nondecreasing h_γ. Note we have used a blue and orange colour coding to emphasise the differences in the setting of γ (best viewed on a computer screen). Recall that $\Phi^{-1}(0) = \min_n y_n$ corresponds to the conventional setting of threshold τ. However, make no mistake, for any $\gamma > 0$,

$$\alpha_{\text{PI}}\left(\mathbf{x};\mathcal{D}_N,\Phi^{-1}(0)\right) \not\propto r\left(\mathbf{x};0,\Phi^{-1}(\gamma)\right).$$

Therefore, given the numerical instabilities associated with this approach, as discussed in Section 5.2.4, there is no advantage to be gained from taking this direction. Moreover, Equation (5.12) only holds for $\gamma > 0$. To see this, suppose $\gamma = 0$, which gives

$$\alpha_{\text{PI}}\left(\mathbf{x};\mathcal{D}_N,\Phi^{-1}(0)\right) \propto r\left(\mathbf{x};0,\Phi^{-1}(0)\right).$$

However, since by definition $\ell\left(\mathbf{x};\Phi^{-1}(0)\right)$ has no mass, the RHS is undefined.

5.B CLASS-POSTERIOR PROBABILITY

We provide an unabridged derivation of the identity of Equation (5.7). First, the γ-relative density ratio is given by

$$\begin{aligned} r_\gamma(\mathbf{x}) &\triangleq \frac{\ell(\mathbf{x})}{\gamma\ell(\mathbf{x}) + (1-\gamma)g(\mathbf{x})} \\ &= \frac{p(\mathbf{x} \mid z=1)}{\gamma \cdot p(\mathbf{x} \mid z=1) + (1-\gamma) \cdot p(\mathbf{x} \mid z=0)} \\ &= \left(\frac{p(z=1 \mid \mathbf{x})\cancel{p(\mathbf{x})}}{p(z=1)}\right)\left(\gamma \cdot \frac{p(z=1 \mid \mathbf{x})\cancel{p(\mathbf{x})}}{p(z=1)} + (1-\gamma) \cdot \frac{p(z=0 \mid \mathbf{x})\cancel{p(\mathbf{x})}}{p(z=0)}\right)^{-1}. \end{aligned}$$

By construction, we have $p(z=1) \triangleq p(y \leq \tau) = \gamma$ and $\pi(\mathbf{x}) \triangleq p(z = 1 \mid \mathbf{x})$. Therefore,

$$\begin{aligned} r_\gamma(\mathbf{x}) &= \gamma^{-1}\pi(\mathbf{x})\left(\cancel{\gamma} \cdot \frac{\pi(\mathbf{x})}{\cancel{\gamma}} + \cancel{(1-\gamma)} \cdot \frac{1-\pi(\mathbf{x})}{\cancel{1-\gamma}}\right)^{-1} \\ &= \gamma^{-1}\pi(\mathbf{x}). \end{aligned}$$

Alternatively, we can also arrive at the same result by writing the ordinary density ratio $r_0(\mathbf{x})$ in terms of $\pi(\mathbf{x})$ and γ, which is well-known to be

$$r_0(\mathbf{x}) = \left(\frac{\gamma}{1-\gamma}\right)^{-1} \frac{\pi(\mathbf{x})}{1-\pi(\mathbf{x})}.$$

Plugging this into function h_γ, we get

$$\begin{aligned} r_\gamma(\mathbf{x}) = h_\gamma(r_0(\mathbf{x})) &= h_\gamma\left(\left(\frac{\gamma}{1-\gamma}\right)^{-1} \frac{\pi(\mathbf{x})}{1-\pi(\mathbf{x})}\right) \\ &= \left(\gamma + (1-\gamma)\left(\frac{\gamma}{1-\gamma}\right)\left(\frac{\pi(\mathbf{x})}{1-\pi(\mathbf{x})}\right)^{-1}\right)^{-1} \\ &= \gamma^{-1}\left(1 + \left(\frac{\pi(\mathbf{x})}{1-\pi(\mathbf{x})}\right)^{-1}\right)^{-1} \\ &= \gamma^{-1}\pi(\mathbf{x}). \end{aligned}$$

5.C LOG LOSS

Recall from Section 2.3.2 that the log loss, also known as the binary cross-entropy (BCE) loss, is given by

$$\mathcal{L}(\boldsymbol{\theta}) \triangleq -\beta \cdot \mathbb{E}_{\ell(\mathbf{x})}[\log \pi_{\boldsymbol{\theta}}(\mathbf{x})] - (1-\beta) \cdot \mathbb{E}_{g(\mathbf{x})}[\log(1 - \pi_{\boldsymbol{\theta}}(\mathbf{x}))]. \quad (5.13)$$

The astute reader will have noticed that, unlike before, we've introduced multipliers involving β, which denotes the class balance rate, to account for potential class imbalance. Indeed, we recover the log loss first introduced in Equation (2.11) when $\beta = 1/2$. In particular, let

N_ℓ and N_g be the sizes of the support of $\ell(\mathbf{x})$ and $g(\mathbf{x})$, respectively. Then, we have

$$\beta = \frac{N_\ell}{N}, \qquad \text{and} \qquad 1 - \beta = \frac{N_g}{N},$$

where $N = N_\ell + N_g$. In practice, we approximate the log loss $\mathcal{L}(\boldsymbol{\theta})$ by the empirical risk of Equation (5.8), given by

$$\hat{\mathcal{L}}(\boldsymbol{\theta}) \triangleq -\frac{1}{N}\left(\sum_{n=1}^{N} z_n \log \pi_{\boldsymbol{\theta}}(\mathbf{x}_n) + (1 - z_n) \log\left(1 - \pi_{\boldsymbol{\theta}}(\mathbf{x}_n)\right)\right).$$

In this section, we show that the approximation of Equation (5.9), that is,

$$\pi_{\boldsymbol{\theta}}(\mathbf{x}) \approx \gamma \cdot r_\gamma(\mathbf{x}),$$

attains equality at $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$.

5.C.1 *Optimum*

Taking the functional derivative of $\mathcal{L}^*$ in Equation (5.13), we get

$$\begin{aligned}\frac{\partial \mathcal{L}^*}{\partial \pi_{\boldsymbol{\theta}}} &= -\mathbb{E}_{\ell(\mathbf{x})}\left[\frac{\beta}{\pi_{\boldsymbol{\theta}}(\mathbf{x})}\right] + \mathbb{E}_{g(\mathbf{x})}\left[\frac{1-\beta}{1-\pi_{\boldsymbol{\theta}}(\mathbf{x})}\right] \\ &= \int \left(-\beta \frac{\ell(\mathbf{x})}{\pi_{\boldsymbol{\theta}}(\mathbf{x})} + (1-\beta)\frac{g(\mathbf{x})}{1-\pi_{\boldsymbol{\theta}}(\mathbf{x})}\right) \mathrm{d}\mathbf{x}\end{aligned}$$

This integral evaluates to zero iff the integrand itself evaluates to zero. Hence, we solve the following for $\pi_{\boldsymbol{\theta}^*}(\mathbf{x})$,

$$\beta \frac{\ell(\mathbf{x})}{\pi_{\boldsymbol{\theta}^*}(\mathbf{x})} = (1-\beta)\frac{g(\mathbf{x})}{1-\pi_{\boldsymbol{\theta}^*}(\mathbf{x})}.$$

We re-arrange this expression to give

$$\frac{1-\pi_{\boldsymbol{\theta}^*}(\mathbf{x})}{\pi_{\boldsymbol{\theta}^*}(\mathbf{x})} = \left(\frac{1-\beta}{\beta}\right)\frac{g(\mathbf{x})}{\ell(\mathbf{x})} \qquad \Leftrightarrow \qquad \frac{1}{\pi_{\boldsymbol{\theta}^*}(\mathbf{x})} - 1 = \frac{\beta\ell(\mathbf{x}) + (1-\beta)g(\mathbf{x})}{\beta\ell(\mathbf{x})} - 1.$$

Finally, we add one to both sides and invert the result to give

$$\begin{aligned}\pi_{\boldsymbol{\theta}^*}(\mathbf{x}) &= \frac{\beta\ell(\mathbf{x})}{\beta\ell(\mathbf{x}) + (1-\beta)g(\mathbf{x})} \\ &= \beta \cdot r_\beta(\mathbf{x}).\end{aligned}$$

Since, by definition $\beta = \gamma$, this leads to $\pi_{\boldsymbol{\theta}^*}(\mathbf{x}) = \gamma \cdot r_\gamma(\mathbf{x})$ as required.

5.C.2 *Empirical Risk Minimisation*

For completeness, we show that the log loss $\mathcal{L}(\boldsymbol{\theta})$ of Equation (5.13) can be approximated by $\hat{\mathcal{L}}(\boldsymbol{\theta})$ of Equation (5.8). First, let ρ be the permutation of the set $\{1, \ldots, N\}$, i.e. the bijection from $\{1, \ldots, N\}$

to itself, such that $y_{\rho(n)} \leq \tau$ if $0 < \rho(n) \leq N_\ell$, and $y_{\rho(n)} > \tau$ if $N_\ell < \rho(n) \leq N_g$. That is to say,

$$\mathbf{x}_{\rho(n)} \sim \begin{cases} \ell(\mathbf{x}) & \text{if } 0 < \rho(n) \leq N_\ell, \\ g(\mathbf{x}) & \text{if } N_\ell < \rho(n) \leq N_g. \end{cases} \quad \text{and} \quad z_{\rho(n)} \triangleq \begin{cases} 1 & \text{if } 0 < \rho(n) \leq N_\ell, \\ 0 & \text{if } N_\ell < \rho(n) \leq N_g. \end{cases}$$

Then, we have

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}) &\triangleq -\frac{1}{N}\left(N_\ell \cdot \mathbb{E}_{\ell(\mathbf{x})}[\log \pi_{\boldsymbol{\theta}}(\mathbf{x})] + N_g \cdot \mathbb{E}_{g(\mathbf{x})}[\log(1-\pi_{\boldsymbol{\theta}}(\mathbf{x}))]\right) \\ &\approx -\frac{1}{N}\left(\cancel{N_\ell} \cdot \frac{1}{\cancel{N_\ell}} \sum_{n=1}^{N_\ell} \log \pi_{\boldsymbol{\theta}}(\mathbf{x}_{\rho(n)}) + \cancel{N_g} \cdot \frac{1}{\cancel{N_g}} \sum_{n=N_\ell+1}^{N_g} \log(1-\pi_{\boldsymbol{\theta}}(\mathbf{x}_{\rho(n)}))\right) \\ &= -\frac{1}{N}\left(\sum_{n=1}^{N} z_{\rho(n)} \log \pi_{\boldsymbol{\theta}}(\mathbf{x}_{\rho(n)}) + (1-z_{\rho(n)}) \log(1-\pi_{\boldsymbol{\theta}}(\mathbf{x}_{\rho(n)}))\right) \\ &= -\frac{1}{N}\left(\sum_{n=1}^{N} z_n \log \pi_{\boldsymbol{\theta}}(\mathbf{x}_n) + (1-z_n) \log(1-\pi_{\boldsymbol{\theta}}(\mathbf{x}_n))\right) \triangleq \hat{\mathcal{L}}(\boldsymbol{\theta}), \end{aligned}$$

as required.

5.D IMPLEMENTATION OF BASELINES

The software implementations of the baseline methods considered in our comparisons are described in Table 5.D.1.

Table 5.D.1: Implementations of baseline methods.

Method	Software Library	URL (github.com/*)	Notes
TPE	HyperOpt	hyperopt/hyperopt	
SMAC	SMAC3	automl/SMAC3	
GP-BO	AutoGluon	awslabs/autogluon	in `autogluon.searcher.GPFIFOSearcher`
DE	-	-	custom implementation
RE	NASBench-101	automl/nas_benchmarks	in `experiment_scripts/run_regularized_evolution.py`

5.E EXPERIMENTAL SET-UP AND IMPLEMENTATION DETAILS

HARDWARE. In our experiments, we employ `m4.xlarge` AWS EC2 instances, which have the following specifications:

- **CPU:** Intel(R) Xeon(R) E5-2676 v3 (4 Cores) @ 2.4 GHz
- **Memory:** 16GiB (DDR3)

SOFTWARE. Our method is implemented as a *configuration generator* plug-in in the HpBandSter library of Falkner, Klein, and Hutter [68]. The code will be released as open-source software upon publication.

The implementations of the classifiers on which the proposed variants of BORE are based are described in Table 5.E.1.

Table 5.E.1: Implementations of classifiers.

Model	Software Library	URL
Multi-layer perceptron (MLP)	Keras	`keras.io`
Random forest (RF)	scikit-learn	`scikit-learn.org`
Extreme gradient-boosting (XGBOOST)	XGBoost	`xgboost.readthedocs.io`

We set out with the aim of devising a practical method that is not only agnostic to the of choice classifier, but also robust to underlying implementation details – down to the choice of algorithmic settings. Ideally, any instantiation of BORE should work well out-of-the-box without the need to tweak the sensible default settings that are typically provided by software libraries. Therefore, unless otherwise stated, we emphasise that made no effort was made to adjust any settings and all reported results were obtained using the defaults. For reproducibility, we explicitly enumerate them in turn for each of the proposed variants.

5.E.1 *BORE-RF*

We limit our description to the most salient hyperparameters. We do not deviate from the default settings which, at the time of this writing, are:

- *number of trees* – 100
- *minimum number of samples required to split an internal node* (`min_samples_split`) – 2
- *maximum depth* – unspecified (nodes are expanded until all leaves contain less than `min_samples_split` samples)

5.E.2 *BORE-XGB*

- *number of trees (boosting rounds)* – 100
- *learning rate* (η) – 0.3
- *minimum sum of instance weight (Hessian) needed in a child* (`min_child_weight`) – 1
- *maximum depth* – 6

5.E.3 *BORE-MLP*

In the BORE-MLP variant, the classifier is a MLP with 2 hidden layers, each with 32 units. We consistently found `elu` activations [44] to be particularly effective for lower-dimensional problems, with `relu` remaining otherwise the best choice. We optimise the weights with ADAM [126] using batch size of $B = 64$. For candidate suggestion, we optimise the input of the classifier wrt to its output using multi-started L-BFGS with three random restarts.

EPOCHS PER ITERATION. To ensure the training time on BO iteration N is nonincreasing as a function of N, instead of directly specifying the number of epochs (i. e. full passes over the data), we specify the number of (batch-wise gradient) steps S to train for in each iteration. Since the number of steps per epoch is $M = \lceil N/B \rceil$, the effective number of epochs on the N-th BO iteration is then $E = \lfloor S/M \rfloor$. For example, if $S = 800$ and $B = 64$, the number of epochs for iteration $N = 512$ would be $E = 100$. As another example, for all $0 < N \leq B$ (i. e. we have yet to observe enough data to fill a batch), we have $E = S = 800$. See Figure 5.E.1 for a plot of the effective number of epochs against iterations for different settings of batch size B and number of steps per epoch S. Across all our experiments, we fix $S = 100$.

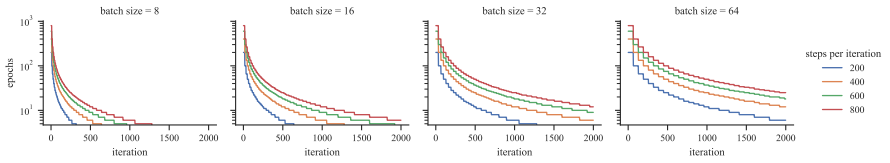


Figure 5.E.1: Effective number of epochs on the nth iteration for different settings of batch size B and number of steps per epoch S.

5.F DETAILS OF BENCHMARKS

5.F.1 *HPOBench*

The hyperparameters for the HPOBench problem and their ranges are summarised in Table 5.F.1. All hyperparameters are discrete – either ordered or unordered. All told, there are $6 \times 2 \times 4 \times 6 \times 2 \times 3 \times 6 \times 2 \times 3 = 66,208$ possible combinations. Further details on this problem can be found in [128].

5.F.2 *NASBench201*

The hyperparameters for the HPOBench problem and their ranges are summarised in Table 5.F.2. The operation associated with each of

Table 5.F.1: Configuration space for HPOBench.

Hyperparameter		Range
Initial learning rate (LR)		$\{5\times10^{-4}, 1\times10^{-3}, 5\times10^{-3}, 1\times10^{-2}, 5\times10^{-2}, 1\times10^{-1}\}$
LR schedule		{`cosine`, `fixed`}
Batch size		$\{2^3, 2^4, 2^5, 2^6\}$
Layer 1	Width	$\{2^4, 2^5, 2^6, 2^7, 2^8, 2^9\}$
	Activation	{`relu`, `tanh`}
	Dropout rate	$\{0.0, 0.3, 0.6\}$
Layer 2	Width	$\{2^4, 2^5, 2^6, 2^7, 2^8, 2^9\}$
	Activation	{`relu`, `tanh`}
	Dropout rate	$\{0.0, 0.3, 0.6\}$

Table 5.F.2: Configuration space for NASBench-201.

Hyperparameter	Range
Arc 0	{`none`, `skip-connect`, `conv-1` $\times$ `1`, `conv-3` $\times$ `3`, `avg-pool-3` $\times$ `3`}
Arc 1	{`none`, `skip-connect`, `conv-1` $\times$ `1`, `conv-3` $\times$ `3`, `avg-pool-3` $\times$ `3`}
Arc 2	{`none`, `skip-connect`, `conv-1` $\times$ `1`, `conv-3` $\times$ `3`, `avg-pool-3` $\times$ `3`}
Arc 3	{`none`, `skip-connect`, `conv-1` $\times$ `1`, `conv-3` $\times$ `3`, `avg-pool-3` $\times$ `3`}
Arc 4	{`none`, `skip-connect`, `conv-1` $\times$ `1`, `conv-3` $\times$ `3`, `avg-pool-3` $\times$ `3`}
Arc 5	{`none`, `skip-connect`, `conv-1` $\times$ `1`, `conv-3` $\times$ `3`, `avg-pool-3` $\times$ `3`}

the $\binom{4}{2} = 6$ arcs can belong to one of five categories. Hence, there are $5^6 = 15,625$ possible combinations of hyperparameter configurations. Further details on this problem can be found in [60].

5.F.3 *Robot pushing control*

This problem is concerned with tuning the controllers of two robot hands, with the goal of each pushing an object to some prescribed goal location $\mathbf{p}_g^{(1)}$ and $\mathbf{p}_g^{(2)}$, respectively. Let $\mathbf{p}_s^{(1)}$ and $\mathbf{p}_s^{(2)}$ denote the specified starting positions, and $\mathbf{p}_f^{(1)}$ and $\mathbf{p}_f^{(2)}$ the final positions (the latter of which are functions of the control parameters $\mathbf{x}$). The reward is defined as

$$R(\mathbf{x}) \triangleq \underbrace{\|\mathbf{p}_g^{(1)} - \mathbf{p}_s^{(1)}\| + \|\mathbf{p}_g^{(2)} - \mathbf{p}_s^{(2)}\|}_{\text{initial distances}} - \underbrace{(\|\mathbf{p}_g^{(1)} - \mathbf{p}_f^{(1)}\| + \|\mathbf{p}_g^{(2)} - \mathbf{p}_f^{(2)}\|)}_{\text{final distances}},$$

which effectively quantifies the amount of progress made toward pushing the objects to the desired goal. For each robot, the control parameters include the location and orientation of its hands, the pushing speed, moving direction and push duration. These parameters and their ranges are summarised in Table 5.F.3.

Further details on this problem can be found in [278]. This simulation is implemented with the Box2D library, and the associated code repository can be found at `https://github.com/zi-w/Ensemble-Bayesian-Optimisation`.

Table 5.F.3: Configuration space for the robot pushing control problem.

Hyperparameter		Range
Robot 1	Position x	$[-5, 5]$
	Position y	$[-5, 5]$
	Angle θ	$[0, 2\pi]$
	Velocity v_x	$[-10, 10]$
	Velocity v_y	$[-10, 10]$
	Push Duration	$[2, 30]$
	Torque	$[-5, 5]$
Robot 2	Position x	$[-5, 5]$
	Position y	$[-5, 5]$
	Angle θ	$[0, 2\pi]$
	Velocity v_x	$[-10, 10]$
	Velocity v_y	$[-10, 10]$
	Push Duration	$[2, 30]$
	Torque	$[-5, 5]$

5.F.4 *Racing Line Optimisation*

This problem is concerned with finding the optimal racing line. Namely, given a racetrack and a vehicle with known dynamics, the task is to determine a trajectory around the track for which the minimum time required to traverse it is minimal. We adopt the set-up of Jain and Morari [113], who consider 1:10 and 1:43 scale miniature remote-controlled cars traversing tracks at UC Berkeley [148] and ETH Zürich [220], respectively.

The trajectory is represented by a cubic spline parameterised by the 2D coordinates of D waypoints, each placed at locations along the length of the track, where the ith waypoint deviates from the centerline of the track by $x_i \in \left[-\frac{W}{2}, \frac{W}{2}\right]$, for some track width W. Hence, the parameters are the distances by which each waypoint deviates from the centerline, $\mathbf{x} = [x_1 \cdots x_D]^\top$.

Our blackbox function of interest, namely, the minimum time to traverse a given trajectory, is determined by the solution to a convex optimisation problem involving partial differential equations (PDEs) [149]. Further details on this problem can be found in [113], and the associated code repository can be found at `https://github.com/jainachin/bayesrace`.

5.G PARAMETERS, HYPERPARAMETERS, AND META-HYPERPARAMETERS

We explicitly identify the parameters $\boldsymbol{\omega}$, hyperparameters $\boldsymbol{\theta}$, and meta-hyperparameters $\boldsymbol{\lambda}$ in our approach, making clear their distinction, examining their roles in comparison with other methods and discuss their treatment.

Table 5.G.1: A taxonomy of parameters, hyperparameters, and meta-hyperparameters.

	BO with Gaussian processes (GPs)	BO with Bayesian neural networks (BNNs)	BORE with neural networks (NNs)
Meta-hyperparameters $\boldsymbol{\lambda}$	kernel family, kernel isotropy (ARD), etc..	layer depth, widths, activations, etc..	prior precision α, likelihood precision β, layer depth, widths, activations, etc..
Hyperparameters $\boldsymbol{\theta}$	kernel lengthscale and amplitude, ℓ and σ, likelihood precision β	prior precision α, likelihood precision β	weights $\mathbf{W}$, biases $\mathbf{b}$
Parameters $\boldsymbol{\omega}$	None $\varnothing$ (nonparametric)	weights $\mathbf{W}$, biases $\mathbf{b}$	None $\varnothing$ (by design)

5.G.1 *Parameters*

Since we seek to *directly* approximate the acquisition function, our method is by design free of parameters $\boldsymbol{\omega}$. By contrast, in classical BO, the acquisition function is derived from the analytical properties of the posterior predictive $p(y \mid \mathbf{x}, \boldsymbol{\theta}, \mathcal{D}_N)$. To compute this, the uncertainty about parameters $\boldsymbol{\omega}$ must be marginalised out

$$p(y \mid \mathbf{x}, \boldsymbol{\theta}, \mathcal{D}_N) = \int p(y \mid \mathbf{x}, \boldsymbol{\omega}, \boldsymbol{\theta}) p(\boldsymbol{\omega} \mid \boldsymbol{\theta}, \mathcal{D}_N) \, \mathrm{d}\boldsymbol{\omega}, \tag{5.14}$$

where

$$p(\boldsymbol{\omega} \mid \boldsymbol{\theta}, \mathcal{D}_N) = \frac{p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\omega}, \boldsymbol{\theta}) p(\boldsymbol{\omega} \mid \boldsymbol{\theta})}{p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta})}.$$

While GPs are free of parameters, the latent function values $\mathbf{f}$ must be marginalised out

$$p(y \mid \mathbf{x}, \boldsymbol{\theta}, \mathcal{D}_N) = \int p(y \mid \mathbf{x}, \mathbf{f}, \boldsymbol{\theta}) p(\mathbf{f} \mid \boldsymbol{\theta}, \mathcal{D}_N) \, \mathrm{d}\mathbf{f}.$$

In the case of GP regression, this is easily computed by applying straightforward rules of Gaussian conditioning. Unfortunately, few other models enjoy this convenience.

CASE STUDY: As a concrete example, consider BNNs. The parameters $\boldsymbol{\omega}$ consist of the weights $\mathbf{W}$ and biases $\mathbf{b}$ in the NN, while the hyperparameters $\boldsymbol{\theta}$ consist of the prior and likelihood precisions, α and β, respectively. In general, $p(\boldsymbol{\omega} \mid \mathcal{D}_N, \boldsymbol{\theta})$ is not analytically tractable.

- To work around this, DNGO [237] and ABLR [200] both constrain the parameters $\boldsymbol{\omega}$ to include the weights and biases of only the *final* layer, $\mathbf{W}_L$ and $\mathbf{b}_L$, and relegate those of all preceding layers, $\mathbf{W}_{1:L-1}$ and $\mathbf{b}_{1:L-1}$, to the hyperparameters $\boldsymbol{\theta}$. This yields an exact (Gaussian) expression for $p(\boldsymbol{\omega} \mid \mathcal{D}_N, \boldsymbol{\theta})$ and $p(y \mid \mathbf{x}, \boldsymbol{\theta}, \mathcal{D}_N)$. To treat the hyperparameters, Perrone et al. [200] estimate $\mathbf{W}_{1:L-1}$,

$\mathbf{b}_{1:L-1}$, α and β using type-II MLE, while Snoek et al. [237] use a combination of type-II MLE and slice sampling [186].

- In contrast, BOHAMIANN [243] makes no such simplifying distinctions regarding the layer weights and biases. Consequently, they must resort to sampling-based approximations of $p(\boldsymbol{\omega} \mid \boldsymbol{\theta}, \mathcal{D}_N)$, in their case by adopting SGHMC [33].

In both approaches, compromises needed to be made in order to negotiate the computation of $p(\boldsymbol{\omega} \mid \boldsymbol{\theta}, \mathcal{D}_N)$. This is not to mention the problem of computing the posterior over the *hyperparameters* $p(\boldsymbol{\theta} \mid \mathcal{D}_N)$, which we discuss next. In contrast, BORE avoids the problems associated with computing the posterior predictive $p(y \mid \mathbf{x}, \boldsymbol{\theta}, \mathcal{D}_N)$, and, by extension, the posterior $p(\boldsymbol{\omega} \mid \boldsymbol{\theta}, \mathcal{D}_N)$ of Equation (5.14). Therefore, such compromises are simply unnecessary.

5.G.2 *Hyperparameters*

For the sake of notational simplicity, we have thus far not been explicit about how the acquisition function depends on the hyperparameters $\boldsymbol{\theta}$ and how they are handled. We first discuss generically how hyperparameters $\boldsymbol{\theta}$ are treated in BO. Refer to [228] for a full discussion. In particular, we rewrite the acquisition function, expressed in Equation (2.45), to explicitly include $\boldsymbol{\theta}$

$$\alpha(\mathbf{x}; \boldsymbol{\theta}, \mathcal{D}_N, \tau) \triangleq \mathbb{E}_{p(y \mid \boldsymbol{\theta}, \mathbf{x}, \mathcal{D}_N)}[U(y; \tau)].$$

MARGINAL ACQUISITION FUNCTION. Ultimately, one wishes to maximise the *marginal* acquisition function $A(\mathbf{x}; \mathcal{D}_N, \tau)$, which marginalises out the uncertainty about the hyperparameters,

$$A(\mathbf{x}; \mathcal{D}_N, \tau) = \int \alpha(\mathbf{x}; \boldsymbol{\theta}, \mathcal{D}_N, \tau) p(\boldsymbol{\theta} \mid \mathcal{D}_N) \, \mathrm{d}\boldsymbol{\theta},$$

where

$$p(\boldsymbol{\theta} \mid \mathcal{D}_N) = \frac{p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathbf{y} \mid \mathbf{X})}.$$

This consists of an expectation over the posterior $p(\boldsymbol{\theta} \mid \mathcal{D}_N)$ which is, generally speaking, analytically intractable. In practice, the most straightforward way to compute $A(\mathbf{x}; \mathcal{D}_N, \tau)$ is to approximate the posterior using a delta measure centered at some point estimate $\hat{\boldsymbol{\theta}}$, either the type-II MLE $\hat{\boldsymbol{\theta}}_{\text{MLE}}$ or the MAP estimate $\hat{\boldsymbol{\theta}}_{\text{MAP}}$. This leads to

$$A(\mathbf{x}; \mathcal{D}_N, \tau) \approx \alpha(\mathbf{x}; \hat{\boldsymbol{\theta}}, \mathcal{D}_N, \tau).$$

Suffice it to say, sound uncertainty quantification is paramount to guiding exploration. Since point estimates fail to capture uncertainty about hyperparameters $\boldsymbol{\theta}$, it is often beneficial to turn instead to MC estimation [236]

$$A(\mathbf{x}; \mathcal{D}_N, \tau) \approx \frac{1}{S} \sum_{s=1}^{S} \alpha(\mathbf{x}; \boldsymbol{\theta}^{(s)}, \mathcal{D}_N, \tau), \qquad \boldsymbol{\theta}^{(s)} \sim p(\boldsymbol{\theta} \mid \mathcal{D}_N).$$

MARGINAL CLASS-POSTERIOR PROBABILITIES. Recall that the likelihood of our model is

$$p(z \mid \mathbf{x}, \boldsymbol{\theta}) \triangleq \text{Bern}(z \mid \pi_{\boldsymbol{\theta}}(\mathbf{x})),$$

or more succinctly $\pi_{\boldsymbol{\theta}}(\mathbf{x}) = p(z = 1 \mid \mathbf{x}, \boldsymbol{\theta})$. We specify a prior $p(\boldsymbol{\theta})$ on hyperparameters $\boldsymbol{\theta}$ and marginalise out its uncertainty to produce our analog to the marginal acquisition function

$$\Pi(\mathbf{x}; \mathcal{D}_N) = \int \pi_{\boldsymbol{\theta}}(\mathbf{x}) p(\boldsymbol{\theta} \mid \mathcal{D}_N) \, \mathrm{d}\boldsymbol{\theta},$$

where

$$p(\boldsymbol{\theta} \mid \mathcal{D}_N) = \frac{p(\mathbf{z} \mid \mathbf{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathbf{z} \mid \mathbf{X})}.$$

As in the generic case, we are ultimately interested in maximising the marginal class-posterior probabilities $\Pi(\mathbf{x}; \mathcal{D}_N)$. However, much like $A(\mathbf{x}; \mathcal{D}_N, \tau)$, the marginal $\Pi(\mathbf{x}; \mathcal{D}_N)$ is analytically intractable in turn due to the intractability of $p(\boldsymbol{\theta} \mid \mathcal{D}_N)$. In this work, we focus on minimising the log loss of Equation (5.8), which is proportional to the negative log-likelihood

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{n=1}^{N} \log p(z_n \mid \mathbf{x}_n, \boldsymbol{\theta}) \propto -\log p(\mathbf{z} \mid \mathbf{X}, \boldsymbol{\theta}).$$

Therefore, we're effectively performing the equivalent of type-II MLE,

$$\hat{\boldsymbol{\theta}}_{\text{MLE}} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \log p(\mathbf{z} \mid \mathbf{X}, \boldsymbol{\theta}).$$

In the interest of improving exploration and, of particular interest in our case, calibration of class-membership probabilities, it may be beneficial to consider MC and other approximate inference methods [18, 74, 136]. This remains fertile ground for future work.

5.G.3 *Meta-hyperparameters*

In the case of BORE-MLP, the meta-hyperparameters might consist of, e. g., layer depth, widths, activations, etc. – the tuning of which is often the reason one appeals to BO in the first place. For improvements in calibration, and therefore sample diversity, it may be beneficial to marginalise out the uncertainty about these, or considering some approximation thereof, such as hyper-deep ensembles [280].

6 CONCLUSION

Through this thesis, we have sought to advance the integration between deep learning and probabilistic ML, with a focus on Gaussian processes (GPs) and Bayesian optimisation (BO). In this chapter, we reflect on our main contributions, discuss directions for future work, and conclude with a few parting words.

6.1 SUMMARY OF CONTRIBUTIONS

First, in Chapter 3, we improved upon prior work hyperspherical sparse GP approximation that uses nonlinear activations as inter-domain features, known as the ACTIVATED SVGP [66, 252], in which a single-layer feed forward NN emerges from the posterior predictive distribution. We provided an analysis of the limitations of this approach which *inter alia* preclude the use of widely-used covariance functions and nonlinear activations. Our key contribution was extending the orthogonally-decoupled sparse GP approximation to accomodate inter-domain features. We demonstrated that the combination of orthogonal inducing points and spherical activation features effectively mitigates the earlier limitations, not only bringing their predictive performance closer to NNs, but achieving superior scalability over alternatives.

Second, in Chapter 4, we provided an interpretation of CYCLEGANS as a Bayesian framework for inferring the hidden representations of entities from one domain as entities in another. Specifically, we framed the problem of learning cross-domain correspondences without paired data as inference in a LVM. First, we introduced the implicit LVM, where the prior over hidden representations is specified flexibly as an implicit distribution. We then introduced a new VI framework that differs from traditional VI in that it directly approximates the joint distribution based on a symmetrised KL divergence. Finally we showed that CYCLEGANS emerges as a variant of this framework, casting new light on this powerful class of deep generative models for image style transfer.

Third, in Chapter 5, we introduced a reformulation of BO based on solving a binary classification problem. We leveraged the connections between the improvement-based acquisition functions, density-ratio estimation (DRE), and class-probability estimation (CPE), to derive a binary classifier of candidate solutions that effectively serves as the acquisition function. By doing away with an explicit probabilistic model of the objective function, we eliminated the impediments posed by tractability requirements, enabling the seamless integration of

deep learning and other powerful modelling paradigms in a manner that does not necessitate approximations or compromise scalability and representational capacity. Overall, this model-agnostic framework substantially expands the applicability of BO to diverse, challenging optimisation problem scenarios.

6.2 FUTURE DIRECTIONS

Looking to the future, promising new research avenues emerge that expand on the contributions made in this thesis.

ORTHOGONAL INTER-DOMAIN INDUCING FEATURES. Future work should explore additional combinations of inter-domain inducing features in the standard and orthogonal bases beyond the traditional inducing points and the spherical NN activation features examined here. In particular, since the orthogonal decoupling of GPS can be seen as a way to leverage different bases to separately represent the predictive mean and variance [223], it is promising to tap into the strengths of the spherical NN activation features [66, 252] and spherical harmonics features [65] to independently capture the predictive mean and variance, respectively. More broadly, to better enable exploration in this direction and accomodate the composition of various inter-domain inducing features, the software developed for this research should be refactored based on principles of modularity and separation of concerns. Adopting this more flexible and extensible design approach will facilitate seamless experimentation with combinations of diverse inter-domain inducing features in orthogonally-decoupled sparse GP approximations.

BORE BY DIRECT DRE. Future work should explore the potential benefits of other direct DRE methods. While the CPE approach is a big improvement from the problematic TPE approach, it is still thought to be a simple baseline in the DRE literature. The RULSIF [292] method may be of particular interest, not least because it is the only method of those discussed in Section 2.3 that directly estimates the *relative* density-ratio. Furthermore, since RULSIF is parameterised by a sum of Gaussian kernels, it enables the use of well-established mode-finding approaches, such as the *mean-shift* algorithm [45], for candidate suggestion. Finally, along the same avenue but in the opposite direction, one may also consider employing other DRE losses [170] for classifier learning, which would accomodate the use of powerful deep learning models.

EXTENDED BO BY CLASSIFICATION. Future work should explore extending BORE with model architectures suited for complex real-world optimisation problems. These "exotic" problems [71] include scenarios

where the function involve *multiple outputs*, e. g., in multi-task, multi-fidelity, and multi-objective problems, where simple feed-forward NNs may be advantageous. They also include problems with *structured* or *sequential inputs*, where graph neural networks (GNNs) or Transformer architectures [274], respectively, may prove beneficial. By upgrading BORE to handle these complex optimisation problems with multiple outputs, structured inputs, and sequential inputs, it would become applicable to a wider range of challenging real-world tasks. Ultimately, the flexibility of the BORE framework means its capabilities continue to grow as researchers creatively integrate it with state-of-the-art modelling paradigms.

6.3 FINAL REFLECTION

Overall, this thesis has laid the necessary groundwork, improved existing frameworks, and offered new perspectives on Bayesian optimisation, Gaussian processes, and deep learning. Our contributions advance the integration between deep learning and probabilistic ML, aiming to make decision support systems more capable, dependable, and equipped to handle the complex, dynamic, and uncertain challenges of the modern world.

In closing, we envision a future where the interplay between deep learning and probabilistic ML continues to evolve, leading to novel applications and breakthroughs that benefit society across a wide range of domains. The quest to realise the grand vision of AI – developing intelligent systems that can perceive, learn, decide, and act autonomously in complex real-world environments remains ongoing, and we are excited to contribute to this journey with our work. As the AI landscape continues to transform, the fusion between deep and probabilistic learning will undoubtedly play a pivotal role in shaping the AI of tomorrow.

A

NUMERICAL METHODS FOR IMPROVED DECOUPLED SAMPLING OF GAUSSIAN PROCESSES

A.1 INTRODUCTION

Sampling from Gaussian processes (GPs) is not only crucial in its own right but also plays a pivotal role in various downstream tasks, notably Thompson sampling [258], as we detailed in Section 2.5.2.4.

Using the standard approach, the computational cost scales cubically with the number of test points. Moreover, samples obtained through this method cannot be straightforwardly evaluated at arbitrary inputs, let alone optimised. To address these challenges, a common strategy involves utilising the weight-space approximation of GPs based on their spectral decomposition. However, this introduces its own issues, particularly when the number of training observations increases, leading to erratic extrapolations [29, 183, 278].

Recent work has proposed a hybrid approach that leverages a simple, effective, yet underutilised method for sampling from Gaussian conditionals [289, 291]. This method enables the combined use of the canonical basis and the spectral basis (also known as Fourier features), to generate samples efficiently. Notably, these samples can be obtained with a linear cost in the number of test points, and are easy to evaluate and optimise.

In existing works, the frequencies are selected through a straightforward MC approximation scheme. In this chapter, we explore the use of various numerical integration techniques to improve upon the selection mechanism. We provide a concise overview of approaches considered for the Fourier feature decomposition of stationary kernels, comparing their effectiveness in approximating the kernel matrix. Subsequently, we introduce variations to existing schemes, extending the applicability of decompositions to kernel classes beyond the SE kernel. We highlight a critical limitation in an existing class of schemes based on Gaussian quadrature when dealing with kernels with small lengthscales. Specifically, small lengthscales result in highly oscillatory integrals that pose challenges for estimation through numerical methods. To address this, we consider a previously untapped technique based on an extension of Newton-Cotes quadrature. Finally, we evaluate how the Fourier feature decompositions derived from the various numerical integration schemes impact the fidelity of the GP posterior samples.

A.2 DECOUPLED SAMPLING OF GAUSSIAN PROCESSES

We give a brief overview of the method proposed by Wilson et al. [289]. Recall that for practical purposes, a GP posterior at T query locations is simply a T-dimensional conditional Gaussian distribution. In general, consider jointly Gaussian random variables $\mathbf{a} \in \mathbb{R}^T$ and $\mathbf{b} \in \mathbb{R}^M$,

$$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_\mathbf{a} \\ \boldsymbol{\mu}_\mathbf{b} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{aa}} & \boldsymbol{\Sigma}_{\mathbf{ab}} \\ \boldsymbol{\Sigma}_{\mathbf{ba}} & \boldsymbol{\Sigma}_{\mathbf{bb}} \end{bmatrix} \right).$$

The distribution of $\mathbf{a}$ conditioned on $\mathbf{b} = \boldsymbol{\beta}$ is given by

$$p(\mathbf{a} \mid \mathbf{b} = \boldsymbol{\beta}) = \mathcal{N}(\mathbf{a} \mid \boldsymbol{\mu}_{\mathbf{a}|\mathbf{b}}, \boldsymbol{\Sigma}_{\mathbf{a}|\mathbf{b}}),$$

where the mean and covariance are given by

$$\boldsymbol{\mu}_{\mathbf{a}|\mathbf{b}} \triangleq \boldsymbol{\mu}_\mathbf{a} + \boldsymbol{\Sigma}_{\mathbf{ab}}\boldsymbol{\Sigma}_{\mathbf{bb}}^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}_\mathbf{b}), \quad \text{and} \quad \boldsymbol{\Sigma}_{\mathbf{a}|\mathbf{b}} \triangleq \boldsymbol{\Sigma}_{\mathbf{aa}} - \boldsymbol{\Sigma}_{\mathbf{ab}}\boldsymbol{\Sigma}_{\mathbf{bb}}^{-1}\boldsymbol{\Sigma}_{\mathbf{ba}}.$$

The standard approach to generating samples from $p(\mathbf{a} \mid \mathbf{b} = \boldsymbol{\beta})$ is to use a location-scale transform of normal random variables, i. e.,

$$\mathbf{a} = \boldsymbol{\mu}_{\mathbf{a}|\mathbf{b}} + \boldsymbol{\Sigma}_{\mathbf{a}|\mathbf{b}}^{1/2}\boldsymbol{\eta}, \quad \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad \Leftrightarrow \quad \mathbf{a} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{a}|\mathbf{b}}, \boldsymbol{\Sigma}_{\mathbf{a}|\mathbf{b}}),$$

where $\boldsymbol{\Sigma}_{\mathbf{a}|\mathbf{b}}^{1/2}$ denotes the Cholesky factor of $\boldsymbol{\Sigma}_{\mathbf{a}|\mathbf{b}}$, whose calculation has a cost of $\mathcal{O}(T^3)$, and is precisely what makes the standard approach so computationally expensive.

Matheron's rule

A powerful alternative for sampling conditional Gaussian variables is Matheron's rule [120],

$$(\mathbf{a} \mid \mathbf{b} = \boldsymbol{\beta}) \overset{D}{=} \mathbf{a} + \boldsymbol{\Sigma}_{\mathbf{ab}}\boldsymbol{\Sigma}_{\mathbf{bb}}^{-1}(\boldsymbol{\beta} - \mathbf{b}), \tag{A.1}$$

where $\overset{D}{=}$ denotes equality *in distribution*. This is straightforward to verify. By computing the mean and covariance of this expression, we get

$$\mathbb{E}[\mathbf{a} + \boldsymbol{\Sigma}_{\mathbf{ab}}\boldsymbol{\Sigma}_{\mathbf{bb}}^{-1}(\boldsymbol{\beta} - \mathbf{b})] = \boldsymbol{\mu}_\mathbf{a} + \boldsymbol{\Sigma}_{\mathbf{ab}}\boldsymbol{\Sigma}_{\mathbf{bb}}^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}_\mathbf{b}) = \boldsymbol{\mu}_{\mathbf{a}|\mathbf{b}}$$

and

$$\begin{aligned} &\operatorname{Cov}[\mathbf{a} + \boldsymbol{\Sigma}_{\mathbf{ab}}\boldsymbol{\Sigma}_{\mathbf{bb}}^{-1}(\boldsymbol{\beta} - \mathbf{b})] \\ &\quad = \boldsymbol{\Sigma}_{\mathbf{aa}} - 2\boldsymbol{\Sigma}_{\mathbf{ab}}\boldsymbol{\Sigma}_{\mathbf{bb}}^{-1}\boldsymbol{\Sigma}_{\mathbf{ba}} + \boldsymbol{\Sigma}_{\mathbf{ab}}\boldsymbol{\Sigma}_{\mathbf{bb}}^{-1}\boldsymbol{\Sigma}_{\mathbf{bb}}\boldsymbol{\Sigma}_{\mathbf{bb}}^{-1}\boldsymbol{\Sigma}_{\mathbf{ba}} \\ &\quad = \boldsymbol{\Sigma}_{\mathbf{aa}} - \boldsymbol{\Sigma}_{\mathbf{ab}}\boldsymbol{\Sigma}_{\mathbf{bb}}^{-1}\boldsymbol{\Sigma}_{\mathbf{ba}} = \boldsymbol{\Sigma}_{\mathbf{a}|\mathbf{b}} \end{aligned}$$

respectively. Recall from Section 2.4.1 that a GP is a random function such that, at a finite set of locations $\mathbf{X}_*$, the vector $\mathbf{f}_* = f(\mathbf{X}_*)$ follows a Gaussian distribution. Specifically, if $f(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'))$, then $\mathbf{f}_* \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{**})$ where $\mathbf{K}_{**} \triangleq k(\mathbf{X}_*, \mathbf{X}_*)$ for some covariance function k. Further recall from Equation (2.16) that the posterior of an exact GP at

test locations $\mathbf{X}_*$ given N observations $\mathbf{y}$ is $p(\mathbf{f}_* \mid \mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}_{*|\mathbf{y}}, \boldsymbol{\Sigma}_{**|\mathbf{y}})$, where

$$\begin{aligned} \boldsymbol{\mu}_{*|\mathbf{y}} &\triangleq \mathbf{K}_{*\mathbf{f}}(\mathbf{K}_{\mathbf{ff}} + \beta^{-1}\mathbf{I})^{-1}\mathbf{y}, \\ \boldsymbol{\Sigma}_{**|\mathbf{y}} &\triangleq \mathbf{K}_{**} - \mathbf{K}_{*\mathbf{f}}(\mathbf{K}_{\mathbf{ff}} + \beta^{-1}\mathbf{I})^{-1}\mathbf{K}_{\mathbf{f}*}, \end{aligned} \tag{A.2}$$

and from Equation (2.22) that the conditional distribution of SVGP models at test locations $\mathbf{X}_*$ given inducing variables $\mathbf{u} \sim p(\mathbf{u})$ is $p(\mathbf{f}_* \mid \mathbf{u}) = \mathcal{N}(\boldsymbol{\mu}_{*|\mathbf{u}}, \boldsymbol{\Sigma}_{**|\mathbf{u}})$, where

$$\begin{aligned} \boldsymbol{\mu}_{*|\mathbf{u}} &\triangleq \mathbf{K}_{*\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{u}, \\ \boldsymbol{\Sigma}_{**|\mathbf{u}} &\triangleq \mathbf{K}_{**} - \mathbf{K}_{*\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}*}. \end{aligned} \tag{A.3}$$

Applying Matheron's rule from Equation (A.1) to these conditionals, we have, for exact GPs,

$$(\mathbf{f}_* \mid \mathbf{y}) \overset{D}{=} \mathbf{f}_* + \mathbf{K}_{*\mathbf{f}}(\mathbf{K}_{\mathbf{ff}} + \beta^{-1}\mathbf{I})^{-1}(\mathbf{y} - \mathbf{f}_N - \boldsymbol{\epsilon}),$$

and, for sparse GPs,

$$(\mathbf{f}_* \mid \mathbf{u}) \overset{D}{=} \mathbf{f}_* + \mathbf{K}_{*\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}(\mathbf{u} - \mathbf{f}_M),$$

where $(\mathbf{f}_*, \mathbf{f}_N)$ and $(\mathbf{f}_*, \mathbf{f}_M)$ respectively are jointly sampled from the GP prior. The astute reader will recognise the absurdity of this approach, as it is in fact considerably more expensive than the conventional one. Specifically, jointly sampling from the prior incurs costs of $\mathcal{O}\left((T+N)^3\right)$ and $\mathcal{O}\left((T+M)^3\right)$, respectively. As we shall see, this is the paradox that Wilson et al. [289] managed to resolve.

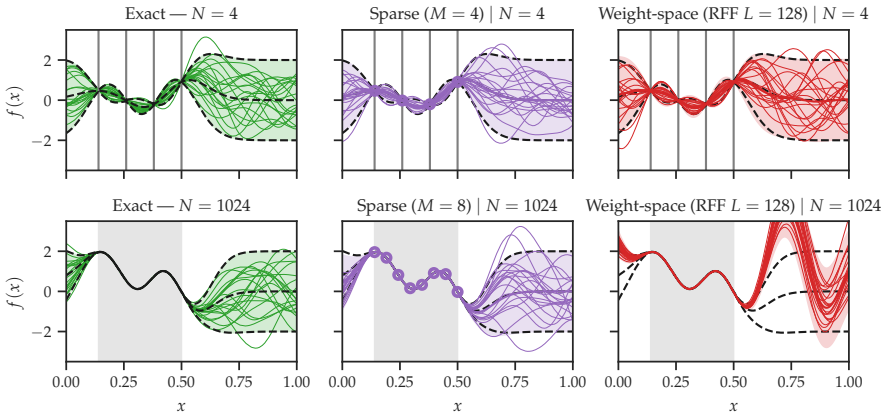


Figure A.1: An illustration of the *variance starvation* phenomenon. Across the columns, we have a comparison of various GP posteriors and their samples, given $n = 4$ (top) and $n = 1,024$ (bottom) observations at locations indicated by the shaded regions. A reproduction of the figures originating from Wilson et al. [289].

Let's consider the weight-space approximation described in Section 2.4.3. Recall from Equation (2.37) that the posterior weight density is $p(\mathbf{w} \mid \mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{w}|\mathbf{y}}, \boldsymbol{\Sigma}_{\mathbf{w}|\mathbf{y}})$, where

$$\begin{aligned}\boldsymbol{\mu}_{\mathbf{w}|\mathbf{y}} &\triangleq (\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \beta^{-1}\mathbf{I})^{-1} \boldsymbol{\Phi}^\top \mathbf{y}, \\ \boldsymbol{\Sigma}_{\mathbf{w}|\mathbf{y}} &\triangleq \beta^{-1} (\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \beta^{-1}\mathbf{I})^{-1}.\end{aligned}$$

Applying Matheron's rule of Equation (A.1) to the weight-space posterior, we have

$$(\mathbf{w} \mid \mathbf{y}) \overset{D}{=} \mathbf{w} + \boldsymbol{\Phi}^\top (\boldsymbol{\Phi}\boldsymbol{\Phi}^\top + \beta^{-1}\mathbf{I})^{-1} (\mathbf{y} - \boldsymbol{\Phi}\mathbf{w} - \boldsymbol{\epsilon}).$$

While possible to sample from efficiently, as alluded to previously, this approach is beset by the general limited expressiveness of finite-dimensional feature maps, which can hamper ability to extrapolate predictions at test time. In particular, for Fourier feature decompositions, this is a phenomenon known as *variance starvation*, whereby extrapolations become erratic as the number of observations N increases. The intuition behind this is that although the Fourier basis is suited for representing stationary GPs, the posterior is generally non-stationary. See Figure A.1 for an illustration of the variance starvation phenomenon.

variance starvation

Wilson et al. [289] seek to combine the best of both worlds, by leveraging the strength of the Fourier basis $\boldsymbol{\phi}(\,\cdot\,)$ at representing stationary priors [206], and the strength of the canonical basis $k(\,\cdot\,, \mathbf{z})$ at representing the data [24].

The decoupled sampling approach for sparse GPs is

$$(\mathbf{f}_* \mid \mathbf{u}) \overset{D}{\approx} \boldsymbol{\Phi}_* \mathbf{w} + \mathbf{K}_{*\mathbf{u}} \mathbf{K}_{\mathbf{uu}}^{-1} (\mathbf{u} - \boldsymbol{\Phi}\mathbf{w}), \tag{A.4}$$

and, for exact GPs, is

$$(\mathbf{f}_* \mid \mathbf{y}) \overset{D}{\approx} \boldsymbol{\Phi}_* \mathbf{w} + \mathbf{K}_{*\mathbf{f}} (\mathbf{K}_{\mathbf{ff}} + \beta^{-1}\mathbf{I})^{-1} (\mathbf{y} - \boldsymbol{\Phi}\mathbf{w} - \boldsymbol{\epsilon}). \tag{A.5}$$

It's important to emphasise that these are in fact only *approximately* equal in distribution. To understand precisely how they differ, let us compute their moments. We focus on the case of sparse GPs in Equation (A.4), the mean and covariance of which are

$$\mathbb{E}[\boldsymbol{\Phi}_* \mathbf{w} + \mathbf{K}_{*\mathbf{u}} \mathbf{K}_{\mathbf{uu}}^{-1} (\mathbf{u} - \boldsymbol{\Phi}\mathbf{w})] = \mathbf{K}_{*\mathbf{u}} \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{u} = \boldsymbol{\mu}_{*|\mathbf{u}}$$

and

$$\begin{aligned}&\operatorname{Cov}[\boldsymbol{\Phi}_* \mathbf{w} + \mathbf{K}_{*\mathbf{u}} \mathbf{K}_{\mathbf{uu}}^{-1} (\mathbf{u} - \boldsymbol{\Phi}\mathbf{w})] \\ &\quad = \boldsymbol{\Phi}_* \boldsymbol{\Phi}_*^\top - 2\mathbf{K}_{*\mathbf{u}} \mathbf{K}_{\mathbf{uu}}^{-1} \boldsymbol{\Phi}\boldsymbol{\Phi}_*^\top + \mathbf{K}_{*\mathbf{u}} \mathbf{K}_{\mathbf{uu}}^{-1} \boldsymbol{\Phi}\boldsymbol{\Phi}^\top \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{K}_{\mathbf{u}*} \\ &\quad \approx \mathbf{K}_{**} - \mathbf{K}_{*\mathbf{u}} \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{K}_{\mathbf{u}*} = \boldsymbol{\Sigma}_{**|\mathbf{u}}\end{aligned} \tag{A.6}$$

We see that mean is exactly equal to the $\boldsymbol{\mu}_{*|\mathbf{u}}$ of Equation (A.3), but the covariance is only equal to $\boldsymbol{\Sigma}_{**|\mathbf{u}}$ if

$$\boldsymbol{\Phi}_* \boldsymbol{\Phi}_*^\top = \mathbf{K}_{**}, \quad \boldsymbol{\Phi}\boldsymbol{\Phi}_*^\top = \mathbf{K}_{\mathbf{u}*}, \quad \text{and} \quad \boldsymbol{\Phi}\boldsymbol{\Phi}^\top = \mathbf{K}_{\mathbf{uu}},$$

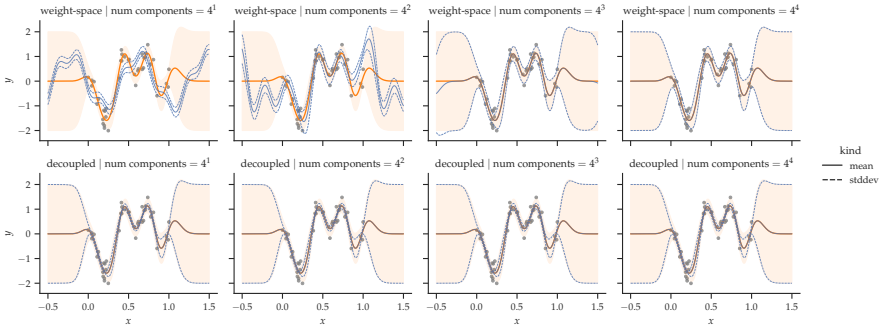


Figure A.2: Posterior predictive distributions.

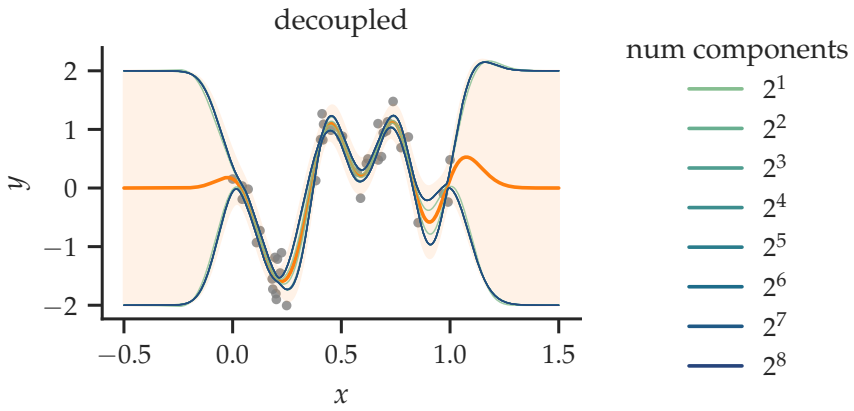


Figure A.3: Posterior predictive distributions from the decoupled approach, overlayed on top of one another.

which are satisfied when $k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\phi}(\mathbf{x}')$ for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$. In other words, we have equality in distribution in Equations (A.4) and (A.5) when the kernel approximation is exact. Thus seen, the quality of decoupled pathwise samples relies crucially on the quality of the kernel approximation in Equation (2.38) itself. In this chapter, we explore various methods from the classical literature on numerical integration [51] to tighten this approximation.

A.3 NUMERICAL INTEGRATION FOR GP PRIOR APPROXIMATIONS

A multitude of *numerical integration* methods [51] can readily be deployed to compute the expectation in Equation (2.44),

$$\begin{aligned} k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}_{p(\omega)}[\varphi_\omega(\mathbf{x})^\top \varphi_\omega(\mathbf{x}')] \\ &\approx \sum_{i=1}^{L} \alpha_i \left(\varphi_{\xi_i}(\mathbf{x})^\top \varphi_{\xi_i}(\mathbf{x}') \right), \end{aligned}$$

where ξ_i are referred to as the *abscissas*, or, *nodes*, and α_i the *weights*, or, *coefficients*. Let us define the mapping $\boldsymbol{\varphi} : \mathbb{R}^D \to \mathbb{R}^{L'}$,

$$\boldsymbol{\varphi}(\mathbf{x}) \triangleq \begin{bmatrix} \sqrt{\alpha_1}\cos \boldsymbol{\xi}_1^\top \mathbf{x} \\ \vdots \\ \sqrt{\alpha_L}\cos \boldsymbol{\xi}_L^\top \mathbf{x} \\ \sqrt{\alpha_1}\sin \boldsymbol{\xi}_1^\top \mathbf{x} \\ \vdots \\ \sqrt{\alpha_L}\sin \boldsymbol{\xi}_L^\top \mathbf{x} \end{bmatrix}, \tag{A.7}$$

where $L' = 2L$. We therefore have

$$\boldsymbol{\varphi}(\mathbf{x})^\top \boldsymbol{\varphi}(\mathbf{x}') = \sum_{i=1}^{L} a_i \left(\varphi_{\xi_i}(\mathbf{x})^\top \varphi_{\xi_i}(\mathbf{x}') \right) \approx k(\mathbf{x}, \mathbf{x}').$$

We can view $\boldsymbol{\varphi}(\mathbf{x}), \boldsymbol{\varphi}(\mathbf{x}')$ as a factorisation, or, decomposition, of the kernel $k(\mathbf{x}, \mathbf{x}')$. Thus, we refer to $\boldsymbol{\varphi}$ as a *Fourier feature decomposition* of k.

A.3.1 *Monte Carlo Estimation*

Let us consider the simple case of MC integration, where $\alpha_i \triangleq 1/L$ and $\boldsymbol{\xi}_i \triangleq \boldsymbol{\omega}^{(i)}$, with $\boldsymbol{\omega}^{(i)} \sim p(\boldsymbol{\omega})$. More explicitly,

$$k(\mathbf{x}, \mathbf{x}') \approx \frac{1}{L} \sum_{i=1}^{L} \varphi_{\boldsymbol{\omega}^{(i)}}(\mathbf{x})^\top \varphi_{\boldsymbol{\omega}^{(i)}}(\mathbf{x}'), \quad \text{where} \quad \boldsymbol{\omega}^{(i)} \sim p(\boldsymbol{\omega}).$$

The corresponding Fourier feature decomposition is then

$$\boldsymbol{\varphi}(\mathbf{x}) \triangleq \sqrt{\frac{2}{L'}} \begin{bmatrix} \cos {\boldsymbol{\omega}^{(1)}}^\top \mathbf{x} \\ \vdots \\ \cos {\boldsymbol{\omega}^{(L'/2)}}^\top \mathbf{x} \\ \sin {\boldsymbol{\omega}^{(1)}}^\top \mathbf{x} \\ \vdots \\ \sin {\boldsymbol{\omega}^{(L'/2)}}^\top \mathbf{x} \end{bmatrix}, \tag{A.8}$$

where $\boldsymbol{\omega}^{(i)} \sim p(\boldsymbol{\omega})$, which we refer to as the MC Fourier features or, more commonly, RFF [206, 207].

phase-shifted cosine features

Let us define $\phi_{(\boldsymbol{\omega},b)} : \mathbb{R}^D \to \mathbb{R}$ to be, as before, the projection in some random direction $\boldsymbol{\omega} \sim p(\boldsymbol{\omega})$, but shifted by some $b \sim \mathcal{U}[0, 2\pi]$,

$$\phi_{(\boldsymbol{\omega},b)}(\mathbf{x}) \triangleq \sqrt{2}\cos(\boldsymbol{\omega}^\top \mathbf{x} + b). \tag{A.9}$$

We take the product of $\phi_{(\boldsymbol{\omega},b)}$ evaluated at inputs $\mathbf{x}$ and $\mathbf{x}'$ to get

$$\phi_{(\boldsymbol{\omega},b)}(\mathbf{x})\phi_{(\boldsymbol{\omega},b)}(\mathbf{x}') = 2\cos(\boldsymbol{\omega}^\top \mathbf{x} + b)\cos(\boldsymbol{\omega}^\top \mathbf{x}' + b) \tag{A.10}$$

$$= \cos(\boldsymbol{\omega}^\top(\mathbf{x} + \mathbf{x}') + 2b) + \cos(\boldsymbol{\omega}^\top(\mathbf{x} - \mathbf{x}')), \tag{A.11}$$

where, in the last line, we've used the *product-to-sum* trigonometric identity (see Appendix A.A for details). By virtue of the periodicity of sinusoids, taking the expectation of Equation (A.10) erases the first term of Equation (A.11), giving

$$\begin{aligned}&\mathbb{E}_{p(\boldsymbol{\omega},b)}[\phi_{(\boldsymbol{\omega},b)}(\mathbf{x})\phi_{(\boldsymbol{\omega},b)}(\mathbf{x}')]\\&\quad= \mathbb{E}_{p(\boldsymbol{\omega})}[\cos\left(\boldsymbol{\omega}^\top(\mathbf{x}-\mathbf{x}')\right)] + \cancel{\mathbb{E}_{p(\boldsymbol{\omega},b)}[\cos\left(\boldsymbol{\omega}^\top(\mathbf{x}+\mathbf{x}')+2b\right)]}\\&\quad= k(\mathbf{x},\mathbf{x}').\end{aligned}$$

See Appendix A.B for details. Hence, the product in Equation (A.10) is also an unbiased estimator of the kernel. For brevity, we shall write $\phi_i(\mathbf{x})$ to signify $\phi_{(\boldsymbol{\omega}^{(i)},b^{(i)})}(\mathbf{x})$ for $\boldsymbol{\omega}^{(i)} \sim p(\boldsymbol{\omega})$ and $b^{(i)} \sim \mathcal{U}[0,2\pi]$. The analogous Fourier feature decomposition $\boldsymbol{\phi} : \mathbb{R}^D \to \mathbb{R}^L$ is given by

$$\boldsymbol{\phi}(\mathbf{x}) \triangleq \sqrt{\frac{2}{L}}\begin{bmatrix}\cos\left({\boldsymbol{\omega}^{(1)}}^\top\mathbf{x}+b^{(1)}\right)\\ \vdots \\ \cos\left({\boldsymbol{\omega}^{(L)}}^\top\mathbf{x}+b^{(L)}\right)\end{bmatrix} = \frac{1}{\sqrt{L}}\begin{bmatrix}\phi_1(\mathbf{x})\\ \vdots \\ \phi_L(\mathbf{x})\end{bmatrix}. \tag{A.12}$$

We refer to this Fourier feature decomposition, originally proposed by Rahimi and Recht [206], as the *phase-shifted cosine* variant of RFF. Both MC estimators outlined in this section introduces error that decays at the rate of $\mathcal{O}(L^{-1/2})$, which, notably, is independent of the input dimensionality. A theoretical comparison of the Fourier feature decompositions of Equations (A.8) and (A.12) is given by Sutherland and Schneider [254], who report that for the SE kernel, the latter produces strictly higher variance and results in worse bounds.

A.3.2 *Quasi-Monte Carlo*

We can readily improve upon the convergence of MC by employing quasi Monte Carlo (QMC), which uses deterministic low-discrepancy sequences to construct samples. We refer to this family of Fourier feature decompositions as quasi-random Fourier features (QRFF) [5, 294].

In particular, QMC approximates following integral over the unit hypercube,

$$\int_{[0,1]^D} f(\mathbf{u})\,\mathrm{d}\mathbf{u} \approx \frac{1}{L}\sum_{i=1}^{L} f(\mathbf{u}^{(i)}), \tag{A.13}$$

by sequentially constructing the samples $\mathbf{u}^{(i)}$ deterministically using low-discrepancy sequences, thereby ameliorating the undesirable effects of samples forming clusters that commonly occurs when sampling independently at random. The interested reader may wish to refer to the manuscripts by Caflisch [27] and Dick, Kuo, and Sloan [58] for a more complete treatment of the topic.

multi-dimensional Gaussian integrals

To approximate multi-dimensional integrals with a Gaussian mea-

sure over $\mathbb{R}^D$, we can apply a change-of-variables based on the Gaussian inverse cumulative distribution function (CDF), or *quantile function*, to reduce it to an integral in the form of Equation (A.13). Suppose we have a multivariate Gaussian density $q(\boldsymbol{\omega})$. Then we can write

$$\int_{\mathbb{R}^D} q(\boldsymbol{\omega}) f(\boldsymbol{\omega}) \, \mathrm{d}\boldsymbol{\omega} = \int_{[0,1]^D} f(\Phi^{-1}(\mathbf{u})) \, \mathrm{d}\mathbf{u}, \tag{A.14}$$

where $\Phi^{-1} : [0,1]^D \to \mathbb{R}$ is the quantile function of q.

importance sampling

For non-Gaussian densities $p(\boldsymbol{\omega})$ in general, we can utilise *importance sampling* to cast our problem into the Gaussian integral of Equation (A.14), by using a Gaussian $q(\boldsymbol{\omega})$ as the *proposal* distribution,

$$\begin{aligned}\int_{\mathbb{R}^D} p(\boldsymbol{\omega}) f(\boldsymbol{\omega}) \, \mathrm{d}\boldsymbol{\omega} &= \int_{\mathbb{R}^D} q(\boldsymbol{\omega}) \left(\frac{p(\boldsymbol{\omega})}{q(\boldsymbol{\omega})} f(\boldsymbol{\omega}) \right) \mathrm{d}\boldsymbol{\omega} \\ &= \int_{[0,1]^D} r(\Phi^{-1}(\mathbf{u})) f(\Phi^{-1}(\mathbf{u})) \, \mathrm{d}\mathbf{u},\end{aligned}$$

where $r(\boldsymbol{\omega}) \triangleq {}^{p(\boldsymbol{\omega})}/_{q(\boldsymbol{\omega})}$ is the *importance weight*, or *likelihood ratio*.

From this, we arrive at the following Fourier feature decomposition,

$$\begin{aligned}k(\mathbf{t}, \mathbf{0}) &\approx \frac{1}{L} \sum_{i=1}^{L} r(\Phi^{-1}(\mathbf{u}^{(i)})) \cos\left(\Phi^{-1}(\mathbf{u}^{(i)}) \cdot \mathbf{t}\right) \\ &= \boldsymbol{\varphi}(\mathbf{x})^\top \boldsymbol{\varphi}(\mathbf{x}'),\end{aligned}$$

where

$$\boldsymbol{\varphi}(\mathbf{x}) \triangleq \sqrt{\frac{2}{L'}} \begin{bmatrix} \sqrt{r(\Phi^{-1}(\mathbf{u}^{(1)}))} \cos\left(\Phi^{-1}(\mathbf{u}^{(1)}) \cdot \mathbf{x}\right) \\ \vdots \\ \sqrt{r(\Phi^{-1}(\mathbf{u}^{(L'/2)}))} \cos\left(\Phi^{-1}(\mathbf{u}^{(L'/2)}) \cdot \mathbf{x}\right) \\ \sqrt{r(\Phi^{-1}(\mathbf{u}^{(1)}))} \sin\left(\Phi^{-1}(\mathbf{u}^{(1)}) \cdot \mathbf{x}\right) \\ \vdots \\ \sqrt{r(\Phi^{-1}(\mathbf{u}^{(L'/2)}))} \sin\left(\Phi^{-1}(\mathbf{u}^{(L'/2)}) \cdot \mathbf{x}\right) \end{bmatrix}.$$

A.3.3 *Quadrature*

We now introduce quadrature Fourier features (QFF) [6, 50, 180, 183]. We first restrict our attention to the one-dimensional case and defer our discussion of the multi-dimensional case when we introduce the multi-dimensional generalisations of numerical quadrature, sometimes referred to as *cubature*.

Gaussian-Christoffel quadrature; Gaussian quadrature

A quadrature formula that approximates the following integral by a finite sum

$$\int_a^b w(u) f(u) \, \mathrm{d}u \approx \sum_{i=1}^{L} \alpha_i f(\xi_i) \tag{A.15}$$

is called a *Gauss-Christoffel quadrature formula* (or simply a *Gaussian quadrature formula*) if it has maximum degree of exactness, i. e., if Equation (A.15) is an exact equality whenever f is a polynomial of degree $2L-1$ [80]. We refer to ξ_i as a the *Christoffel abscissas* and α_i the *Christoffel weights* associated with the weight function $w(u)$. The case of $w(u) \triangleq 1$ on the interval $[-1,1]$ was first studied by Gauss [79], and is now referred to as Gauss-Legendre quadrature. Other classical cases are associated with the names of Jacobi, Laguerre, and Hermite. The formulation based on orthogonal polynomials was advanced by Jacobi [112]. A comprehensive though possibly now outdated review of the topic of Gauss-Christoffel quadrature can be found in the landmark survey of Gautschi [81].

A.3.3.1 *Gauss-Hermite Quadrature*

The weight function serves to help factor out unruly behaviour in the integrand. Particularly relevant is the case of Gauss-Hermite quadrature, in which the weight function of interest is $w(u) \triangleq e^{-u^2}$ and the interval of integration is $(-\infty, \infty)$. That is, we're interested in approximating integrals of the form

Gauss-Hermite quadrature

$$\int_{-\infty}^{\infty} e^{-u^2} f(u)\,\mathrm{d}u \tag{A.16}$$

The nodes ξ_i are roots of $H_L(u)$, the Hermite polynomial of degree L, and the associated weights α_i are given by

$$\alpha_i \triangleq \frac{2^{L-1} L! \sqrt{\pi}}{L^2 [H_{L-1}(\xi_i)]^2}.$$

It is not hard to appreciate the power of this quadrature formula, for it is trivial to apply it to the calculation of expectations under Gaussian distributions, a quantity upon which many problems in statistical ML rely. In particular, we are often interested in computing the expected value of $f(\omega)$ under $p(\omega) = \mathcal{N}(\omega \,|\, \mu, \sigma^2)$,

Gaussian expectations

$$\begin{aligned} \mathbb{E}_{p(\omega)}[f(\omega)] &= \int_{-\infty}^{\infty} \mathcal{N}(\omega \,|\, \mu, \sigma^2) f(\omega)\,\mathrm{d}\omega \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\infty} e^{-\left(\frac{\omega-\mu}{\sqrt{2}\sigma}\right)^2} f(\omega)\,\mathrm{d}\omega. \end{aligned} \tag{A.17}$$

Simply by making a change-of-variable $u = \frac{\omega-\mu}{\sqrt{2}\sigma} \Leftrightarrow \omega = \sqrt{2}\sigma u + \mu$, we can rewrite Equation (A.17) in the form of Equation (A.16),

$$\mathbb{E}_{p(\omega)}[f(\omega)] = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-u^2} g(u)\,\mathrm{d}u, \tag{A.18}$$

where we've defined $g(u) \triangleq f(\sqrt{2}\sigma u + \mu)$. We thus have the following quadrature formula:

$$\mathbb{E}_{p(\omega)}[f(\omega)] \approx \frac{1}{\sqrt{\pi}} \sum_{i=1}^{L} \alpha_i g(\xi_i).$$

Kernel decomposition of the SE kernel based on Gauss-Hermite quadrature

Recall that by Equation (2.41), assuming its spectral density $p(\omega)$ is even symmetric, we can express a stationary kernel $k(t,0)$ as the expected value of function $f(\omega) = \cos(\omega t)$ under $p(\omega)$. Let us first focus on the case of the SE kernel, the spectral density of which is given in Equation (2.40) as a Gaussian, $p(\omega) = \mathcal{N}\left(\omega \,|\, 0, \ell^{-2}\right)$. Therefore, by Equation (A.18), with $g(u) = f\left(\sqrt{2}u/\ell\right) = \cos\left(\sqrt{2}ut/\ell\right)$, we can write

$$k(t,0) = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-u^2} \cos\left(\frac{\sqrt{2}ut}{\ell}\right) \mathrm{d}u \tag{A.19}$$

$$\approx \frac{1}{\sqrt{\pi}} \sum_{i=1}^{L} \alpha_i \cos\left(\frac{\sqrt{2}\xi_i t}{\ell}\right). \tag{A.20}$$

By Equation (2.43), we have

$$\begin{aligned}\cos\left(\frac{\sqrt{2}\xi_i t}{\ell}\right) &= \cos\left(\frac{\sqrt{2}\xi_i (x - x')}{\ell}\right) \\ &= \varphi_{(\sqrt{2}\xi_i/\ell)}(x)^\top \varphi_{(\sqrt{2}\xi_i/\ell)}(x'),\end{aligned}$$

where $\varphi_{(\cdot)}(x)$ is defined in Equation (2.42). Accordingly, as in Equation (A.7), we have the Fourier feature decomposition $\boldsymbol{\varphi} : \mathbb{R} \to \mathbb{R}^{L'}$,

$$\boldsymbol{\varphi}(x) \triangleq \frac{1}{\sqrt[4]{\pi}} \begin{bmatrix} \sqrt{\alpha_1} \cos\left(\frac{\sqrt{2}\xi_1 x}{\ell}\right) \\ \vdots \\ \sqrt{\alpha_L} \cos\left(\frac{\sqrt{2}\xi_L x}{\ell}\right) \\ \sqrt{\alpha_1} \sin\left(\frac{\sqrt{2}\xi_1 x}{\ell}\right) \\ \vdots \\ \sqrt{\alpha_L} \sin\left(\frac{\sqrt{2}\xi_L x}{\ell}\right) \end{bmatrix}.$$

Extending this to kernels with non-Gaussian spectral densities can be done using the importance sampling technique described in the preceding section.

A.3.3.2 *Gauss-Legendre Quadrature*

Gauss-Legendre quadrature

Let us consider the classical case of Gauss-Legendre quadrature, in which the weight function of interest is $w(u) \triangleq 1$ and the interval of integration is $[-1, 1]$. That is, we're interested in approximating integrals of the form

$$\int_{-1}^{1} f(u) \, \mathrm{d}u$$

The nodes ξ_i are roots of $P_L(u)$, the Legendre polynomial of degree L normalized to give $P_L(1) = 1$, and the associated weights a_i are given by

$$\alpha_i \triangleq \frac{2}{\left(1 - \xi_i^2\right)\left[P_L'(\xi_i)\right]^2}.$$

Change of interval

An integral over $[a, b]$ must be changed into an integral over $[-1, 1]$ before applying Gauss-Legendre quadrature,

$$\int_a^b f(\omega)\,\mathrm{d}\omega = \int_{T(-1)}^{T(1)} f(\omega)\,\mathrm{d}\omega = \int_{-1}^{1} f(T(u))T'(u)\,\mathrm{d}u,$$

where $T : [a, b] \to [-1, 1]$ is a differentiable function with a continuous derivative. In particular, for integration over the infinite interval, we can use the substitution $T(u) = \tan\left(\frac{\pi}{2}u\right)$ to give

$$\int_{-\infty}^{\infty} f(\omega)\,\mathrm{d}\omega = \frac{\pi}{2}\int_{-1}^{1} \frac{f\left(\tan\left(\frac{\pi}{2}u\right)\right)}{\cos^2\left(\frac{\pi}{2}u\right)}\,\mathrm{d}u,$$

where we've used $T'(u) = \frac{\pi}{2}\frac{1}{\cos^2\left(\frac{\pi}{2}u\right)}$. A myriad other choices are also available. For instance, Mutnỳ and Krause [184] use $T(u) \triangleq \cot\left(\frac{\pi}{2}(u+1)\right)$ to give

$$T'(u) = -\frac{\pi}{2}\frac{1}{\sin^2\left(\frac{\pi}{2}(u+1)\right)},$$

and

$$\int_{-\infty}^{\infty} f(\omega)\,\mathrm{d}\omega = \frac{\pi}{2}\int_{-1}^{1} \frac{f\left[\cot\left(\frac{\pi}{2}(u+1)\right)\right]}{\sin^2\left(\frac{\pi}{2}(u+1)\right)}\,\mathrm{d}u.$$

Recall that in Gauss-Hermite quadrature, our integrand of interest is $f(\omega) = \cos(\omega t)$. That is, the contribution of the spectral density $p(\omega)$ is absorbed into the weight function. In contrast, when using Gauss-Legendre quadrature, our integrand explicitly includes the contribution from the spectral density, $f(\omega) = p(\omega)\cos(\omega t)$. Therefore, we can directly incorporate non-Gaussian spectral densities $p(\omega)$ without needing to resort to importance sampling. However, unlike in Gauss-Hermite quadrature, we will not be able to isolate potentially deleterious effects of the spectral density from our approximation.

All in all, we have the Fourier feature decomposition $\boldsymbol{\varphi} : \mathbb{R} \to \mathbb{R}^{L'}$,

$$\boldsymbol{\varphi}(x) \triangleq \sqrt{\frac{\pi}{2}}\begin{bmatrix} \sqrt{\frac{\alpha_1 p\left(\tan\left(\frac{\pi}{2}\xi_1\right)\right)}{\cos^2\left(\frac{\pi}{2}\xi_1\right)}}\cos\left(\tan\left(\frac{\pi}{2}\xi_1\right)\cdot x\right) \\ \vdots \\ \sqrt{\frac{\alpha_L p\left(\tan\left(\frac{\pi}{2}\xi_L\right)\right)}{\cos^2\left(\frac{\pi}{2}\xi_L\right)}}\cos\left(\tan\left(\frac{\pi}{2}\xi_L\right)\cdot x\right) \\ \sqrt{\frac{\alpha_1 p\left(\tan\left(\frac{\pi}{2}\xi_1\right)\right)}{\cos^2\left(\frac{\pi}{2}\xi_1\right)}}\sin\left(\tan\left(\frac{\pi}{2}\xi_1\right)\cdot x\right) \\ \vdots \\ \sqrt{\frac{\alpha_L p\left(\tan\left(\frac{\pi}{2}\xi_L\right)\right)}{\cos^2\left(\frac{\pi}{2}\xi_L\right)}}\sin\left(\tan\left(\frac{\pi}{2}\xi_L\right)\cdot x\right) \end{bmatrix}.$$

Gaussian quadrature error analysis

The error of a Gaussian quadrature formula is as follows [248],

$$\int_a^b w(u)f(u)\,\mathrm{d}u - \sum_{i=1}^{L} a_i f(\xi_i) = \frac{f^{(2L)}(\theta)}{(2L)!}\langle p_L, p_L\rangle, \tag{A.21}$$

for some $\theta \in (a, b)$ where p_L is a monic orthogonal polynomial of degree L, and $\langle \cdot, \cdot \rangle$ is the scalar product associated with the weight function $w(u)$,

$$\langle p, q \rangle = \int_a^b w(u) p(u) q(u) \, \mathrm{d}u.$$

Cubature: quadrature in multiple dimensions

Let us now consider quadrature for functions of several variables,

$$f(\mathbf{u}) = f(u_1, \ldots, u_D).$$

For weight functions w that factorise as

$$w\left(\|\mathbf{u}\|_2\right) = \prod_{d=1}^{D} w(u_d), \tag{A.22}$$

we have

$$\begin{aligned}
&\int_{\mathbb{R}^D} w\left(\|\mathbf{u}\|_2\right) f(\mathbf{u}) \, \mathrm{d}\mathbf{u} = \int \cdots \int \prod_{d=1}^{D} w(u_d) f(u_1, \ldots, u_D) \, \mathrm{d}u_1 \cdots \mathrm{d}u_D \\
&= \int w(u_D) \left(\int w(u_{D-1}) \cdots \left(\int w(u_1) f(u_1, \ldots, u_D) \, \mathrm{d}u_1 \right) \cdots \mathrm{d}u_{D-1} \right) \mathrm{d}u_D \\
&\approx \sum_{i=1}^{L_D} \alpha_i^{(D)} \left(\int w(u_{D-1}) \cdots \left(\int w(u_1) f(u_1, \ldots, \xi_i^{(D)}) \, \mathrm{d}u_1 \right) \cdots \mathrm{d}u_{D-1} \right) \\
&\vdots \\
&\approx \sum_{i_1=1}^{L_1} \cdots \sum_{i_{D-1}=1}^{L_{D-1}} \sum_{i_D=1}^{L_D} \alpha_{i_1}^{(1)} \cdots \alpha_{i_{D-1}}^{(D-1)} \alpha_{i_D}^{(D)} f(\xi_{i_1}^{(1)}, \ldots, \xi_{i_{D-1}}^{(D-1)}, \xi_{i_D}^{(D)}),
\end{aligned} \tag{A.23}$$

where $\xi_i^{(d)}$ and $\alpha_i^{(d)}$ are the $L_d > 0$ abscissa and weights for quadrature along the dth dimension. In other words, for weight functions that satisfy Equation (A.22), we can decompose its multi-dimensional quadrature through repeated application of one-dimensional quadrature along each dimension.

The nested sum of Equation (A.23) can be written as a single sum over the elements of the D-ary Cartesian product of the quadrature nodes along each dimension $(\xi_1^{(1)} \cdots \xi_{L_1}^{(1)}), \ldots, (\xi_1^{(D)} \cdots \xi_{L_D}^{(D)})$, of which there are in total $\prod_{d=1}^{D} L_d$. Assuming for simplicity that $L_d = L$ for all $d = 1, \ldots, D$ and some $L > 0$, then there are a total of of L^D quadrature nodes. That is, the number of nodes grows exponentially in the input dimensionality.

The weight function in Gauss-Legendre quadrature trivially satisfies Equation (A.22). It is easy to verify that it is also satisfied by the weight function in Gauss-Hermite quadrature. Namely, for $w(u) \triangleq e^{-u^2}$, we have

$$e^{-\|\mathbf{u}\|_2^2} = e^{-\sum_{d=1}^{D} u_d^2} = \prod_{d=1}^{D} e^{-u_d^2}.$$

A.3.3.3 *Newton-Cotes Quadrature*

Newton-Cotes quadrature

Let us now consider an alternative to Gaussian quadrature, namely, the quadrature rules of Newton and Cotes, which is obtained by replacing the integrand with a suitable interpolating polynomial $P(u)$. Consult the text of Stoer and Bulirsch [248] for a more complete treatment of the subject. Consider a uniform partition of the closed interval $[a, b]$ with

$$\xi_i \triangleq a + ih,$$

and step width $h \triangleq \frac{b-a}{m}$, for some integer $m > 0$, and let P_m be the interpolating polynomial of degree m or less with

$$P_m(\xi_i) = f_i \triangleq f(\xi_i)$$

for $i = 0, 1, \ldots, m$. Lagrange's interpolation formula gives

$$P_m(u) \triangleq \sum_{i=0}^{m} f_i \mathcal{L}_i(u), \qquad \mathcal{L}_i(u) \triangleq \prod_{\substack{j=0 \\ j \neq i}}^{m} \frac{u - \xi_j}{\xi_i - \xi_j}.$$

Integration gives

$$\int_a^b P_m(u)\,\mathrm{d}u = h \sum_{i=0}^{m} \alpha_i f(\xi_i)$$

where the weights α_i are some function strictly of m, and crucially not dependent on the integrand f, nor on the boundaries of the interval, a, b.

Simpson's rule

In the case of $m = 2$, we obtain the approximation

$$\int_a^b f(u)\,\mathrm{d}u \approx \int_a^b P_2(u)\,\mathrm{d}u = \frac{h}{3}(f(\xi_0) + 4f(\xi_1) + f(\xi_2)),$$

which is commonly known as *Simpson's rule*.

Consider a step width $h > 0$ such that

$$b = a + L'h$$

for some positive even integer $L' = 2L, L > 0$. We can apply Simpson's rule to each subinterval $[\xi_{2k-2}, \xi_{2k-1}, \xi_{2k}]$, where $k = 1, \ldots, L$.

Compound Simpson's rule

$$\begin{aligned} \int_a^b f(u)\,\mathrm{d}u &= \sum_{k=1}^{L} \int_{\xi_{2k-2}}^{\xi_{2k}} f(u)\,\mathrm{d}u \\ &\approx \frac{h}{3} \sum_{k=1}^{L} [f(\xi_{2k-2}) + 4f(\xi_{2k-1}) + f(\xi_{2k})] \\ &\triangleq \mathcal{S}[f]. \end{aligned} \tag{A.24}$$

We can rearrange by odd and even terms to get

$$\mathcal{S}[f] = \frac{h}{3}\left(4 \sum_{k=1}^{L} f(\xi_{2k-1}) + 2 \sum_{k=1}^{L} f(\xi_{2k}) + f(\xi_0) - f(\xi_{L'})\right).$$

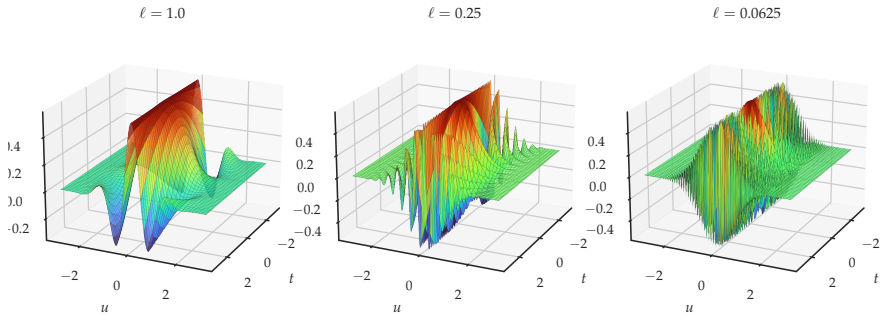


Figure A.4: The integrand $H(u) = e^{-u^2} \cos\left(\frac{\sqrt{2}ut}{\ell}\right)$ which becomes increasingly oscillatory as the lengthscale decreases $\ell = 4^{-k}$ where $k = 0, 1, 2$.

We can also write

$$\mathcal{S}[f] = \sum_{i=0}^{L'} \gamma_i f(\xi_i)$$

where

$$\gamma_i = \begin{cases} \frac{4}{3}h & i \text{ odd}, \\ \frac{2}{3}hc_i & i \text{ even}. \end{cases} \quad \text{and} \quad c_i = \begin{cases} \frac{1}{2} & i = 0 \text{ or } i = L', \\ 1 & 0 < i < L'. \end{cases}$$

for $i = 0, \ldots, L'$. Denoting the odd and even terms as

$$S_{\text{odd}} \triangleq \sum_{k=1}^{L} f(\xi_{2k-1}), \tag{A.25}$$

$$\begin{aligned} S_{\text{even}} &\triangleq \sum_{k=1}^{L} f(\xi_{2k}) + \frac{f(\xi_0) - f(\xi_{L'})}{2} \\ &= \sum_{k=0}^{L} f(\xi_{2k}) - \frac{f(\xi_0) + f(\xi_{L'})}{2}, \end{aligned} \tag{A.26}$$

respectively, we can simplify Equation (A.24) as

$$\mathcal{S}[f] = \frac{h}{3}\left(4 \cdot S_{\text{odd}} + 2 \cdot S_{\text{even}}\right). \tag{A.27}$$

Newton-Cotes error analysis

A.3.3.4 *Filon's rule for highly-oscillatory integrals*

Highly-oscillatory integrals

Recall from Equation (A.19) that the integrand in which we're inter-

ested is

$$F(u) = e^{-u^2} \cos\left(\frac{\sqrt{2}ut}{\ell}\right) \tag{A.28}$$

See Figure A.4 for surface plots of this function at varying settings of ℓ. More broadly, consider integrals of the form

$$\int_a^b g(u)\cos(ut)\,\mathrm{d}u, \quad \text{and} \quad \int_a^b g(u)\sin(ut)\,\mathrm{d}u, \tag{A.29}$$

or, more generally,

$$\int_a^b g(u)e^{iut}\,\mathrm{d}u, \tag{A.30}$$

of which Equation (A.19) is clearly an instance.

Filon's rule

An extension of Simpson's rule that is aimed at dealing with highly-oscillatory integrals, known as Filon's rule [69]. Consider integrands of the form

$$f(u) = g(u)\cos(ut)$$

Let $\theta \triangleq ht$ so that $1/t = h/\theta$. We have

$$\begin{aligned}
&\int_a^b g(u)\cos(ut)\,\mathrm{d}u = \sum_{k=1}^{L}\int_{\xi_{2k-2}}^{\xi_{2k}} g(u)\cos(ut)\,\mathrm{d}u \\
&\approx \frac{1}{t}\left[\frac{4}{\theta}\left(\frac{\sin\theta}{\theta} - \cos\theta\right)\sum_{k=1}^{L} f(\xi_{2k-1})\right. \\
&+\frac{1}{\theta}\left(1+\cos^2\theta - \frac{2\cos\theta\sin\theta}{\theta}\right)\left(2\sum_{k=1}^{L} f(\xi_{2k}) + f(\xi_0) - f(\xi_{L'})\right) \\
&\left.+\left(1+\frac{\cos\theta\sin\theta}{\theta} - \frac{2\sin^2\theta}{\theta^2}\right)\left[g(\xi_{L'})\sin(\xi_{L'}t) - g(\xi_0)\sin(\xi_0 t)\right]\right] \\
&= \frac{h}{\theta^3}\left[4\left(\sin\theta - \theta\cos\theta\right)\sum_{k=1}^{L} f(\xi_{2k-1})\right. \\
&+\left(\theta(1+\cos^2\theta) - 2\cos\theta\sin\theta\right)\left(2\sum_{k=1}^{L} f(\xi_{2k}) + f(\xi_0) - f(\xi_{L'})\right) \\
&\left.+\left(\theta^2 + \theta\cos\theta\sin\theta - 2\sin^2\theta\right)\left[g(\xi_{L'})\sin(\xi_{L'}t) - g(\xi_0)\sin(\xi_0 t)\right]\right] \\
&\triangleq \mathcal{F}[g]
\end{aligned} \tag{A.31}$$

Note that we can also write

$$\begin{aligned}
\mathcal{F}[g] = &\sum_{i=0}^{L'}\gamma_i g(\xi_i) \\
&+\gamma\left(g(\xi_{L'})\sin\left(\xi_{L'}t\right) - g(\xi_0)\sin\left(\xi_0 t\right)\right)
\end{aligned}$$

where

$$\gamma_i = \begin{cases} \alpha h & i \text{ odd} \\ \beta h c_i & i \text{ even} \end{cases},$$
$$c_i = \begin{cases} \frac{1}{2} & i = 0 \text{ or } i = L' \\ 1 & 0 < i < L' \end{cases}.$$

for $i = 0, \ldots, L'$.

Using Equations (A.25) and (A.26), we can simplify Equation (A.31) to

$$\begin{aligned} \mathcal{F}[g] = & h \left[\alpha \cdot S_{\text{odd}} + \beta \cdot S_{\text{even}} \right. \\ & \left. + \gamma \left(g(\xi_{L'}) \sin(\xi_{L'} t) - g(\xi_0) \sin(\xi_0 t) \right) \right] \end{aligned} \tag{A.32}$$

where

$$\begin{aligned} \alpha &\triangleq \frac{4}{\theta^3} \left[\sin\theta - \theta \cdot \cos\theta \right], \\ \beta &\triangleq \frac{2}{\theta^3} \left[\theta \cdot (1 + \cos^2\theta) - 2\cos\theta \sin\theta \right], \\ \gamma &\triangleq \frac{1}{\theta^3} \left[\theta^2 + \theta \cdot \cos\theta \sin\theta - 2\sin^2\theta \right]. \end{aligned}$$

Now, by expanding α, β, and γ in powers of θ, we get

$$\begin{aligned} \alpha &= \frac{4}{3} - \frac{2\theta^2}{15} + \frac{\theta^4}{210} - \frac{\theta^6}{11340} + \frac{\theta^8}{997920} - \frac{\theta^{10}}{129729600} + \cdots \\ \beta &= \frac{2}{3} + \frac{2\theta^2}{15} - \frac{4\theta^4}{105} + \frac{2\theta^6}{567} - \frac{4\theta^8}{22275} + \frac{4\theta^{10}}{675675} - \cdots \\ \gamma &= \frac{2\theta^3}{45} - \frac{2\theta^5}{315} + \frac{2\theta^7}{4725} - \frac{8\theta^9}{467775} + \frac{4\theta^{11}}{8513505} - \cdots \end{aligned}$$

It is clear to see that as $\theta \to 0$ (or, equivalently, as $t \to 0$) we have

$$\alpha \to \frac{4}{3}, \qquad \beta \to \frac{2}{3}, \qquad \gamma \to 0.$$

In other words, Filon's rule of Equation (A.32) reduces exactly to Simpson's rule of Equation (A.27). This suggests that for sufficiently small values of t, Simpson's rule is just as good as Filon's rule when it comes to dealing with highly-oscillatory integrals. Of course, another way to look at it is that Filon's rule may actually be no better than Simpson's rule in this setting.

A.3.4 *Other Approaches*

Another notable approach is the orthogonal random features (ORF) [41, 42, 298], which selects frequencies according a an appropriately scaled random orthogonal matrix instead of a random Gaussian matrix. In

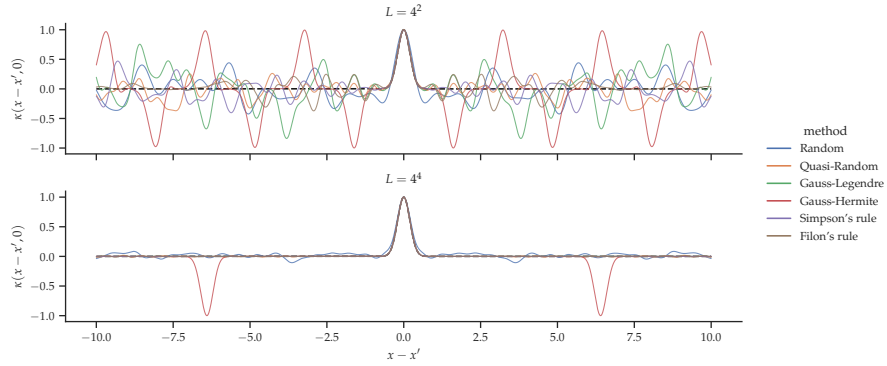


Figure A.5: SE kernel with variance 1 and lengthscale $\ell = 1/5$, and various approximations thereof, visualized on the domain $[-3, 3]$. In this domain, apart from Gauss-Hermite quadrature, the difference between quadrature methods is virtually indistinguishable for $L = 4^4 = 256$.

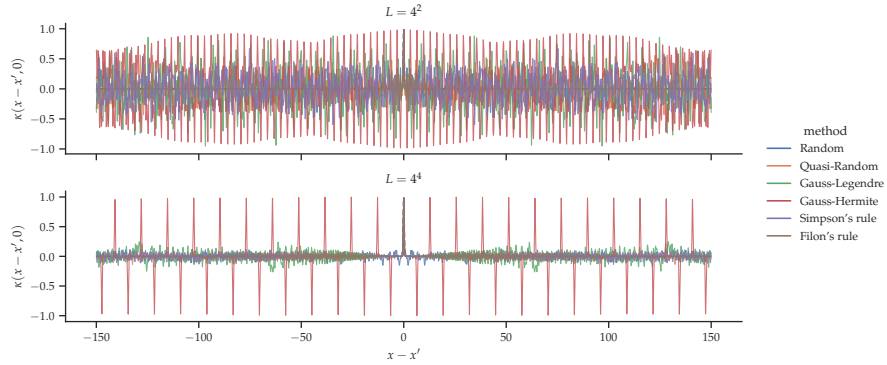


Figure A.6: SE kernel with variance 1 and lengthscale $\ell = 1/5$, and various approximations thereof, visualized on the domain $[-150, 150]$. The advantages of Filon's rule appear only to be realized when $|x - x'| > 100$ where the spurious oscillations begin to attenuate.

particular, let's define matrix $\mathbf{W}$ as the collection of L frequencies sampled from the kernel's spectral density, $\mathbf{W} = \begin{bmatrix}\boldsymbol{\omega}_1 \cdots \boldsymbol{\omega}_L\end{bmatrix}^\top \in \mathbb{R}^{L \times D}$. We can write Equation (A.8) as

$$\boldsymbol{\varphi}(\mathbf{x}) \triangleq \sqrt{\frac{2}{L'}} \begin{bmatrix}\cos(\mathbf{W}\mathbf{x}) \\ \sin(\mathbf{W}\mathbf{x})\end{bmatrix}.$$

Furthre, by Equation (A.8) and Table 2.1, the matrix $\mathbf{W}$ of RFF can be written as

$$\mathbf{W} = \mathbf{M}^{-\frac{1}{2}}\mathbf{G}$$

where $\mathbf{G}$ is a Gaussian random matrix. For ORF, we assume $L = D$ so that $\mathbf{G}$ is a square matrix, and set

$$\mathbf{W} \triangleq \mathbf{M}^{-\frac{1}{2}}\mathbf{S}\mathbf{Q},$$

where $\mathbf{Q}$ is the orthogonal matrix such that $\mathbf{QR} = \mathbf{G}$ for some upper triangular matrix $\mathbf{R}$, and $\mathbf{S} = \text{diag}(s_1, \cdots, s_D)$ with $s_i \sim \chi_D$, where χ_D denotes the χ-distribution[1] with D degrees of freedom. The transformation by $\mathbf{S}$ has the effect of making the rows of $\mathbf{SQ}$ and $\mathbf{G}$ identically distributed.

A number of relevant approaches, such as Fastfood [141], À la Carte [297] and the Nyström approximation [287, 296], have been excluded from the scope this work.

A.4 EXPERIMENTS

A.4.1 *Prior Approximation*

We're interested in the *relative error*, defined as the Frobenius norm of the difference between the kernel's exact Gram matrix $\mathbf{K}$, and its approximation based on an Fourier feature decomposition $\mathbf{\Phi\Phi}^\top$, normalized by the Frobenius norm of $\mathbf{K}$,

$$\text{relative error} \triangleq \frac{\|\mathbf{K} - \mathbf{\Phi\Phi}^\top\|_F}{\|\mathbf{K}\|_F}.$$

We restrict our focus to the Fourier feature decompositions that have been outlined in this report: RFF and its phase-shifted cosine variant, QFF, specifically its variants based on Gaussian quadrature (Gauss-Legendre, Gauss-Hermite), and Newton-Cotes quadrature (Simpson's rule), QRFF with Sobol sequences, and finally, ORF. We consider a number of datasets, namely, MOTORCYCLE, IRIS, DIABETES, BOSTON, WINE, and BREAST CANCER, and look at two kernels: the SE and Matérn-5/2 kernels both with decreasing lengthscales $\ell = 4^{-k}$ for $k = 0, 1, 2$.

See Figures A.7 and A.8 for results on the SE and Matérn-5/2 kernels, respectively. For methods with an inherent source of randomness, we report the mean and 95% confidence interval across 25 repetitions.

For the SE kernel, the picture is clear: for problems of moderate dimensionality (say, $D < 5$), Gaussian quadrature methods are far more efficient than any competing method. Furthermore, in the case of $D = 1$, the deleterious effects of small lengthscales are barely noticeable. In all settings of the lengthscale, kernel decompositions based on quadrature rapidly converges to the exact kernel, and require orders of magnitude fewer features.

1 not to be confused with the χ^2-distribution

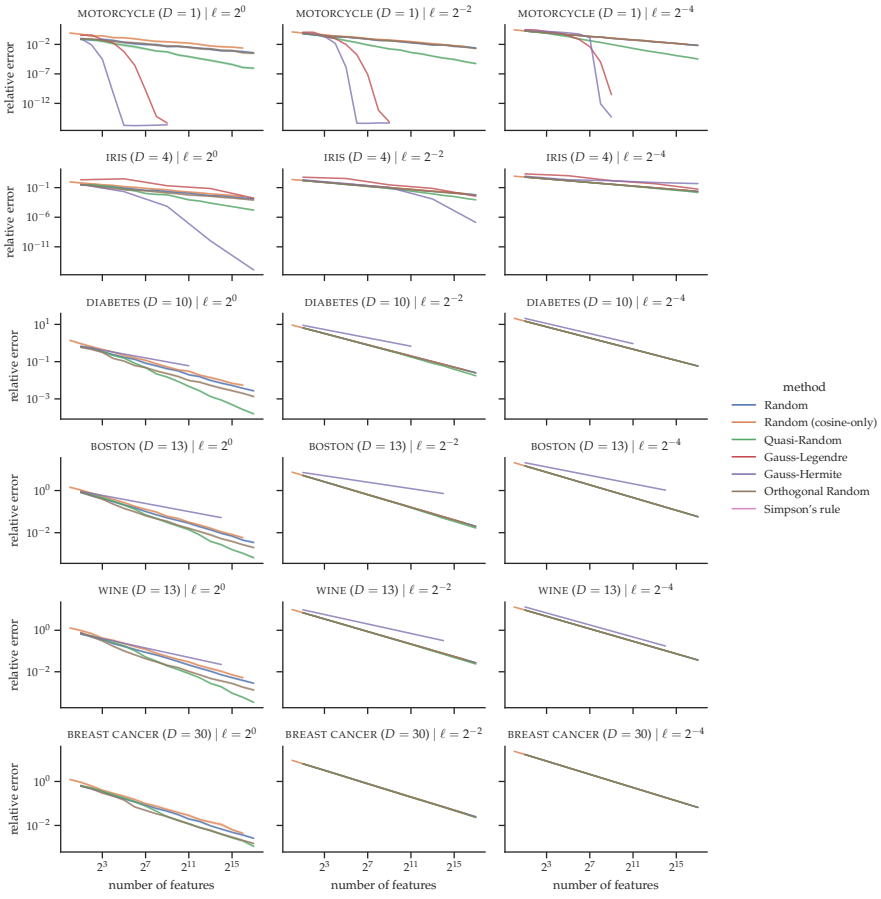


Figure A.7: Comparing the efficiency of various Fourier feature decompositions for the SE kernel.

In the case of $D = 4$, We see that Gauss-Hermite still outperforms all other methods for settings of the lengthscale above $\ell = 2^{-4}$, though perhaps less dramatically,

On the other hand, for the same dimensionality, Gauss-Legendre already begins to perform worse than all other methods. In fact, in all remaining problems (where dimensionality $D > 5$), Gauss-Legendre performs orders of magnitude worse (by orders of 10^2 or more). For the sake of readability, we've omitted the results of Gauss-Legendre and Simpson as these distort the scale of the plot dramatically.

As we move on to higher dimensions, particular in $D = 10$, the performance of Gauss-Hermite quadrature already degrades so significantly that it has become the worst of all competing methods. Furthermore, it becomes practically inapplicable beyond $D = 13$. In dimensionality $D \geq 10$ where Gauss-Hermite quadrature is still feasible, we see that its curves are truncated. This is because the errors are reported for just two settings of number of features. Recall that the number of features in multidimensional quadrature is L^D for some $L > 0$. When $D \geq 10$, for any setting of $L > 2$, this clearly becomes prohibitively large. Therefore, in such high-dimensional problems, we are restricted to setting $L \leq 2$. However, this is amounts to computing up to just two abscissa along the real line and then taking their D-ary Cartesian power to form D-dimensional quadrature nodes. Thus seen, it is no surprise that it fails to yield good results.

Outside of quadrature methods, we see that Quasi-random performs consistently well, in both low- and high-dimensional regimes. In low dimensions, it is second only to quadrature methods; in high dimensions, it consistently outperforms all competing methods. Therefore, it's safe to conclude that for the SE kernel in low-dimensional settings, one should prefer Gauss-Hermite QFF and in high-dimensional settings one should resort to QRFF.

For the Matérn-5/2 kernel, the story is quite similar, with one major exception: Quasi-random performs considerably worse and with far higher variance, particularly in high dimensions. Recall that to extend QRFF to kernels non-Gaussian spectral densities we are required to resort to importance sampling. Although this still results in an unbiased estimator, the variance is now a function of the likelihood ratio $r(\cdot)$, which is prone to taking on large values in high-dimensional settings.

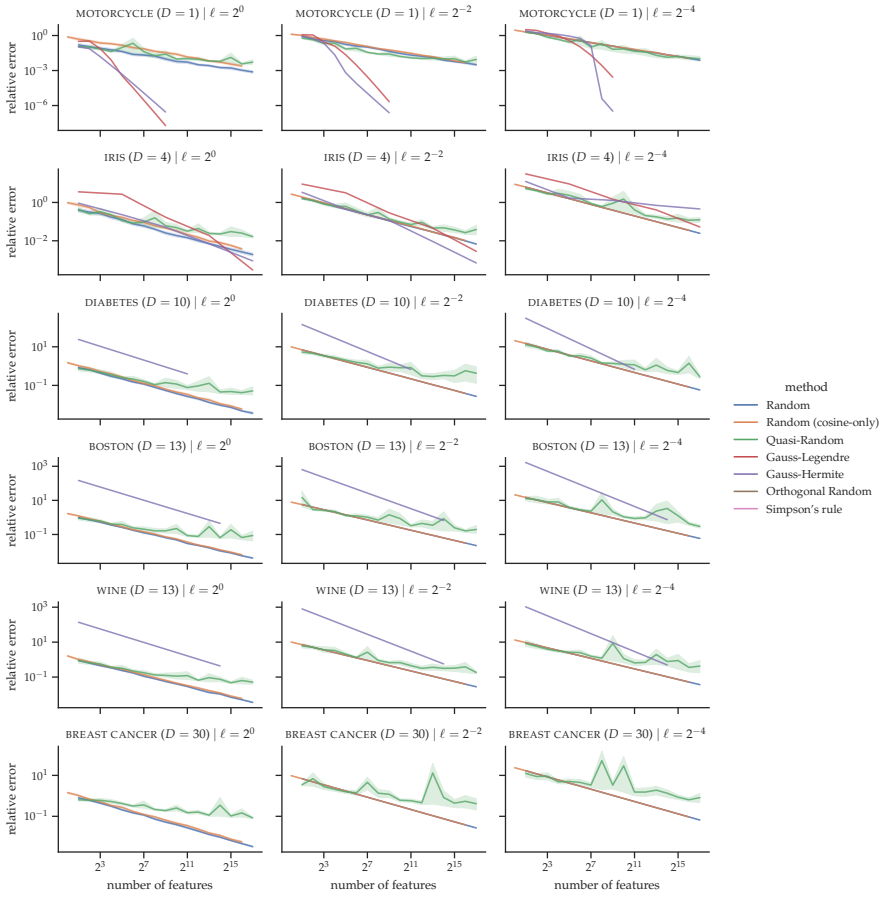


Figure A.8: Comparing the efficiency of various Fourier feature decompositions for the Matérn-5/2 kernel.

A.4.2 *Posterior Sample Approximation*

To assess the quality of posterior samples, we follow the approach of Wilson et al. [289], namely, by measuring the 2-Wasserstein distance [161] between the exact GP posterior and an empirical distribution constructed from samples.

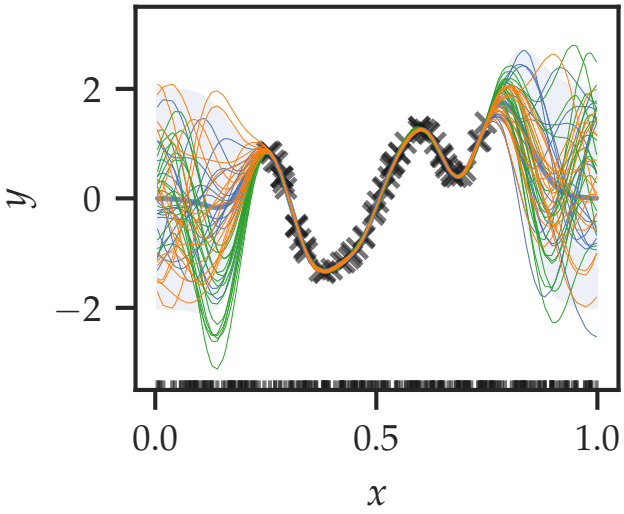


Figure A.9: An example a synthetic problem in 1D. In this illustration, there are $N = 2^6$ crosses ('$\times$') which represent the observations, and $T = 2^8$ vertical notches along the horizontal axis which represent the test, or query, points. The observations are generated using a GP with a SE kernel with unit variance and lengthscale $\ell = 2^{-4}$, while the test points are sampled uniformly at random. The *blue* curves are samples drawn from the exact GP posterior at the test points. Similarly, the *green* curves are 2^4 samples drawn from the weight-space approximate posterior, and the *orange* curves are samples generated with decoupled sampling. The kernel approximations are based on an RFF decomposition using $L = 256$ samples.

Toy datasets are synthesized as follows. The N training locations $\mathbf{X}$ are sampled uniformly at random and their corresponding observations are generated from the prior $\mathbf{y} \sim \mathcal{GP}(\mathbf{0}, \mathbf{K_{ff}} + \beta^{-1}\mathbf{I})$ with observation noise variance $\beta^{-1} = 10^{-3}$, using the SE kernel with unit amplitude and lengthscales of decreasing order $\ell = 4^{-k}$ for $k = 0, 1, 2$. Likewise, the T test locations $\mathbf{X}_*$ are sampled uniformly at random from $\mathcal{U}[0, 1]^{T \times D}$, where we set $T = 2^6 = 64$. The above is repeated to generate D-dimensional datasets for $D = 2^0, \ldots, 2^4$. See Figure A.9 for an example problem in one dimension.

Consistent with the findings of Wilson et al. [289], we observe the decoupled sampling scheme to be robust against variance starvation. In particular, the distance remains largely the same irrespective of the

training size N. Consequently, we only report results for the setting $N = 2^7 = 128$.

To eliminate confounding factors, we restrict our attention to exact GPs using the SE kernel with known and fixed hyperparameters, i. e. the hyperparameters that were used to synthesise the observed data. In total, $2^{12} = 4,096$ samples of $\mathbf{f}_* \mid \mathbf{y}$ are used as unbiased estimates $(\hat{\boldsymbol{\mu}}_{*|\mathbf{f}}, \hat{\boldsymbol{\Sigma}}_{**|\mathbf{f}})$ of the exact posterior moments $(\boldsymbol{\mu}_{*|\mathbf{f}}, \boldsymbol{\Sigma}_{**|\mathbf{f}})$ given in Equation (A.2). The 2-Wasserstein distances are then computed based on these moments,

$$\mathcal{W}_2 \left(\mathcal{N}(\boldsymbol{\mu}_{*|\mathbf{f}}, \boldsymbol{\Sigma}_{**|\mathbf{f}}), \mathcal{N}(\hat{\boldsymbol{\mu}}_{*|\mathbf{f}}, \hat{\boldsymbol{\Sigma}}_{**|\mathbf{f}}) \right)^2 .$$

This is computed both for samples from the weight-space approximate posterior, and for samples generated with decoupled sampling, shown in Figures A.10 and A.11, respectively.

We restrict our focus to the Fourier feature decompositions that have been outlined in this chapter: RFF, QFF, specifically its variants based on Gaussian quadrature (Gauss-Legendre, Gauss-Hermite), and Newton-Cotes quadrature (Simpson's rule), QRFF with Sobol sequences, and finally, ORF.

Lastly, we report, for each combination of dimensionality D and kernel lengthscale ℓ, the mean and 95% confidence interval across 5 repetitions.

As expected, for the weight-space approximation as pictured in Figure A.10, having a tighter approximation of the Fourier feature decomposition to the kernel seems to have a large positive effect. Particularly, we see that in low dimensionalities ($D < 5$) with sufficiently large lengthscales, the distances are considerably lower when using QFF with Gauss-Hermite quadrature.

On the other hand, for the samples generated with decoupled sampling as pictured in Figure A.11, the distances are far less discernible from one another.

In the weight-space view, neither the mean nor the variance match that of the exact posterior. However, as we improve the kernel approximation (specifically, as we double the number of quadrature nodes), we observe a dramatic improvement in the approximation. In the last two panes (reading from left to right) with 2^7 and 2^8 nodes, the difference is virtually indistinguishable to the naked eye. In contrast, in decoupled pathwise sampling, we have equality *in expectation* – that is, the mean match up regardless of the quality of the kernel approximation. On the other hand, we do not have equality *in distribution*, so the variance is still dependent on the quality of the kernel approximation. We see that in the very beginning, with a few nodes, it already does a fairly good job of approximating the variance outside the regions in which the the observations are located. On the other hand, inside such regions it appears to severely underestimate the variance. What worse is that it doesn't seen to get better as we improve the kernel

approximation. Indeed, doubling the number of nodes does not seem to change anything.

Increasing the number of nodes does seem to help, but only up to a point. Beyond 2^8 nodes, it is doubtful whether the approximation will improve. It is also unknowable, as this is around the limits of numerical precision. Yet, even with this amount of nodes, the understimation of the variance still persists.

We note, however, that it is difficult to draw conclusions using the 2-Wasserstein distance with empirical distributions. Beyond the numerical stability issues, note that even with samples from the exact GP drawn using the conventional location-scale transform approach, the distance based on empirical estimates are on the order of 10^{-2} (in theory, it should be 0).

Alternatively, it may be worthwhile to instead consider the NLPD of the samples under the exact GP posterior, or using the KL divergence. In particular, the KL divergence between Gaussian distributions with the same mean $\mathbf{m}$ but different covariances $\mathcal{N}(\mathbf{m}, \boldsymbol{\Sigma}_0)$ and $\mathcal{N}(\mathbf{m}, \boldsymbol{\Sigma}_1)$ is

$$\mathrm{KL}\left[\mathcal{N}_0 \parallel \mathcal{N}_1\right] = \frac{1}{2}\left[\ln|\boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_0^{-1}| + \operatorname{tr}\left(\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{\Sigma}_0 - \boldsymbol{\Sigma}_1)\right)\right]. \tag{A.33}$$

Recall from Equations (A.3) and (A.6) that the covariance of a decoupled pathwise sample from a sparse GP posterior is

$$\boldsymbol{\Sigma}_0 \triangleq \boldsymbol{\Phi}_*\boldsymbol{\Phi}_*^\top - 2\mathbf{K}_{*\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\boldsymbol{\Phi}\boldsymbol{\Phi}_*^\top + \mathbf{K}_{*\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\boldsymbol{\Phi}\boldsymbol{\Phi}^\top\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}*},$$

while that of a sparse GP posterior is

$$\boldsymbol{\Sigma}_1 \triangleq \boldsymbol{\Sigma}_{**|\mathbf{u}} = \mathbf{K}_{**} - \mathbf{K}_{*\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}*}.$$

Further, the mean of both is

$$\mathbf{m} \triangleq \boldsymbol{\mu}_{*|\mathbf{u}} = \mathbf{K}_{*\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{u}.$$

Taking the KL divergence between Gaussians with these means and covariances is an attractive alternative to the 2-Wasserstein distance described above, since it is more numerically stable and can be computed analytically without resorting to empirical estimates. In fact, this author perceives no good reason to use empirical estimates, let alone the 2-Wasserstein distance, when we have the exact moments $\mathbf{m}$, $\boldsymbol{\Sigma}_0$, and $\boldsymbol{\Sigma}_1$ readily available to us.

A.5 SUMMARY

We motivated the work summarised in this chapter by showing that the quality of decoupled pathwise samples still depends crucially on the quality of the kernel approximation.

We conducted a survey of existing Fourier feature decompositions for approximating stationary kernels to provide a better understanding

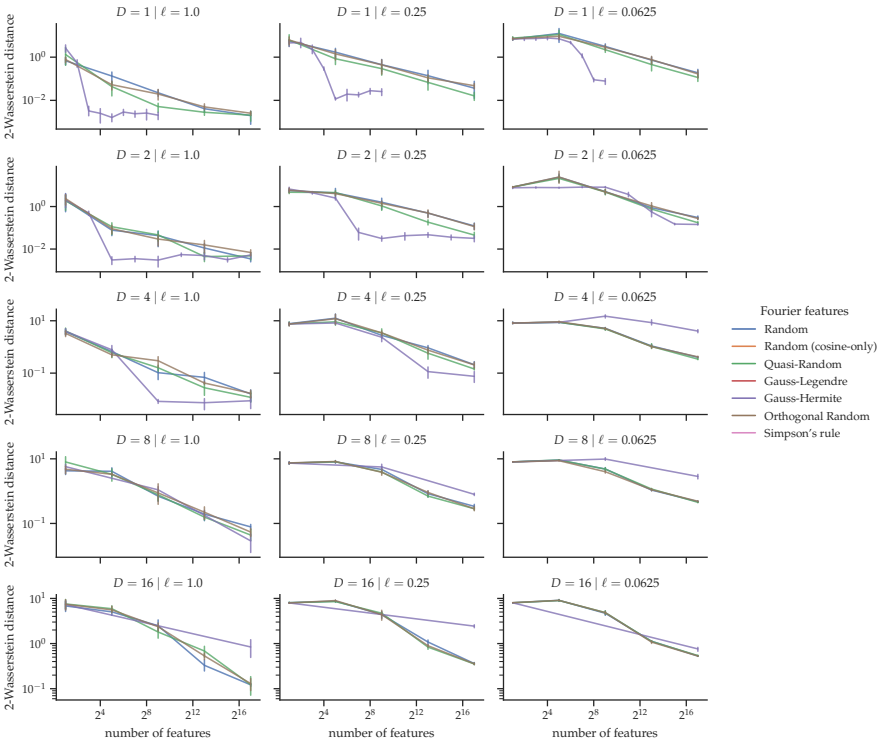


Figure A.10: Weight-space approximate posterior samples (SE kernel).

of the tightness of these various approximations. In doing so, we also made variations on existing schemes to construct new decompositions, or expanded the applicability of existing decompositions to classes of kernels beyond the SE kernel.

We also highlighted a significant shortcoming with an existing class of schemes, namely Gaussian quadrature, in dealing with small lengthscales. Small lengthscales in effect lead to highly-oscillatory integrals that are difficult to approximate. Unfortunately, efforts to ameliorate these shortcomings came up short, as it is not analytically possible to factorise the approximation into an inner product of feature maps. Furthermore, there is evidence to suggest that the benefits of more sophisticated schemes to deal with high-oscillations are only realised at scales well outside the input domains in which we're typically interested for practical purposes.

Lastly, contrary to our motivating hypothesis, we did not find a strong positive correlation between tightness of kernel approximation and quality of decoupled pathwise samples. However, we also underscored the potential flaws of the existing methods to assess sample quality, and emphasised that our empirical findings be taken with a pinch of salt. We suggest in future work that more attention be devoted to devising more principled methods for assessing sample quality.

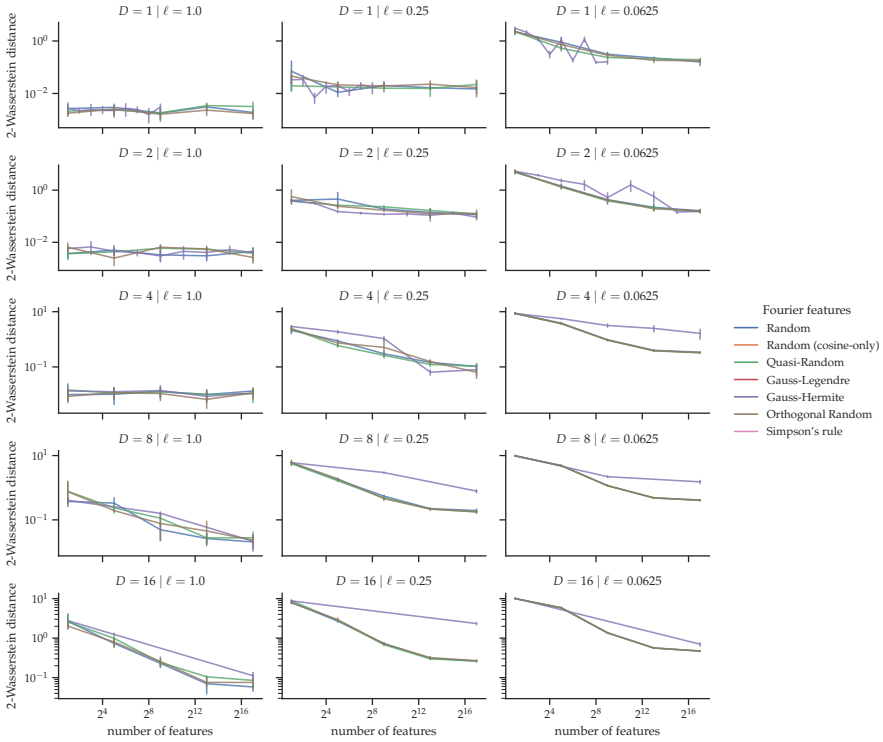


Figure A.11: Samples generated using decoupled sampling (SE kernel).

ADDENDUM

A.A PRODUCT-TO-SUM IDENTITY

The product-to-sum identity, which follows as an immediate consequence of Equation (2.52), is given by

$$2\cos\alpha\cos\beta = \cos(\alpha+\beta) + \cos(\alpha-\beta). \tag{A.34}$$

A.B ZERO IN EXPECTATION

By the *law of total expectation*, we can rewrite the expectation as

$$\begin{aligned}&\mathbb{E}_{p(\boldsymbol{\omega},b)}[\cos(\boldsymbol{\omega}^\top(\mathbf{x}+\mathbf{x}')+2b)]\\ &\quad= \mathbb{E}_{p(\boldsymbol{\omega})}\Big[\mathbb{E}[\cos(\boldsymbol{\omega}^\top(\mathbf{x}+\mathbf{x}')+2b)\,|\,\boldsymbol{\omega}]\Big].\end{aligned}$$

For notational convenience, we set $\theta \triangleq \boldsymbol{\omega}^\top(\mathbf{x}+\mathbf{x}')$. The inner expectation evaluates to

$$\begin{aligned}\mathbb{E}[\cos(\theta+2b)\,|\,\boldsymbol{\omega}] &= \int_0^{2\pi}\cos(\theta+2b)p(b)\,\mathrm{d}b\\ &= \frac{1}{2\pi}\int_0^{2\pi}\cos(\theta+2b)\,\mathrm{d}b\\ &= \frac{1}{2\pi}\sin(\theta+2b)\Big|_0^{2\pi}\\ &= \frac{1}{2\pi}[\sin(\theta+4\pi)-\sin(\theta)] = 0\end{aligned}$$

since the sine function is 2π-periodic, i. e., $\sin(\theta+2\pi\cdot k) = \sin(\theta)$ for any integer k.

BIBLIOGRAPHY

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. "Tensorflow: a system for large-scale machine learning." In: *Osdi*. Vol. 16. 2016. Savannah, GA, USA. 2016, pp. 265–283.

[2] Syed Mumtaz Ali and Samuel D Silvey. "A general class of coefficients of divergence of one distribution from another". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 28.1 (1966), pp. 131–142.

[3] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. "Palm 2 technical report". In: *arXiv preprint arXiv:2305.10403* (2023).

[4] Peter M Attia, Aditya Grover, Norman Jin, Kristen A Severson, Todor M Markov, Yang-Hung Liao, Michael H Chen, Bryan Cheong, Nicholas Perkins, Zi Yang, et al. "Closed-loop optimization of fast-charging protocols for batteries with machine learning". In: *Nature* 578.7795 (2020), pp. 397–402.

[5] Haim Avron, Vikas Sindhwani, Jiyan Yang, and Michael W Mahoney. "Quasi-Monte Carlo feature maps for shift-invariant kernels". In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 4096–4133.

[6] Francis Bach. "On the equivalence between kernel quadrature rules and random feature expansions". In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 714–751.

[7] Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. "BoTorch: A framework for efficient Monte-Carlo Bayesian optimization". In: *Advances in neural information processing systems* 33 (2020), pp. 21524–21538.

[8] David J Bartholomew, Martin Knott, and Irini Moustaki. *Latent variable models and factor analysis: A unified approach*. John Wiley & Sons, 2011.

[9] Matthias Bauer, Mark van der Wilk, and Carl Edward Rasmussen. "Understanding probabilistic sparse Gaussian process approximations". In: *Advances in neural information processing systems* 29 (2016).

[10] Thomas Bayes. "LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFR S". In: *Philosophical transactions of the Royal Society of London* 53 (1763), pp. 370–418.

[11] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. "Mutual information neural estimation". In: *International conference on machine learning*. PMLR. 2018, pp. 531–540.

[12] James O Berger. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013.

[13] James Bergstra and Yoshua Bengio. "Random search for hyper-parameter optimization". In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 281–305.

[14] James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. "Algorithms for hyper-parameter optimization". In: *Advances in Neural Information Processing Systems*. 2011, pp. 2546–2554.

[15] Steffen Bickel, Michael Brückner, and Tobias Scheffer. "Discriminative learning for differing training and test distributions". In: *Proceedings of the 24th international conference on Machine learning*. 2007, pp. 81–88.

[16] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN: 0387310738.

[17] 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.

[18] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. "Weight uncertainty in neural network". In: *International conference on machine learning*. PMLR. 2015, pp. 1613–1622.

[19] Leo Breiman. "Random forests". In: *Machine Learning* 45.1 (2001), pp. 5–32.

[20] Eric Brochu, Vlad M Cora, and Nando De Freitas. "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning". In: *arXiv preprint arXiv:1012.2599* (2010).

[21] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[22] Thang D Bui, Josiah Yan, and Richard E Turner. "A unifying framework for Gaussian process pseudo-point approximations using power expectation propagation". In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 3649–3720.

[23] Thang Duc Bui. "Efficient Deterministic Approximate Bayesian Inference for Gaussian Process models". PhD thesis. University of Cambridge, 2018.

[24] David Burt, Carl Edward Rasmussen, and Mark Van Der Wilk. "Rates of convergence for sparse variational Gaussian process regression". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 862–871.

[25] David R Burt, Carl Edward Rasmussen, and Mark van der Wilk. "Variational orthogonal features". In: *arXiv preprint arXiv:2006.13170* (2020).

[26] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. "A limited memory algorithm for bound constrained optimization". In: *SIAM Journal on scientific computing* 16.5 (1995), pp. 1190–1208.

[27] Russel E Caflisch. "Monte carlo and quasi-monte carlo methods". In: *Acta numerica* 7 (1998), pp. 1–49.

[28] Roberto Calandra, Jan Peters, Carl Edward Rasmussen, and Marc Peter Deisenroth. "Manifold Gaussian processes for regression". In: *2016 International joint conference on neural networks (IJCNN)*. IEEE. 2016, pp. 3338–3345.

[29] Daniele Calandriello, Luigi Carratino, Alessandro Lazaric, Michal Valko, and Lorenzo Rosasco. "Gaussian process optimization with adaptive sketching: Scalable and no regret". In: *Conference on Learning Theory*. PMLR. 2019, pp. 533–557.

[30] Peter E Castro, W H_ Lawton, and EA Sylvestre. "Principal modes of variation for processes with continuous sample curves". In: *Technometrics* 28.4 (1986), pp. 329–337.

[31] Liqun Chen, Shuyang Dai, Yunchen Pu, Erjin Zhou, Chunyuan Li, Qinliang Su, Changyou Chen, and Lawrence Carin. "Symmetric variational autoencoder and connections to adversarial learning". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2018, pp. 661–669.

[32] Peng Chen, Brian M Merrick, and Thomas J Brazil. "Bayesian optimization for broadband high-efficiency power amplifier designs". In: *IEEE Transactions on Microwave Theory and Techniques* 63.12 (2015), pp. 4263–4272.

[33] Tianqi Chen, Emily Fox, and Carlos Guestrin. "Stochastic gradient hamiltonian monte carlo". In: *International conference on machine learning*. PMLR. 2014, pp. 1683–1691.

[34] Tianqi Chen and Carlos Guestrin. "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794.

[35] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. "Infogan: Interpretable representation learning by information maximizing generative adversarial nets". In: *Advances in neural information processing systems* 29 (2016).

[36] Ching-An Cheng and Byron Boots. "Variational inference for Gaussian process models with linear complexity". In: *Advances in Neural Information Processing Systems* 30 (2017).

[37] Kuang Fu Cheng, Chih-Kang Chu, et al. "Semiparametric density estimation under a two-sample density ratio model". In: *Bernoulli* 10.4 (2004), pp. 583–604.

[38] Youngmin Cho and Lawrence Saul. "Kernel methods for deep learning". In: *Advances in neural information processing systems* 22 (2009).

[39] Kristy Choi, Aditya Grover, Trisha Singh, Rui Shu, and Stefano Ermon. "Fair generative modeling via weak supervision". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 1887–1898.

[40] Francois Chollet. *Deep learning with Python*. Simon and Schuster, 2021.

[41] Krzysztof Choromanski, Mark Rowland, Tamás Sarlós, Vikas Sindhwani, Richard Turner, and Adrian Weller. "The geometry of random features". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2018, pp. 1–9.

[42] Krzysztof Choromanski, Mark Rowland, and Adrian Weller. "The unreasonable effectiveness of structured random orthogonal embeddings". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017, pp. 218–227.

[43] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. "A downsampled variant of imagenet as an alternative to the cifar datasets". In: *arXiv preprint arXiv:1707.08819* (2017).

[44] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. "Fast and accurate deep network learning by exponential linear units (elus)". In: *arXiv preprint arXiv:1511.07289* (2015).

[45] Dorin Comaniciu and Peter Meer. "Mean shift: A robust approach toward feature space analysis". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.5 (2002), pp. 603–619.

[46] Lehel Csató and Manfred Opper. "Sparse on-line Gaussian processes". In: *Neural computation* 14.3 (2002), pp. 641–668.

[47] Imre Csiszár. "On information-type measure of difference of probability distributions and indirect observations". In: *Studia Sci. Math. Hungar.* 2 (1967), pp. 299–318.

[48] Imre Csiszár, Paul C Shields, et al. "Information theory and statistics: A tutorial". In: *Foundations and Trends® in Communications and Information Theory* 1.4 (2004), pp. 417–528.

[49] Andreas Damianou and Neil D Lawrence. "Deep gaussian processes". In: *Artificial intelligence and statistics*. PMLR. 2013, pp. 207–215.

[50] Tri Dao, Christopher De Sa, and Christopher Ré. "Gaussian quadrature for kernel features". In: *Advances in neural information processing systems* 30 (2017), p. 6109.

[51] Philip J Davis and Philip Rabinowitz. *Methods of numerical integration*. Courier Corporation, 2007.

[52] Peter Dayan, Geoffrey E Hinton, Radford M Neal, and Richard S Zemel. "The helmholtz machine". In: *Neural computation* 7.5 (1995), pp. 889–904.

[53] George De Ath, Tinkle Chugh, and Alma AM Rahat. "MBORE: multi-objective Bayesian optimisation by density-ratio estimation". In: *Proceedings of the Genetic and Evolutionary Computation Conference*. 2022, pp. 776–785.

[54] Morris H DeGroot. *Optimal statistical decisions*. John Wiley & Sons, 2005.

[55] Marc Deisenroth and Carl E Rasmussen. "PILCO: A model-based and data-efficient approach to policy search". In: *Proceedings of the 28th International Conference on machine learning (ICML-11)*. 2011, pp. 465–472.

[56] Marc Peter Deisenroth, A Aldo Faisal, and Cheng Soon Ong. *Mathematics for machine learning*. Cambridge University Press, 2020.

[57] Amir Dezfouli and Edwin V Bonilla. "Scalable inference for Gaussian process models with black-box likelihoods". In: *Advances in Neural Information Processing Systems* 28 (2015).

[58] Josef Dick, Frances Y Kuo, and Ian H Sloan. "High-dimensional integration: the quasi-Monte Carlo way". In: *Acta Numerica* 22 (2013), pp. 133–288.

[59] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. "Adversarial feature learning". In: *arXiv preprint arXiv:1605.09782* (2016).

[60] Xuanyi Dong and Yi Yang. "Nas-bench-102: Extending the scope of reproducible neural architecture search". In: *arXiv preprint arXiv:2001.00326* (2020).

[61] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: http://archive.ics.uci.edu/ml.

[62] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. "Adversarially learned inference". In: *arXiv preprint arXiv:1606.00704* (2016).

[63] Joseph Duris, Dylan Kennedy, Adi Hanuka, Jane Shtalenkova, Auralee Edelen, P Baxevanis, Adam Egger, T Cope, M McIntire, S Ermon, et al. "Bayesian optimization of a free-electron laser". In: *Physical review letters* 124.12 (2020), p. 124801.

[64] Conor Durkan, Iain Murray, and George Papamakarios. "On contrastive learning for likelihood-free inference". In: *International conference on machine learning*. PMLR. 2020, pp. 2771–2781.

[65] Vincent Dutordoir, Nicolas Durrande, and James Hensman. "Sparse Gaussian processes with spherical harmonic features". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 2793–2802.

[66] Vincent Dutordoir, James Hensman, Mark van der Wilk, Carl Henrik Ek, Zoubin Ghahramani, and Nicolas Durrande. "Deep neural networks as point estimates for deep Gaussian processes". In: *Advances in Neural Information Processing Systems* 34 (2021).

[67] Costas Efthimiou and Christopher Frye. *Spherical harmonics in p dimensions*. World Scientific, 2014.

[68] Stefan Falkner, Aaron Klein, and Frank Hutter. "BOHB: Robust and Efficient Hyperparameter Optimization at Scale". In: *International Conference on Machine Learning*. 2018, pp. 1437–1446.

[69] Louis Napoleon George Filon. "III.–On a quadrature formula for trigonometric integrals". In: *Proceedings of the Royal Society of Edinburgh* 49 (1928), pp. 38–47.

[70] Alexander IJ Forrester and Andy J Keane. "Recent advances in surrogate-based optimization". In: *Progress in aerospace sciences* 45.1-3 (2009), pp. 50–79.

[71] Peter I Frazier. "A tutorial on Bayesian optimization". In: *arXiv preprint arXiv:1807.02811* (2018).

[72] Brendan J Frey and Geoffrey E Hinton. "Variational learning in nonlinear Gaussian belief networks". In: *Neural Computation* 11.1 (1999), pp. 193–213.

[73] Keinosuke Fukunaga. *Introduction to statistical pattern recognition*. Elsevier, 2013.

[74] Yarin Gal and Zoubin Ghahramani. "Dropout as a bayesian approximation: Representing model uncertainty in deep learning". In: *international conference on machine learning*. PMLR. 2016, pp. 1050–1059.

[75] Roman Garnett. *Bayesian Optimization*. to appear. Cambridge University Press, 2023.

[76] Roman Garnett, Michael A Osborne, and Stephen J Roberts. "Bayesian optimization for sensor set selection". In: *Proceedings of the 9th ACM/IEEE international conference on information processing in sensor networks*. 2010, pp. 209–219.

[77] Eduardo C Garrido-Merchán and Daniel Hernández-Lobato. "Dealing with categorical and integer-valued variables in Bayesian optimization with Gaussian processes". In: *Neurocomputing* 380 (2020), pp. 20–35.

[78] Adrià Garriga-Alonso, Carl Edward Rasmussen, and Laurence Aitchison. "Deep convolutional networks as shallow gaussian processes". In: *arXiv preprint arXiv:1808.05587* (2018).

[79] Carl Friedrich Gauss. *Methodvs nova integralivm valores per approximationem inveniendi*. apvd Henricvm Dieterich, 1815.

[80] Walter Gautschi. "Construction of Gauss-Christoffel quadrature formulas". In: *Mathematics of Computation* 22.102 (1968), pp. 251–270.

[81] Walter Gautschi. "A survey of Gauss-Christoffel quadrature formulae". In: *EB Christoffel*. Springer, 1981, pp. 72–147.

[82] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. CRC press, 2013.

[83] Samuel Gershman and Noah Goodman. "Amortized inference in probabilistic reasoning". In: *Proceedings of the annual meeting of the cognitive science society*. Vol. 36. 2014.

[84] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[85] Tilmann Gneiting and Adrian E Raftery. "Strictly proper scoring rules, prediction, and estimation". In: *Journal of the American Statistical Association* 102.477 (2007), pp. 359–378.

[86] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative Adversarial Networks". In: *arXiv preprint arXiv:1406.2661* (2014).

[87] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks". In: *2013 IEEE international conference on acoustics, speech and signal processing*. Ieee. 2013, pp. 6645–6649.

[88] Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. "Covariate shift by kernel mean matching". In: *Dataset Shift in Machine Learning* 3.4 (2009), p. 5.

[89] Ryan-Rhys Griffiths, Alexander A Aldrick, Miguel Garcia-Ortegon, Vidhi Lalchand, et al. "Achieving robustness to aleatoric uncertainty with heteroscedastic Bayesian optimisation". In: *Machine Learning: Science and Technology* 3.1 (2021), p. 015004.

[90] Aditya Grover and Stefano Ermon. "Boosted generative models". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[91] Aditya Grover, Jiaming Song, Ashish Kapoor, Kenneth Tran, Alekh Agarwal, Eric J Horvitz, and Stefano Ermon. "Bias correction of learned generative models using likelihood-free importance weighting". In: *Advances in neural information processing systems* 32 (2019).

[92] Michael U Gutmann and Aapo Hyvärinen. "Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics." In: *Journal of machine learning research* 13.2 (2012).

[93] Heikki Haario, Eero Saksman, and Johanna Tamminen. "Adaptive proposal distribution for random walk Metropolis algorithm". In: *Computational statistics* 14 (1999), pp. 375–395.

[94] Marton Havasi, José Miguel Hernández-Lobato, and Juan José Murillo-Fuentes. "Deep Gaussian processes with decoupled inducing inputs". In: *arXiv preprint arXiv:1801.02939* (2018).

[95] Philipp Hennig, Michael A Osborne, and Hans P Kersting. *Probabilistic Numerics*. Cambridge University Press, 2022.

[96] Philipp Hennig and Christian J Schuler. "Entropy Search for Information-Efficient Global Optimization." In: *Journal of Machine Learning Research* 13.6 (2012).

[97] James Hensman, Nicolas Durrande, and Arno Solin. "Variational Fourier Features for Gaussian Processes". In: *Journal of Machine Learning Research* 18.151 (2018), pp. 1–52. URL: http://jmlr.org/papers/v18/16-579.html.

[98] James Hensman, Nicolo Fusi, and Neil D Lawrence. "Gaussian processes for big data". In: *arXiv preprint arXiv:1309.6835* (2013).

[99] James Hensman, Alexander Matthews, and Zoubin Ghahramani. "Scalable variational Gaussian process classification". In: *Artificial Intelligence and Statistics*. PMLR. 2015, pp. 351–360.

[100] James Hensman, Alexander G Matthews, Maurizio Filippone, and Zoubin Ghahramani. "MCMC for variationally sparse Gaussian processes". In: *Advances in Neural Information Processing Systems* 28 (2015).

[101] Daniel Hernández-Lobato, Jose Hernandez-Lobato, Amar Shah, and Ryan Adams. "Predictive entropy search for multi-objective Bayesian optimization". In: *International Conference on Machine Learning*. 2016, pp. 1492–1501.

[102] José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. "Predictive entropy search for efficient global optimization of black-box functions". In: *Advances in neural information processing systems* 27 (2014).

[103] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups". In: *IEEE Signal processing magazine* 29.6 (2012), pp. 82–97.

[104] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models". In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851.

[105] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. "Stochastic variational inference". In: *Journal of Machine Learning Research* (2013).

[106] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. "Training compute-optimal large language models". In: *arXiv preprint arXiv:2203.15556* (2022).

[107] Kurt Hornik, Maxwell Stinchcombe, Halbert White, et al. "Multilayer feedforward networks are universal approximators." In: *Neural Networks* 2.5 (1989), pp. 359–366.

[108] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. "Bayesian active learning for classification and preference learning". In: *arXiv preprint arXiv:1112.5745* (2011).

[109] Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, and Eric P Xing. "On unifying deep generative models". In: *arXiv preprint arXiv:1706.00550* (2017).

[110] Ferenc Huszár. "Variational inference using implicit distributions". In: *arXiv preprint arXiv:1702.08235* (2017).

[111] Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. "Sequential model-based optimization for general algorithm configuration". In: *International Conference on Learning and Intelligent Optimization*. Springer. 2011, pp. 507–523.

[112] Carl Gustav Jakob Jacobi. "Über Gauss neue Methode, die Werthe der Integrale näherungsweise zu finden." In: (1826).

[113] Achin Jain and Manfred Morari. "Computing the racing line using Bayesian optimization". In: *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE. 2020, pp. 6192–6197.

[114] Eric Jang, Shixiang Gu, and Ben Poole. "Categorical reparameterization with gumbel-softmax". In: *arXiv preprint arXiv:1611.01144* (2016).

[115] Edwin T Jaynes. "Prior probabilities". In: *IEEE Transactions on systems science and cybernetics* 4.3 (1968), pp. 227–241.

[116] Rodolphe Jenatton, Cedric Archambeau, Javier González, and Matthias Seeger. "Bayesian optimization with tree-structured dependencies". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1655–1664.

[117] Donald R Jones, Matthias Schonlau, and William J Welch. "Efficient global optimization of expensive black-box functions". In: *Journal of Global Optimization* 13.4 (1998), pp. 455–492.

[118] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. "An introduction to variational methods for graphical models". In: *Learning in graphical models* (1998), pp. 105–161.

[119] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. "An introduction to variational methods for graphical models". In: *Machine learning* 37 (1999), pp. 183–233.

[120] Andre G Journel and Charles J Huijbregts. "Mining geostatistics". In: (1976).

[121] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (2021), pp. 583–589.

[122] Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. "A least-squares approach to direct importance estimation". In: *The Journal of Machine Learning Research* 10 (2009), pp. 1391–1445.

[123] Takafumi Kanamori, Taiji Suzuki, and Masashi Sugiyama. "Theoretical analysis of density ratio estimation". In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 93.4 (2010), pp. 787–798.

[124] Kirthevasan Kandasamy, Gautam Dasarathy, Jeff Schneider, and Barnabás Póczos. "Multi-fidelity Bayesian optimisation with continuous approximations". In: *arXiv preprint arXiv:1703.06240* (2017).

[125] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. "Learning to discover cross-domain relations with generative adversarial networks". In: *International conference on machine learning*. PMLR. 2017, pp. 1857–1865.

[126] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[127] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[128] Aaron Klein and Frank Hutter. "Tabular benchmarks for joint architecture and hyperparameter optimization". In: *arXiv preprint arXiv:1905.04970* (2019).

[129] Steven Kleinegesse and Michael U Gutmann. "Efficient Bayesian experimental design for implicit models". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 476–485.

[130] Steven Kleinegesse and Michael U Gutmann. "Bayesian experimental design for implicit models by mutual information neural estimation". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 5316–5326.

[131] Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, and Vincent Fortuin. "Promises and Pitfalls of the Linearized Laplace in Bayesian Optimization". In: *arXiv preprint arXiv:2304.08309* (2023).

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

[133] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012).

[134] Harold J Kushner. "A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise". In: *Joint Automatic Control Conference*. 1. 1963, pp. 69–79.

[135] Tze Leung Lai, Herbert Robbins, et al. "Asymptotically efficient adaptive allocation rules". In: *Advances in applied mathematics* 6.1 (1985), pp. 4–22.

[136] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and scalable predictive uncertainty estimation using deep ensembles". In: *Advances in neural information processing systems* 30 (2017).

[137] Rémi Lam, Matthias Poloczek, Peter Frazier, and Karen E Willcox. "Advances in Bayesian optimization with applications in aerospace engineering". In: *2018 AIAA Non-Deterministic Approaches Conference*. 2018, p. 1656.

[138] Pierre Simon Laplace. *Théorie analytique des probabilités*. Courcier, 1814.

[139] Harri Lappalainen and Antti Honkela. "Bayesian non-linear independent component analysis by multi-layer perceptrons". In: *Advances in independent component analysis* (2000), pp. 93–121.

[140] Miguel Lázaro-Gredilla and Anibal Figueiras-Vidal. "Inter-domain Gaussian processes for sparse inference using inducing features". In: *Advances in Neural Information Processing Systems* 22 (2009).

[141] Quoc Le, Tamás Sarlós, Alex Smola, et al. "Fastfood-approximating kernel expansions in loglinear time". In: *Proceedings of the international conference on machine learning*. Vol. 85. 2013.

[142] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. "Efficient backprop". In: *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 9–48.

[143] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. "Deep neural networks as gaussian processes". In: *arXiv preprint arXiv:1711.00165* (2017).

[144] Chunyuan Li, Hao Liu, Changyou Chen, Yuchen Pu, Liqun Chen, Ricardo Henao, and Lawrence Carin. "Alice: Towards understanding adversarial learning for joint distribution matching". In: *Advances in neural information processing systems* 30 (2017).

[145] Yucen Lily Li, Tim GJ Rudner, and Andrew Gordon Wilson. "A Study of Bayesian Neural Network Surrogates for Bayesian Optimization". In: *arXiv preprint arXiv:2305.20028* (2023).

[146] Friedrich Liese and Igor Vajda. "On divergences and informations in statistics and information theory". In: *IEEE Transactions on Information Theory* 52.10 (2006), pp. 4394–4412.

[147] 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: *arXiv preprint arXiv:1509.02971* (2015).

[148] Alexander Liniger, Alexander Domahidi, and Manfred Morari. "Optimization-based autonomous racing of 1: 43 scale RC cars". In: *Optimal Control Applications and Methods* 36.5 (2015), pp. 628–647.

[149] Thomas Lipp and Stephen Boyd. "Minimum-time speed optimisation over a fixed path". In: *International Journal of Control* 87.6 (2014), pp. 1297–1311.

[150] Dong C Liu and Jorge Nocedal. "On the limited memory BFGS method for large scale optimization". In: *Mathematical Programming* 45.1-3 (1989), pp. 503–528.

[151] Qiang Liu and Dilin Wang. "Stein variational gradient descent: A general purpose bayesian inference algorithm". In: *Advances in neural information processing systems* 29 (2016).

[152] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. "Deep learning face attributes in the wild". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 3730–3738.

[153] Wenlong Lyu, Pan Xue, Fan Yang, Changhao Yan, Zhiliang Hong, Xuan Zeng, and Dian Zhou. "An efficient bayesian optimization approach for automated optimization of analog circuits". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 65.6 (2017), pp. 1954–1967.

[154] David JC MacKay. "A practical Bayesian framework for backpropagation networks". In: *Neural computation* 4.3 (1992), pp. 448–472.

[155] David JC MacKay. "The evidence framework applied to classification networks". In: *Neural computation* 4.5 (1992), pp. 720–736.

[156] David JC MacKay et al. "Introduction to Gaussian processes". In: *NATO ASI series F computer and systems sciences* 168 (1998), pp. 133–166.

[157] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[158] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. "The concrete distribution: A continuous relaxation of discrete random variables". In: *arXiv preprint arXiv:1611.00712* (2016).

[159] Anastasia Makarova, Ilnura Usmanova, Ilija Bogunovic, and Andreas Krause. "Risk-averse heteroscedastic bayesian optimization". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 17235–17245.

[160] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. "Adversarial autoencoders". In: *arXiv preprint arXiv:1511.05644* (2015).

[161] Anton Mallasto and Aasa Feragen. "Learning from uncertain curves: The 2-Wasserstein metric for Gaussian processes". In: *Advances in Neural Information Processing Systems* 30 (2017).

[162] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. "Least squares generative adversarial networks". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2794–2802.

[163] Roman Marchant and Fabio Ramos. "Bayesian optimisation for intelligent environmental monitoring". In: *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. 2012, pp. 2242–2249.

[164] Bertil Matérn. "Spatial variation: Stachastic models and their application to some problems in forest surveys and other sampling investigations". PhD thesis. Stockholm University, 1960.

[165] 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: *Artificial Intelligence and Statistics*. PMLR. 2016, pp. 231–239.

[166] Alexander G de G Matthews, Mark Rowland, Jiri Hron, Richard E Turner, and Zoubin Ghahramani. "Gaussian process behaviour in wide deep neural networks". In: *arXiv preprint arXiv:1804.11271* (2018).

[167] Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke. Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. "GPflow: A Gaussian process library using TensorFlow". In: *Journal of Machine Learning Research* 18.40 (Apr. 2017), pp. 1–6. URL: http://jmlr.org/papers/v18/16-537.html.

[168] Alexander Graeme de Garis Matthews. "Scalable Gaussian process inference using variational methods". PhD thesis. University of Cambridge, 2017.

[169] Warren S McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5 (1943), pp. 115–133.

[170] Aditya Menon and Cheng Soon Ong. "Linking losses for density ratio and class-probability estimation". In: *International Conference on Machine Learning*. 2016, pp. 304–313.

[171] Lassi Meronen, Christabella Irwanto, and Arno Solin. "Stationary activations for uncertainty calibration in deep learning". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 2338–2350.

[172] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. "Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks". In: *International conference on machine learning*. PMLR. 2017, pp. 2391–2400.

[173] Thomas P Minka. "Expectation propagation for approximate Bayesian inference". In: *arXiv preprint arXiv:1301.2294* (2013).

[174] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. "Playing atari with deep reinforcement learning". In: *arXiv preprint arXiv:1312.5602* (2013).

[175] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. "Human-level control through deep reinforcement learning". In: *nature* 518.7540 (2015), pp. 529–533.

[176] Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. "The application of Bayesian methods for seeking the extremum". In: *Towards Global Optimization* 2.117-129 (1978), p. 2.

[177] Shakir Mohamed and Balaji Lakshminarayanan. "Learning in implicit generative models". In: *arXiv preprint arXiv:1610.03483* (2016).

[178] Henry B Moss, Daniel Beck, Javier González, David S Leslie, and Paul Rayson. "BOSS: Bayesian Optimization over String Spaces". In: *arXiv preprint arXiv:2010.00979* (2020).

[179] Henry B Moss, Sebastian W Ober, and Victor Picheny. "Inducing point allocation for sparse gaussian processes in high-throughput bayesian optimisation". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2023, pp. 5213–5230.

[180] Marina Munkhoeva, Yermek Kapushev, Evgeny Burnaev, and Ivan Oseledets. "Quadrature-based features for kernel approximation". In: *arXiv preprint arXiv:1802.03832* (2018).

[181] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[182] Iain Murray and Ryan P Adams. "Slice sampling covariance hyperparameters of latent Gaussian models". In: *Advances in neural information processing systems* 23 (2010).

[183] Mojmir Mutny and Andreas Krause. "Efficient high dimensional bayesian optimization with additivity and quadrature fourier features". In: *Advances in Neural Information Processing Systems* 31 (2018).

[184] Mojmír Mutnỳ and Andreas Krause. "Efficient high dimensional Bayesian optimization with additivity and quadrature Fourier features". In: *Advances in Neural Information Processing Systems 31* (2019), pp. 9005–9016.

[185] Radford M Neal. "BAYESIAN LEARNING FOR NEURAL NETWORKS". PhD thesis. University of Toronto, 1995.

[186] Radford M Neal. "Slice sampling". In: *The annals of statistics* 31.3 (2003), pp. 705–767.

[187] Radford M Neal et al. "MCMC using Hamiltonian dynamics". In: *Handbook of markov chain monte carlo* 2.11 (2011), p. 2.

[188] XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. "Estimating divergence functionals and the likelihood ratio by convex risk minimization". In: *IEEE Transactions on Information Theory* 56.11 (2010), pp. 5847–5861.

[189] Alexandru Niculescu-Mizil and Rich Caruana. "Predicting good probabilities with supervised learning". In: *Proceedings of the 22nd International Conference on Machine Learning*. 2005, pp. 625–632.

[190] Roman Novak, Lechao Xiao, Jaehoon Lee, Yasaman Bahri, Greg Yang, Jiri Hron, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. "Bayesian deep convolutional networks with many channels are gaussian processes". In: *arXiv preprint arXiv:1810.05148* (2018).

[191] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. "f-gan: Training generative neural samplers using variational divergence minimization". In: *Advances in neural information processing systems* 29 (2016).

[192] Sebastian W Ober, Carl E Rasmussen, and Mark van der Wilk. "The promises and pitfalls of deep kernel learning". In: *Uncertainty in Artificial Intelligence*. PMLR. 2021, pp. 1206–1216.

[193] Rafael Oliveira, Louis C Tiao, and Fabio T Ramos. "Batch Bayesian Optimisation via Density-Ratio Estimation with Guarantees". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 29816–29829. URL: https://bit.ly/oliveira2022batch.

[194] R OpenAI. "GPT-4 technical report". In: *arXiv* (2023), pp. 2303–08774.

[195] Manfred Opper and Ole Winther. "Gaussian processes and SVM: Mean field results and leave-one-out". In: (2000).

[196] Byeong U Park and James S Marron. "Comparison of data-driven bandwidth selectors". In: *Journal of the American Statistical Association* 85.409 (1990), pp. 66–72.

[197] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems* 32 (2019).

[198] Karl Pearson. "LIII. On lines and planes of closest fit to systems of points in space". In: *The London, Edinburgh, and Dublin philosophical magazine and journal of science* 2.11 (1901), pp. 559–572.

[199] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[200] Valerio Perrone, Rodolphe Jenatton, Matthias W Seeger, and Cédric Archambeau. "Scalable hyperparameter transfer learning". In: *Advances in neural information processing systems* 31 (2018).

[201] John Platt et al. "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods". In: *Advances in Large Margin Classifiers* 10.3 (1999), pp. 61–74.

[202] Yuchen Pu, Weiyao Wang, Ricardo Henao, Liqun Chen, Zhe Gan, Chunyuan Li, and Lawrence Carin. "Adversarial symmetric variational autoencoder". In: *Advances in neural information processing systems* 30 (2017).

[203] Jing Qin. "Inferences for case-control and semiparametric two-sample density ratio models". In: *Biometrika* 85.3 (1998), pp. 619–630.

[204] Joaquin Quinonero-Candela and Carl Edward Rasmussen. "A unifying view of sparse approximate Gaussian process regression". In: *The Journal of Machine Learning Research* 6 (2005), pp. 1939–1959.

[205] Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. "Scaling language models: Methods, analysis & insights from training gopher". In: *arXiv preprint arXiv:2112.11446* (2021).

[206] Ali Rahimi and Benjamin Recht. "Random Features for Large-Scale Kernel Machines." In: *NIPS*. Vol. 3. 4. Citeseer. 2007, p. 5.

[207] Ali Rahimi and Benjamin Recht. "Weighted Sums of Random Kitchen Sinks: Replacing minimization with randomization in learning". In: *Advances in Neural Information Processing Systems* 21 (2008), pp. 1313–1320.

[208] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. "Hierarchical text-conditional image generation with clip latents". In: *arXiv preprint arXiv:2204.06125* 1.2 (2022), p. 3.

[209] Rajesh Ranganath, Sean Gerrish, and David Blei. "Black box variational inference". In: *Artificial intelligence and statistics*. PMLR. 2014, pp. 814–822.

[210] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. "Regularized evolution for image classifier architecture search". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 4780–4789.

[211] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[212] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic backpropagation and approximate inference in deep generative models". In: *International conference on machine learning*. PMLR. 2014, pp. 1278–1286.

[213] R Tyrrell Rockafellar. *Convex analysis*. Vol. 11. Princeton university press, 1997.

[214] Joseph Lee Rodgers, W Alan Nicewander, and Larry Toothaker. "Linearly independent, orthogonal, and uncorrelated variables". In: *The American Statistician* 38.2 (1984), pp. 133–134.

[215] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-resolution image synthesis with latent diffusion models". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 10684–10695.

[216] Philip A Romero, Andreas Krause, and Frances H Arnold. "Navigating the protein fitness landscape with Gaussian processes". In: *Proceedings of the National Academy of Sciences* 110.3 (2013), E193–E201.

[217] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer. 2015, pp. 234–241.

[218] Lorenzo Rosasco, Ernesto De Vito, Andrea Caponnetto, Michele Piana, and Alessandro Verri. "Are loss functions all the same?" In: *Neural computation* 16.5 (2004), pp. 1063–1076.

[219] Frank Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6 (1958), p. 386.

[220] Ugo Rosolia and Francesco Borrelli. "Learning how to autonomously race a car: a predictive control approach". In: *IEEE Transactions on Control Systems Technology* 28.6 (2019), pp. 2713–2719.

[221] Sam Roweis and Zoubin Ghahramani. "A unifying review of linear Gaussian models". In: *Neural computation* 11.2 (1999), pp. 305–345.

[222] Amir Saffari, Christian Leistner, Jakob Santner, Martin Godec, and Horst Bischof. "On-line random forests". In: *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*. IEEE. 2009, pp. 1393–1400.

[223] Hugh Salimbeni, Ching-An Cheng, Byron Boots, and Marc Deisenroth. "Orthogonally decoupled variational Gaussian processes". In: *Advances in Neural Information Processing Systems* 31 (2018).

[224] David Salinas, Matthias Seeger, Aaron Klein, Valerio Perrone, Martin Wistuba, and Cedric Archambeau. "Syne tune: A library for large scale hyperparameter tuning and reproducible research". In: *International Conference on Automated Machine Learning*. PMLR. 2022, pp. 16–1.

[225] Warren Scott, Peter Frazier, and Warren Powell. "The correlated knowledge gradient for simulation optimization of continuous parameters using gaussian process regression". In: *SIAM Journal on Optimization* 21.3 (2011), pp. 996–1026.

[226] Matthias W Seeger, Christopher KI Williams, and Neil D Lawrence. "Fast forward selection to speed up sparse Gaussian process regression". In: *International Workshop on Artificial Intelligence and Statistics*. PMLR. 2003, pp. 254–261.

[227] Atsuto Seko, Atsushi Togo, Hiroyuki Hayashi, Koji Tsuda, Laurent Chaput, and Isao Tanaka. "Prediction of low-thermal-conductivity compounds with first-principles anharmonic lattice-dynamics calculations and Bayesian optimization". In: *Physical review letters* 115.20 (2015), p. 205901.

[228] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. "Taking the human out of the loop: A review of Bayesian optimization". In: *Proceedings of the IEEE* 104.1 (2015), pp. 148–175.

[229] Simon J Sheather and Michael C Jones. "A reliable data-based bandwidth selection method for kernel density estimation". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 53.3 (1991), pp. 683–690.

[230] Jiaxin Shi, Michalis Titsias, and Andriy Mnih. "Sparse orthogonal variational inference for Gaussian processes". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 1932–1942.

[231] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. "Megatron-lm: Training multi-billion parameter language models using model parallelism". In: *arXiv preprint arXiv:1909.08053* (2019).

[232] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. "Mastering the game of Go with deep neural networks and tree search". In: *nature* 529.7587 (2016), pp. 484–489.

[233] Bernard W Silverman. *Density estimation for statistics and data analysis*. Vol. 26. CRC Press, 1986.

[234] Edward Snelson and Zoubin Ghahramani. "Sparse Gaussian processes using pseudo-inputs". In: *Advances in neural information processing systems* 18 (2005).

[235] Edward Snelson and Zoubin Ghahramani. "Local and global sparse Gaussian process approximations". In: *Artificial Intelligence and Statistics*. PMLR. 2007, pp. 524–531.

[236] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. "Practical Bayesian optimization of machine learning algorithms". In: *Advances in Neural Information Processing Systems* 25 (2012), pp. 2951–2959.

[237] Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Mostofa Patwary, Mr Prabhat, and Ryan Adams. "Scalable bayesian optimization using deep neural networks". In: *International conference on machine learning*. PMLR. 2015, pp. 2171–2180.

[238] Jasper Snoek, Kevin Swersky, Rich Zemel, and Ryan Adams. "Input warping for Bayesian optimization of non-stationary functions". In: *International Conference on Machine Learning*. PMLR. 2014, pp. 1674–1682.

[239] Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. "Amortised map inference for image super-resolution". In: *arXiv preprint arXiv:1610.04490* (2016).

[240] Jiaming Song and Stefano Ermon. "Likelihood-free Density Ratio Acquisition Functions are not Equivalent to Expected Improvements". In: *Bayesian Deep Learning Workshop at NeurIPS*. 2021.

[241] Jiaming Song, Lantao Yu, Willie Neiswanger, and Stefano Ermon. "A general recipe for likelihood-free Bayesian optimization". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 20384–20404.

[242] C Spearman. ""General Intelligence," Objectively Determined and Measured". In: *The American Journal of Psychology* 15.2 (1904), pp. 201–292.

[243] Jost Tobias Springenberg, Aaron Klein, Stefan Falkner, and Frank Hutter. "Bayesian optimization with robust Bayesian neural networks". In: *Advances in neural information processing systems* 29 (2016).

[244] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. "Gaussian process optimization in the bandit setting: No regret and experimental design". In: *arXiv preprint arXiv:0912.3995* (2009).

[245] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. "Veegan: Reducing mode collapse in gans using implicit variational learning". In: *Advances in neural information processing systems* 30 (2017).

[246] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[247] Michael L Stein. *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media, 1999.

[248] Josef Stoer and Roland Bulirsch. *Introduction to numerical analysis*. Vol. 12. Springer Science & Business Media, 2013.

[249] R. Storn and K. Price. "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces". In: *Journal of Global Optimization* (1997).

[250] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul V Buenau, and Motoaki Kawanabe. "Direct importance estimation with model selection and its application to covariate shift adaptation". In: *Advances in Neural Information Processing Systems*. 2008, pp. 1433–1440.

[251] Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density Ratio Estimation in Machine Learning*. Cambridge University Press, 2012.

[252] Shengyang Sun, Jiaxin Shi, and Roger Baker Grosse. "Neural Networks as Inter-Domain Inducing Points". In: *Third Symposium on Advances in Approximate Bayesian Inference*. 2020.

[253] Shengyang Sun, Jiaxin Shi, Andrew Gordon Gordon Wilson, and Roger B Grosse. "Scalable Variational Gaussian Processes via Harmonic Kernel Decomposition". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 9955–9965.

[254] Dougal J Sutherland and Jeff Schneider. "On the error of random Fourier features". In: *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*. 2015, pp. 862–871.

[255] Kevin Swersky, Jasper Snoek, and Ryan P Adams. "Multi-task Bayesian optimization". In: *Advances in Neural Information Processing Systems*. 2013, pp. 2004–2012.

[256] George R Terrell and David W Scott. "Variable kernel density estimation". In: *The Annals of Statistics* (1992), pp. 1236–1265.

[257] Owen Thomas, Ritabrata Dutta, Jukka Corander, Samuel Kaski, and Michael U Gutmann. "Likelihood-free inference by ratio estimation". In: *Bayesian Analysis* 17.1 (2022), pp. 1–31.

[258] William R Thompson. "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples". In: *Biometrika* 25.3-4 (1933), pp. 285–294.

[259] Louis C Tiao, Aaron Klein, Matthias W Seeger, Edwin V Bonilla, Cedric Archambeau, and Fabio T Ramos. "BORE: Bayesian Optimization by Density-Ratio Estimation". 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. 10289–10300. URL: https://proceedings.mlr.press/v139/tiao21a.html. **(Accepted as *Oral* presentation).**

[260] Robert Tibshirani. "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 58.1 (1996), pp. 267–288.

[261] Michael E Tipping and Christopher M Bishop. "Probabilistic principal component analysis". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.3 (1999), pp. 611–622.

[262] Michalis Titsias. "Variational learning of inducing variables in sparse Gaussian processes". In: *Artificial Intelligence and Statistics*. PMLR. 2009, pp. 567–574.

[263] Michalis K Titsias. "Variational model selection for sparse Gaussian process regression". In: *Report, University of Manchester, UK* (2009).

[264] Michalis K Titsias. "Learning model reparametrizations: Implicit variational inference by fitting MCMC distributions". In: *arXiv preprint arXiv:1708.01529* (2017).

[265] Hakki Mert Torun, Madhavan Swaminathan, Anto Kavungal Davis, and Mohamed Lamine Faycal Bellaredj. "A global Bayesian optimization algorithm and its application to integrated system design". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 26.4 (2018), pp. 792–802.

[266] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. "Llama 2: Open foundation and fine-tuned chat models". In: *arXiv preprint arXiv:2307.09288* (2023).

[267] Dustin Tran, Rajesh Ranganath, and David Blei. "Hierarchical implicit models and likelihood-free variational inference". In: *Advances in Neural Information Processing Systems* 30 (2017).

[268] Yuta Tsuboi, Hisashi Kashima, Shohei Hido, Steffen Bickel, and Masashi Sugiyama. "Direct density ratio estimation for large-scale covariate shift adaptation". In: *Journal of Information Processing* 17 (2009), pp. 138–155.

[269] Zhuowen Tu. "Learning generative models via discriminative approaches". In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2007, pp. 1–8.

[270] Ryan Turner, David Eriksson, Michael McCourt, Juha Kiili, Eero Laaksonen, Zhen Xu, and Isabelle Guyon. "Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020". In: *NeurIPS 2020 Competition and Demonstration Track*. PMLR. 2021, pp. 3–26.

[271] Masatoshi Uehara, Issei Sato, Masahiro Suzuki, Kotaro Nakayama, and Yutaka Matsuo. "Generative adversarial nets from a density ratio estimation perspective". In: *arXiv preprint arXiv:1610.02920* (2016).

[272] Mark Van der Wilk. "Sparse Gaussian process approximations and applications". PhD thesis. University of Cambridge, 2019.

[273] Vladimir Vapnik. *The nature of statistical learning theory*. Springer Science & Business Media, 2013.

[274] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[275] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[276] Martin J Wainwright, Michael I Jordan, et al. "Graphical models, exponential families, and variational inference". In: *Foundations and Trends® in Machine Learning* 1.1–2 (2008), pp. 1–305.

[277] Ke Wang, Geoff Pleiss, Jacob Gardner, Stephen Tyree, Kilian Q Weinberger, and Andrew Gordon Wilson. "Exact Gaussian processes on a million data points". In: *Advances in neural information processing systems* 32 (2019).

[278] Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. "Batched large-scale Bayesian optimization in high-dimensional spaces". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2018, pp. 745–754.

[279] Zi Wang and Stefanie Jegelka. "Max-value entropy search for efficient Bayesian optimization". In: *arXiv preprint arXiv:1703.01968* (2017).

[280] Florian Wenzel, Jasper Snoek, Dustin Tran, and Rodolphe Jenatton. "Hyperparameter ensembles for robustness and uncertainty quantification". In: *arXiv preprint arXiv:2006.13570* (2020).

[281] Colin White, Willie Neiswanger, and Yash Savani. "Bananas: Bayesian optimization with neural architectures for neural architecture search". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 12. 2021, pp. 10293–10301.

[282] Paul B Wigley, Patrick J Everitt, Anton van den Hengel, John W Bastian, Mahasen A Sooriyabandara, Gordon D McDonald, Kyle S Hardman, Ciaron D Quinlivan, P Manju, Carlos CN Kuhn, et al. "Fast machine-learning online optimization of ultra-cold-atom experiments". In: *Scientific reports* 6.1 (2016), p. 25890.

[283] Mark van der Wilk, Vincent Dutordoir, ST John, Artem Artemev, Vincent Adam, and James Hensman. "A framework for interdomain and multioutput Gaussian processes". In: *arXiv preprint arXiv:2003.01115* (2020).

[284] Christopher Williams. "Computing with infinite networks". In: *Advances in neural information processing systems* 9 (1996).

[285] Christopher KI Williams and David Barber. "Bayesian classification with Gaussian processes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.12 (1998), pp. 1342–1351.

[286] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. Vol. 2. 3. MIT press Cambridge, MA, 2006.

[287] Christopher KI Williams and Matthias Seeger. "Using the Nyström method to speed up kernel machines". In: *Proceedings of the 13th International Conference on Neural Information Processing Systems*. 2000, pp. 661–667.

[288] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. "Deep kernel learning". In: *Artificial intelligence and statistics*. PMLR. 2016, pp. 370–378.

[289] James Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Deisenroth. "Efficiently sampling functions from Gaussian process posteriors". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 10292–10302.

[290] James Wilson, Frank Hutter, and Marc Deisenroth. "Maximizing acquisition functions for Bayesian optimization". In: *Advances in Neural Information Processing Systems* 31 (2018), pp. 9884–9895.

[291] James T Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Peter Deisenroth. "Pathwise conditioning of gaussian processes". In: *The Journal of Machine Learning Research* 22.1 (2021), pp. 4741–4787.

[292] Makoto Yamada, Taiji Suzuki, Takafumi Kanamori, Hirotaka Hachiya, and Masashi Sugiyama. "Relative density-ratio estimation for robust distribution comparison". In: *Advances in Neural Information Processing Systems*. 2011, pp. 594–602.

[293] Greg Yang. "Wide feedforward or recurrent neural networks of any architecture are gaussian processes". In: *Advances in Neural Information Processing Systems* 32 (2019).

[294] Jiyan Yang, Vikas Sindhwani, Haim Avron, and Michael Mahoney. "Quasi-Monte Carlo feature maps for shift-invariant kernels". In: *International Conference on Machine Learning*. PMLR. 2014, pp. 485–493.

[295] Kevin K Yang, Zachary Wu, and Frances H Arnold. "Machine-learning-guided directed evolution for protein engineering". In: *Nature methods* 16.8 (2019), pp. 687–694.

[296] Tianbao Yang, Yu-Feng Li, Mehrdad Mahdavi, Rong Jin, and Zhi-Hua Zhou. "Nyström method vs random fourier features: A theoretical and empirical comparison". In: *Advances in neural information processing systems* 25 (2012), pp. 476–484.

[297] Zichao Yang, Andrew Wilson, Alex Smola, and Le Song. "A la carte–learning fast kernels". In: *Artificial Intelligence and Statistics*. PMLR. 2015, pp. 1098–1106.

[298] Felix Xinnan X Yu, Ananda Theertha Suresh, Krzysztof M Choromanski, Daniel N Holtmann-Rice, and Sanjiv Kumar. "Orthogonal random features". In: *Advances in neural information processing systems* 29 (2016), pp. 1975–1983.

[299] Bianca Zadrozny and Charles Elkan. "Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers". In: *ICML*. Vol. 1. Citeseer. 2001, pp. 609–616.

[300] Bianca Zadrozny and Charles Elkan. "Transforming classifier scores into accurate multiclass probability estimates". In: *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2002, pp. 694–699.

[301] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization". In: *ACM Transactions on mathematical software (TOMS)* 23.4 (1997), pp. 550–560.

[302] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2223–2232.