Diagnostic Methods for Bayesian Inference

Hanwen Xing
St Cross College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

December 2022

Statement of Originality

This thesis is my own work except as specified in the text and the attached Statement of Authorship forms. I declare the materials in this thesis have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university.

Hanwen Xing

October 2022

Acknowledgements

I would like to thank my supervisor Professor Geoff Nicholls for his enthusiasm and guidance. I would like to thank my family and friends for their support. I would also like to thank all my colleagues and collaborators for the many enlightening and constructive discussions that we have had.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Bayesian statistics provides a systematic approach to update the knowledge about parameters in a statistical model using the information in observed data. Given finite observed data y_{obs}, the Bayesian inference pipeline typically proceeds by 1) specifying a statistical model $p(y|\phi)$ alongside with the corresponding likelihood function, where ϕ denotes the parameters governing the statistical model, 2) specifying a prior distribution on ϕ which appropriately incorporates the practitioner's prior knowledge on it, and 3) updating the knowledge about the parameters by inferring the posterior distribution of ϕ conditioned on y_{obs} using Bayes theorem.

This Bayesian inferential framework provides a principled way to combine the observational model with any prior information. Despite of its natural formulation, carrying out the inferential procedure is a challenging task from a computational perspective. Various approaches have been proposed to address the computational challenges arising from different aspects of the inferential framework such as posterior sampling, evidence estimation and model averaging. For example, Monte Carlo methods such as Markov Chain Monte Carlo (Andrieu et al., 2003; Neal et al., 2011), Importance sampling and Sequential Monte Carlo (Kahn, 1950; Landau and Binder, 2021; Doucet et al., 2001; Chopin et al., 2020) have been widely used to approximately draw samples from the posterior distributions. In addition to the problem of posterior sampling, various computational techniques (Chib, 1995; Meng and Wong, 1996; Gelman and Meng, 1998; Friel and Pettitt, 2008; Wang et al., 2022; Dai and Liu, 2022) have also been proposed for estimating marginal likelihood or evidence, a key quantity for model diagnostic and comparison (Kass and Raftery, 1995; Raftery, 1999).

Recent developments in machine learning shed new light on the computational challenges in Bayesian statistics. On one hand, new model architectures such as Normalizing flow (Papamakarios et al., 2021) have been successfully applied to probabilistic modelling and inference problems such as variational inference (Rezende and

Mohamed, 2015), density estimation (Papamakarios et al., 2017) and evidence estimation (Jia and Seljak, 2020). On the other hand, powerful and versatile computational platforms and libraries such as `jax` (Bradbury et al., 2018), `Pytorch` (Paszke et al., 2017) and `Tensorflow` (Abadi et al., 2016) allow the design and training of complex models to be easily and efficiently implemented. In this thesis, we focus on tackling computational challenges arising from Bayesian inference with the help of recent developments in machine learning. For instance, in Chapter 2, we are interested in applying machine learning methods to the problem of Bayesian model diagnostics and comparison. In particular, we consider the problem of marginal likelihood and Bayes factor estimation, which can be computationally costly and challenging depending on the complexity and dimensionality of the statistical models. In Chapter 2, we use flexible generative models to guide us finding statistically more efficient estimators of Bayes factor.

In addition to Bayes factor estimation, we also use machine learning approaches to construct generic diagnostic tools for approximate Bayesian inference. Since posterior sampling in Bayesian inference is almost always carried out through approximate methods, generic computational frameworks capable of accurately capturing the approximation error in an approximate inference procedure are important for practitioners who make decisions based on those approximate results. In Chapter 3 and 4, we discuss various strategies to design generic diagnostic tools using methods taken from machine learning literature.

In brief, the main contributions of this thesis can be summarized as

- A novel computational framework for estimating the asymptotic error and improving the statistical efficiency of the optimal Bridge estimator, a Monte Carlo estimator of the ratio of normalizing constants between two unnormalized densities. This method is applicable to Bayesian model diagnostics and Bayes factor estimation.

- A generic diagnostic tool for estimating the approximation error in approximate credible sets. This method can be used to calibrate the uncertainty estimation in approximate Bayesian inference.

- A generic diagnostic tool for detecting approximation error in the marginals of approximate posteriors. It is a easy-to-interpret visual diagnostic tool which can be used to identify the details of approximation error in the marginal approximate posteriors.

1.1 Thesis outline

This thesis is in an integrated format and consists of five chapters, with the first chapter being an introduction, and the last chapter being conclusion and further discussion. Each of the remaining three chapters is based on a published paper, and hence is self-contained and consists of a separate introduction and literature review specific to the topics it covers. In this section, we give a general overview of Chapter 2, 3, 4 of the thesis.

Chapter 2 is based on Xing (2022). In Chapter 2, we focus on estimating the asymptotic error and improving the statistical efficiency of the optimal Bridge estimator (Bennett, 1976; Meng and Wong, 1996), a Monte Carlo estimator of the ratio of normalizing constants between two unnormalized densities, using machine learning approaches. In this chapter, we first propose a new estimator of the asymptotic relative mean square error (RMSE) of the optimal Bridge estimator. We show estimating the RMSE of the optimal Bridge estimator is equivalent to estimating an f-divergence between the two densities using the variational framework given in Nguyen et al. (2010). We then give a new computational framework that aims to find a bijective transformation that maps one density to the other and directly minimizes the asymptotic RMSE of the optimal Bridge estimator with respect to the transformed density using an f-GAN (Nowozin et al., 2016). Our proposed method is optimal in the sense that asymptotically, it can achieve a RMSE lower than that achieved by Bridge estimators based on any transformed density within the class of densities generated by the candidate transformations we consider. In contrast, existing improvement strategies such as Jia and Seljak (2020) and Wang et al. (2022) do not offer such guarantee. Simulation studies show that our proposed estimator achieves state-of-the-art accuracy, and outperforms existing methods. In addition, we also discuss the connection between f-divergence estimation and Bridge estimators, and show how Bridge estimators naturally arise from the problem of estimating an f-divergence between two probability distributions. This work is carried out by myself.

Chapter 3 is based on Xing et al. (2019). In Chapter 3, we change the topic from estimating ratio of normalizing constants to diagnosing approximation error in approximate Bayesian inference. We extend the diagnostic tools in Lee et al. (2019), and give two computationally more efficient methods for assessing the approximation quality of approximate credible sets. Bayesian credible sets with coverage level α provide a natural way to convey uncertainty in the parameter of interest. Let y_{obs} be the observed data and $\pi(\cdot|y_{obs})$ be the *exact* posterior conditioned on y_{obs}. Let

$\tilde{\pi}(\cdot|y_{obs})$ be a generic approximation of $\pi(\cdot|y_{obs})$. When we report an approximate credible set $\tilde{C}_{y_{obs}}$ with nominal level α based on the approximate posterior $\tilde{\pi}(\cdot|y_{obs})$, we only know $\Pr(\theta \in \tilde{C}_{y_{obs}}) = \alpha$, where $\theta \sim \tilde{\pi}(\cdot|y_{obs})$. However, there is no guarantee that the statement $\Pr(\phi \in \tilde{C}_{y_{obs}}) = \alpha$, where $\phi \sim \pi(\cdot|y_{obs})$, holds. So, what coverage does $\tilde{C}_{y_{obs}}$ actually achieve? We call the coverage that $\tilde{C}_{y_{obs}}$ actually achieves (i.e. $\Pr(\phi \in \tilde{C}_{y_{obs}})$, where $\phi \sim \pi(\cdot|y_{obs})$) the *operational coverage*, and show how to estimate this quantity without recourse to the exact posteriors. We recommend that wherever an approximate credible set with nominal coverage α is reported, the corresponding operational coverage, which correctly reflects the coverage and uncertainty level, should be given as well. This is joint work with Prof. Geoff Nicholls and Prof. Jeong Eun Lee. In particular, I contributed to the design and implementation of the proposed methods and led the drafting of the paper. In addition, I also implemented all the simulated and real world examples.

Chapter 4 is based on Xing et al. (2020). In Chapter 4, we focus on diagnosing the approximation error in the marginals of an approximate posterior. Here we introduce and estimate a "distortion map" $D_{y_{obs}} : [0,1] \longrightarrow [0,1]$ which acts on an univariate *approximate* posterior CDF conditioned on the observed data y_{obs} and brings it closer to the corresponding *exact* posterior CDF. The distortion map $D_{y_{obs}}$ encodes details of approximation error in the entire univariate approximate CDF (recall that in Chapter 3, we only focus on credible sets), and any deviation in $D_{y_{obs}}$ from an identity map indicates "distortion" in posterior due to approximation error. One advantage of this approach is that $D_{y_{obs}}$ is an invertible function with bounded domain and image, and we show that in fact $D_{y_{obs}}$ itself is a CDF over the unit interval, and hence can be estimated using a maximum-likelihood approach. This makes the parameterization and estimation of $D_{y_{obs}}$ easier. We give diagnostic procedures for both univariate and bivariate approximate marginals using the proposed distortion map. Compared with existing methods such as Prangle et al. (2014) and Talts et al. (2020), our approach is less likely to be fooled by any approximation error. We demonstrate that the proposed distortion map is able to identify approximation error that is overlooked by existing diagnostic methods using a real world social network model. Zhao et al. (2021) also propose visual diagnostic tools similar to ours. Their local P-P plot has the same interpretation as our distortion map. The main distinction is that Zhao et al. (2021) do not utilize the fact that the distortion map itself is a CDF over the unit interval, and their visual diagnostic is estimated by performing logistic regression repeatedly. In addition to uni- and bivariate cases, we also discuss possible strategies that extend the distortion map to multivariate settings in Chapter 4.9. This chapter is joint work

with Prof. Geoff Nicholls and Prof. Jeong Eun Lee. In particular, I proposed the method and led the design and implementation of it. I also designed and implemented all the simulated and real world examples, and contributed to the drafting of the paper. Additionally, I contributed to the unpublished extension of the proposed method in Chapter 4.9.

1.2 Background

In this section, we provide general overviews of a small selection of relevant background and concepts that are not detailed in the following chapters.

1.2.1 Deep generative models

Deep generative models such as finite (Rezende and Mohamed, 2015; Dinh et al., 2016) and continuous Normalizing flow (Grathwohl et al., 2018; Onken et al., 2021), Variational Autoencoder (Kingma and Welling, 2013; Rezende et al., 2014), diffusion model (Dhariwal and Nichol, 2021; Kingma et al., 2021) and Generative Adversarial Network (Goodfellow et al., 2014; Goodfellow, 2017; Nowozin et al., 2016) are neural network models that aim to model or approximate probability distributions in high dimensional spaces or with complicated structures. When trained successfully (the training process is usually carried out by approximately minimizing some statistical divergence between the generative model and a target distribution), one can use these models to estimate the density function of a probability distribution with complicated structures, or approximately draw samples from it. These methods have been successfully applied to fields such as image generation (Radford et al., 2015), computational physics (Carleo et al., 2019; Brehmer et al., 2020) and chemistry (Zhang et al., 2021).

In Chapter 2, our proposed method involves increasing the overlap between two distributions of interest by mapping samples from one distribution to the other using a smooth and bijective transformation. We parameterize such transformation using a finite Normalizing flow model. A finite Normalizing flow (Rezende and Mohamed, 2015; Dinh et al., 2016; Papamakarios et al., 2017, 2021) parameterizes a continuous probability distribution by mapping a simple base distribution (e.g. standard Normal) to the more complex target using a bijective transformation T, which is parameterized as a composition of a series of smooth and invertible mappings $f_1, ..., f_K$ with easy-to-compute Jacobians. This $T = f_K \circ f_{K-1} \circ ... \circ f_1$ is applied to the "base" random

variable $z_0 \sim p_0$, where $z_0 \in \mathbb{R}^d$ and p_0 is the known base density. Let

$$z_k = f_k \circ f_{k-1} \circ ... \circ f_1(z_0), \quad k = 1, ..., K \tag{1.1}$$

Since the transformation T is smooth and invertible, by applying change of variable repeatedly, the final output z_K has density

$$p_K(z_K) = p_0(z_0) \prod_{k=1}^{K} |\det J_k(z_{k-1})|^{-1} \tag{1.2}$$

where J_k is the Jacobian of the mapping f_k. The final density p_K can be used to approximate target distributions with complicated structures, and one can sample from p_K easily by applying $T = f_K \circ f_{K-1} \circ ... \circ f_1$ to $z_0 \sim p_0$. In order to evaluate p_K efficiently, we are restricted to transformations f_k whose $\det J_k(z)$ is easy to compute. For example, Real-NVP (Dinh et al., 2016) uses the following transformation: For $m \in \mathbb{N}$ such that $1 < m < d$, let $z_{1:m}$ be the first m entries of $z \in \mathbb{R}^d$, let $\times$ be element-wise multiplication and let $\mu_k, \sigma_k : \mathbb{R}^m \to \mathbb{R}^{d-m}$ be two mappings (usually parameterized by neural nets). The smooth and invertible transformation $y = f_k(z)$ for each step k in a Real-NVP is defined as

$$y_{1:m} = z_{1:m}, \quad y_{m+1:d} = \mu_k(z_{1:m}) + \sigma_k(z_{1:m}) \times z_{m+1:d} \tag{1.3}$$

This means f_k keeps the first m entries of input z, while shifting and scaling the remaining ones. The Jacobian J_k of f_k is lower triangular, hence $\det J_k(z) = \prod_{i=1}^{d-m} \sigma_{ik}(z_{1:m})$, where $\sigma_{ik}(z_{1:m})$ is the ith entry of $\sigma_k(z_{1:m})$. Each transformation f_k is also called a coupling layer. When composing a series of coupling layers $f_1, ..., f_K$, the authors also swap the ordering of indices in (1.3) so that the dimensions that are kept unchanged in one step k are to be scaled and shifted in the next step.

1.2.2 f-divergence

f-divergence (Ali and Silvey, 1966) is a broad class of statistical divergences that measures the discrepancy between two probability distributions. Let Q_1, Q_2 be two probability distributions defined on a common support Ω. The f-divergence between Q_1, Q_2 is defined as follows

Definition 1.2.1 (f-divergence)**.** *Suppose the two probability distributions Q_1, Q_2 have absolutely continuous density functions q_1 and q_2 with respect to a base measure μ on a common support Ω. Let the generator function $f : \mathbb{R}^+ \to \mathbb{R}$ be a convex and lower*

semi-continuous function satisfying $f(1) = 0$. *The* f*-divergence* $D_f(q_1, q_2)$ *defined by* f *takes the form*

$$D_f(q_1, q_2) = \int_\Omega f\left(\frac{q_1(\omega)}{q_2(\omega)}\right) q_2(\omega) d\mu(\omega) \tag{1.4}$$

By choosing f accordingly, one can recover various common divergences between probability distributions. Here we give two examples.

Example 1.2.1 (Kullback–Leibler divergence)

Kullback–Leibler (KL) divergence is defined as

$$KL(q_1, q_2) = \int_\Omega \log\left(\frac{q_1(\omega)}{q_2(\omega)}\right) q_1(\omega) d\mu(\omega) \tag{1.5}$$

$KL(q_1, q_2)$ is an f-divergence with $f(u) = u \log u$. It is also straightforward to verify that $KL(q_2, q_1)$ is an f-divergence with $f(u) = -\log(u)$.

Example 1.2.2 (Squared Hellinger distance)

Squared Hellinger distance is defined as

$$H^2(q_1, q_2) = \int_\Omega \left(\sqrt{q_1(\omega)} - \sqrt{q_2(\omega)}\right)^2 d\mu(\omega) \tag{1.6}$$

$H^2(q_1, q_2)$ is an f-divergence with $f(u) = (\sqrt{u} - 1)^2$.

A table of common f-divergences with their corresponding generator functions f can be found in Nowozin et al. (2016). In Chapter 2, we show how to estimate the RMSE of the optimal Bridge estimator and improve its statistical efficiency by utilizing its connection with a particular choice of f-divergence between the two distributions of interest. In addition, we show how Bridge estimators naturally arise from the problem of f-divergence estimation.

1.2.3 Diagnostics for Bayesian inference procedures

Bayesian inference is typically carried out in an approximate fashion and is usually computationally intensive. Therefore it is important for the practitioners to check the reliability of the approximation scheme they use. The diagnostic tools discussed in Chapter 3 and 4 are inspired by Geweke (2004) and Cook et al. (2006), which were setup for checking MCMC software implementation. Let $\pi(\phi)$ be the prior of parameter $\phi \in \Omega$, let $p(y|\phi)$ be the observation model for generic data $y \in \mathcal{Y}$ and let $\pi(\phi|y) \propto \pi(\phi)p(y|\phi)$ be the posterior for ϕ given the generic data point y. Let $p(\phi, y) = \pi(\phi)p(y|\phi)$ be the joint density of the pair $\{\phi, y\}$.

Geweke (2004) show that one can check the correctness and convergence of an MCMC algorithm targeting $\pi(\phi|y)$ by comparing the outputs from two separate procedures that both aim to generate samples from the joint distribution $p(\phi, y)$: One of the procedure (the marginal-conditional simulator) is designed to mimic the data generating process. In other words, it draws i.i.d. samples from $p(\phi, y)$ by first drawing $\phi \sim \pi(\phi)$, then $y \sim p(y|\phi)$ conditioned on ϕ. This procedure is straightforward to implement, and is assumed to be correct. The second procedure (the successive-conditional simulator) involves iterating between calling the MCMC algorithm that targets $\pi(\phi|y)$ conditioned on y, and sampling y from the generative model $p(y|\phi)$ conditioned on ϕ obtained from the MCMC algorithm, where the initial $\{\phi, y\}$ pair is drawn from $p(\phi, y)$ using the marginal-conditional simulator. If the MCMC algorithm is correctly implemented, then samples generated from both procedures should follow the same distribution $p(\phi, y)$. Hence one can check the implementation or convergence of MCMC by checking the discrepancy between samples drawn from the two procedures. We summarize this diagnostic procedure in Algorithm 1.1.

Algorithm 1.1 Diagnostic procedure in Geweke (2004)

Require: Sample size N, Prior distribution $\pi(\phi)$; Generative model $p(y|\phi)$; To-be-tested sampler M_y that aims to draw samples from the posterior $\pi(\phi|y)$.
for $n = 1, ..., N$ (the marginal-conditional simulator) **do**
Draw $\phi_n \sim \pi(\phi)$, $y_n \sim p(y|\phi_n)$
end for
for $n = N + 1, ..., 2N$ (the successive-conditional simulator) **do**
Draw ϕ_n from the to-be-tested sampler $M_{y_{n-1}}$, then draw $y_n \sim p(y|\phi_n)$
end for
Test if $\{(\phi_n, y_n)\}_{n=1}^{N}$ and $\{(\phi_n, y_n)\}_{n=N+1}^{2N}$ follow the same joint distribution.

Cook et al. (2006) provide an alternative strategy for checking MCMC implementations. Let L be a positive integer. Suppose we first draw $\{\phi^{(0)}, y\} \sim p(\phi, y)$ using the marginal-conditional simulator described above, then generate $\{\phi^{(1)}, ..., \phi^{(L)}\}$ by calling the MCMC algorithm targeting $\pi(\phi|y)$. By Bayes theorem, we have $\phi^{(0)} \sim \pi(\phi|y)$. Let $g : \Omega \to \mathbb{R}$ be a summary statistics that maps ϕ to a real number. If the MCMC algorithm is correctly implemented, then we expect $\phi^{(0)}$ and each element in $\{\phi^{(1)}, ..., \phi^{(L)}\}$ to follow the same conditional distribution $\pi(\phi|y)$. In this case, the empirical quantile $q = \frac{1}{L}\sum_{l=1}^{L} \mathbb{1}(g(\phi^{(0)}) > g(\phi^{(l)}))$ must follow a (discrete) uniform distribution. The authors utilize this fact and propose an alternative diagnostic procedure: Users can identify error in their MCMC software by repeating the above process N times, generating a collection of quantiles $\{q_n\}_{n=1}^{N}$ and then testing if $\{q_n\}_{n=1}^{N}$ follows a uniform

distribution. If $\{q_n\}_{n=1}^N$ strongly deviates from uniformity, then it can be viewed as evidence of presence of error in the software. We summarize this procedure in Algorithm 1.2. However, having perfectly uniformly distributed $\{q_n\}_{n=1}^N$ does not guarantee that the MCMC software is correctly implemented. To see this, suppose the sampler erroneously return samples from the prior regardless of the given data. In this case, both $\phi^{(0)}$ and $\{\phi^{(1)}, ..., \phi^{(L)}\}$ would (marginally) follow the same *prior* distribution $\pi(\phi)$ regardless of the value of y, and therefore the quantiles would still follow a uniform distribution.

Algorithm 1.2 Diagnostic procedure in Cook et al. (2006)

Require: Sample size N; Number of testing samples L; Prior distribution $\pi(\phi)$; Generative model $p(y|\phi)$; To-be-tested sampler M_y that aims to draw samples from the posterior $\pi(\phi|y)$; Summary statistics $g(\phi) : \Omega \to \mathbb{R}$.

for $n = 1, ..., N$ **do**

Draw $\phi_n^{(0)} \sim \pi(\phi)$, $y_n \sim p(y|\phi_n^{(0)})$

Draw $\{\phi_n^{(1)}, ..., \phi_n^{(L)}\}$ from the to-be-tested sampler M_{y_n}

Compute and store the empirical quantiles $q_n = \frac{1}{L} \sum_{l=1}^{L} \mathbb{1}(g(\phi_n^{(0)}) > g(\phi_n^{(l)}))$

end for

Test if $\{q_n\}_{n=1}^N$ follow a discrete uniform distribution.

Yao et al. (2018) extend Cook et al. (2006) from checking MCMC implementation to diagnosing approximation error in any approximation scheme. In Yao et al. (2018), the samples $\{\phi^{(1)}, ..., \phi^{(L)}\}$ are now drawn from a generic approximate posterior. In this case, any deviation from uniformity in the quantiles $\{q_n\}_{n=1}^N$ would indicate discrepancy between the exact and approximate posteriors. Talts et al. (2020) utilize the same idea, and propose generic diagnostic tools based on the uniformity of *rank statistics* instead of quantiles.

Chapter 2

Improving Bridge estimators using f-GAN

Estimating the normalizing constant of an unnormalized probability density, or the ratio of normalizing constants between two unnormalized densities is a challenging and important task. In Bayesian inference, such problems are closely related to estimating the marginal likelihood of a model or the Bayes factor between two competing models, and can arise from fields such as econometrics (Geweke, 1999), astronomy (Bridges et al., 2009), phylogenetics (Fourment et al., 2020), etc. Monte Carlo methods such as Bridge sampling (Bennett, 1976; Meng and Wong, 1996), path sampling (Gelman and Meng, 1998), reverse logistic regression (Geyer, 1994), nested sampling (Skilling et al., 2006) and reverse Annealed Importance Sampling (Burda et al., 2015) have been proposed to address this problem. See Friel and Wyse (2012) for an overview of some popular algorithms. Fourment et al. (2020) also compare the empirical performance of 19 algorithms for estimating normalizing constants in the context of phylogenetics.

In this chapter, we develop a statistically more efficient estimator of Bayes factor using methods taken form machine learning literature. In particular, we focus on improving the statistical efficiency of the Bridge estimator (Meng and Wong, 1996) using deep generative models such as Normalizing flow (Rezende and Mohamed, 2015; Papamakarios et al., 2017, 2021) and Generative Adversarial Network (Goodfellow et al., 2014; Nowozin et al., 2016; Gui et al., 2021). Bridge sampling is a powerful Monte Carlo method for estimating the ratio of normalizing constants between two unnormalized densities. Various methods have been introduced to improve its efficiency. These methods aim to reduce the asymptotic error of the Bridge estimator by applying appropriate transformations to the densities of interest without changing their normalizing constants, and increasing the overlap between them. In this chapter, we first give a new estimator of the asymptotic relative mean square error (RMSE) of

the optimal Bridge estimator by equivalently estimating an f-divergence between the two densities. We then utilize this framework and propose f-GAN-Bridge estimator (f-GB) based on a bijective transformation that maps one density to the other and minimizes the asymptotic RMSE of the resulting optimal Bridge estimator with respect to the transformed density. This transformation is chosen by minimizing a specific f-divergence between the densities. We show f-GB is optimal in the sense that within any given set of candidate transformations, the f-GB estimator can asymptotically achieve an RMSE lower than or equal to that achieved by Bridge estimators based on any other transformed densities. Numerical experiments show that f-GB outperforms existing methods in both simulated and real world examples. In addition, we discuss how Bridge estimators naturally arise from the problem of f-divergence estimation.

2.1 Introduction and background

Bridge sampling (Bennett, 1976; Meng and Wong, 1996) is a powerful, easy-to-implement Monte Carlo method for estimating the ratio of normalizing constants. Let $\tilde{q}_i(\omega), \omega \in \Omega_i,\ i = 1, 2$ be two unnormalized probability densities with respect to a common measure μ. Let $q_i(\omega) = \tilde{q}_i(\omega)/Z_i$ be the corresponding normalized density, where Z_i is the normalizing constant. Bridge sampling estimates $r = Z_1/Z_2$ using samples from q_1, q_2 and the unnormalized density functions $\tilde{q}_1, \tilde{q}_2$. Meng and Schilling (2002) point out that Bridge sampling is equally useful for estimating a single normalizing constant. The relative mean square error (RMSE) of a Bridge estimator depends on the overlap or "distance" between q_1, q_2. The overlap can be quantified by some divergence between them. When q_1, q_2 share little overlap, the corresponding Bridge estimator has large RMSE and therefore is unreliable. In order to improve the efficiency of Bridge estimators, various methods such as Warp Bridge sampling (Meng and Schilling, 2002), Warp-U Bridge sampling (Wang et al., 2022) and Gaussianized Bridge sampling (Jia and Seljak, 2020) have been introduced. These methods first apply transformations T_i to q_i in a tractable way without changing the normalizing constant Z_i for $i = 1, 2$, then compute Bridge estimators based on the transformed densities $q_i^{(T)}$ and the corresponding samples for $i = 1, 2$. If $q_1^{(T)}, q_2^{(T)}$ have greater overlap than the original ones, then the resulting Bridge estimator of r based on $q_1^{(T)}, q_2^{(T)}$ would have a lower RMSE.

In this chapter, we first demonstrate the connection between Bridge estimators and f-divergence (Ali and Silvey, 1966). We show that one can estimate the asymptotic RMSE of the optimal Bridge estimator by equivalently estimating a specific

f-divergence between q_1, q_2. Nguyen et al. (2010) propose a general variational framework for f-divergence estimation. We apply this framework to our problem and obtain a new estimator of the asymptotic RMSE of the optimal Bridge estimator using the unnormalized densities $\tilde{q}_1, \tilde{q}_2$ and the corresponding samples. We also find a connection between Bridge estimators and the variational lower bound of f-divergence given by Nguyen et al. (2010). In particular, we show that the problem of estimating an f-divergence between q_1, q_2 using this variational framework naturally leads to a Bridge estimator of $r = Z_1/Z_2$. Kong et al. (2003) observe that the optimal Bridge estimator is a maximum likelihood estimator under a semi-parametric formulation. We use this f-divergence estimation framework to extend this observation and show that many special cases of Bridge estimators such as the geometric Bridge estimator can also be interpreted as maximizers of some objective functions that are related to the f-divergence between q_1, q_2. This formulation also connects Bridge estimators and density ratio estimation problems: Since we can evaluate the unnormalized densities $\tilde{q}_1, \tilde{q}_2$, we know the true density ratio up to a multiplicative constant $r = Z_1/Z_2$. Hence estimating r can be viewed as a problem of estimating the density ratio between q_1, q_2. A similar idea has been explored in e.g. Noise Contrastive Estimation (Gutmann and Hyvärinen, 2010), where the authors treat the unknown normalizing constant as a model parameter, and cast the estimation problem as a classification problem. Similar ideas have also been discussed in e.g. Geyer (1994) and Uehara et al. (2016).

We then utilize the connection between the asymptotic RMSE of the optimal Bridge estimator and a specific f-divergence between q_1, q_2, and propose f-GAN-Bridge estimator (f-GB), which improves the efficiency of the optimal Bridge estimator of r by directly minimizing the first order approximation of its asymptotic RMSE with respect to the *densities* using an f-GAN. f-GAN (Nowozin et al., 2016) is a class of generative model that aims to approximate the target distribution by minimizing an f-divergence between the generative model and the target. Let $\mathcal{T}$ be a collection of transformations T such that $\tilde{q}_1^{(T)}$, the transformed unnormalized density of q_1 is computationally tractable and have the same normalizing constant Z_1 as the original $\tilde{q}_1$. The f-GAN-Bridge estimator is obtained using a two-step procedure: We first use the f-GAN framework to find the transformation T^* that minimizes a specific f-divergence between $q_1^{(T)}$ and q_2 with respect to $T \in \mathcal{T}$. Once T^* and $q_1^{(T^*)}$ are chosen, we then compute the optimal Bridge estimator of r based on $q_1^{(T^*)}$ and q_2 as the f-GAN-Bridge estimator. We show T^* asymptotically minimizes the first order approximation of the asymptotic RMSE of the optimal Bridge estimator based on $q_1^{(T)}$ and q_2 with respect to T. In contrast, existing methods such as Warp Bridge sampling

(Meng and Schilling, 2002; Wang et al., 2022) and Gaussianized Bridge sampling (Jia and Seljak, 2020) do not offer such theoretical guarantee. The transformed $q_1^{(T)}$ can be parameterized in any way as long as it is computationally tractable and preserves the normalizing constant Z_1. In this paper, we parameterize $q_1^{(T)}$ as a Normalizing flow (Rezende and Mohamed, 2015; Dinh et al., 2016) with base density q_1 because of its flexibility.

2.1.1 Summary of our contributions

The main contribution of this chapter is that we give a computational framework to improve the optimal Bridge estimator by minimizing the first order approximation of its asymptotic RMSE with respect to the densities. We also give a new estimator of the asymptotic RMSE of the optimal Bridge estimator using the variational framework proposed by Nguyen et al. (2010). This formulation allows us to cast the estimation problem as a 1-d optimization problem. We find the f-GAN-Bridge estimator outperforms existing methods significantly in both simulated and real-world examples. Numerical experiments show that the proposed method provides not only a reliable estimate of r, but also an accurate estimate of its RMSE. In addition, we also find a connection between Bridge estimators and the problem of f-divergence estimation, which allows us to view Bridge estimators from a different perspective.

This chapter is structured as follows: In Chapter 2.2, we briefly review Bridge sampling and existing improvement strategies. In Chapter 2.3, we give a new estimator of the asymptotic RMSE of the optimal Bridge estimator using the variational framework for f-divergence estimation (Nguyen et al., 2010). We also demonstrate the connection between Bridge estimators and the problem of f-divergence estimation. In Chapter 2.4, we introduce the f-GAN-Bridge estimator and give implementation details. We give both simulated and real-world examples in Chapter 2.5, 2.6. Chapter 2.7 concludes the paper with a discussion. A Python implementation of the proposed method alongside with examples can be found at `https://github.com/hwxing3259/Bridge_sampling_and_fGAN`.

2.2 Bridge sampling and related works

Let Q_1, Q_2 be two probability distributions of interest. Let $q_i(\omega), \omega \in \Omega_i, i = 1, 2$ be the densities of Q_1, Q_2 with respect to a common measure μ defined on $\Omega_1 \cup \Omega_2$, where Ω_1 and Ω_2 are the corresponding supports. We use $\tilde{q}_i(\omega), i = 1, 2$ to denote the unnormalized densities and $Z_i, i = 1, 2$ to denote the corresponding normalizing

constants, i.e. $q_i(\omega) = \tilde{q}_i(\omega)/Z_i$ for $i = 1, 2$. Suppose we have samples from q_1, q_2, but we are only able to evaluate the *unnormalized* densities $\tilde{q}_i(\omega), i = 1, 2$. Our goal is to estimate the ratio of normalizing constants $r = Z_1/Z_2$ using only $\tilde{q}_i(\omega), i = 1, 2$ and samples from the two distributions. Bridge sampling (Bennett, 1976; Meng and Wong, 1996) is a powerful method for this task.

Definition 2.2.1 (Bridge estimator)**.** *Suppose* $\mu(\Omega_1 \cap \Omega_2) > 0$ *and* $\alpha : \Omega_1 \cap \Omega_2 \to \mathbb{R}$ *satisfies* $0 < \left| \int_{\Omega_1 \cap \Omega_2} \alpha(\omega) q_1(\omega) q_2(\omega) d\mu(\omega) \right| < \infty$. *Given samples* $\{\omega_{ij}\}_{j=1}^{n_i} \sim q_i$ *for* $i = 1, 2$*, the Bridge estimator* $\hat{r}_\alpha$ *of* $r = Z_1/Z_2$ *is defined as*

$$\hat{r}_\alpha = \frac{{n_2}^{-1} \sum_{j=1}^{n_2} \alpha(\omega_{2j}) \tilde{q}_1(\omega_{2j})}{{n_1}^{-1} \sum_{j=1}^{n_1} \alpha(\omega_{1j}) \tilde{q}_2(\omega_{1j})} \tag{2.1}$$

The choice of free function α directly affects the quality of $\hat{r}_\alpha$, which is quantified by the relative mean square error (RMSE) $E(\hat{r}_\alpha - r)^2/r^2$. Let $n = n_1 + n_2$ and $s_i = n_i/n$ for $i = 1, 2$. Let $RE^2(\hat{r}_\alpha)$ denote the asymptotic RMSE of $\hat{r}_\alpha$ as $n_1, n_2 \to \infty$. Under the assumption that the samples from q_1, q_2 are i.i.d., Meng and Wong (1996) show the optimal α which minimizes the first order approximation of $RE^2(\hat{r}_\alpha)$ takes the form

$$\alpha_{opt}(\omega) \propto \frac{1}{s_1 \tilde{q}_1(\omega) + s_2 \tilde{q}_2(\omega) r}, \quad \omega \in \Omega_1 \cap \Omega_2 \tag{2.2}$$

The resulting $RE^2(\hat{r}_{\alpha_{opt}})$ with the optimal free function α_{opt} is

$$RE^2(\hat{r}_{\alpha_{opt}}) = \frac{1}{n s_1 s_2} \left[\left(\int_{\Omega_1 \cap \Omega_2} \frac{q_1(\omega) q_2(\omega)}{s_1 q_1(\omega) + s_2 q_2(\omega)} d\mu(\omega) \right)^{-1} - 1 \right] + o\left(\frac{1}{n}\right). \tag{2.3}$$

Note that the optimal α_{opt} is not directly usable as it depends on the unknown quantity r we would like to estimate in the first place. To resolve this problem, Meng and Wong (1996) give an iterative procedure

$$\hat{r}^{(t+1)} = \frac{n_2^{-1} \sum_{j=1}^{n_2} \tilde{q}_1(\omega_{2j}) / (s_1 \tilde{q}_1(\omega_{2j}) + s_2 \tilde{q}_2(\omega_{2j}) \hat{r}^{(t)})}{n_1^{-1} \sum_{j=1}^{n_1} \tilde{q}_2(\omega_{1j}) / (s_1 \tilde{q}_1(\omega_{1j}) + s_2 \tilde{q}_2(\omega_{1j}) \hat{r}^{(t)})}, \quad t = 0, 1, 2, ... \tag{2.4}$$

The authors show that for any initial value $\hat{r}^{(0)}$, $\hat{r}^{(t)}$ is a consistent estimator of r for all $t \geq 1$, and the sequence $\{\hat{r}^{(t)}\}$, $t = 0, 1, 2, ...$ converges to the unique limit $\hat{r}_{opt}$. Let $MSE(\log \hat{r}_{opt})$ denote the asymptotic mean square error of $\log \hat{r}_{opt}$.

Under the i.i.d. assumption, the authors also show $RE^2(\hat{r}_{opt})$ and $MSE(\log \hat{r}_{opt})$ are asymptotically equivalent to $RE^2(\hat{r}_{\alpha_{opt}})$ in (2.3) up to the first order (i.e. they have the same leading term). Note that $\hat{r}_{opt}$ can be found numerically while $\hat{r}_{\alpha_{opt}}$ is not directly computable. We will focus on the asymptotically optimal Bridge estimator $\hat{r}_{opt}$ for the rest of the paper.

2.2.1 Improving Bridge estimators via transformations

From (2.3) and the fact that $RE^2(\hat{r}_{opt})$ and $RE^2(\hat{r}_{\alpha_{opt}})$ are asymptotically equivalent, we see $RE^2(\hat{r}_{opt})$ depends on the overlap between q_1 and q_2. Even when $\Omega_1 = \Omega_2$, if q_1 and q_2 put their probability mass on very different regions, the integral in (2.3) would be close to 0, leading to large RMSE and unreliable estimators. In order to improve the performance of $\hat{r}_{opt}$, one may apply transformations to q_1, q_2 (and to the corresponding samples) to increase their overlap while keeping the *transformed* unnormalized densities computationally tractable and the normalizing constants unchanged. We assume that we are dealing with unconstrained, continuous random variables with a common support, i.e. $\Omega_1 = \Omega_2 = \mathbb{R}^d$. When the supports Ω_1, Ω_2 are constrained or different from each other, we can usually match them by applying simple invertible transformations to q_1, q_2. When Ω_1,Ω_2 have different dimensions, Chen and Shao (1997) suggest matching the dimensions of q_1, q_2 by augmenting the lower dimensional distribution using some completely known random variables (See Appendix 2.8.2 for details).

Voter (1985) gives a method to increase the overlap in the context of free energy estimation by shifting the samples from one distribution to the other and matching their modes. Meng and Schilling (2002) extends this idea and consider more general mappings. Let $T_i : \mathbb{R}^d \to \mathbb{R}^d$, $i = 1, 2$ be two smooth and invertible transformations that aim to bring q_1, q_2 "closer". For $\omega_i \sim q_i$, define $\omega_i^{(T)} = T_i(\omega_i)$, $i = 1, 2$. Then for $i = 1, 2$, the distribution of the transformed sample $\omega_i^{(T)}$ has density

$$q_i^{(T)}(\omega_i^{(T)}) = \tilde{q}_i(T_i^{-1}(\omega_i^{(T)})) \left|\det J_i(\omega_i^{(T)})\right| / Z_i \tag{2.5}$$

$$\equiv \tilde{q}_i^{(T)}(\omega_i^{(T)})/Z_i, \quad i = 1, 2 \tag{2.6}$$

where $\tilde{q}_i^{(T)}$ is the unnormalized version of $q_i^{(T)}$, T_i^{-1} is the inverse transformation of T_i and $J_i(\omega)$ is its Jacobian. One can then apply (2.1) to the *transformed* samples and the corresponding unnormalized densities $\tilde{q}_1^{(T)}, \tilde{q}_2^{(T)}$, and obtain a Bridge estimator

$$\hat{r}_\alpha^{(T)} = \frac{n_2^{-1} \sum_{j=1}^{n_2} \tilde{q}_1^{(T)}(T_2(\omega_{2j}))\alpha(T_2(\omega_{2j}))}{n_1^{-1} \sum_{j=1}^{n_1} \tilde{q}_2^{(T)}(T_1(\omega_{1j}))\alpha(T_1(\omega_{1j}))} \tag{2.7}$$

without the need to sample from $\tilde{q}_1^{(T)}$ or $\tilde{q}_2^{(T)}$ separately. Let $\hat{r}_{opt}^{(T)}$ denote the asymptotically optimal Bridge estimator based on the transformed densities. We stress that the superscript of $\hat{r}^{(t)}$ in (2.4) indicates the number of iterations, while the superscript in $\hat{r}_{opt}^{(T)}$ means it is based on the transformed densities. If the transformed $q_1^{(T)}, q_2^{(T)}$ have a greater overlap than the original q_1, q_2, then $\hat{r}_{opt}^{(T)}$ should be a more reliable

estimator of r with a lower RMSE. Meng and Schilling (2002) further extend this idea and propose the Warp transformation, which aims to increase the overlap by centering, scaling and symmetrizing the two densities q_1, q_2. One limitation of the Warp transformation is that it does not work well for multimodal distributions. Wang et al. (2022) propose the Warp-U transformation to address this problem. The key idea of the Warp-U transformation is to first approximate q_i by a mixture of Normal or t distributions, then construct a coupling between them which allows us to map q_i into a unimodal density in the same way as mapping the mixture back to a single standard Normal or t distribution.

An alternative to the Warp transformation (Meng and Schilling, 2002) is Normalizing flow models (Rezende and Mohamed, 2015; Dinh et al., 2016; Papamakarios et al., 2017, 2021), which are introduced it in Chapter 1. Jia and Seljak (2020) utilize the idea of transforming q_i using a Normalizing flow, and propose Gaussianzed Bridge sampling (GBS) for estimating a single normalizing constant. The authors set q_1 to be a completely known density, e.g. standard multivariate Normal, and aim to approximate the target q_2 using a Normalizing flow with base density q_1. The transformed density $q_1^{(T)}$ is estimated by matching the marginal distributions between $q_1^{(T)}$ and q_2. Once $q_1^{(T)}$ is chosen, the authors use (2.7) and the iterative procedure (2.4) to form the asymptotically optimal Bridge estimator of Z_2 based on the transformed $q_1^{(T)}$ and the original q_2.

The idea of increasing overlap via transformations is also applicable to discrete random variables. For example, Meng and Schilling (2002) suggest using swapping and permutation to increase the overlap between two discrete distributions. Tran et al. (2019) also give Normalizing flow models applicable to discrete random variables based on modulo operations. We give a toy example of increasing the overlap between two discrete distributions using Normalizing flows in Appendix 2.8.7. In the later sections, we will extend the idea of increasing overlap via transformations, and propose a new strategy to improve $\hat{r}_{opt}^{(T)}$ by directly minimizing the first order approximation of $RE^2(\hat{r}_{opt}^{(T)})$ with respect to the transformed densities.

2.3 Bridge estimators and f-divergence estimation

Frühwirth-Schnatter (2004) gives an MC estimator of $RE^2(\hat{r}_{opt})$. In this section, we introduce an alternative estimator of $RE^2(\hat{r}_{opt})$ and $MSE(\log \hat{r}_{opt})$ by equivalently estimating an f-divergence between q_1, q_2. This formulation allows us to utilize the variational lower bound of f-divergence given by Nguyen et al. (2010), and cast

the problem of estimating $RE^2(\hat{r}_{opt})$ as a 1-d optimization problem. In the later section, we will also show how to use this new estimator to improve the efficiency of $\hat{r}_{opt}^{(T)}$. In addition, we find that estimating different choices of f-divergences under the variational framework proposed by Nguyen et al. (2010) naturally leads to Bridge estimators of r with different choices of free function $\alpha(\omega)$.

2.3.1 Estimating $RE^2(\hat{r}_{opt})$ via f-divergence estimation

f-divergence (Ali and Silvey, 1966) is a broad class of divergences between two probability distributions. By choosing f accordingly, one can recover common divergences between probability distributions such as KL divergence $KL(q_1, q_2)$, Squared Hellinger distance $H^2(q_1, q_2)$ and total variation distance $d_{TV}(q_1, q_2)$.

Definition 2.3.1 (f-divergence)**.** *Suppose the two probability distributions Q_1, Q_2 have absolutely continuous density functions q_1 and q_2 with respect to a base measure μ on a common support Ω. Let the generator function $f : \mathbb{R}^+ \to \mathbb{R}$ be a convex and lower semi-continuous function satisfying $f(1) = 0$. The f-divergence $D_f(q_1, q_2)$ defined by f takes the form*

$$D_f(q_1, q_2) = \int_\Omega f\left(\frac{q_1(\omega)}{q_2(\omega)}\right) q_2(\omega) d\mu(\omega) \tag{2.8}$$

Unless otherwise stated, we assume $\Omega = \mathbb{R}^d$ where $d \in \mathbb{N}$ i.e. both q_1 and q_2 are defined on $\mathbb{R}^d$. If the densities q_1, q_2 have different or disjoint supports Ω_1, Ω_2, then we apply appropriate transformations and augmentations discussed in the previous sections to ensure that the transformed and augmented densities (if necessary) are defined on the common support $\Omega = \mathbb{R}^d$. In this paper, we focus on a particular choice of f-divergence that is closely related to $RE^2(\hat{r}_{opt})$ in (2.3).

Definition 2.3.2. *(Weighted harmonic divergence) Let q_1, q_2 be continuous densities with respect to a base measure μ on the common support Ω. The weighted harmonic divergence is defined as*

$$H_\pi(q_1, q_2) = 1 - \int_\Omega \left(\pi q_1^{-1}(\omega) + (1-\pi) q_2^{-1}(\omega)\right)^{-1} d\mu(\omega) \tag{2.9}$$

where $\pi \in (0, 1)$ is the weight parameter.

Wang et al. (2022) observe that the weighted harmonic divergence $H_\pi(q_1, q_2)$ is an f-divergence with generator $f(u) = 1 - \frac{u}{\pi + (1-\pi)u}$, and $RE^2(\hat{r}_{opt})$ can be rearranged as

$$RE^2(\hat{r}_{opt}) = (s_1 s_2 n)^{-1} \left((1 - H_{s_2}(q_1, q_2))^{-1} - 1\right) + o\left(\frac{1}{n}\right). \tag{2.10}$$

The same statement also holds for $MSE(\log \hat{r}_{opt})$ since $MSE(\log \hat{r}_{opt})$ is asymptotically equivalent to $RE^2(\hat{r}_{opt})$ (Meng and Wong, 1996). This means if we have an estimator of $H_{s_2}(q_1, q_2)$, then we can plug it into the leading term of the right hand side of (2.10) and obtain an estimator of the first order approximation of $RE^2(\hat{r}_{opt})$ and $MSE(\log \hat{r}_{opt})$. Before we give the estimator of $H_{s_2}(q_1, q_2)$, we first introduce the variational framework for f-divergence estimation proposed by Nguyen et al. (2010). Every convex, lower semi-continuous function $f : \mathbb{R}^+ \to \mathbb{R}$ has a convex conjugate f^* which is defined as follows,

Definition 2.3.3. *(Convex conjugate) Let $f : \mathbb{R}^+ \to \mathbb{R}$ be a convex and lower semi-continuous function. The convex conjugate of f is defined as*

$$f^*(t) = \sup_{u \in \mathbb{R}^+} \{ut - f(u)\} \tag{2.11}$$

Nguyen et al. (2010) show that an f-divergence $D_f(q_1, q_2)$ satisfies

$$D_f(q_1, q_2) \geq \sup_{V \in \mathcal{V}} \Big(E_{q_1}[V(\omega)] - E_{q_2}[f^*(V(\omega))] \Big), \tag{2.12}$$

where $\mathcal{V}$ is an arbitrary class of functions $V : \Omega \to \mathbb{R}$, and $f^*(t)$ is the convex conjugate of the generator f which characterizes the f-divergence $D_f(q_1, q_2)$. A table of common f-divergences with their generator f and the corresponding convex conjugate f^* can be found in Nowozin et al. (2016). Nguyen et al. (2010) show that if f is differentiable and strictly convex, then $D_f(q_1, q_2)$ is equal to $E_{q_1}[V(\omega)] - E_{q_2}[f^*(V(\omega))]$ in (2.12) if and only if $V(\omega) = f'\left(\frac{q_1(\omega)}{q_2(\omega)}\right)$, the first order derivative of f evaluated at $q_1(\omega)/q_2(\omega)$. The authors then give a new strategy of estimating the f-divergence $D_f(q_1, q_2)$ by finding the maximum of an empirical estimate of $E_{q_1}[V(\omega)] - E_{q_2}[f^*(V(\omega))]$ in (2.12) with respect to the variational function $V \in \mathcal{V}$. We now use this framework to give an estimator of $H_\pi(q_1, q_2)$.

Proposition 2.3.1 (Estimating $H_\pi(q_1, q_2)$)**.** *Let q_1, q_2 be continuous densities with respect to a base measure μ on the common support Ω. Let $\{\omega_{ij}\}_{j=1}^{n_i}$ be samples from q_i for $i = 1, 2$. Let $\pi \in (0, 1)$ be the weight parameter. Let r be the true ratio of normalizing constants between q_1, q_2, and $C_2 > C_1 > 0$ be constants such that $r \in [C_1, C_2]$. For $\tilde{r} \in [C_1, C_2]$, define*

$$G(\tilde{r}; \pi) = 1 - \tfrac{1}{\pi} E_{q_1}\left(\tfrac{\pi \tilde{q}_2(\omega)\tilde{r}}{(1-\pi)\tilde{q}_1(\omega) + \pi \tilde{q}_2(\omega)\tilde{r}}\right)^2 - \tfrac{1}{1-\pi} E_{q_2}\left(\tfrac{(1-\pi)\tilde{q}_1(\omega)}{(1-\pi)\tilde{q}_1(\omega) + \pi \tilde{q}_2(\omega)\tilde{r}}\right)^2 \tag{2.13}$$

Then $H_\pi(q_1, q_2)$ satisfies

$$H_\pi(q_1, q_2) \geq G(\tilde{r}; \pi) \quad \forall \tilde{r} \in [C_1, C_2], \tag{2.14}$$

and equality holds if and only if $\tilde{r} = r$. In addition, let

$$\hat{G}(\tilde{r}; \pi, \{\omega_{ij}\}_{j=1}^{n_i}) = 1 - \frac{1}{\pi n_1} \sum_{j=1}^{n_1} \left(\frac{\pi \tilde{q}_2(\omega_{1j}) \tilde{r}}{(1-\pi)\tilde{q}_1(\omega_{1j}) + \pi \tilde{q}_2(\omega_{1j}) \tilde{r}} \right)^2 - \frac{1}{(1-\pi) n_2} \sum_{j=1}^{n_2} \left(\frac{(1-\pi)\tilde{q}_1(\omega_{2j})}{(1-\pi)\tilde{q}_1(\omega_{2j}) + \pi \tilde{q}_2(\omega_{2j}) \tilde{r}} \right)^2 \tag{2.15}$$

be the empirical estimate of $G(\tilde{r}; \pi)$ based on $\{\omega_{ij}\}_{j=1}^{n_i} \sim q_i$ for $i = 1, 2$.

If $\hat{r}_\pi = \arg\max_{\tilde{r} \in [C_1, C_2]} \hat{G}(\tilde{r}; \pi, \{\omega_{ij}\}_{j=1}^{n_i})$, then $\hat{r}_\pi$ is a consistent estimator of r, and $\hat{G}(\hat{r}_\pi; \pi, \{\omega_{ij}\}_{j=1}^{n_i})$ is a consistent estimator of $H_\pi(q_1, q_2)$ as $n_1, n_2 \to \infty$.

Proof. See Appendix 2.8.1. □

Note that (2.14) is a special case of the variational lower bound (2.12) with the f-divergence $D_f(q_1, q_2) = H_\pi(q_1, q_2)$, the corresponding generator $f(u) = 1 - \frac{u}{\pi + (1-\pi)u}$ and variational function $V_{\tilde{r}}(\omega) = f'\left(\frac{\tilde{q}_1(\omega)}{\tilde{q}_2(\omega)\tilde{r}}\right)$ with $\mathcal{V} = \{V_{\tilde{r}}(\omega) | \tilde{r} \in [C_1, C_2]\}$, i.e. $\tilde{r} \in [C_1, C_2]$ is the is the sole parameter of $V_{\tilde{r}}(\omega)$. Note that $V_r(\omega) = f'\left(\frac{q_1(\omega)}{q_2(\omega)}\right)$ since r is the ratio of normalizing constants between q_1, q_2. We parameterize the variational function in this specific form because we would like to take the advantage of knowing the unnormalized densities $\tilde{q}_1, \tilde{q}_2$ in our setup. Here we assume that $\tilde{r} \in [C_1, C_2]$ instead of $\tilde{r} \in \mathbb{R}^+$. This is not a strong assumption, since we can set C_1 (C_2) to be arbitrarily small (large). We take $\hat{G}(\hat{r}_{s_2}; s_2, \{\omega_{ij}\}_{j=1}^{n_i})$ as an estimator of $H_{s_2}(q_1, q_2)$, and define our estimator of the first order approximation of $RE^2(\hat{r}_{opt})$ as follows:

Definition 2.3.4 (Estimator of $RE^2(\hat{r}_{opt})$)**.** *Let $\{\omega_{ij}\}_{j=1}^{n_i}$ be samples from q_i for $i = 1, 2$. Define*

$$\widehat{RE}^2(\hat{r}_{opt}) = (s_1 s_2 n)^{-1} \left((1 - \hat{G}(\hat{r}_{s_2}; s_2, \{\omega_{ij}\}_{j=1}^{n_i}))^{-1} - 1 \right) \tag{2.16}$$

as an estimator of the first order approximation of both $RE^2(\hat{r}_{opt})$ and $MSE(\log \hat{r}_{opt})$ in (2.10)*.*

Even though $\hat{G}(\hat{r}_{s_2}; s_2, \{\omega_{ij}\}_{j=1}^{n_i})$ is a consistent estimator of $H_{s_2}(q_1, q_2)$, it suffers from a positive bias (See Appendix 2.8.3 for details). We have not found a practical strategy to correct it so far. On the other hand, we believe this bias does not prevent our

proposed error estimator $\widehat{RE}^2(\hat{r}_{opt})$ from being useful in practice. Since our estimator of $RE^2(\hat{r}_{opt})$ in (2.16) is a monotonically increasing function of $\hat{G}(\hat{r}_\pi; \pi, \{\omega_{ij}\}_{j=1}^{n_i})$ in Prop 2.3.1, the positive bias in $\hat{G}(\hat{r}_\pi; \pi, \{\omega_{ij}\}_{j=1}^{n_i})$ leads to a positive bias in $\widehat{RE}^2(\hat{r}_{opt})$. Therefore $\widehat{RE}^2(\hat{r}_{opt})$ will systemically overestimate the true error $RE^2(\hat{r}_{opt})$, which will lead to more conservative conclusions (e.g. wider error bars). This is certainly not ideal, but we believe in practice, it is less harmful than underestimating the variability in $\hat{r}_{opt}$. In addition, we see the proposed error estimator provides accurate estimates of $RE^2(\hat{r}_{opt})$ in both examples in Chapter 2.5 and 2.6, indicating the effectiveness of it.

2.3.2 f-divergence estimation and Bridge estimators

In the last section, we focus on estimating $H_\pi(q_1, q_2)$. We now extend the estimation framework to other choices of f-divergence, and show how Bridge estimators naturally arise from this estimation problem. Let an f-divergence $D_f(q_1, q_2)$ with the corresponding generator $f(u)$ be given. Similar to Proposition 2.3.1, under our parameterization of the variational function $V_{\tilde{r}}$, the empirical estimate of $E_{q_1}[V(\omega)] - E_{q_2}[f^*(V(\omega))]$ in (2.12) becomes

$$\hat{G}_f(\tilde{r}; \{\omega_{ij}\}_{j=1}^{n_i}) = \frac{1}{n_1}\sum_{j=1}^{n_1} V_{\tilde{r}}(\omega_{1j}) - \frac{1}{n_2}\sum_{j=1}^{n_2} f^*(V_{\tilde{r}}(\omega_{2j})) \tag{2.17}$$

$$= \frac{1}{n_1}\sum_{j=1}^{n_1} f'\left(\frac{\tilde{q}_1(\omega_{1j})}{\tilde{q}_2(\omega_{1j})\tilde{r}}\right) - \frac{1}{n_2}\sum_{j=1}^{n_2} f^* \circ f'\left(\frac{\tilde{q}_1(\omega_{2j})}{\tilde{q}_2(\omega_{2j})\tilde{r}}\right), \tag{2.18}$$

where $\{\omega_{ij}\}_{j=1}^{n_i} \sim q_i$ for $i = 1, 2$. Let $\hat{r}^{(f)} = \arg\max_{\tilde{r}\in\mathbb{R}^+} \hat{G}_f(\tilde{r}; \{\omega_{ij}\}_{j=1}^{n_i})$. By Nguyen et al. (2010), $V_{\hat{r}^{(f)}} = f'\left(\frac{\tilde{q}_1(\omega)}{\tilde{q}_2(\omega)\hat{r}^{(f)}}\right)$ is an estimator of $V_r(\omega) = f'\left(\frac{\tilde{q}_1(\omega)}{\tilde{q}_2(\omega)r}\right)$, and $\hat{G}_f(\hat{r}^{(f)}; \{\omega_{ij}\}_{j=1}^{n_i})$ is an estimator of $D_f(q_1, q_2)$. In Proposition 2.3.1 we have shown that $\hat{r}^{(f)}$ and $\hat{G}_f(\hat{r}^{(f)}; \{\omega_{ij}\}_{j=1}^{n_i})$ are consistent estimators of r and $D_f(q_1, q_2)$ when $D_f(q_1, q_2)$ is the weighted Harmonic divergence $H_\pi(q_1, q_2)$ [1]. Here we show the connection between $\hat{r}^{(f)}$ and the Bridge estimators of r with different choices of free function $\alpha(\omega)$.

Proposition 2.3.2 (Connection between $\hat{r}^{(f)}$ and Bridge estimators)**.** *Suppose $f(u) : \mathbb{R}^+ \to \mathbb{R}$ is strictly convex, twice differentiable and satisfies $f(1) = 0$. Let $\{\omega_{ij}\}_{j=1}^{n_i}$ be samples from q_i for $i = 1, 2$. If $\hat{r}^{(f)} = \arg\max_{\tilde{r}\in\mathbb{R}^+} \hat{G}_f(\tilde{r}; \{\omega_{ij}\}_{j=1}^{n_i})$ is a stationary*

[1] It is of interest to see if $\hat{r}^{(f)}$ and $\hat{G}_f(\hat{r}^{(f)}; \{\omega_{ij}\}_{j=1}^{n_i})$ are consistent for all generator functions f and the corresponding f-divergences. We have not considered this general problem here.

point of $\hat{G}_f(\tilde{r}; \{\omega_{ij}\}_{j=1}^{n_i})$ *in* (2.18)*, then* $\hat{r}^{(f)}$ *satisfies the following equation*

$$\hat{r}^{(f)} = \frac{\frac{1}{n_2}\sum_{j=1}^{n_2} f''\left(\frac{\tilde{q}_1(\omega_{2j})}{\tilde{q}_2(\omega_{2j})\hat{r}^{(f)}}\right)\frac{\tilde{q}_1(\omega_{2j})}{\tilde{q}_2(\omega_{2j})^2}\tilde{q}_1(\omega_{2j})}{\frac{1}{n_1}\sum_{j=1}^{n_1} f''\left(\frac{\tilde{q}_1(\omega_{1j})}{\tilde{q}_2(\omega_{1j})\hat{r}^{(f)}}\right)\frac{\tilde{q}_1(\omega_{1j})}{\tilde{q}_2(\omega_{1j})^2}\tilde{q}_2(\omega_{1j})} \tag{2.19}$$

where f'' *is the second order derivative of* f*.*

Proof. See Appendix 2.8.1. □

In Equation (2.19), $f''\left(\frac{\tilde{q}_1(\omega)}{\tilde{q}_2(\omega)\hat{r}^{(f)}}\right)\frac{\tilde{q}_1(\omega)}{\tilde{q}_2(\omega)^2}$ plays the role of the free function $\alpha(\omega)$ in a Bridge estimator (2.1). Common Bridge estimators such as the asymptotically optimal Bridge estimator $\hat{r}_{opt}$ and the geometric Bridge estimator can be recovered by choosing f accordingly (See Appendix 2.8.4). Kong et al. (2003) observe that $\hat{r}_{opt}$ can be viewed as a semi-parametric maximum likelihood estimator. Proposition 2.3.2 extends this observation and show that in addition to $\hat{r}_{opt}$, a large class of Bridge estimators can also be viewed as maximizers of some objective functions that are related to the variational lower bound of some f-divergences. In the next section, we will show how to use this variational framework to minimize the first order approximation of $RE^2(\hat{r}_{opt}^{(T)})$ with respect to the transformed densities.

2.4 Improving $\hat{r}_{opt}$ via f-GAN

From Chapter 2.2.1, we see that one can improve $\hat{r}_{opt}$ and reduce its RMSE by first transforming q_1, q_2 appropriately, then computing $\hat{r}_{opt}^{(T)}$ using the transformed densities and samples. From Chapter 2.3 we also see the first order approximation of $RE^2(\hat{r}_{opt})$ is a monotonic function of $H_{s_2}(q_1, q_2)$. In this section, we utilize this observation and introduce the f-GAN-Bridge estimator (f-GB) that aims to improve $\hat{r}_{opt}^{(T)}$ by minimizing the first order approximation of $RE^2(\hat{r}_{opt}^{(T)})$ with respect to the transformed densities. We show it is equivalent to minimizing $H_{s_2}(q_1^{(T)}, q_2)$ with respect to $q_1^{(T)}$ using the variational lower bound of $H_\pi(q_1, q_2)$ (2.14) and f-GAN (Nowozin et al., 2016).

2.4.1 The f-GAN framework

We start by introducing the GAN and f-GAN models. A Generative Adversarial Network (GAN) (Goodfellow et al., 2014) is an expressive class of generative models. Let p_{tar} be the target distribution of interest. In the original GAN, Goodfellow et al. (2014) estimate a generative model p_ϕ parameterized by a real vector ϕ by

approximately minimizing the Jensen-Shannon divergence between p_ϕ and p_{tar}. The key idea of the original GAN is to introduce a separate discriminator which tries to distinguish between "true samples" from p_{tar} and artificially generated samples from p_ϕ. This discriminator is then optimized alongside with the generative model p_ϕ in the training process. See Creswell et al. (2018) for an overview of GAN models.

f-GAN (Nowozin et al., 2016) extends the original GAN model using the variational lower bound of f-divergence (2.12), and introduces a GAN-type framework that generalizes to minimizing any f-divergence between p_{tar} and p_ϕ. Let an f-divergence with the generator f be given. Nowozin et al. (2016) parameterize the variational function V_ξ and the generative model p_ϕ as two neural nets with parameters ξ and ϕ respectively, and propose

$$G(\phi, \xi) = E_{p_{tar}}(V_\xi(\omega)) - E_{p_\phi}(f^*(V_\xi(\omega))) \tag{2.20}$$

as the objective function of the f-GAN model, where f^* is the convex conjugate of the generator f of the chosen f-divergence. Recall that $G(\phi, \xi)$ is in the form of the variational lower bound (2.12) of $D_f(p_\phi, p_{tar})$. Nowozin et al. (2016) show that $D_f(p_\phi, p_{tar})$ can be minimized by solving $\min_\phi \max_\xi G(\phi, \xi)$. Intuitively, we can view $\max_\xi G(\phi, \xi)$ as an estimate of $D_f(p_\phi, p_{tar})$ (Nguyen et al., 2010). This means minimizing $\max_\xi G(\phi, \xi)$ with respect to ϕ can be interpreted as minimizing an estimate of $D_f(p_\phi, p_{tar})$.

Now we show how to use the f-GAN framework to construct the f-GAN-Bridge estimator (f-GB). Suppose q_1, q_2 are defined on a common support $\Omega = \mathbb{R}^d$. Let $T_\phi : \Omega \to \Omega$ be a transformation parameterized by a real vector $\phi \in \mathbb{R}^l$ that aims to map q_1 to q_2. Let $q_1^{(\phi)}$ be the transformed density obtained by applying T_ϕ to q_1, and $\tilde{q}_1^{(\phi)}$ be the corresponding unnormalized density. We also require $\tilde{q}_1^{(\phi)}$ to be computationally tractable, and $\tilde{q}_1^{(\phi)} = q_1^{(\phi)} Z_1$, i.e. $\tilde{q}_1^{(\phi)}$ and $\tilde{q}_1$ have the same normalizing constant Z_1. Let $\mathcal{T} = \{T_\phi : \phi \in \mathbb{R}^l\}$ be a collection of such transformations. Define $\hat{r}_{opt}^{(\phi)}$ to be the asymptotically optimal Bridge estimator of r based on the unnormalized densities $\tilde{q}_1^{(\phi)}, \tilde{q}_2$ and corresponding samples $\{T_\phi(\omega_{1j})\}_{j=1}^{n_1}, \{\omega_{2j}\}_{j=1}^{n_2}$. Let $\pi \in (0, 1)$. Define

$$G(\phi, \tilde{r}; \pi) = 1 - \tfrac{1}{\pi} E_{q_1^{(\phi)}} \left(\frac{\pi \tilde{q}_2(\omega)\tilde{r}}{(1-\pi)\tilde{q}_1^{(\phi)}(\omega) + \pi \tilde{q}_2(\omega)\tilde{r}} \right)^2 - \tfrac{1}{1-\pi} E_{q_2} \left(\frac{(1-\pi)\tilde{q}_1^{(\phi)}(\omega)}{(1-\pi)\tilde{q}_1^{(\phi)}(\omega) + \pi \tilde{q}_2(\omega)\tilde{r}} \right)^2. \tag{2.21}$$

By Proposition 2.3.1, $G(\phi, \tilde{r}; \pi)$ is the variational lower bound of $H_\pi(q_1^{(\phi)}, q_2)$. In order to illustrate our idea, we first give an idealized Algorithm 2.1 to find the f-GAN-Bridge estimator. A practical version will be given in the next section.

Algorithm 2.1 f-GAN-Bridge estimator (Idealized version)

Require: Samples $\{\omega_{ij}\}_{j=1}^{n_i} \sim q_i$ for $i = 1, 2$; Candidate transformations $T_\phi \in \mathcal{T}$ parameterized by $\phi \in \mathbb{R}^l$.
Set $n = n_1 + n_2$, $s_i = n_i/n$ for $i = 1, 2$.
Find $(\phi^*, \tilde{r}^*)$, a solution of $\min_{\phi\in\mathbb{R}^l} \max_{\tilde{r}\in\mathbb{R}^+} G(\phi, \tilde{r}; s_2)$ defined in (2.21).
Use the iterative procedure in (2.4) to compute the asymptotically optimal Bridge estimator $\hat{r}_{opt}^{(\phi^*)}$ based on $\tilde{q}_1^{(\phi^*)}$, $\tilde{q}_2$ and the samples $\{T_{\phi^*}(\omega_{1j})\}_{j=1}^{n_1}$,$\{\omega_{2j}\}_{j=1}^{n_2}$.
Compute $\widehat{RE}^2(\hat{r}_{opt}^{(\phi^*)}) = (s_1 s_2 n)^{-1}\left((1 - G(\phi^*, \tilde{r}^*; s_2))^{-1} - 1\right)$.
return $\hat{r}_{opt}^{(\phi^*)}$ as the f-GAN-Bridge estimate of r, $\widehat{RE}^2(\hat{r}_{opt}^{(\phi^*)})$ as an estimate of $RE^2(\hat{r}_{opt}^{(\phi^*)})$ and $MSE(\log \hat{r}_{opt}^{(\phi^*)})$.

Since $\tilde{q}_1^{(\phi)}$ and $\tilde{q}_1$ have the same normalizing constant by (2.6), $\hat{r}_{opt}^{(\phi)}$ is an asymptotically optimal Bridge estimator of r for any transformation $T_\phi \in \mathcal{T}$. We show that within the given family of transformations $\mathcal{T}$, Algorithm 2.1 is able to find T_{ϕ^*} that minimizes the first order approximation of $RE^2(\hat{r}_{opt}^{(\phi)})$ with respect to $T_\phi \in \mathcal{T}$ under the i.i.d. assumption.

Proposition 2.4.1 (Minimizing $RE^2(\hat{r}_{opt}^{(\phi)})$ using Algorithm 2.1)**.** *If $(\phi^*, \tilde{r}^*)$ is a solution of $\min_{\phi\in\mathbb{R}^l} \max_{\tilde{r}\in\mathbb{R}^+} G(\phi, \tilde{r}; s_2)$ defined in Algorithm 2.1, then $G(\phi, \tilde{r}^*; s_2) = H_{s_2}(q_1^{(\phi)}, q_2)$ for all $\phi \in \mathbb{R}^l$, T_{ϕ^*} minimizes $H_{s_2}(q_1^{(\phi)}, q_2)$ with respect to $T_\phi \in \mathcal{T}$. If the samples $\{\omega_{ij}\}_{j=1}^{n_i} \overset{i.i.d.}{\sim} q_i$ for $i = 1, 2$, then T_{ϕ^*} also minimizes $RE^2(\hat{r}_{opt}^{(\phi)})$ with respect to $T_\phi \in \mathcal{T}$ up to the first order.*

Proof. See Appendix 2.8.1. □

From Proposition 2.4.1 we see that under the i.i.d. assumption, T_{ϕ^*} and the corresponding f-GAN-Bridge estimator $\hat{r}_{opt}^{(\phi^*)}$ are optimal in the sense that $\hat{r}_{opt}^{(\phi^*)}$ attains the minimal RMSE (up to the first order) among all possible transformations $T_\phi \in \mathcal{T}$ and their corresponding $\hat{r}_{opt}^{(\phi)}$. Since $G(\phi^*, \tilde{r}^*; s_2) = H_{s_2}(q_1^{(\phi^*)}, q_2)$, $\widehat{RE}^2(\hat{r}_{opt}^{(\phi^*)})$ in Algorithm 2.1 is exactly the leading term of $RE^2(\hat{r}_{opt}^{(\phi^*)})$ in the form of (2.10). Note that by Proposition 2.3.1, $\tilde{r}^*$ is equal the true ratio of normalizing constants r. This means if we have $(\phi^*, \tilde{r}^*)$ in the idealized Algorithm 2.1, it seems there is no need to carry out the following Bridge sampling step. However, $(\phi^*, \tilde{r}^*)$ is not computable in practice as $G(\phi, \tilde{r}; s_2)$ depends on the unknown normalizing constants Z_1, Z_2. Therefore $G(\phi, \tilde{r}; s_2)$ has to be approximated by an empirical estimate, and its corresponding optimizer w.r.t. $\tilde{r}$ is no longer equal to r. In the next section, we will give a practical implementation of Algorithm 2.1 and discuss the role of $\tilde{r}^*$ when $G(\phi, \tilde{r}; s_2)$ is replaced by an empirical estimate of it.

In Algorithm 2.1, we use the f-GAN framework to minimize $H_{s_2}(q_1^{(\phi)}, q_2)$ with respect to $T_\phi \in \mathcal{T}$. We can also apply this f-GAN framework to minimizing other choices of f-divergences such as KL divergence, Squared Hellinger distance and weighted Jensen-Shannon divergence. However, these choices of f-divergence are less efficient compared to the weighted Harmonic divergence $H_{s_2}(q_1^{(\phi)}, q_2)$ if our goal is to improve the efficiency of $\hat{r}_{opt}^{(\phi)}$, as we show that minimizing these choices of f-divergence between $q_1^{(\phi)}$ and q_2 can be viewed as minimizing some *upper bounds* of the first order approximation of $RE^2(\hat{r}_{opt}^{(\phi)})$ (See Appendix 2.8.5).

2.4.2 Implementation details

In this section, we give a practical implementation of the idealized Algorithm 2.1 based on an alternative objective function. We first describe the practical version of Algorithm 2.1 in Chapter 2.4.2.1, then justify the choice of this alternative objective in Chapter 2.4.2.2.

2.4.2.1 A practical implementation of Algorithm 2.1

In this paper, we parameterize $q_1^{(\phi)}$ as a Normalizing flow. In particular, we parameterize $q_1^{(\phi)}$ as a Real-NVP (Dinh et al., 2016) with base density q_1 and a smooth, invertible transformation T_ϕ, where T_ϕ is parameterized by a real vector $\phi \in \mathbb{R}^l$. See Chapter 2.2.1 for a brief description of Real-NVP. Given samples $\{\omega_{ij}\}_{j=1}^{n_i} \sim q_i$ for $i = 1, 2$, define

$$\hat{G}(\phi, \tilde{r}; \pi, \{\omega_{ij}\}_{j=1}^{n_i}) = 1 - \frac{1}{\pi n_1} \sum_{j=1}^{n_1} \left(\frac{\pi \tilde{q}_2(T_\phi(\omega_{1j}))\tilde{r}}{(1-\pi)\tilde{q}_1^{(\phi)}(T_\phi(\omega_{1j})) + \pi \tilde{q}_2(T_\phi(\omega_{1j}))\tilde{r}} \right)^2 - \frac{1}{(1-\pi)n_2} \sum_{j=1}^{n_2} \left(\frac{(1-\pi)\tilde{q}_1^{(\phi)}(\omega_{2j})}{(1-\pi)\tilde{q}_1^{(\phi)}(\omega_{2j}) + \pi \tilde{q}_2(\omega_{2j})\tilde{r}} \right)^2 \quad (2.22)$$

to be the empirical estimate of $G(\phi, \tilde{r}; \pi)$ in (2.21). Unlike Algorithm 2.1, we do not aim to solve $\min_{\phi \in \mathbb{R}^l} \max_{\tilde{r} \in \mathbb{R}^+} \hat{G}(\phi, \tilde{r}; \pi, \{\omega_{ij}\}_{j=1}^{n_i})$ directly. Instead, we define our objective function as

$$L_{\lambda_1, \lambda_2}(\phi, \tilde{r}; \pi, \{\omega_{ij}\}_{j=1}^{n_i}) = -\log(1 - \hat{G}(\phi, \tilde{r}; \pi, \{\omega_{ij}\}_{j=1}^{n_i})) - \frac{\lambda_1}{n_1} \sum_{j=1}^{n_1} \left(\log \tilde{q}_2(T_\phi(\omega_{1j})) - \log \tilde{q}_1^{(\phi)}(T_\phi(\omega_{1j})) \right) - \frac{\lambda_2}{n_2} \sum_{j=1}^{n_2} \log \tilde{q}_1^{(\phi)}(\omega_{2j}), \quad (2.23)$$

where $\lambda_1, \lambda_2 \geq 0$ are two hyperparameters. We first give Algorithm 2.2, a practical implementation of Algorithm 2.1, then justify the choice of the objective function (2.23) and give implementation details in the following section.

Algorithm 2.2 f-GAN-Bridge estimator (Practical version)

Require: Training samples $\{\omega_{ij}\}_{j=1}^{n_i}$ and estimating samples $\{\omega'_{ij}\}_{j=1}^{n'_i}$ from q_i for $i = 1, 2$; Initial parameters $\phi_0 \in \mathbb{R}^l$, $\tilde{r}_0 > 0$; Learning rate $\eta_\phi, \eta_{\tilde{r}} > 0$; Tolerance level $\epsilon_1, \epsilon_2 > 0$; Hyperparameters $\lambda_1, \lambda_2 \geq 0$.

Transform and augment q_1, q_2 appropriately so that both densities are on a common support.

Set $t = 0, n' = n'_1 + n'_2, s_i = n'_i/n'$ for $i = 1, 2$

while $\left|L_{\lambda_1,\lambda_2}(\phi_t, \tilde{r}_t; s_2, \{\omega_{ij}\}_{j=1}^{n_i}) - L_{\lambda_1,\lambda_2}(\phi_{t-1}, \tilde{r}_{t-1}; s_2, \{\omega_{ij}\}_{j=1}^{n_i})\right| > \epsilon_1$ or $|\tilde{r}_t - \tilde{r}_{t-1}| > \epsilon_2$ or $t = 0$ **do**

Update $\phi_{t+1} = \phi_t - \eta_\phi \nabla_\phi L_{\lambda_1,\lambda_2}(\phi_t, \tilde{r}_t; s_2, \{\omega_{ij}\}_{j=1}^{n_i})$

Update $\tilde{r}_{t+1} = \tilde{r}_t + \eta_{\tilde{r}} \nabla_{\tilde{r}} L_{\lambda_1,\lambda_2}(\phi_t, \tilde{r}_t; s_2, \{\omega_{ij}\}_{j=1}^{n_i})$

Update $t = t + 1$

end while

Use $\tilde{r}_t$ as the initial value of the iterative procedure in (2.4), compute $\hat{r}_{opt}^{\prime(\phi_t)}$ based on $\tilde{q}_1^{(\phi_t)}, \tilde{q}_2$ and the estimating samples $\{T_{\phi_t}(\omega'_{1j})\}_{j=1}^{n'_1}$, $\{\omega'_{2j}\}_{j=1}^{n'_2}$.

Compute $\widehat{RE}^2(\hat{r}_{opt}^{\prime(\phi_t)}) = \max_{\tilde{r}\in\mathbb{R}^+}(s_1 s_2 n')^{-1}\left((1 - \hat{G}(\phi_t, \tilde{r}; s_2, \{\omega'_{ij}\}_{j=1}^{n'_i}))^{-1} - 1\right)$.

return $\hat{r}_{opt}^{\prime(\phi_t)}$ as the f-GAN-Bridge estimate of r; $\widehat{RE}^2(\hat{r}_{opt}^{\prime(\phi_t)})$ as an estimate of $RE^2(\hat{r}_{opt}^{\prime(\phi_t)})$ and $MSE(\log \hat{r}_{opt}^{\prime(\phi_t)})$.

In Algorithm 2.2, most of the computational cost is spent on estimating $q_1^{(\phi)}$. Since we parameterize $q_1^{(\phi)}$ as a Real-NVP in this paper, we leverage the GPU computing framework for neural networks. In particular, we implement Algorithm 2.2 using PyTorch (Paszke et al., 2017) and CUDA (NVIDIA et al., 2020). As a result, most of the computation of Algorithm 2.2 is parallelized and carried out on the GPU. This greatly accelerates the training process in Algorithm 2.2. We will further compare the computational cost of Algorithm 2.2 with existing improvement strategies for Bridge sampling (Meng and Schilling, 2002; Jia and Seljak, 2020; Wang et al., 2022) in Chapter 2.5 and 2.6.

2.4.2.2 Choosing the objective function

Note that the original empirical estimate $\hat{G}(\phi, \tilde{r}; s_2, \{\omega_{ij}\}_{j=1}^{n_i})$ can be extremely close to 1 when $q_1^{(\phi)}$ and q_2 share little overlap. In order to improve its numerical stability, we first transform $\hat{G}(\phi, \tilde{r}; s_2, \{\omega_{ij}\}_{j=1}^{n_i})$ to log scale using a monotonic function $h(x) = -\log(1 - x)$, then apply the log-sum-exp trick on the transformed

$-\log(1-\hat{G}(\phi,\tilde{r};s_2,\{\omega_{ij}\}_{j=1}^{n_i}))$. Since $h(x)$ is monotonically increasing on $(-\infty,1)$, applying this transformation does not change the optimizers of $\hat{G}(\phi,\tilde{r};s_2,\{\omega_{ij}\}_{j=1}^{n_i})$.

In addition, GAN-type models can be difficult to train in practice (Arjovsky and Bottou, 2017). Grover et al. (2018) suggest one can stabilize the adversarial training process of GAN-type models by incorporating a log likelihood term into the original objective function when the generative model $q_1^{(\phi)}$ is a Normalizing flow. Since both $\tilde{q}_1^{(\phi)}$ and $\tilde{q}_2$ are computationally tractable in our setup, we are able to extend this idea and stabilize the alternating training process by incorporating two "likelihood" terms that are asymptotically equivalent to $\lambda_1 KL(q_1^{(\phi)},q_2), \lambda_2 KL(q_2,q_1^{(\phi)})$ up to additive constants into the transformed f-GAN objective $-\log(1-\hat{G}(\phi,\tilde{r};s_2,\{\omega_{ij}\}_{j=1}^{n_i}))$. Our proposed objective function $L_{\lambda_1,\lambda_2}(\phi,\tilde{r};s_2,\{\omega_{ij}\}_{j=1}^{n_i})$ is then a weighted combination of $-\log(1-\hat{G}(\phi,\tilde{r};s_2,\{\omega_{ij}\}_{j=1}^{n_i}))$ and the two "likelihood" terms, where the hyper parameters $\lambda_1,\lambda_2\geq 0$ control the contribution of the "likelihood" terms.

Similar to Algorithm 2.1, let $(\phi_L^*,\tilde{r}_L^*)$ be a solution of the min-max problem

$$\min_{\phi\in\mathbb{R}^l}\max_{\tilde{r}\in\mathbb{R}^+} L_{\lambda_1,\lambda_2}(\phi,\tilde{r};s_2,\{\omega_{ij}\}_{j=1}^{n_i}).$$

Note that regardless of the choice of λ_1,λ_2, the scalar parameter $\tilde{r}$ only depends on $L_{\lambda_1,\lambda_2}(\phi,\tilde{r};s_2,\{\omega_{ij}\}_{j=1}^{n_i})$ through $\hat{G}(\phi,\tilde{r};s_2,\{\omega_{ij}\}_{j=1}^{n_i})$. Therefore by Proposition 2.3.2, if $\tilde{r}_L^*$ is a stationary point of $L_{\lambda_1,\lambda_2}(\phi_L^*,\tilde{r};s_2,\{\omega_{ij}\}_{j=1}^{n_i})$ w.r.t. $\tilde{r}\in\mathbb{R}^+$, then $\tilde{r}_L^*$ can be viewed as a Bridge estimator of r based on the transformed $\tilde{q}_1^{(\phi_L^*)}$ and the original $\tilde{q}_2$ with a specific choice of the free function $\alpha(\omega)$. However, $\tilde{r}_L^*$ is sub-optimal since the free function $\alpha(\omega)$ it uses is different from the optimal $\alpha_{opt}(\omega)$ in (2.2). This means $\tilde{r}_L^*$ will have greater asymptotic error than the asymptotically optimal Bridge estimator. In addition, $\tilde{r}_L^*$ suffers from an adaptive bias (Wang et al., 2022). Such bias arises from the fact that the estimated transformed density $q_1^{(\phi_t)}$ in Algorithm 2.2 is chosen based on the training samples $\{\omega_{ij}\}_{j=1}^{n_i}$ for $i=1,2$. This means the density of the distribution of the transformed training samples $\{T_{\phi_t}(\omega_{1j})\}_{j=1}^{n_1}$ is no longer proportional to $\tilde{q}_1^{(\phi_t)}(T_{\phi_t}(\omega_{1j}))$ for $j=1,...,n_1$, as ϕ_t can be viewed as a function of $\{\omega_{ij}\}_{j=1}^{n_i}$. Hence we do not use $\tilde{r}_L^*$ as our final estimator of r. Instead, once we have obtained $\tilde{r}_L^*$, we use it as a sensible initial value of the iterative procedure in (2.4), and compute the asymptotically optimal Bridge estimator $\hat{r}_{opt}^{\prime(\phi_L^*)}$ using a separate set of estimating samples $\{\omega'_{ij}\}_{j=1}^{n'_i}$, $i=1,2$. The resulting $\hat{r}_{opt}^{\prime(\phi_L^*)}$ does not suffer from the adaptive bias as the estimating samples are independent to the transformation $q_1^{(\phi_t)}$. When $n'_i=n_i$ for $i=1,2$, $\hat{r}_{opt}^{\prime(\phi_L^*)}$ is also statistically more efficient than $\tilde{r}_L^*$.

On the other hand, if ϕ_L^* is a minimizer of $L_{\lambda_1,\lambda_2}(\phi,\tilde{r}_L^*;s_2,\{\omega_{ij}\}_{j=1}^{n_i})$ with respect to ϕ, then it asymptotically minimizes a mixture of $-\log(1-H_{s_2}(q_1^{(\phi)},q_2))$, $KL(q_1^{(\phi)},q_2)$

and $KL(q_2, q_1^{(\phi)})$. Recall that as $n_1, n_2 \to \infty$, the additional log likelihood terms in (2.23) is asymptotically equivalent to $\lambda_1 KL(q_1^{(\phi)}, q_2), \lambda_2 KL(q_2, q_1^{(\phi)})$ up to additive constants. We have demonstrated that minimizing $-\log(1 - H_{s_2}(q_1^{(\phi)}, q_2))$ with respect to ϕ is equivalent to minimizing the first order approximation of $RE^2(\hat{r}_{opt}^{(\phi)})$ under the i.i.d. assumption. We can also show that minimizing $KL(q_1^{(\phi)}, q_2), KL(q_2, q_1^{(\phi)})$ correspond to minimizing upper bounds of the first order approximation of $RE^2(\hat{r}_{opt}^{(\phi)})$ w.r.t. ϕ under the same assumption (See Appendix 2.8.5). Note that when $\lambda_1, \lambda_2 \neq 0$, Proposition 2.4.1 no longer holds for this hybrid objective asymptotically, i.e. $T_{\phi_L^*}$ no longer asymptotically minimizes the first order approximation of $RE^2(\hat{r}_{opt}^{(\phi)})$ w.r.t. T_ϕ. However, we find Algorithm 2.2 with the hybrid objective works well in the numerical examples in Chapter 2.5, 2.6 for any value of $\lambda_1, \lambda_2 \in (10^{-2}, 10^{-1})$. We want to keep λ_1, λ_2 small since we do not want the log likelihood terms to dominate $\hat{G}(\phi, \tilde{r}; s_2, \{\omega_{ij}\}_{j=1}^{n_i})$ in the hybrid objective $L_{\lambda_1,\lambda_2}(\phi, \tilde{r}; s_2, \{\omega_{ij}\}_{j=1}^{n_i})$. In addition, we would like to stress that even though the final ϕ_t in Algorithm 2.2 does not asymptotically minimize the first order approximation of $RE^2(\hat{r}_{opt}^{(\phi)})$ w.r.t. ϕ when $\lambda_1, \lambda_2 > 0$, $\widehat{RE}^2(\hat{r}_{opt}^{\prime(\phi_t)})$ in Algorithm 2.2 is still a consistent estimator of the first order approximation of $RE^2(\hat{r}_{opt}^{\prime(\phi_t)})$ by Proposition 2.3.1 and the fact that $\hat{r}_{opt}^{\prime(\phi_t)}$ is the asymptotically optimal Bridge estimator based on the transformed $q_1^{(\phi_t)}$ and the original q_2.

2.4.2.3 Choosing the transformation T_ϕ

We parameterize $\tilde{q}_1^{(\phi)}$ as a Real-NVP (Dinh et al., 2016) with base density $\tilde{q}_1$ (See Chapter 2.2.1 for a brief description of Normalizing flow models and Real-NVP). As we have discussed before, this ensures that $\tilde{q}_1^{(\phi)}$ is both flexible and computationally tractable, and its normalizing constant is unchanged. It is possible to specify $\tilde{q}_1^{(\phi)}$ using a simpler parameterization, e.g. Warp-III transformation (Meng and Schilling, 2002). However, such parameterization is not as flexible comparing to a Normalizing flow. It is also possible to replace a Real-NVP by more sophisticated Normalizing flow architectures e.g. Autoregressive flows (Papamakarios et al., 2017) or Neural Spline flows (Durkan et al., 2019). But we find a Real-NVP is sufficient for us to illustrate our ideas and achieve satisfactory results in both simulated and real world examples. In addition, both the froward and inverse transformation of a Real-NVP can be computed efficiently. This is an appealing feature since we need both T_ϕ and T_ϕ^{-1} for evaluating $L_{\lambda_1,\lambda_2}(\phi, \tilde{r}; s_2, \{\omega_{ij}\}_{j=1}^{n_i})$. Therefore we choose to use a Real-NVP in Algorithm 2.2, as it has a good balance of flexibility and computational efficiency.

2.4.2.4 Splitting the samples from q_1, q_2

In Algorithm 2.2, we first estimate $\{\phi_t, \tilde{r}_t\}$ using the training samples $\{\omega_{ij}\}_{j=1}^{n_i}$, then compute the optimal Bridge estimator based on the separate estimating samples $\{\omega'_{ij}\}_{j=1}^{n'_i}$. We use separate samples for the Bridge sampling step because the estimated transformed density $q_1^{(\phi_t)}$ in Algorithm 2.2 is chosen based on the training samples $\{\omega_{ij}\}_{j=1}^{n_i}$ for $i = 1, 2$. This means the density of the distribution of the transformed training samples $\{T_{\phi_t}(\omega_{1j})\}_{j=1}^{n_1}$ is no longer proportional to $\tilde{q}_1^{(\phi_t)}(T_{\phi_t}(\omega_{1j}))$ for $j = 1, ..., n_1$ as ϕ_t can be viewed as a function of $\{\omega_{ij}\}_{j=1}^{n_i}$. If we apply the iterative procedure (2.4) to densities $q_1^{(\phi_t)}, q_2$ and the transformed training samples $\{T_{\phi_t}(\omega_{1j})\}_{j=1}^{n_1}, \{\omega_{2j}\}_{j=1}^{n_2}$, then the resulting $\hat{r}_{opt}^{(\phi_t)}$ will be a biased estimate of r. See also Wong et al. (2020) for a detailed discussion under a similar setting. One way to correct this bias is to split the samples from q_1, q_2 into training samples $\{\omega_{ij}\}_{j=1}^{n_i}$ and estimating samples $\{\omega'_{ij}\}_{j=1}^{n'_i}$ for $i = 1, 2$. We first estimate the transformation T_{ϕ_t} using the training samples $\{\omega_{ij}\}_{j=1}^{n_i}$, $i = 1, 2$. Once we have obtained the estimated ϕ_t, we apply the iterative procedure (2.4) to $\tilde{q}_1^{(\phi_t)}, \tilde{q}_2$ and the transformed estimating samples $\{T_{\phi_t}(\omega'_{1j})\}_{j=1}^{n'_1}, \{\omega'_{2j}\}_{j=1}^{n'_2}$, $i = 1, 2$. Then the resulting estimate $\hat{r}_{opt}^{\prime(\phi_t)}$ will not suffer from this bias. The same approach is used in Wang et al. (2022) and Jia and Seljak (2020). The idea of eliminating this bias by splitting the samples from q_1, q_2 is further discussed in Wong et al. (2020). The above argument also applies to the estimation of $RE^2(\hat{r}_{opt}^{\prime(\phi_t)})$. We compute $\widehat{RE}^2(\hat{r}_{opt}^{\prime(\phi_t)})$ based on the independent estimating samples using (2.16). Since finding $\widehat{RE}^2(\hat{r}_{opt}^{\prime(\phi_t)})$ is a 1-d optimization problem, the additional computational cost is negligible compared with the rest of Algorithm 2.2. In practice, we recommend setting $n_i = n'_i$ for $i = 1, 2$, i.e. splitting the samples from q_1, q_2 into equally sized training samples and estimating samples.

2.4.2.5 Finding the saddle point using alternating gradient method

In Algorithm 2.2, we aim to find a saddle point of $L_{\lambda_1,\lambda_2}(\phi, \tilde{r}; s_2, \{\omega_{ij}\}_{j=1}^{n_i})$ using the alternating gradient method. This approach is adapted from the Algorithm 1 of Nowozin et al. (2016). The authors show that their Algorithm 1 converges geometrically to a saddle point under suitable conditions. In the alternating training process of Algorithm 2.2, updating $\tilde{r}_{t+1}$ is a 1-d optimization problem when ϕ_t is treated as fixed for any step t. Hence we can also directly find $\hat{r}_{\phi_t} = \arg\max_{\tilde{r}\in\mathbb{R}^+} L_{\lambda_1,\lambda_2}(\phi_t, \tilde{r}; s_2, \{\omega_{ij}\}_{j=1}^{n_i})$ instead of performing a single step gradient ascent on $\tilde{r}_t$. By Proposition 2.3.1 and 2.3.2, $\hat{r}_{\phi_t}$ can be viewed as a (biased) Bridge estimator of r given ϕ_t. However, such estimator $\hat{r}_{\phi_t}$ is not reliable when $q_1^{(\phi_t)}$ and q_2 share little overlap. Therefore directly optimizing

$\tilde{r}$ at each iteration t is not always necessary in practice, especially at the early stage of training when $q_1^{(\phi_t)}$ is not yet a sensible approximation of q_2. In addition, the gradient ascent update of $\tilde{r}_t$ is computationally cheaper than finding the optimizer $\hat{r}_{\phi_t}$ directly. Therefore we follow Nowozin et al. (2016) and use the alternating gradient method to find the saddle point of $L_{\lambda_1,\lambda_2}(\phi, \tilde{r}; s_2, \{\omega_{ij}\}_{j=1}^{n_i})$. We only recommend optimizing $\tilde{r}_t$ directly in Algorithm 2.2 when we know $q_1^{(\phi_t)}$ and q_2 have at least some degree of overlap.

Note that $\{\phi_t, \tilde{r}_t\}$ being approximately a saddle point of the objective function does not necessarily imply that it solves $\min_{\phi\in\mathbb{R}^l} \max_{\tilde{r}\in\mathbb{R}^+} L_{\lambda_1,\lambda_2}(\phi, \tilde{r}; s_2, \{\omega_{ij}\}_{j=1}^{n_i})$. For $\tilde{r}_t$, it is easy to verify if $\tilde{r}_t$ is indeed the maximizer of $L_{\lambda_1,\lambda_2}(\phi_t, \tilde{r}; s_2, \{\omega_{ij}\}_{j=1}^{n_i})$ w.r.t. $\tilde{r} \in \mathbb{R}^+$ since it is a 1-d optimization problem. However, for ϕ_t there is no guarantee that it is the global minimizer of $L_{\lambda_1,\lambda_2}(\tilde{r}_t, \phi; s_2, \{\omega_{ij}\}_{j=1}^{n_i})$ w.r.t. $\phi \in \mathbb{R}^l$. One way to address this problem is to run Algorithm 2.2 multiple times and choose the $q_1^{(\phi_t)}$ that attains the smallest objective function value. In the numerical examples, we find $q_1^{(\phi_t)}$ returned from Algorithm 2.2 is almost always a good approximation of q_2. Therefore we do not worry about this problem in practice.

In the alternating training process, seeing the absolute difference between $L_{\lambda_1,\lambda_2}(\phi_t, \tilde{r}_t; s_2, \{\omega_{ij}\}_{j=1}^{n_i})$ and $L_{\lambda_1,\lambda_2}(\phi_{t-1}, \tilde{r}_{t-1}; s_2, \{\omega_{ij}\}_{j=1}^{n_i})$ being less than the tolerance level ϵ_1 at an iteration t does not necessarily imply that it has reached a saddle point. Therefore we also need to monitor the sequence $\{\tilde{r}_t\}$, $t = 0, 1, 2, ...$ in the training process. If $|\tilde{r}_t - \tilde{r}_{t-1}| > \epsilon_2$, then $\tilde{r}_t$ has not converged to a stationary point regardless of the value of the objective function. In other words, we know $\{\phi_t, \tilde{r}_t\}$ has approximately converged to a saddle point only if both the objective function $L_{\lambda_1,\lambda_2}(\phi_t, \tilde{r}_t; s_2, \{\omega_{ij}\}_{j=1}^{n_i})$ and $\tilde{r}_t$ have stopped changing. In practice, we recommend setting $\epsilon_1 \in (10^{-3}, 10^{-1})$ depending on the scale of the objective function, and $\epsilon_2 \in (10^{-3}, 10^{-2})$.

2.5 Example 1: Mixture of Rings

We first demonstrate the effectiveness of the f-GAN-Bridge estimator and Algorithm 2.2 using a simulated example. Since this paper focuses on improving the original Bridge estimator (Meng and Wong, 1996) rather than giving a new estimator of the normalizing constant or the ratio of normalizing constants, we will focus on comparing the performance of the proposed f-GAN-Bridge estimator to existing improvement strategies for Bridge sampling (Meng and Schilling, 2002; Wang et al., 2022; Jia and Seljak, 2020) in this and the following section. We do not include other classes of

methods such as path sampling (Gelman and Meng, 1998; Lartillot and Philippe, 2006), nested sampling (Skilling et al., 2006), variational approaches (Ranganath et al., 2014), etc. in the examples. Empirical study (Fourment et al., 2020) finds evidence that Bridge sampling was competitive with a wide range of methods, including the methods mentioned above, in the context of phylogenetics.

In this example, we set q_1, q_2 to be mixtures of ring-shaped distributions, and we would like to estimate the ratio of their normalizing constants. We choose this example because such mixture has a multi-modal structure, and its normalizing constant is available in closed form. Let $\boldsymbol{x} \in \mathbb{R}^2$. In order to define the pdf of q_1, q_2 for this example, we first define the pdf of a 2-d ring distribution as

$$R(\boldsymbol{x}; \boldsymbol{\mu}, b, \sigma) = \frac{1}{\sqrt{2\pi^3\sigma^2}\Phi(b/\sigma)} \exp\left(-\frac{(\|\boldsymbol{x}-\boldsymbol{\mu}\|_2^2 - b)^2}{2\sigma^2}\right); \quad \boldsymbol{\mu} \in \mathbb{R}^2,\ b, \sigma > 0 \quad (2.24)$$

where $\Phi(\cdot)$ is the standard Normal CDF and $\boldsymbol{\mu}, b, \sigma$ controls the location, radius and thickness of the ring respectively. Let $\tilde{R}(\boldsymbol{x}; \boldsymbol{\mu}, b, \sigma) = \exp\left(-\frac{(\|\boldsymbol{x}-\boldsymbol{\mu}\|_2^2-b)^2}{2\sigma^2}\right)$ be the corresponding unnormalized density. Let $\boldsymbol{\omega} \in \mathbb{R}^p$ where p is an even integer. For $i = 1, 2$, let the unnormalized density $\tilde{q}_i$ be

$$\tilde{q}_i(\boldsymbol{\omega}; \boldsymbol{\mu}_{i1}, \boldsymbol{\mu}_{i2}, b_i, \sigma_i) = \prod_{j=1}^{p/2} \left(\frac{1}{2}\tilde{R}\left(\{\omega_{2j-1}, \omega_{2j}\}; \boldsymbol{\mu}_{i1}, b_i, \sigma_i\right) + \frac{1}{2}\tilde{R}\left(\{\omega_{2j-1}, \omega_{2j}\}; \boldsymbol{\mu}_{i2}, b_i, \sigma_i\right)\right) \quad (2.25)$$

where ω_j is the jth entry of $\boldsymbol{\omega}$. This means for $i = 1, 2$, if $\boldsymbol{\omega} \sim q_i$, then every two entries of $\boldsymbol{\omega}$ are independent and identically distributed, and follow an equally weighted mixture of 2-d ring distributions with different location parameters $\boldsymbol{\mu}_{i1}, \boldsymbol{\mu}_{i2}$ and the same radius and thickness parameter b_i, σ_i. It is straightforward to verify that Z_i, the normalizing constant of $\tilde{q}_i$ is $\left(\sqrt{2\pi^3\sigma_i^2}\Phi(b_i/\sigma_i)\right)^{p/2}$. In this example, we consider dimension $p = \{12, 18, 24, 30, 36, 42, 48\}$, and set $\boldsymbol{\mu}_{11} = (2, 2), \boldsymbol{\mu}_{12} = (-2, -2), \boldsymbol{\mu}_{21} = (3, -3), \boldsymbol{\mu}_{22} = (-3, 3)$, $b_1 = 3, b_2 = 6, \sigma_1 = 1, \sigma_2 = 2$.

In this example, we estimate $\log r = \log Z_1 - \log Z_2$ using the f-GAN-Bridge estimator (f-GB, Algorithm 2.2), Warp-III Bridge estimator (Meng and Schilling, 2002), Warp-U Bridge estimator (Wang et al., 2022) and Gaussianzed Bridge Sampling (GBS) (Jia and Seljak, 2020). We fix N_i, the number of samples from q_i, to be 2000 for $i = 1, 2$, and compare the performance of these methods as we increase the dimension p. For each value of p, we run each methods 100 times. For Algorithm 2.2, we set $\lambda_1, \lambda_2 = 0.05$, and $\tilde{q}_1^{(\phi)}$ to be a Real-NVP with 4 coupling layers. For

Warp-III and GBS, we use the recommended or default settings. For Warp-U, we adopt the cross splitting strategy suggested by the authors: We first estimate the Warp-U transformation using first half of the samples as the training set, and compute the Warp-U Bridge estimator using the second half as the estimating set. We then swap the role of the training and estimating set to compute another Warp-U Bridge estimator. The final output would then be the average of the two Warp-U Bridge estimators. This idea has also been discussed in Wong et al. (2020). Let $\hat{r}$ be a generic estimator of r. For each method and each value of p, we compute a MC estimate of the MSE of $\log \hat{r}$ based on the results from the repeated runs. We use it as the benchmark of performance. From Figure 2.1 we see f-GB outperforms all other methods for all choices of p. We also include a scatter plot of the first two dimensions of samples from q_1, q_2 and the transformed $q_1^{(\phi_t)}$ when $p = 48$, where $q_1^{(\phi_t)}$ is estimated using Algorithm 2.2 with $n_i = n_i' = N_i/2$ for $i = 1, 2$. We see the transformed $q_1^{(\phi_t)}$ captures the structure of q_2 accurately, and they share much greater overlap than the original q_1, q_2.

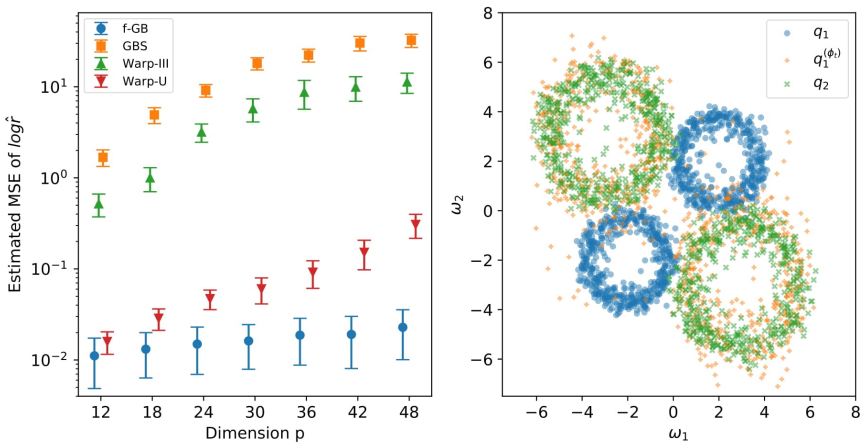


Figure 2.1: Left: MC estimates of MSE of $\log \hat{r}$ for each methods. Vertical segments are 2σ error bars. Note that the y-axis is on log scale. Right: Scatter plot of the first two dimensions of samples from q_1, q_2 and $q_1^{(\phi_t)}$ when $p = 48$. $q_1^{(\phi_t)}$ is obtained from Algorithm 2.2 with $n_i = n_i' = 1000$ for $i = 1, 2$.

We now compare the computational cost of these methods. Recall that our Algorithm 2.2 utilizes GPU acceleration. Because of the difference in GPU and CPU computing, it is not straightforward to compare the computational cost of Algorithm 2.2 with GBS, Warp-III and Warp-U, which are CPU based, using benchmarks such as CPU seconds or number of function calls. We simply report the averaged running time

for each method on our machine in Figure 2.2. Similar to Wang et al. (2022), we will also report the average "precision per second", which is the reciprocal of the product of the running time and the estimated MSE of $\log \hat{r}$, for each method (higher precision per second means better efficiency). We see that for all methods, the computation time is approximately a linear function of the dimension p. Even though f-GB takes roughly twice longer to run compared to GBS and $30 \sim 40$ times longer compared to Warp-III, it achieves the highest precision per second for all dimension p we consider. In addition, we also run further simulations with larger sample sizes. We find that when $p = 48$, Warp-U needs around $N_1 = N_2 = 7500$ samples to reach a similar level of precision as f-GB based on $N_1 = N_2 = 2000$ samples. In this case, Warp-U takes around $3 \sim 4$ times longer to run compared to f-GB. For Warp-III and GBS, we further increase the sample size to $N_1 = N_2 = 5 \times 10^4$, but find that their performance is still worse than f-GB and Warp-U, and both take more than three times longer to run. For Warp-III and Warp-U, it is not obvious how they would benefit from GPU computation. Although GBS may benefit from GPU acceleration in principle, it would require careful implementation and optimization. Therefore we compare our Algorithm 2.2 to these methods based on their publicly available implementations.

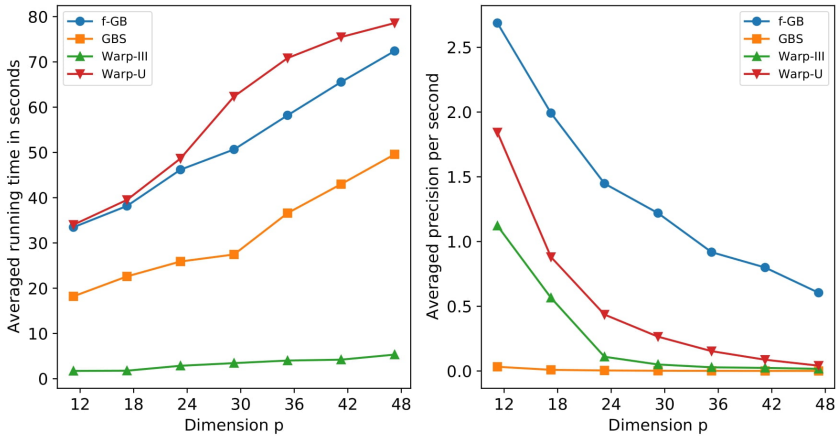


Figure 2.2: Left: Averaged running time for each method. Right: Averaged precision per second (i.e. reciprocal of the product of running time and the estimated MSE of $\log \hat{r}$) for each method.

Recall that $MSE(\log \hat{r}_{opt})$ is asymptotically equivalent to $RE^2(\hat{r}_{opt})$ (Meng and Wong, 1996). Therefore $\widehat{RE}^2(\hat{r}_{opt}^{\prime(\phi_t)})$ returned from Algorithm 2.2 can also be viewed as an estimate of $MSE(\log \hat{r}_{opt}^{\prime(\phi_t)})$. In order to assess its accuracy, we compare it with both the error estimator given in Frühwirth-Schnatter (2004) and a direct MC

estimator of $MSE(\log \hat{r}_{opt}^{\prime(\phi_t)})$: For each value of p, we first run Algorithm 2.2 with $N_1 = N_2 = 2000$ samples as before (i.e. we set $n_i = n_i' = 1000$ for $i = 1, 2$). We then fix the transformed density $\tilde{q}_1^{(\phi_t)}$ obtained from Algorithm 2.2, repeatedly draw $n_1' = n_2' = 1000$ independent samples from $\tilde{q}_1^{(\phi_t)}, q_2$ and record $\hat{r}_{opt}^{\prime(\phi_t)}$, $\widehat{RE}^2(\hat{r}_{opt}^{\prime(\phi_t)})$ and the error estimate given in Frühwirth-Schnatter (2004) (F-S) based on these new samples. We repeat this process 100 times, and report the box plots of $\widehat{RE}^2(\hat{r}_{opt}^{\prime(\phi_t)})$ and the error estimates given in Frühwirth-Schnatter (2004) (F-S) based on the repeated runs. We also compare the results with the direct MC estimate of $MSE(\log \hat{r}_{opt}^{\prime(\phi_t)})$ based on the repeated estimates $\log \hat{r}_{opt}^{\prime(\phi_t)}$ and the ground truth $\log r$. Note that here we fix the transformed $\tilde{q}_1^{(\phi_t)}$ and only repeat the Bridge sampling step in Algorithm 2.2. We summarize the results in Figure 2.3. We see that $\widehat{RE}^2(\hat{r}_{opt}^{\prime(\phi_t)})$ returned from Algorithm 2.2 agrees with the error estimator given in Frühwirth-Schnatter (2004) (F-S), and provides a sensible estimate of $MSE(\log \hat{r}_{opt}^{\prime(\phi_t)})$ for all choices of p.

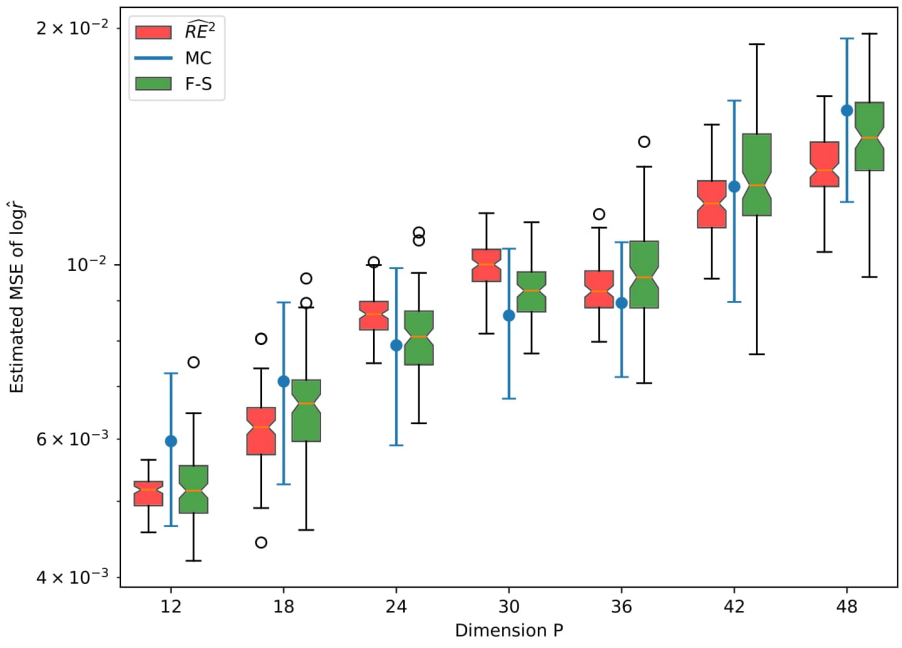


Figure 2.3: Box plots of 100 repetitions of $\widehat{RE}^2(\hat{r}_{opt}^{\prime(\phi_t)})$ based on Algorithm 2.2 and the error estimator given in Frühwirth-Schnatter (2004) (F-S) for each dimension P. Blue vertical segments are the 2σ error bars of the corresponding MC estimates of $MSE(\log \hat{r}_{opt}^{\prime(\phi_t)})$ based on 100 repetitions.

2.6 Example 2: Comparing two Bayesian GLMMs

In this Chapter we demonstrate the effectiveness of the f-GAN-Bridge estimator and Algorithm 2.2 by considering a Bayesian model comparison problem based on the six cities dataset (Fitzmaurice and Laird, 1993), where q_1, q_2 are the posterior densities of the parameters of two Bayesian GLMMs M_1, M_2. This example is adapted from Overstall and Forster (2010). We choose this example because it is based on real world dataset, and the posteriors q_1, q_2 are relatively high dimensional and are defined on disjoint supports with different dimensions.

The six cities dataset consists of the wheezing status y_{ij} (1 = wheezing, 0 otherwise) of child i at time j for $i = 1, ..., n$, $n = 537$ and $j = 1, ...4$. It also includes x_{ij}, the smoking status (1 = smoke, 0 otherwise) of the i-th child's mother at time-point j as a covariate. We compare two mixed effects logistic regression models M_1, M_2 with different linear predictors. Define

$$M1 : \eta_{ij}^{(1)} = \beta_0 + u_i; \quad u_i \overset{i.i.d.}{\sim} N(0, \sigma^2) \tag{2.26}$$

$$M2 : \eta_{ij}^{(2)} = \beta_0 + \beta_1 x_{ij} + u_i; \quad u_i \overset{i.i.d.}{\sim} N(0, \sigma^2) \tag{2.27}$$

where β_0, β_1 are regression parameters, u_i is the random effect of the i-th child and σ^2 controls the variance of the random effects. We use the default prior given by Overstall and Forster (2010) for both models, i.e. we take $\beta_0 \sim N(0, 4)$, $\sigma^{-2} \sim \Gamma(0.5, 0.5)$ for M_1 and $(\beta_0, \beta_1) \sim N(0, 4n(\boldsymbol{X}^T\boldsymbol{X})^{-1})$, $\sigma^{-2} \sim \Gamma(0.5, 0.5)$ for M_2 where $\boldsymbol{X} = [\mathbf{1}_{4n}^T, (\boldsymbol{x}_1, ..., \boldsymbol{x}_n)^T]$, $\boldsymbol{x}_i = (x_{i1}, ..., x_{i4})$ for $i = 1, ..., n$.

Let $\boldsymbol{y} = (\boldsymbol{y}_1, ...\boldsymbol{y}_n)$ with $\boldsymbol{y}_i = (y_{i1}, ..., y_{i4})$. Let $\boldsymbol{u} = (u_1, ...u_n)$ be the vector of random effects. Let $q_1(\beta_0, \boldsymbol{u}) = p(\beta_0, \boldsymbol{u}|\boldsymbol{X}, \boldsymbol{y}, M_1)$ be the marginal posterior of $(\beta_0, \boldsymbol{u})$ under M_1, and $\tilde{q}_1(\beta_0, \boldsymbol{u})$ be the corresponding unnormalized density. Let $q_2(\beta_0, \beta_1, \boldsymbol{u})$, $\tilde{q}_2(\beta_0, \beta_1, \boldsymbol{u})$ be defined in a similar fashion under M_2. Samples of q_1, q_2 are obtained using MCMC package `R2WinBUGS` (Sturtz et al., 2005; Lunn et al., 2000). For $k = 1, 2$, the normalizing constant Z_k of $\tilde{q}_k$ is the marginal likelihood under M_k. We first generate 2×10^5 MCMC samples from q_1, q_2 and estimate $\log Z_1, \log Z_2$ using the method described in Overstall and Forster (2010). The estimated log marginal likelihoods of M_1, M_2 based on 2×10^5 MCMC samples are -808.139 and -809.818 respectively. The results are consistent with the estimated log marginal likelihoods reported in Overstall and Forster (2010) based on 5×10^4 MCMC samples. We take them as the baseline "true values" of $\log Z_1$ and $\log Z_2$. See Overstall and Forster (2010) for R codes and technical details.

Similar to the previous example, we use f-GB to estimate the log Bayes factor $\log r = \log Z_1 - \log Z_2$ between M_1, M_2. Note that q_1, q_2 are defined on disjoint support $\mathbb{R}^{n+1}, \mathbb{R}^{n+2}$ respectively. In order to apply our Algorithm 2.2 to this problem, we first augment q_1 using a standard Normal to match up the difference in dimension between q_1 and q_2: Let $q_{1,aug}(\beta_0, \gamma, \boldsymbol{u}) = q_1(\beta_0, \boldsymbol{u})N(\gamma; 0, 1)$ be the augmented density where $N(\cdot; 0, 1)$ is the standard Normal pdf. Let $\tilde{q}_{1,aug}$ be the corresponding unnormalized augmented density. Note that $\tilde{q}_{1,aug}$ and $\tilde{q}_1$ have the same normalizing constant Z_1. We can then apply Algorithm 2.2 to $q_{1,aug}$ and q_2 since $q_{1,aug}$ and q_2 are now defined on a common support $\mathbb{R}^{n+2}$. We can sample from $q_{1,aug}$ by simply concatenating a sample $(\beta_0, \boldsymbol{u}) \sim q_1$ and a sample $\gamma \sim N(0, 1)$.

Let N_k be the number of MCMC samples drawn from q_k for $k = 1, 2$. In this example, we compare the performance of the f-GAN-Bridge estimator with the Warp-III Bridge estimator and the Warp-U Bridge estimator as we increase the number of MCMC samples N_1, N_2. We consider sample size $N = \{1000, 2000, 3000, 4000, 5000\}$. This is a challenging task since the sample size N is limited compared to the dimension of the problem (Recall that q_1, q_2 are defined on $\mathbb{R}^{n+1}, \mathbb{R}^{n+2}$ respectively with $n = 537$). For each choice of N, we repeatedly draw $N_1 = N_2 = N$ MCMC samples from q_1, q_2 respectively and estimate the MSE of $\log \hat{r}$ for each method in the same way as in the previous example. For our Algorithm 2.2, we augment q_1 as described above, set $\lambda_1, \lambda_2 = 0.1$ and $q_{1,aug}^{(\phi)}$ to be a Real-NVP with 10 coupling layers. For the Warp-U and Warp-III Bridge estimator, we still use the recommended or default settings. We do not include GBS in this example since we find that for all values of N, it does not converge for most of the repetitions. From Figure 2.4 we see our Algorithm 2.2 outperforms the Warp-III and the Warp-U Bridge estimator for all sample size N. We also include a scatter plot of the first two dimensions of samples from $q_{1,aug}, q_2$ and the transformed $q_{1,aug}^{(\phi_t)}$, where $q_{1,aug}^{(\phi_t)}$ is obtained from Algorithm 2.2 with $N = 3000$. We see $q_{1,aug}^{(\phi_t)}$ an q_2 share much greater overlap than the original $q_{1,aug}, q_2$. From Figure 2.5 we see for the same sample size N, the running time of f-GB is $4 \sim 6$ times as long as Warp-III, and roughly $30\% \sim 40\%$ shorter than Warp-U. On the other hand, f-GB achieves the highest precision per second for all sample size N in this example. We further increase the sample size N, and find that Warp-U requires around 10^4 MCMC samples to reach a similar level of precision achieved by f-GB with $N = 5000$ samples, and takes around 2 times longer to run. Similarly, Warp-III requires around 8×10^4 samples to get a similar level of precision, and takes around three times longer to run.

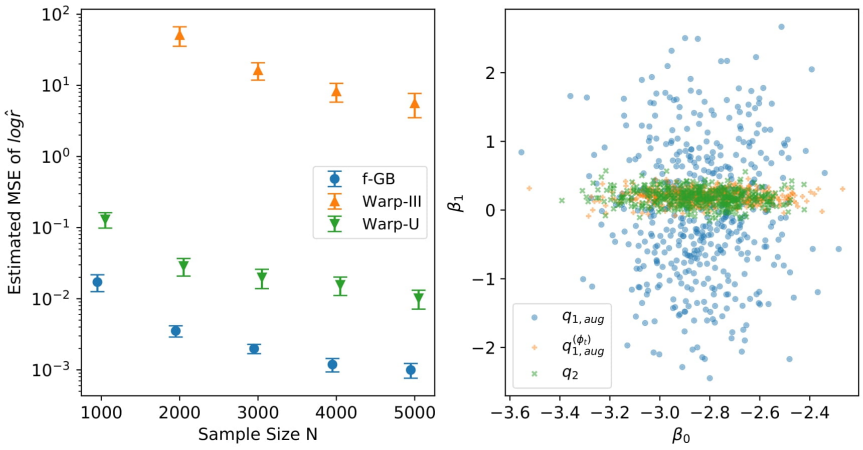


Figure 2.4: Left: MC estimates of MSE of $\log \hat{r}$ for each methods. Vertical segments are 2σ error bars. Note that the y-axis is on log scale. Warp-III does not converge for most of the repetitions when $N = 1000$. Right: Scatter plot of the first two dimensions of samples from $q_{1,aug}, q_2$ and $q_{1,aug}^{(\phi_t)}$, where $q_{1,aug}^{(\phi_t)}$ is obtained from Algorithm 2.2 with $n_1 = n_i' = 1500$ for $i = 1, 2$. The first two dimensions of $q_{1,aug}$ and q_2 are $(\beta_0, \gamma), (\beta_0, \beta_1)$ respectively.

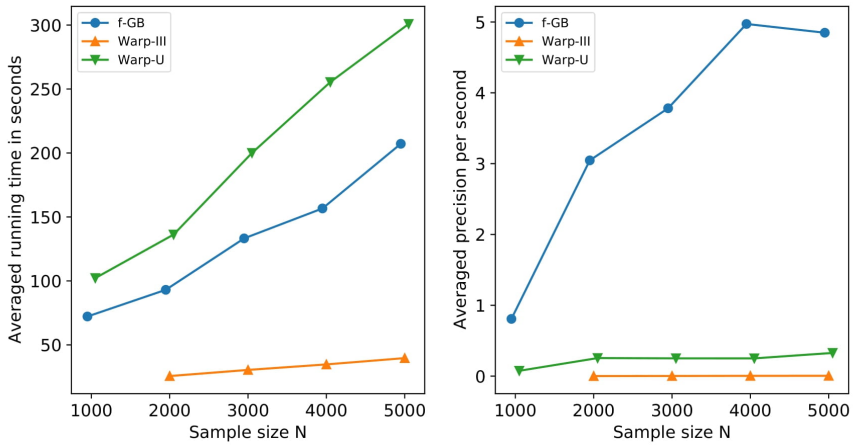


Figure 2.5: Left: Averaged running time for each method. Warp-III does not converge for most of the repetitions when $N = 1000$. Right: Averaged precision per second (i.e. reciprocal of the product of running time and the estimated MSE of $\log \hat{r}$) for each method.

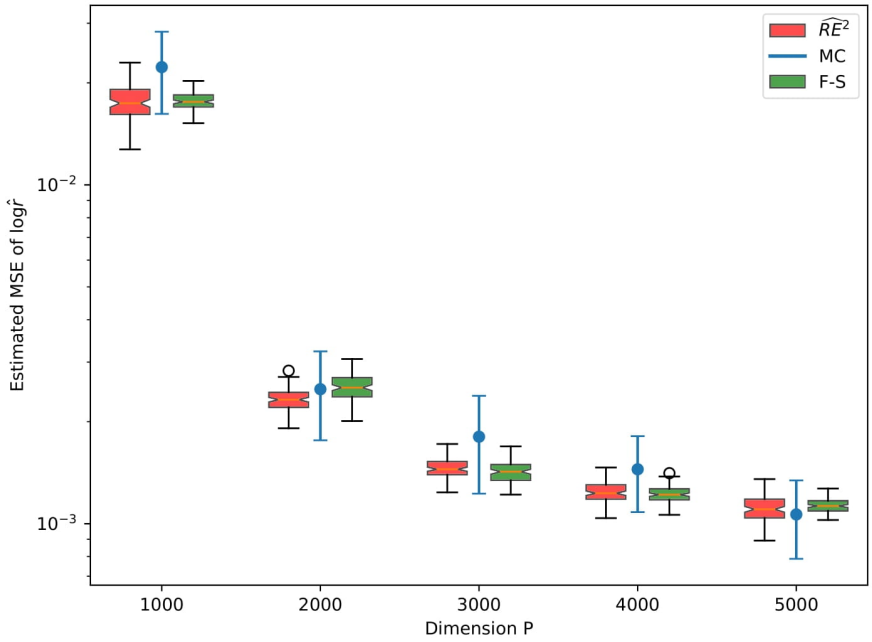


Figure 2.6: Box plots of 100 repetitions of $\widehat{RE}^2(\hat{r}_{opt}^{\prime(\phi_t)})$ based on Algorithm 2.2 and the error estimator given in Frühwirth-Schnatter (2004) (F-S) for each sample size N. Blue vertical segments are the 2σ error bars of the corresponding MC estimates of $MSE(\log \hat{r}_{opt}^{\prime(\phi_t)})$ based on 100 repetitions.

For each choice of N, we also compare $\widehat{RE}^2(\hat{r}_{opt}^{\prime(\phi_t)})$ returned from Algorithm 2.2 with the error estimator given in Frühwirth-Schnatter (2004) (F-S) and a direct MC estimator of $MSE(\log \hat{r}_{opt}^{\prime(\phi_t)})$ in the same way as in the last example. We summarize the results in Figure 2.6. In principle, it is not appropriate to use $\widehat{RE}^2(\hat{r}_{opt}^{\prime(\phi_t)})$ as an estimate of $MSE(\log \hat{r}_{opt}^{\prime(\phi_t)})$ in this example as the MCMC samples are correlated. However, from Figure 2.6 we see it agrees with the error estimator given in Frühwirth-Schnatter (2004), which does take autocorrelation into account, and still provides sensible estimate of $MSE(\log \hat{r}_{opt}^{\prime(\phi_t)})$ for all choices of N. This is likely due to the fact that the autocorrelation in our MCMC samples is weak, as we find that for all N, the effective sample sizes for all dimensions of the MCMC samples from q_1, q_2 are greater than $0.8N$. When working with weakly correlated MCMC samples, we recommend users to compute both our $\widehat{RE}^2(\hat{r}_{opt}^{\prime(\phi_t)})$ and the error estimator given in Frühwirth-Schnatter (2004), which does take autocorrelation into account, and check if they agree with each other. When the MCMC samples are strongly correlated, we do not recommend using $\widehat{RE}^2(\hat{r}_{opt}^{\prime(\phi_t)})$ as the error estimate of $\hat{r}_{opt}^{\prime(\phi_t)}$.

2.7 Conclusion and further discussion

In this chapter, we give a new estimator of $RE^2(\hat{r}_{opt})$ based on the variational lower bound of f-divergence proposed by Nguyen et al. (2010), discuss the connection between Bridge estimators and the problem of f-divergence estimation, and give a computational framework to improve the optimal Bridge estimator using an f-GAN (Nowozin et al., 2016). We show that under the i.i.d. assumption, our f-GAN-Bridge estimator is optimal in the sense that it asymptotically minimizes the first order approximation of $RE^2(\hat{r}_{opt}^{(\phi)})$ with respect to the transformed density $q_1^{(\phi)}$. We see that in both simulated and real world examples, our f-GB estimator provides accurate estimate of r and outperforms existing methods significantly. In addition, Algorithm 2.2 also provides accurate estimates of $RE^2(\hat{r}_{opt}^{(\phi)})$ and $MSE(\log \hat{r}_{opt}^{(\phi)})$. In our experience, Algorithm 2.2 (f-GB) is computationally more demanding than the existing methods. In the numerical examples, the running time of Algorithm 2.2 is roughly 1 to 3 times as long as the existing methods such as Warp-U and GBS when the sample size are the same. We have not attempted to formalize the difference in computational cost because of the very different nature of GPU and CPU computing. Although in our examples, it is possible for a competing method to match the performance of the f-GB estimator by increasing the number of samples drawn from q_1, q_2, it takes longer to run, and can be inefficient or impractical when sampling from q_1, q_2 is computationally expensive. This also means the f-GB estimator is especially appealing when we only have a limited amount of samples from q_1, q_2. In summary, when q_1, q_2 are relatively simple-structured and low dimensional, the extra computational cost required by f-GB may not be worthwhile. However, when q_1, q_2 are high dimensional or have complicated multi-modal structure, we recommend the users to choose the more accurate f-GB estimator of r, given the key summary role it plays in many applications and publications.

2.7.1 Computational cost of GPU-accelerated Algorithm 2.2

Comparing the computational cost of our Algorithm 2.2 with existing methods and their existing implementations is not straightforward because of the very different nature of GPU and CPU computing. In both examples, we compare the existing CPU implementations of Warp-III, Warp-U, GBS and a GPU implementation of our Algorithm 2 in term of wall clock time. We think this comparison is not unfair because there is no simple way to accelerate existing algorithms with a GPU, while training neural nets on GPU was a design element in implementing Algorithm 2.2 using deep

learning frameworks such as Torch (Paszke et al., 2017). If a user have access to both CPU and GPU, then we believe the wall clock time to some extent can be viewed as a natural metric of the time cost a user has to pay for the estimator. And the Precision per Second benchmark can be viewed as the cost-performance ratio of these methods. This measure is not a rigorous metric for comparing computation costs, but we believe it is at least an intuitive one for the users to get a rough idea of the time cost and efficiency of these algorithms.

From the numerical examples we see f-GB scaled better with dimension than its competitors. For example, from Example 1 we see that even though Warp-III can be $30 \sim 40$ faster to compute than our proposed method given the same amount of samples from q_1, q_2, its accuracy (measured in $MSE(\log \hat{r})$) is orders of magnitude greater (worse) than our approach. When $p = 48$, we find that Warp-III is not able to return a sensible estimate of r even with 25 times more samples from q_1, q_2 than f-GB. In Example 2 we also find that Warp-III requires around 18 times more samples to achieve a similar level of accuracy as f-GB, and takes around 3 times longer to run. Therefore we believe the extra computational cost of our f-GB estimator is "well-spent" as the numerical examples show that our Algorithm 2.2 is able to return an estimate of r with much higher precision than GBS, Warp-III and Warp-U and scales better with the dimension of the distributions.

As we acknowledged, if q_1, q_2 are simple structured and low dimensional, then users can get adequate precision more quickly using Warp-III or Warp-U. On the other hand, in speaking to users who report Bayes factors in applied Bayesian work, the overwhelming requirement was that the estimate be reliable, as the Bayes Factor value is sometimes the crux. In all the numerical examples we considered, f-GB never broke and produced accurate estimates. Warp III and GBS did break on larger problems. Warp-U took a similar amount of time to run compared with f-GB, but was less accurate. Therefore users may still prefer a method which is "over-powered" but more reliable.

2.7.2 Limitations and future works

One limitation of the f-GB estimator is the computational cost. In this paper we parameterize $q_1^{(\phi)}$ as a Normalizing flow. A possible direction of future work is to explore different choices of parameterizations of $q_1^{(\phi)}$. We expect that we can speed up our Algorithm 2.2 by replacing a Normalizing flow with simpler transformations such as Warp-I and Warp-II transformation (Meng and Schilling, 2002) at the expense of flexibility. Another limitation is that Algorithm 2.1 is only optimal when samples from

q_1, q_2 are i.i.d. Recall that $RE^2(\hat{r}_{opt})$ in (2.10) is derived based on the i.i.d. assumption. Therefore if the samples from q_1, q_2 are correlated, then Proposition 2.4.1 no longer holds, and minimizing $H_{s_2}(q_1^{(\phi)}, q_2)$ with respect to $q_1^{(\phi)}$ is no longer equivalent to minimizing the first order approximation of $RE^2(\hat{r}_{opt}^{(\phi)})$. Therefore it is of interest to see if it is possible to give an algorithm that minimizes the first order approximation of $RE^2(\hat{r}_{opt}^{(\phi)})$ when the samples are correlated. In addition, our approach only focuses on estimating the ratio of normalizing constants between two densities. When we have multiple unnormalized densities and would like to estimate the ratios between their normalizing constants, our approach needs to estimate these quantities separately in a pairwise fashion, which can be inefficient. Meng and Schilling (1996) and (Geyer, 1994) show that one can estimate multiple normalizing constants simultaneously up to a common multiplicative constant. We are also interested in extending our improvement strategy to this multiple densities setup.

2.8 Appendix of Chapter 2

2.8.1 Proofs

Here we give proof of Proposition 2.3.1, 2.3.2 and 2.4.1.

Proposition 2.3.1 (Estimating $H_\pi(q_1, q_2)$). *Let q_1, q_2 be continuous densities with respect to a base measure μ on the common support Ω. Let $\{\omega_{ij}\}_{j=1}^{n_i}$ be samples from q_i for $i = 1, 2$. Let $\pi \in (0, 1)$ be the weight parameter. Let r be the true ratio of normalizing constants between q_1, q_2, and $C_2 > C_1 > 0$ be constants such that $r \in [C_1, C_2]$. For $\tilde{r} \in [C_1, C_2]$, define*

$$G(\tilde{r}; \pi) = 1 - \frac{1}{\pi} E_{q_1}\left(\frac{\pi \tilde{q}_2(\omega)\tilde{r}}{(1-\pi)\tilde{q}_1(\omega) + \pi\tilde{q}_2(\omega)\tilde{r}}\right)^2 - \frac{1}{1-\pi} E_{q_2}\left(\frac{(1-\pi)\tilde{q}_1(\omega)}{(1-\pi)\tilde{q}_1(\omega) + \pi\tilde{q}_2(\omega)\tilde{r}}\right)^2. \tag{2.28}$$

Then $H_\pi(q_1, q_2)$ satisfies

$$H_\pi(q_1, q_2) \geq \sup_{\tilde{r} \in [C_1, C_2]} G(\tilde{r}; \pi), \tag{2.29}$$

and equality holds if and only if $\tilde{r} = r$. In addition, let $\hat{G}(\tilde{r}; \pi, \{\omega_{ij}\}_{j=1}^{n_i})$ be an empirical estimate of $G(\tilde{r}; \pi)$ based on $\{\omega_{ij}\}_{j=1}^{n_i} \sim q_i$ for $i = 1, 2$. If $\hat{r}_\pi = \arg\max_{\tilde{r} \in [C_1, C_2]} \hat{G}(\tilde{r}; \pi, \{\omega_{ij}\}_{j=1}^{n_i})$, then $\hat{r}_\pi$ is a consistent estimator of r, and $\hat{G}(\hat{r}_\pi; \pi, \{\omega_{ij}\}_{j=1}^{n_i})$ is a consistent estimator of $H_\pi(q_1, q_2)$ as $n_1, n_2 \to \infty$.

Proof. By definition, we know $0 \leq H_\pi(q_1, q_2) \leq 1$. And by setting $D_f(q_1, q_2) = H_\pi(q_1, q_2)$, $f(u) = 1 - \frac{u}{\pi+(1-\pi)u}$ and variational function $V_{\tilde{r}}(\omega) = f'\left(\frac{\tilde{q}_1(\omega)}{\tilde{q}_2(\omega)\tilde{r}}\right)$ with $\mathcal{V} = \{V_{\tilde{r}}(\omega)|\tilde{r} \in [C_1, C_2]\}$, we see $G(\tilde{r}; \pi)$ exists for all $\tilde{r} \in [C_1, C_2]$ and is the variational lower bound of $H_\pi(q_1, q_2)$ in the form of (2.12). Then by Nguyen et al. (2010), equality holds if and only if $V_{\tilde{r}}(\omega) = f'\left(\frac{\tilde{q}_1(\omega)}{\tilde{q}_2(\omega)r}\right)$. Since $f(u)$ is strictly convex, $f'(u)$ is monotonically increasing. By assumption, we also know $q_1(\omega), q_2(\omega) > 0$ for all $\omega \in \Omega$. Therefore by applying the inverse of f' to both side, we see $V_{\tilde{r}}(\omega) = f'\left(\frac{\tilde{q}_1(\omega)}{\tilde{q}_2(\omega)r}\right)$ if and only if $\tilde{r} = r$. Therefore $G(r; \pi) = H_\pi(q_1, q_2)$, and $\tilde{r} = r$ is the unique maximizer of $G(\tilde{r}; \pi)$.

Now we show the consistency of $\hat{r}_\pi$. It can be shown in a similar fashion to the proof of the consistency of an extremum estimator in e.g. Newey and McFadden (1994) Theorem 2.1.

We first check $\hat{G}(\tilde{r}; \pi, \{\omega_{ij}\}_{j=1}^{n_i})$ satisfies the uniform law of large number (ULLN). Let

$$g_1(\omega, \tilde{r}) = \frac{1}{\pi}\left(\frac{\pi\tilde{q}_2(\omega)\tilde{r}}{(1-\pi)\tilde{q}_1(\omega) + \pi\tilde{q}_2(\omega)\tilde{r}}\right)^2$$

and

$$g_2(\omega, \tilde{r}) = \frac{1}{1-\pi}\left(\frac{(1-\pi)\tilde{q}_1(\omega)}{(1-\pi)\tilde{q}_1(\omega) + \pi\tilde{q}_2(\omega)\tilde{r}}\right)^2.$$

Since $0 < g_1(\omega, \tilde{r}), g_2(\omega, \tilde{r}) < \max(\frac{1}{\pi}, \frac{1}{1-\pi})$ for any $\omega \in \Omega$ and $\tilde{r} \in [C_1, C_2]$, by Jennrich (1969) Theorem 2, we have

$$\sup_{\tilde{r}\in[C_1,C_2]}\left|\frac{1}{n_1}\sum_{j=1}^{n_1} g_1(\omega_{1j}) - E_{q_1}g_1(\omega, \tilde{r})\right| \to_p 0$$

and

$$\sup_{\tilde{r}\in[C_1,C_2]}\left|\frac{1}{n_2}\sum_{j=1}^{n_2} g_2(\omega_{2j}) - E_{q_2}g_2(\omega, \tilde{r})\right| \to_p 0$$

as $n_1, n_2 \to \infty$. Since $G(\tilde{r}; \pi) = 1 - E_{q_1}g_1(\omega, \tilde{r}) - E_{q_2}g_2(\omega, \tilde{r})$, by triangle inequality, we have

$$\sup_{\tilde{r}\in[C_1,C_2]}\left|\hat{G}(\tilde{r}; \pi, \{\omega_{ij}\}_{j=1}^{n_i}) - G(\tilde{r}; \pi)\right| \to_p 0$$

as $n_1, n_2 \to \infty$. Hence $\hat{G}(\tilde{r}; \pi, \{\omega_{ij}\}_{j=1}^{n_i})$ satisfies the uniform law of large number (ULLN).

We also need to check $G(\hat{r}_\pi;\pi) \to_p G(r;\pi)$:

$$G(r;\pi) \geq G(\hat{r}_\pi;\pi) \quad \text{since } r \text{ is the unique maximizer of } G(\tilde{r};\pi) \tag{2.30}$$

$$= \hat{G}(\hat{r}_\pi;\pi,\{\omega_{ij}\}_{j=1}^{n_i}) + (G(\hat{r}_\pi;\pi) - \hat{G}(\hat{r}_\pi;\pi,\{\omega_{ij}\}_{j=1}^{n_i})) \tag{2.31}$$

$$\geq \hat{G}(r;\pi,\{\omega_{ij}\}_{j=1}^{n_i}) + (G(\hat{r}_\pi;\pi) - \hat{G}(\hat{r}_\pi;\pi,\{\omega_{ij}\}_{j=1}^{n_i})) \tag{2.32}$$

$$= G(r;\pi) + (\hat{G}(r;\pi,\{\omega_{ij}\}_{j=1}^{n_i}) - G(r;\pi)) + (G(\hat{r}_\pi;\pi) - \hat{G}(\hat{r}_\pi;\pi,\{\omega_{ij}\}_{j=1}^{n_i})) \tag{2.33}$$

Since the last two terms converge in probability to 0 by ULLN, we have $G(r;\pi) \geq G(\hat{r}_\pi;\pi) \geq G(r;\pi) + o_p(1)$. This implies $G(\hat{r}_\pi;\pi) \to_p G(r;\pi)$.

Since $[C_1, C_2]$ is compact and $G(\tilde{r};\pi)$ is continuous, for every open interval $A \subset [C_1, C_2]$ containing r, we have $\sup_{\tilde{r}\notin A} G(\tilde{r};\pi) < G(r;\pi)$. On the other hand, $G(\hat{r}_\pi;\pi) \to_p G(r;\pi)$ implies that $Pr(G(\hat{r}_\pi;\pi) > \sup_{\tilde{r}\notin A} G(\tilde{r};\pi))$ converges to 1. Therefore $Pr(\hat{r}_\pi \in A)$ also converges to 1, i.e. $\hat{r}_\pi$ is a consistent estimator of r.

Finally we show $\hat{G}(\hat{r}_\pi;\pi,\{\omega_{ij}\}_{j=1}^{n_i})$ is a consistent estimator of $H_\pi(q_1,q_2)$. Recall that $G(r;\pi) = H_\pi(q_1,q_2)$. By triangle inequality,

$$\left|\hat{G}(\hat{r}_\pi;\pi,\{\omega_{ij}\}_{j=1}^{n_i}) - H_\pi(q_1,q_2)\right| \leq \left|\hat{G}(\hat{r}_\pi;\pi,\{\omega_{ij}\}_{j=1}^{n_i}) - G(\hat{r}_\pi;\pi)\right| + |G(\hat{r}_\pi;\pi) - G(r;\pi)| \tag{2.34}$$

The first term on the RHS converges to 0 in probability by ULLN. The second term on the RHS converges to 0 in probability by continuous mapping theorem and the fact that $\hat{r}_\pi$ is a consistent estimator of r. Hence $\hat{G}(\hat{r}_\pi;\pi,\{\omega_{ij}\}_{j=1}^{n_i})$ is a consistent estimator of $H_\pi(q_1,q_2)$. □

Proposition 2.3.2 (Connection between $\hat{r}^{(f)}$ and Bridge sampling). *Suppose $f(u) : \mathbb{R}^+ \to \mathbb{R}$ is strictly convex, twice differentiable and satisfies $f(1) = 0$. Let $\{\omega_{ij}\}_{j=1}^{n_i}$ be samples from q_i for $i = 1, 2$. If $\hat{r}^{(f)} = \arg\max_{\tilde{r}\in\mathbb{R}^+} \hat{G}_f(\tilde{r};\{\omega_{ij}\}_{j=1}^{n_i})$ is a stationary point of $\hat{G}_f(\tilde{r};\{\omega_{ij}\}_{j=1}^{n_i})$ in (2.18), then $\hat{r}^{(f)}$ satisfies the following equation*

$$\hat{r}^{(f)} = \frac{\frac{1}{n_2}\sum_{j=1}^{n_2} f''\left(\frac{\tilde{q}_1(\omega_{2j})}{\tilde{q}_2(\omega_{2j})\hat{r}^{(f)}}\right)\frac{\tilde{q}_1(\omega_{2j})}{\tilde{q}_2(\omega_{2j})^2}\tilde{q}_1(\omega_{2j})}{\frac{1}{n_1}\sum_{j=1}^{n_1} f''\left(\frac{\tilde{q}_1(\omega_{1j})}{\tilde{q}_2(\omega_{1j})\hat{r}^{(f)}}\right)\frac{\tilde{q}_1(\omega_{1j})}{\tilde{q}_2(\omega_{1j})^2}\tilde{q}_2(\omega_{1j})}. \tag{2.35}$$

where f'' is the second order derivative of f.

Proof. Note that the objective function can be written as

$$\hat{G}_f(\tilde{r};\{\omega_{ij}\}_{j=1}^{n_i}) = \frac{1}{n_1}\sum_{j=1}^{n_1} f'\left(\frac{\tilde{q}_1(\omega_{1j})}{\tilde{q}_2(\omega_{1j})\tilde{r}}\right) - \frac{1}{n_2}\sum_{j=1}^{n_2} f^*\circ f'\left(\frac{\tilde{q}_1(\omega_{2j})}{\tilde{q}_2(\omega_{2j})\tilde{r}}\right) \tag{2.36}$$

$$= \frac{1}{n_1}\sum_{j=1}^{n_1} f'\left(\frac{\tilde{q}_1(\omega_{1j})}{\tilde{q}_2(\omega_{1j})\tilde{r}}\right) - \frac{1}{n_2}\sum_{j=1}^{n_2}\frac{\tilde{q}_1(\omega_{2j})}{\tilde{q}_2(\omega_{2j})\tilde{r}} f'\left(\frac{\tilde{q}_1(\omega_{2j})}{\tilde{q}_2(\omega_{2j})\tilde{r}}\right) - f\left(\frac{\tilde{q}_1(\omega_{2j})}{\tilde{q}_2(\omega_{2j})\tilde{r}}\right) \tag{2.37}$$

using the equation $f^* \circ f'(u) = uf'(u) - f(u)$ (Uehara et al., 2016). Let $S(\tilde{r}; \{\omega_{ij}\}_{j=1}^{n_i}) = \frac{d}{d\tilde{r}} \hat{G}_f(\tilde{r}; \{\omega_{ij}\}_{j=1}^{n_i})$. If $\hat{r}^{(f)}$ is the stationary point, then it satisfies the "score" equation

$$0 = S(\hat{r}^{(f)}; \{\omega_{ij}\}_{j=1}^{n_i}) \tag{2.38}$$

$$= -\frac{1}{n_1} \sum_{j=1}^{n_1} f'' \left(\frac{\tilde{q}_1(\omega_{1j})}{\tilde{q}_2(\omega_{1j}) \hat{r}^{(f)}} \right) \frac{\tilde{q}_1(\omega_{1j})}{\tilde{q}_2(\omega_{1j}) (\hat{r}^{(f)})^2} + \frac{1}{n_2} \sum_{j=1}^{n_2} f'' \left(\frac{\tilde{q}_1(\omega_{2j})}{\tilde{q}_2(\omega_{2j}) \hat{r}^{(f)}} \right) \frac{\tilde{q}_1(\omega_{2j})^2}{\tilde{q}_2(\omega_{2j})^2 (\hat{r}^{(f)})^3} \tag{2.39}$$

The above equation can be rearranged as

$$\hat{r}^{(f)} = \frac{\frac{1}{n_2} \sum_{j=1}^{n_2} f'' \left(\frac{\tilde{q}_1(\omega_{2j})}{\tilde{q}_2(\omega_{2j}) \hat{r}^{(f)}} \right) \frac{\tilde{q}_1(\omega_{2j})}{\tilde{q}_2(\omega_{2j})^2} \tilde{q}_1(\omega_{2j})}{\frac{1}{n_1} \sum_{j=1}^{n_1} f'' \left(\frac{\tilde{q}_1(\omega_{1j})}{\tilde{q}_2(\omega_{1j}) \hat{r}^{(f)}} \right) \frac{\tilde{q}_1(\omega_{1j})}{\tilde{q}_2(\omega_{1j})^2} \tilde{q}_2(\omega_{1j})} \tag{2.40}$$

□

Proposition 2.4.1 (Minimizing $RE^2(\hat{r}_{opt}^{(\phi)})$ using Algorithm 2.1). *If $(\phi^*, \tilde{r}^*)$ is the solution of $\min_{\phi \in \mathbb{R}^l} \max_{\tilde{r} \in \mathbb{R}^+} G(\phi, \tilde{r}; s_2)$ defined in Algorithm 2.1, then $G(\phi, \tilde{r}^*; s_2) = H_{s_2}(q_1^{(\phi)}, q_2)$ for all $\phi \in \mathbb{R}^l$, T_{ϕ^*} minimizes $H_{s_2}(q_1^{(\phi)}, q_2)$ with respect to $T_\phi \in \mathcal{T}$. If the samples $\{\omega_{ij}\}_{j=1}^{n_i} \overset{i.i.d.}{\sim} q_i$ for $i = 1, 2$, then T_{ϕ^*} also minimizes $RE^2(\hat{r}_{opt}^{(\phi)})$ with respect to $T_\phi \in \mathcal{T}$ up to the first order.*

Proof. For every $\phi \in \mathbb{R}^l$, $G(\phi, \tilde{r}; s_2)$ is the variational lower bound of $H_{s_2}(q_1^{(\phi)}, q_2)$ in the form of (2.12). By Proposition 2.3.1, we know $G(\phi, \tilde{r}; s_2)$ is uniquely maximized at $\tilde{r} = r$ w.r.t $\tilde{r} > 0$, and $G(\phi, r; s_2) = H_{s_2}(q_1^{(\phi)}, q_2)$. Since $(\phi^*, \tilde{r}^*)$ is the solution of $\min_{\phi \in \mathbb{R}^l} \max_{\tilde{r} \in \mathbb{R}^+} G(\phi, \tilde{r}; s_2)$, it is straightforward to verify that $\tilde{r}^* = r$, and $H_{s_2}(q_1^{(\phi^*)}, q_2) = G(\phi^*, \tilde{r}^*; s_2) \leq G(\phi, \tilde{r}^*; s_2)$ for any $\phi \in \mathbb{R}^l$. Hence T_{ϕ^*} minimizes $H_{s_2}(q_1^{(\phi)}, q_2)$ with respect to $T_\phi \in \mathcal{T}$.

Since the leading term of $RE^2(\hat{r}_{opt}^{(\phi)})$ in (2.10) is a monotonically increasing function of $H_{s_2}(q_1^{(\phi)}, q_2)$, T_{ϕ^*} minimizes $H_{s_2}(q_1^{(\phi)}, q_2)$ w.r.t. $T_\phi \in \mathcal{T}$ implies T_{ϕ^*} minimizes the leading term of $RE^2(\hat{r}_{opt}^{(\phi)})$ w.r.t. $T_\phi \in \mathcal{T}$ under the assumption that samples $\{\omega_{ij}\}_{j=1}^{n_i} \sim q_i$ are i.i.d. for $i = 1, 2$. □

2.8.2 Dimension matching

The standard Bridge estimator (2.1) can not be applied directly when Ω_1,Ω_2 have different dimensions. This is a common and important case. For example, if we would like to compare two models M_1, M_2 by estimating the Bayes factor between them, the standard Bridge estimator (2.1) is not directly applicable when M_1, M_2 are controlled by parameters that live in different dimensions.

Assume $\Omega_1 = \mathbb{R}^{d_1}$, $\Omega_2 = \mathbb{R}^{d_2}$ and $d_1 < d_2$. Discrete cases work similarly. Chen and Shao (1997) resolve the problem of unequal dimensions by first augmenting the lower dimensional density $q_1(\omega_1)$ by some completely known, normalized density $p(\theta|\omega_1)$ where $\theta \in \mathbb{R}^{d_2-d_1}$. This ensures the augmented density

$$q_1^*(\omega_1, \theta) = \tilde{q}_1^*(\omega_1, \theta)/Z_1 \tag{2.41}$$

$$= \tilde{q}_1(\omega_1)p(\theta|\omega_1)/Z_1 \tag{2.42}$$

matches the dimension of the q_2, where $\tilde{q}_1^*(\omega_1, \theta)$ is the unnormalized augmented density. Let Ω_1^* be the augmented support of q_1^*. Since the augmented density $q_1^*(\omega_1, \theta)$ and the original $q_1(\omega_1)$ have the same normalizing constant, we can then treat $r = Z_1/Z_2$ as the ratio between the normalizing constants of $q_1^*(\omega_1, \theta)$ and $q_2(\omega_2)$, and form an "augmented" Bridge estimator $\hat{r}_\alpha^*$ based on the augmented densities. Chen and Shao (1997) also show that when the free function $\alpha(\omega) = \alpha_{opt}(\omega)$, the optimal augmenting density $p_{opt}(\theta|\omega_1)$ which attains the minimal $RE^2(\hat{r}^*_{\alpha_{opt}})$ is

$$p_{opt}(\theta|\omega_1) = q_2(\theta|\omega_1) \tag{2.43}$$

i.e. $p_{opt}(\theta|\omega_1)$ is the conditional distribution of the remaining $d_2 - d_1$ entries of $\omega_2 \sim q_2$ given that the first d_1 entries are ω_1. However, $q_2(\theta|\omega_1)$ is difficult to evaluate or sample from in general. One way to approximate the optimal augmenting distribution $q_2(\theta|\omega_1)$ is to incorporate the augmented density $q_1^*(\omega_1, \theta)$ with a Normalizing flow (see Chapter 1.2.1). Assume we start with an arbitrary augmenting density $p(\theta|\omega_1)$, e.g. standard Normal $N(0, I_{d_2-d_1})$. Consider a Normalizing flow with base density q_1^* and a smooth and invertible transformation $T_1^* : \Omega_1^* \to \Omega_2$ that aims to map the augmented q_1^* to the target q_2. Let $(\omega_1^{(T)}, \theta^{(T)}) = T_1^*(\omega_1, \theta)$. If $q_1^{*(T)}(\omega_1^{(T)}, \theta^{(T)}) \approx q_2(\omega_1^{(T)}, \theta^{(T)})$ for all $(\omega_1^{(T)}, \theta^{(T)}) \in \Omega_2$, i.e. $q_1^{*(T)}$ is a good approximation to q_2, then for the transformed augmenting density, we expect $q_1^{*(T)}(\theta^{(T)}|\omega_1^{(T)}) \approx q_2(\theta^{(T)}|\omega_1^{(T)})$ as well. This means the transformed $q_1^{*(T)}$ automatically learns the optimal augmenting density.

2.8.3 Bias in the estimator of $H_\pi(q_1, q_2)$ given in Proposition 2.3.1

In Proposition 2.3.1, the estimator of $H_\pi(q_1, q_2)$ is given in the form of the maximum of the function $\hat{G}(\tilde{r}; \pi, \{\omega_{ij}\}_{j=1}^{n_i})$ w.r.t. $\tilde{r}$. Let $r = Z_1/Z_2$ be the true ratio of normalizing constants. Even though $\hat{G}(r; \pi, \{\omega_{ij}\}_{j=1}^{n_i})$ is an unbiased estimator of $H_\pi(q_1, q_2)$, our proposed estimator $\hat{G}(\hat{r}_\pi; \pi, \{\omega_{ij}\}_{j=1}^{n_i})$ sufferes from a positive bias. Intuitively speaking, this bias is analogous to the fact that the training error of a model is an underestimate

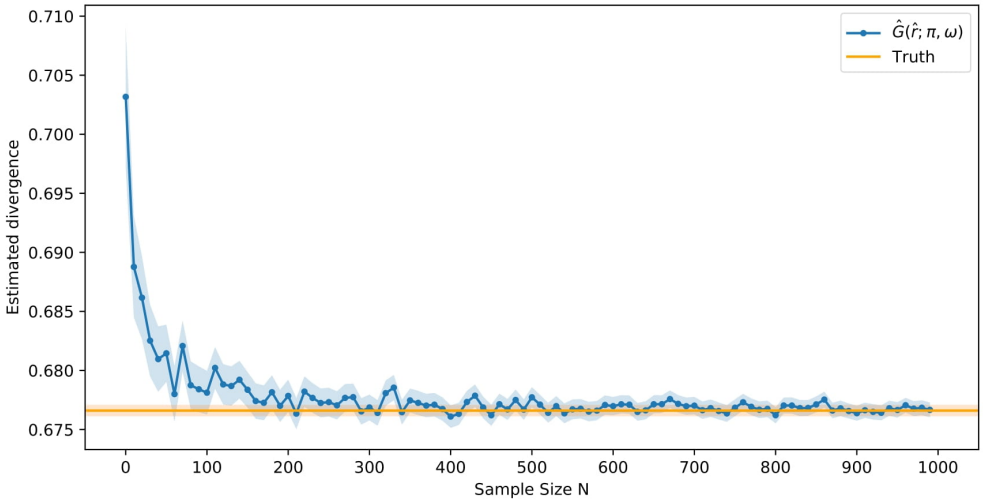


Figure 2.7: Sample mean of the estimated $H_\pi(q_1, q_2)$ for each sample size N. The blue band represents the 2σ error bars of the sample means. Orange line represents a high precision unbiased MC estimator of $H_\pi(q_1, q_2)$. Orange band represents the 2σ error bar of the MC estimate.

of the true error. We use a toy example to illustrate this bias. Let $x \in \mathbb{R}^3$, $\sigma_1 = 1$ and $\sigma_2 = 3$. Let

$$\tilde{q}_1 = \exp\left(-\frac{\|x\|_2^2}{2\sigma_1^2}\right) \tag{2.44}$$

$$\tilde{q}_2 = \exp\left(-\frac{\|x\|_2^2}{2\sigma_2^2}\right). \tag{2.45}$$

In other words, $\tilde{q}_1, \tilde{q}_2$ are the unnormalized pdf of two Gaussian distributions with zero mean and covaraince $\sigma_1 I_3, \sigma_2 I_3$ respectively, where I_p is the $p \times p$ identity matrix. Let q_1, q_2 be the corresponding normalized densities. Let $\pi = 0.5$. It is straightforward to form an unbiased MC estimate of $H_\pi(q_1, q_2)$ using (2.9). Let $N = \{10, 20, 30, ..., 1000\}$. For each value of N, we repeatedly compute the proposed estimator $\hat{G}(\hat{r}_\pi; \pi, \{\omega_{ij}\}_{j=1}^{n_i})$ 1000 times based on $n_1 = n_2 = N$ i.i.d. samples from q_1, q_2 respectively. We then report the sample mean of the repeated estimates for each N, and compare it with a high precision unbiased MC estimator of $H_\pi(q_1, q_2)$. From Figure 2.7 we see $\hat{G}(\hat{r}_\pi; \pi, \{\omega_{ij}\}_{j=1}^{n_i})$ does exhibit a positive bias when $N < 500$, and this bias gradually vanishes as sample size increases.

Even though we have not found a practical strategy to correct this bias, we believe this bias does not prevent our proposed estimator from being useful in practice. Since our estimator of $RE^2(\hat{r}_{opt})$ in (2.16) is a monotonically increasing function of

$\hat{G}(\hat{r}_\pi; \pi, \{\omega_{ij}\}_{j=1}^{n_i})$, the positive bias in $\hat{G}(\hat{r}_\pi; \pi, \{\omega_{ij}\}_{j=1}^{n_i})$ leads to a positive bias in $\widehat{RE}^2(\hat{r}_{opt})$. Therefore $\widehat{RE}^2(\hat{r}_{opt})$ will systemically overestimate the true error $RE^2(\hat{r}_{opt})$, which will lead to more conservative conclusions (e.g. wider error bars). This is certainly not ideal, but we believe in practice, it is less harmful than underestimating the variability in $\hat{r}_{opt}$. In addition, we see that our proposed error estimator provides accurate estimates of the MSE of $\log \hat{r}_{opt}^{\prime(\phi^t)}$ in both examples in Chapter 2.5 and 2.6, indicating the effectiveness of it.

2.8.4 f-divergence and Bridge estimators

Here we give some examples of Proposition 2.3.2. We demonstrate how the Bridge estimators with different choices of free function $\alpha(\omega)$ arise from estimating different f-divergences.

Example 2.8.1 (KL divergence and the Importance sampling estimator)
KL divergence

$$KL(q_1, q_2) = \int_\Omega \log\left(\frac{q_1(\omega)}{q_2(\omega)}\right) q_1(\omega) d\mu(\omega) \tag{2.46}$$

is an f-divergence with $f(u) = u \log u$, $f'(u) = 1 + \log u$ and $f^*(t) = \exp(t-1)$. This specification corresponds to $V_{\tilde{r}}(\omega) = 1 + \log \frac{\tilde{q}_1(x)}{\tilde{q}_2(x)\tilde{r}}$. Suppose we have $\{\omega_{1j}\}_{j=1}^{n_1} \sim q_1$ and $\{\omega_{2j}\}_{j=1}^{n_2} \sim q_2$. The maximizer $\hat{r}_{KL}$ of equation (2.17) under this specification is

$$\hat{r}_{KL} = \arg\max_{\tilde{r}\in\mathbb{R}^+} \frac{1}{n_1}\sum_{j=1}^{n_1}\left(1 + \log\frac{\tilde{q}_1(\omega_{1j})}{\tilde{q}_2(\omega_{1j})\tilde{r}}\right) - \frac{1}{n_2}\sum_{j=1}^{n_2}\frac{\tilde{q}_1(\omega_{2j})}{\tilde{q}_2(\omega_{2j})\tilde{r}} \tag{2.47}$$

$$= \frac{1}{n_2}\sum_{j=1}^{n_2}\frac{\tilde{q}_1(\omega_{2j})}{\tilde{q}_2(\omega_{2j})} \tag{2.48}$$

Note that this is the Importance sampling estimator of r using q_2 as the proposal, which is a special case of a Bridge estimator with free function $\alpha(\omega) = \tilde{q}_2(\omega)^{-1}$. Therefore we recover the Importance sampling estimator from the problem of estimating $KL(q_1, q_2)$. It is also straightforward to verify that estimating $KL(q_2, q_1)$ leads to $\hat{r}_{KL} = \left(\frac{1}{n_1}\sum_{j=1}^{n_1}\frac{\tilde{q}_2(\omega_{1j})}{\tilde{q}_1(\omega_{1j})}\right)^{-1}$, the Reciprocal Importance sampling estimator of r based on a similar argument.

Example 2.8.2 (Weighted Jensen-Shannon divergence and the optimal Bridge estimator)

Weighted Jensen-Shannon divergence is defined as

$$JS_\pi(q_1, q_2) = \pi KL(q_1, q_\pi) + (1-\pi) KL(q_2, q_\pi) \tag{2.49}$$

where $\pi \in (0,1)$ is the weight parameter and $q_\pi = \pi q_1 + (1-\pi)q_2$ is a mixture of q_1 and q_2. Weighted Jensen-Shannon divergence is an f-divergence with $f(u) = \pi u \log u - (1-\pi+\pi u)\log(1-\pi+\pi u)$, $f'(u) = \pi \log \frac{u}{1-\pi+\pi u}$ and $f^*(t) = (1-\pi)\log \frac{1-\pi}{1-\pi\exp(t/\pi)}$. This corresponds to $V_{\tilde{r}}(\omega) = \pi \log \frac{\tilde{q}_1(\omega)}{\pi\tilde{q}_1(\omega)+(1-\pi)\tilde{q}_2(\omega)\tilde{r}}$. Suppose we have $\{\omega_{1j}\}_{j=1}^{n_1} \sim q_1$ and $\{\omega_{2j}\}_{j=1}^{n_2} \sim q_2$. Let the weight $\pi = \frac{n_1}{n_1+n_2} = s_1$, then under this specification, the maximizer $\hat{r}_{JS}$ of Equation (2.17) is defined as

$$\hat{r}_{JS} = \arg\max_{\tilde{r}\in\mathbb{R}^+} \frac{\pi}{n_1}\sum_{j=1}^{n_1} \log \frac{\tilde{q}_1(\omega_{1j})}{\pi\tilde{q}_1(\omega_{1j}) + (1-\pi)\tilde{q}_2(\omega_{1j})\tilde{r}} + \frac{1-\pi}{n_2}\sum_{j=1}^{n_2} \log \frac{\tilde{q}_2(\omega_{2j})\tilde{r}}{\pi\tilde{q}_1(\omega_{2j}) + (1-\pi)\tilde{q}_2(\omega_{2j})\tilde{r}} \quad (2.50)$$

It is straightforward to verify that $\hat{r}_{JS}$ satisfies

$$\hat{r}_{JS} = \frac{\frac{1}{n_2}\sum_{j=1}^{n_2} \frac{\pi\tilde{q}_1(\omega_{2j})}{\pi\tilde{q}_1(\omega_{2j})+(1-\pi)\tilde{q}_2(\omega_{2j})\hat{r}_{JS}}}{\frac{1}{n_1}\sum_{j=1}^{n_1} \frac{(1-\pi)\tilde{q}_2(\omega_{1j})}{\pi\tilde{q}_1(\omega_{1j})+(1-\pi)\tilde{q}_2(\omega_{1j})\hat{r}_{JS}}} \quad (2.51)$$

On the other hand, recall that the asymptotically optimal Bridge estimator $\hat{r}_{opt}$ must be a fixed point of the iterative procedure (2.4). Therefore $\hat{r}_{opt}$ satisfies the following "score equation" (Meng and Wong, 1996)

$$S(\hat{r}_{opt}) = -\sum_{j=1}^{n_1} \frac{s_2\tilde{q}_2(\omega_{1j})\hat{r}_{opt}}{s_1\tilde{q}_1(\omega_{1j}) + s_2\tilde{q}_2(\omega_{1j})\hat{r}_{opt}} + \sum_{j=1}^{n_2} \frac{s_1\tilde{q}_1(\omega_{2j})}{s_1\tilde{q}_1(\omega_{2j}) + s_2\tilde{q}_2(\omega_{2j})\hat{r}_{opt}} \quad (2.52)$$

$$= 0 \quad (2.53)$$

When $\pi = s_1$, Equation (2.51) is precisely the score equation (2.52) of $\hat{r}_{opt}$. This implies $\hat{r}_{JS} = \hat{r}_{opt}$ because the root of the score function $S(r)$ in (2.52) is unique (Meng and Wong, 1996). Therefore $\hat{r}_{JS}$ is equivalent to the asymptotically optimal Bridge estimator $\hat{r}_{opt}$, and we recover $\hat{r}_{opt}$ from the problem of estimating the weighted Jensen-Shannon divergence between q_1, q_2.

Example 2.8.3 (Squared Hellinger distance and the geometric Bridge estimator) Squared Hellinger distance

$$H^2(q_1,q_2) = \int_\Omega \left(\sqrt{q_1(\omega)} - \sqrt{q_2(\omega)}\right)^2 d\mu(\omega) \quad (2.54)$$

is an f-divergence with $f(u) = (\sqrt{u}-1)^2$, $f'(u) = 1 - u^{-\frac{1}{2}}$ and $f^*(t) = \frac{t}{1-t}$. This specification corresponds to $V_{\tilde{r}}(\omega) = 1 - \sqrt{\frac{\tilde{q}_2(\omega)\tilde{r}}{\tilde{q}_1(\omega)}}$. Again suppose we have $\{\omega_{1j}\}_{j=1}^{n_1} \sim q_1$ and $\{\omega_{2j}\}_{j=1}^{n_2} \sim q_2$. The maximizer $\hat{r}_{H^2}$ of equation (2.17) under this specification is

$$\hat{r}_{H^2} = \arg\max_{\tilde{r}\in\mathbb{R}^+} \frac{1}{n_1}\sum_{j=1}^{n_1}\left(1-\sqrt{\frac{\tilde{q}_2(\omega_{1j})\tilde{r}}{\tilde{q}_1(\omega_{1j})}}\right) - \frac{1}{n_2}\sum_{j=1}^{n_2}\left(\sqrt{\frac{\tilde{q}_1(\omega_{2j})}{\tilde{q}_2(\omega_{2j})\tilde{r}}}-1\right) \tag{2.55}$$

$$= \frac{\frac{1}{n_2}\sum_{j=1}^{n_2}\sqrt{\tilde{q}_1(\omega_{2j})/\tilde{q}_2(\omega_{2j})}}{\frac{1}{n_1}\sum_{j=1}^{n_1}\sqrt{\tilde{q}_2(\omega_{1j})/\tilde{q}_1(\omega_{1j})}} \tag{2.56}$$

This is precisely the geometric Bridge estimator $\hat{r}_{geo}$ in Meng and Wong (1996) with free function $\alpha(\omega) = (\tilde{q}_1(\omega)\tilde{q}_2(\omega))^{-\frac{1}{2}}$.

In addition to the fact that Bridge estimators with different choices of free function $\alpha(\omega)$ can arise from estimating different f-divergences, the asymptotic RMSE of $\hat{r}_{KL}, \hat{r}_{opt}$ and $\hat{r}_{geo}$ can also be written as functions of some f-divergences between the two distributions. For example, Meng and Wong (1996) show that $RE^2(\hat{r}_{geo})$ is a function of the Hellinger distance between q_1, q_2, Wang et al. (2022) show that $RE^2(\hat{r}_{opt})$ is a function of $H_\pi(q_1, q_2)$ in (2.10). It is also straightforward to show $RE^2(\hat{r}_{KL})$ is a function of the Rényi's 2-divergence between q_1, q_2 using the formula of $RE^2(\hat{r}_\alpha)$ given by (3.2) in Meng and Wong (1996). However, the general connection between $RE^2(\hat{r}_\alpha)$ and the f-divergence between the two distributions is not obvious. For example, suppose we choose the constant free function $\alpha(\omega) = 1$ discussed in Meng and Wong (1996). Then we can work out the asymptotic RMSE of the corresponding Bridge estimator $\hat{r}_1$ using the formula of $RE^2(\hat{r}_\alpha)$ in Meng and Wong (1996). Suppose q_1, q_2 are defined on a common support Ω, the resulting $RE^2(\hat{r}_1)$ takes the form

$$RE^2(\hat{r}_1) = (s_1 s_2 n)^{-1}\frac{\int_\Omega q_1(\omega)q_2(\omega)(s_1 q_1(\omega) + s_2 q_2(\omega))d\omega}{\left(\int_\Omega q_1(\omega)q_2(\omega)d\omega\right)^2} + o\left(\frac{1}{n}\right) \tag{2.57}$$

It is not obvious how this expression can be rearranged into a function of some f-divergence between q_1, q_2, as the leading term of $RE^2(\hat{r}_1)$ is in the form of ratio of integrals, which is different from the general functional form of an f-divergence. This example suggests that there may not be a general connection between the f-divergence between two distributions and the RMSE of a Bridge estimator apart from common Bridge estimators such as the optimal Bridge estimator and the geometric Bridge estimator. We have also tried the other direction. We started from an f-divergence. By Proposition 2.3.2, estimating the f-divergence leads to a Bridge estimator with a specific free function in the form of $\alpha_f(\omega) = f''\left(\frac{\tilde{q}_1(\omega)}{\tilde{q}_2(\omega)\hat{r}^{(f)}}\right)\frac{\tilde{q}_1(\omega)}{\tilde{q}_2(\omega)^2}$. We then substitute this $\alpha_f(\omega)$ into the formula of $RE^2(\hat{r}_\alpha)$ in Meng and Wong (1996). The functional form of the resulting expression is still also very different from the functional form of an f-divergence in the general case, and it is not obvious to see how it can be

rearranged into a function of some f-divergence between the two distributions. This also suggests that there may not be a general connection between $RE^2(\hat{r}_\alpha)$ and the f-divergence between two distributions.

2.8.5 Other choices of f-divergence

The weighted Harmonic divergence $H_\pi(q_1^{(\phi)}, q_2)$ is not the only choice of f divergence to minimize if our goal is to increase the overlap between $q_1^{(\phi)}$ and q_2. Recall that in Algorithm 2.2 we parameterize $q_1^{(\phi)}$ as a Normalizing flow. Since both $\tilde{q}_1, \tilde{q}_2$ are available, it is also possible to estimate $q_1^{(\phi)}$ by variational inference approaches without using the f-GAN framework. For example, one may approximate q_2 using $q_1^{(\phi)}$ by asymptotically minimizing $KL(q_1^{(\phi)}, q_2)$ or $KL(q_2, q_1^{(\phi)})$. In addition to the KL divergence, other common f-divergences such as the Squared Hellinger distance and the weighted Jensen-Shannon divergence are also sensible measures of overlap between densities, and we can minimize these divergences using the f-GAN framework in a similar fashion to Algorithm 2.1. However, f-divergences such as KL divergence, Squared Hellinger distance and the weighted Jensen-Shannon divergence are inefficient compared to the weighted Harmonic divergence $H_{s_2}(q_1^{(\phi)}, q_2)$ if our goal is to minimize $RE^2(\hat{r}_{opt}^{(\phi)})$. In Proposition 2.4.1 we have shown that under the i.i.d. assumption, minimizing $H_{s_2}(q_1^{(\phi)}, q_2)$ with respect to $q_1^{(\phi)}$ is equivalent to minimizing the first order approximation of $RE^2(\hat{r}_{opt}^{(\phi)})$ directly. On the other hand, Meng and Wong (1996) show that asymptotically,

$$RE^2(\hat{r}_{opt}) \leq (ns_1s_2)^{-1} \left(\left(1 - \frac{1}{2} H^2(q_1, q_2) \right)^{-2} - 1 \right) \tag{2.58}$$

up to the first order, where $n = n_1 + n_2$ and $s_i = n_i/n$ for $i = 1, 2$ under the same i.i.d. assumption. Note that $H^2(q_1^{(\phi)}, q_2) \to 0$ also implies $RE^2(\hat{r}_{opt}^{(\phi)}) \to 0$, but minimizing $H^2(q_1^{(\phi)}, q_2)$ with respect to the density $q_1^{(\phi)}$ can be viewed as minimizing an *upper bound* of the first order approximation of $RE^2(\hat{r}_{opt}^{(\phi)})$, which is less efficient. Here we show minimizing $JS_\pi(q_1^{(\phi)}, q_2)$, $KL(q_1^{(\phi)}, q_2)$ or $KL(q_2, q_1^{(\phi)})$ with respect to $q_1^{(\phi)}$ can also be viewed as minimizing some upper bounds of the first order approximation of $RE^2(\hat{r}_{opt}^{(\phi)})$.

Proposition 2.8.1 (Upper bounds of $RE^2(\hat{r}_{opt}^{(\phi)})$)**.** *Let q_1, q_2 be continuous densities with respect to a base measure μ on the common support Ω. If $\pi \in (0, 1)$ is the weight parameter, then $JS_\pi(q_1, q_2) \to 0$, $KL(q_1, q_2) \to 0$ or $KL(q_2, q_1) \to 0$ implies*

$RE^2(\hat{r}_{opt}) \to 0$, *and asymptotically,*

$$RE^2(\hat{r}_{opt}) \leq \frac{1}{s_1 s_2 n}\left(\left(1 - \min(1, \sqrt{JS_\pi(q_1, q_2)/\min(\pi, 1-\pi)})\right)^{-2} - 1\right), \quad (2.59)$$

$$RE^2(\hat{r}_{opt}) \leq \frac{1}{s_1 s_2 n}\left(\left(1 - \min(1, \sqrt{2KL(q_1, q_2)})\right)^{-2} - 1\right), \quad (2.60)$$

$$RE^2(\hat{r}_{opt}) \leq \frac{1}{s_1 s_2 n}\left(\left(1 - \min(1, \sqrt{2KL(q_2, q_1)})\right)^{-2} - 1\right). \quad (2.61)$$

up to the first order, where $n = n_1 + n_2$ *and* $s_i = n_i/n$ *for* $i = 1, 2$.

Proof. Recall that $JS_\pi(q_1, q_2) = \pi KL(q_1, q_\pi) + (1-\pi)KL(q_2, q_\pi)$ where $q_\pi = \pi q_1 + (1-\pi)q_2$ is a mixture of q_1, q_2. Let $d_{TV}(q_1, q_2)$ be the total variation distance between q_1 and q_2. By Pinsker's inequality we have $KL(q_i, q_\pi) \geq 2d^2_{TV}(q_i, q_\pi)$ for $i = 1, 2$ (Pinsker, 1964). Then

$$\begin{aligned}
JS_\pi(q_1, q_2) &= \pi KL(q_1, q_\pi) + (1-\pi)KL(q_2, q_\pi) && (2.62)\\
&\geq 2\pi d^2_{TV}(q_1, q_\pi) + 2(1-\pi)d^2_{TV}(q_2, q_\pi) && (2.63)\\
&\geq 2\min(\pi, 1-\pi)(d^2_{TV}(q_1, q_\pi) + d^2_{TV}(q_2, q_\pi)) && (2.64)\\
&\geq 2\min(\pi, 1-\pi)\left(\frac{1}{2}(d_{TV}(q_1, q_\pi) + d_{TV}(q_2, q_\pi))^2\right) && (2.65)\\
&\geq \min(\pi, 1-\pi)d^2_{TV}(q_1, q_2) && (2.66)
\end{aligned}$$

by the algebraic-geometric mean inequality and the triangle inequality. Since $d_{TV}(q_1, q_2) \geq \frac{1}{2}H^2(q_1, q_2)$ (Le Cam, 1969), we have $JS_\pi(q_1, q_2) \geq \min(\pi, 1-\pi)\left(\frac{1}{2}H^2(q_1, q_2)\right)^2$. Since both $JS_\pi(q_1, q_2)$ and $H^2(q_1, q_2)$ are non-negative, $JS_\pi(q_1, q_2) \to 0$ implies $H^2(q_1, q_2) \to 0$ and $RE^2(\hat{r}_{opt}) \to 0$ by (2.58). On the other hand, since $H^2(q_1, q_2) \leq 2$, we have

$$\frac{1}{2}H^2(q_1, q_2) \leq \min(1, \sqrt{JS_\pi(q_1, q_2)/\min(\pi, 1-\pi)}).^2 \quad (2.67)$$

Substituting it into the right hand side of (2.58) yields (2.59).

From the last paragraph, we have $KL(q_1, q_2) \geq 2d^2_{TV}(q_1, q_2) \geq \frac{1}{2}\left(H^2(q_1, q_2)\right)^2$. Therefore $KL(q_1, q_2) \to 0$ also implies $H^2(q_1, q_2) \to 0$ and $RE^2(\hat{r}_{opt}) \to 0$. We also have $\frac{1}{2}H^2(q_1, q_2) \leq \min(1, \sqrt{2KL(q_1, q_2)})$. Substituting it into the right hand side of (2.58) yields (2.60). We can show (2.61) using the same argument. □

[2]Since $JS_\pi(q_1, q_2) \leq \log 2$ for all $\pi \in (0, 1)$ (Lin, 1991), $\sqrt{JS_\pi(q_1, q_2)/\min(\pi, 1-\pi)}$ does not exceed 1 by large amount when π is close to 1/2. For example, when $\pi = 1/2$, $\sqrt{JS_\pi(q_1, q_2)/\min(\pi, 1-\pi)} < 1.18$.

From Proposition 2.8.1 we see minimizing these choices of f-divergences are also effective for reducing the $RE^2(\hat{r}_{opt}^{\phi})$. However, these choices of f-divergence are inefficient compared with $H_{s_2}(q_1^{(\phi)}, q_2)$ since minimizing these f-divergences only corresponds to minimizing some upper bounds of the first order approximation of $RE^2(\hat{r}_{opt}^{\phi})$, while minimizing $H_{s_2}(q_1^{(\phi)}, q_2)$ is equivalent to minimizing the first order approximation of $RE^2(\hat{r}_{opt}^{(\phi)})$ directly.

2.8.6 Effectiveness of the hybrid objective in Algorithm 2.2

As we have discussed previously, we introduce the hybrid objective to stabilize the alternating training process and accelerate the convergence of Algorithm 2.2. Here we demonstrate the effectiveness of the hybrid objective in Algorithm 2.2 using the mixture of rings example in Chapter 2.5 with $p = 12$, $\boldsymbol{\mu}_{11} = (2, 2), \boldsymbol{\mu}_{12} = (-2, -2), \boldsymbol{\mu}_{21} = (2, -2), \boldsymbol{\mu}_{21} = (-2, 2)$, $b_1 = 3, b_2 = 6, \sigma_1 = 1, \sigma_2 = 2$. We set $q_1^{(\phi)}$ to be a Real-NVP with 5 coupling layers. We first run Algorithm 2.2 50 times with $n_i = n_i' = 1000$ for $i = 1, 2$ and $\lambda_1 = \lambda_2 = 0.05$. We record the values of the objective function and $\tilde{r}_t$ of the first 25 iterations. Then we run Algorithm 2.2 50 times with $n_i = n_i' = 1000$ for $i = 1, 2$ and $\lambda_1 = \lambda_2 = 0$, and record the same values. Recall that setting $\lambda_1 = \lambda_2 = 0$ is equivalent to using the original f-GAN objective (2.22). From Figure 2.8 we see most of the hybrid objectives and the corresponding $\tilde{r}_t$ values have stabilized after 20 iterations. The stand alone f-GAN objective with $\lambda_1 = \lambda_2 = 0$ also demonstrate a decreasing trend, but the objective values are much more wiggly compared to the hybrid objective due to the adversarial training process, and there is no sign of convergence in 25 iterations. Note that for both the hybrid objective and the original f-GAN objective, the corresponding $\tilde{r}_t$ tend to converge to a value slightly different from the true r as the number of iteration increases. This is likely due to the bias we discussed previously.

2.8.7 Additional simulations

Simulated example: Quantized Mixture of Gaussians

Here we illustrate how Normalizing flows can be used to increase the overlap between discrete random variables in the context of estimating a single normalizing constant (i.e. one of q_1, q_2 is completely known). We take the quantized Mixture of Gaussian in Tran et al. (2019) and Metz et al. (2017) as a toy example.

Following Tran et al. (2019), we first define the completely known "base" distribution q_1. Let $\omega^{(1)}, \omega^{(2)}$ be two categorical variables each with 90 states. Let

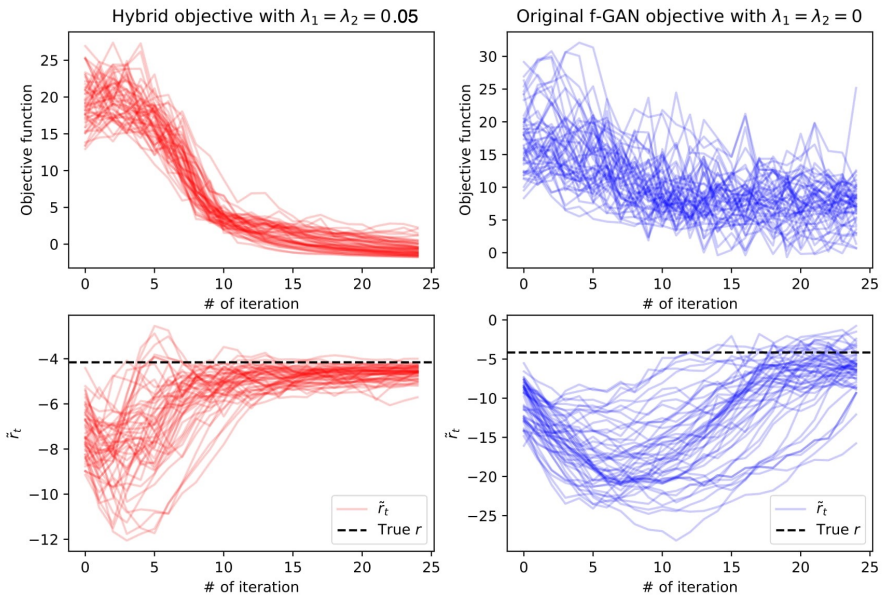


Figure 2.8: Left: The objective function and $\tilde{r}_t$ of the first 25 iterations of Algorithm 2.2 with $\lambda_1 = \lambda_2 = 0.05$. Right: The objective function and $\tilde{r}_t$ of the first 25 iterations of Algorithm 2.2 with $\lambda_1 = \lambda_2 = 0$.

$\omega = (\omega^{(1)}, \omega^{(2)})$. Let q_1 be a uniform distribution over all possible states of ω. The probability mass function of q_1 is then

$$q_1(\omega^{(1)} = u, \omega^{(2)} = v) = \frac{1}{90 \times 90} \qquad u, v \in \{1, 2, ..., 90\} \tag{2.68}$$

We then define the quantized Mixture of Gaussian distribution as our "target" distribution q_2. In order to define the quantized Mixture of Gaussian, we first define $\tilde{g}(x)$, the unnormalized density of a mixture of 2D Gaussian distributions, to be

$$\tilde{g}(x) = \sum_{k=1}^{K} \pi_k \tilde{p}(x; \mu_k, \sigma^2 I_2) \tag{2.69}$$

where $x \in \mathbb{R}^2$, I_2 is the 2×2 identity matrix, $\tilde{p}(\cdot; \mu, \Sigma) = \exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu))$ is the unnormalized 2D Gaussian density with mean μ and covariance Σ, $K = 4$, $\sigma = 0.1$, $\mu_1 = (2, 0)$, $\mu_2 = (-2, 0)$, $\mu_3 = (0, 2)$, $\mu_4 = (0, -2)$ and $\pi_k = \frac{1}{K}$ for $k = 1, ..., K$. We then truncate the support of $\tilde{g}(x)$ to be $[-2.25, 2.25]^2$. We now define $q_2(\omega)$, the quantized 2D Mixture of Gaussian distribution, by discretizing this square at the 0.05 level (i.e. forming a 90×90 equally spaced grid). This discretization step leads to two categorical variables $\omega^{(1)}, \omega^{(2)}$ each with 90 states. For $u, v \in \{1, 2, ..., 90\}$, let $B_{uv} \subseteq [-2.25, 2.25]^2$ be the cell of the grid that corresponds to

the state $\{\omega^{(1)} = u, \omega^{(2)} = v\}$. Then the unnormalized probability mass function of q_2 can be written as

$$\tilde{q}_2(\omega^{(1)} = u, \omega^{(2)} = v) = \int_{x \in B_{uv}} \tilde{g}(x)dx \qquad u, v \in \{1, 2, ..., 90\}. \tag{2.70}$$

See Figure 2.10 for a 2D histogram of samples from q_2. Let

$$Z_2 = \sum_{u=1}^{90} \sum_{v=1}^{90} \int_{x \in B_{uv}} \tilde{g}(x)dx \tag{2.71}$$

be the normalizing constant of $\tilde{q}_2(\omega)$, which can be computed easily. Let $q_2(\omega) = \tilde{q}_2(\omega)/Z_2$ be the corresponding normalized pmf. Since q_1 is completely known, its normalizing constant Z_1 is equal to 1 and therefore $\tilde{q}_1(\omega) = q_1(\omega)$.

Our goal is to estimate $\log r = \log Z_1 - \log Z_2 = -\log Z_2$ by first increasing the overlap between q_1 and q_2 using a Normalizing flow, then compute the asymptotically optimal Bridge estimator of r based of the transformed distributions. Let $N = \{500, 1000, 1500, 2000, 2500\}$. To demonstrate the effectiveness of this approach, for each value of N, we first draw $n_1 = n_2 = N$ training samples $\{\omega_{1j}\}_{j=1}^{n_1}$ and $\{\omega_{2j}\}_{j=1}^{n_2}$ from q_1, q_2 respectively, and use an autoregressive discrete flow Tran et al. (2019) to estimate a bijective transformation T that maps q_1 to q_2 based on the training samples and the training procedure given by Tran et al. (2019). One key distinction between discrete flows Tran et al. (2019) and their continuous counterparts (e.g. Dinh et al. (2016); Kingma et al. (2016)) is that for discrete flows, the base distribution q_1 is treated as a model parameter and is estimated jointly with the transformation T. In our example, this means the parameterization (i.e. the 90×90 probability table) we chose for q_1 in (2.68) is only treated as the "initial values" of the model parameters, and is updated alongside with the transformation T. (Note that when q_1, q_2 have a large number of discrete states, storing or updating the probability table of the base q_1 is computationally infeasible. To alleviate this problem, Tran et al. (2019) also considered more sophisticated parameterizations of the "trainable base" q_1 such as the autoregressive Categorical distribution.) Let T_1 be the estimated transformation, $\bar{q}_1$ be the updated base distribution (which is also completely known and easy to sample from). Let $\bar{q}_1^{(T)}$ be the transformed distribution obtained by applying T_1 to the samples from the updated $\bar{q}_1$. We then draw $n_1' = n_2' = N$ estimating samples $\{\bar{\omega}_{1j}'\}_{j=1}^{n_1'}$ and $\{\omega_{2j}'\}_{j=1}^{n_2'}$ from $\bar{q}_1, q_2$ respectively, and compute $\hat{r}_{opt}^{(T)}$ in (2.7) based on the transformed $\{T_1(\bar{\omega}_{1j}')\}_{j=1}^{n_1'}$ and the original $\{\omega_{2j}'\}_{j=1}^{n_2'}$. For each value of N, we repeat this process 100 times, and report the MC estimate of $MSE(\hat{r}_{opt}^{(T)})$ based on the repeated $\hat{r}_{opt}^{(T)}$s and the ground truth r. Let $\hat{r}_{opt}$ be the optimal Bridge estimator based on the original

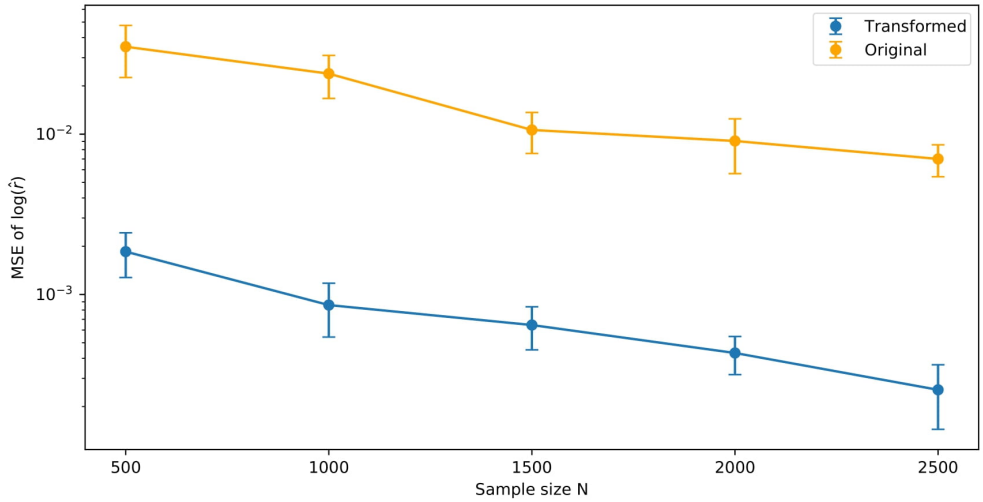


Figure 2.9: MC estimate of MSE of $\log \hat{r}_{opt}^{(T)}$ and $\log \hat{r}_{opt}$ for each value of N. Vertical segments represent the 2σ error bars.

$\tilde{q}_1, \tilde{q}_2$. For each value of N, we compare $MSE(\log \hat{r}_{opt}^{(T)})$ with $MSE(\log \hat{r}_{opt})$, which is also estimated based on 100 repetitions in a similar fashion. From Figure 2.9 we see $\hat{r}_{opt}^{(T)}$ is a reliable estimator of r for all choice of N and is much more accurate than the optimal Bridge estimator based on the original $\tilde{q}_1, \tilde{q}_2$. From Figure 2.10 we also see that the transformed $\bar{q}_1^{(T)}$ accurately captures the multimodal structure of q_2.

In addition to the quantized mixture of Gaussian example, more substantial applications of discrete flows can also be found in Tran et al. (2019). However, the discrete flows in Tran et al. (2019) are in general not directly applicable to our proposed Algorithm 2.2. This is because in our Algorithm 2.2, the unnormalized densities $\tilde{q}_1$ and $\tilde{q}_2$ are specified by the users and therefore can be arbitrary. However, for discrete flows, the "base" distribution has to be completely known, and is treated as a trainable model parameter (as in this example). This means we are not able to use it to directly estimate the ratio of normalizing constants between two arbitrary unnormalized probability mass functions in the same way as in Algorithm 2.2. Nevertheless, one may obtain an estimate of the ratio of normalizing constants between two discrete distributions by estimating their normalizing constants separately using discrete flows and the procedure described in this example. For future work, we are interested in extending Algorithm 2.2 so that it is able to handle arbitrary unnormalized pmfs using e.g. more sophisticated Normalizing flow architectures.

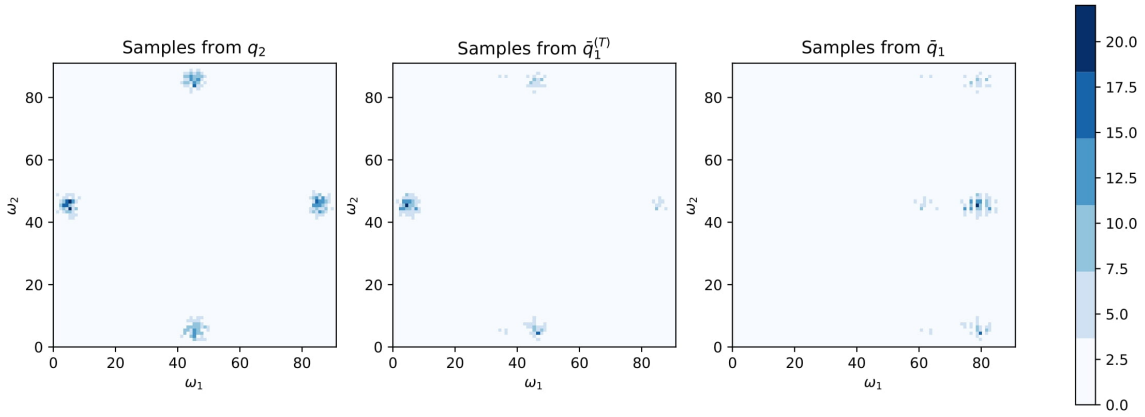


Figure 2.10: Left: 2D histogram of 10^3 samples from q_2. Mid: 2D histogram of 10^3 samples from the transformed $\bar{q}_1^{(T)}$, which is estimated using $N = 10^3$ training samples. Right: 2D histogram of 10^3 samples from the corresponding updated base distribution $\bar{q}_1$ of the transformed $\bar{q}_1^{(T)}$.

Simulated example: Mixture of t-distributions

In this example, we let q_1 and q_2 be two mixtures of p dimensional t-distributions. We are interested in this example because both q_1, q_2 are multimodal and have heavy tails. For $i = 1, 2$, let

$$q_i(\omega) = \sum_{k=1}^{K} \pi_{ik} p_t(\omega; \mu_{ik}, \Sigma_i, \nu_i), \tag{2.72}$$

where K is the number of components, $\pi_i = \{\pi_{ik}\}_{k=1}^K$ are the mixing weights and $p_t(\cdot; \mu_{ik}, \Sigma_i, \nu_i)$, the kth component of q_i is the pdf of a multivariate t-distribution with mean $\mu_{ik} \in \mathbb{R}^p$, positive-definite scale matrix $\Sigma_i \in \mathbb{R}^{p \times p}$ and degree of freedom $\nu_i \in \mathbb{R}^+$. Note that all K components of q_i have the same covariance structure and degree of freedom. Let

$$Z_i = \frac{\Gamma((\nu_i + p)/2)}{\Gamma(\nu_i/2)\nu_i^{p/2}\pi^{p/2}|\Sigma_i|^{1/2}} \tag{2.73}$$

be the normalizing constant of $p_t(\cdot; \mu_{ik}, \Sigma_i, \nu_i)$. Note that this quantity does not depend on μ_{ik}. Let $\tilde{p}_t(\cdot; \mu_{ik}, \Sigma_i, \nu_i) = Z_i p_t(\cdot; \mu_{ik}, \Sigma_i, \nu_i)$ be the unnormalized density of each component $p_t(\cdot; \mu_{ik}, \Sigma_i, \nu_i)$. Let $\tilde{q}_i(\omega) = \sum_{k=1}^K \pi_{ik} \tilde{p}_t(\omega; \mu_{ik}, \Sigma_i, \nu_i)$ be the unnormalized density of $q_i(\omega)$. It is easy to verify that $\tilde{q}_i(\omega) = Z_i q_i(\omega)$, i.e. the normalizing constant of $\tilde{q}_i(\omega)$ is Z_i.

For this example, we consider $p = \{5, 10, 20, 40, 60, 80, 100\}$. For each choice of p, the parameters of q_1, q_2 are chosen in the following way: We fix the degree of freedom $\nu_1 = 1, \nu_2 = 4$, and number of component $K = 7$. The mixing weights π_i for $i = 1, 2$ are sampled independently from a $Dir(\alpha_1, ..., \alpha_K)$ where $\alpha_k = 1$ for $k = 1, ..., K$. The

location parameters μ_{ik} for $i = 1, 2$, $k = 1, ..., K$ are sampled from a standard Normal $N(0, I_p)$ independently. For the scale matrices Σ_i, we first sample Σ_1, Σ_2 independently from a inverse Wishart distribution $\mathcal{W}^{-1}(I_p, p)$, then rescale Σ_1, Σ_2 so that $|\Sigma_1| = 1$ and $|\Sigma_2| = 1000$. This ensures the components of q_1 are more concentrated than the components in q_2.

We estimate $\log r = \log Z_1 - \log Z_2$ in a similar fashion to Chapter 2.5 and 2.6. For each choice of p, we run each method 30 times. Let $\hat{r}$ be a generic estimate of r. We use the MC estimate of RMSE of $\log \hat{r}$, i.e. $E((\log \hat{r} - r)^2)/(\log r)^2$, based on the repeated runs as the benchmark of performance for this example. For each repetition, we run each method with $N_1 = N_2 = 6000$ independent samples from q_1, q_2 respectively. For our Algorithm 2.2, we parameterize $q_1^{(\phi)}$ as a Real-NVP with 20 coupling layers, and set $\lambda_1 = \lambda_2 = 0.01$. For the rest of the methods, we use the default or recommended settings. The results are summarized in Table 2.1. We see our Algorithm 2.2 outperforms all methods when $p \geq 40$.

p	f-GB	GBS	Warp-III	Warp-U
5	3.69e-5	8.22e-4	1.14e-3	**3.54e-5**
10	**6.21e-5**	1.74e-3	5.15e-3	6.42e-5
20	4.12e-3	4.96e-3	8.87e-3	**1.69e-3**
40	**1.23e-2**	4.05e-2	9.01e-2	5.75e-2
60	**1.21e-2**	3.88e-2	9.26e-2	7.64e-2
80	**1.81e-2**	5.20e-2	1.59e-1	6.05e-2
100	**2.46e-2**	8.14e-2	-	4.78e-1

Table 2.1: The estimated RMSE of the $\log \hat{r}$ of each methods based on 30 repeated runs. The lowest estimated RMSE for each p is in boldface. Warp-III does not converge for most of the repeated runs when $p = 100$ so we are not able to estimate its RMSE.

Chapter 3

Estimating operational coverage

In the last chapter, we proposed a new strategy for constructing statistically more efficient estimators of Bayes factors using deep generative models. We now change the focus from Bayes factor estimation and Bayesian model selection to diagnosing approximation error in approximate Bayesian inference. When we implement Bayesian inference for even moderately large datasets or complicated models, some approximation is usually inescapable. Approximation schemes suitable for different Bayesian applications include Approximate Bayesian Computation (Pritchard et al., 1999; Beaumont, 2010), Variational Inference (Jordan et al., 1999; Hoffman et al., 2013), loss-calibrated inference (Lacoste-Julien et al., 2011; Kuśmierczyk et al., 2019) and synthetic likelihood (Wood, 2010; Price et al., 2018). New applications suggest new approximation schemes, and generic diagnostic tools measuring the reliability of these approximations are necessary.

A number of methods have been developed to check the quality of approximation: Rodrigues et al. (2018) give a post-processing algorithm specifically for diagnosing and recalibrating ABC posteriors. Kang et al. (2021) focus on diagnosing inexact algorithms for doubly intractable distributions. The generic diagnostic tools given in Yao et al. (2018) and Talts et al. (2020) focus on checking the average performance of an approximation scheme over data space $\mathcal{Y}$ and are related to Prangle et al. (2014), which focuses on ABC posterior diagnostics. Their methods can be seen as an extension of Cook et al. (2006), Geweke (2004) and Monahan and Boos (1992), which were setup for checking MCMC software implementations. In contrast, we are interested in developing generic computational tools to assess the quality of approximation *at the observed data* y_{obs}. If one posterior approximation scheme works poorly in some region of $\mathcal{Y}$ but works well at y_{obs}, we may reject a useful approximation using any diagnostic tool based on average performance. We may also conversely accept a poor approximation for the same reason.

In this chapter, we focus on diagnosing approximation error in the approximate credible sets. In particular, we are interested in calibrating the *coverage* of approximate credible sets obtained from approximate posteriors conditioned on the observed y_{obs}, where *coverage* is referred to the probability that credible sets (either approximate or exact) cover samples from the *exact* posterior distribution. We give computational frameworks for estimating the bias in coverage resulting from making approximations in Bayesian inference, and show how to estimate the coverage an approximate credible set achieves at y_{obs} when the observation model and the prior are computationally intractable but can be simulated.

3.1 Introduction and background

Bayesian credible sets with stated nominal coverage are a fundamental way to communicate statistical uncertainty. However, as some approximation is inevitable for large data sets and complex models, we usually report approximate credible sets with uncalibrated coverage. Approximation may have controlled or "fixed" precision. For example, in MCMC, samples are only asymptotically distributed according to the posterior. In this case, the "precision parameter", controlling the accuracy of the approximation, is the run length. Approximate Bayesian Computation (Pritchard et al., 1999) typically has two precision parameters: a distance threshold and a Monte Carlo sample size. There are also "fixed" approximations with no precision parameters. For example, one may replace a likelihood by an approximation which cannot be improved by varying a control parameter. Pseudo-likelihood (Besag, 1975) and Variational Inference (Jordan et al., 1999; Hoffman et al., 2013) often lead to a fixed approximation.

Various approaches have been proposed to check if the approximation is acceptable. Rodrigues et al. (2018) give diagnostic and post-processing procedures ABC posteriors. Kang et al. (2021) give diagnostic tools for checking the reliability of inexact algorithms for doubly intractable distributions. Recent new generic diagnostic tools given in Talts et al. (2020) and Yao et al. (2018) are related to earlier work in Prangle et al. (2014) and exploit an idea, developed in Geweke (2004) and Cook et al. (2006) as an MCMC convergence diagnostic, going back to Monahan and Boos (1992). In early related work, Menendez et al. (2014) give procedures for correcting credible sets under conditions stronger than those required here.

We consider an approximate Bayesian credible set with given nominal coverage level α. What coverage does the credible set actually achieve? Wherever approximate

Bayesian inference reports a credible set, an associated coverage measure should be given. Let $\pi(\phi)$ be the prior for $\phi \in \Omega$, let $p(y|\phi)$ be the observation model (i.e. likelihood) for generic data $y \in \mathcal{Y}$ and let $\pi(\phi|y) \propto \pi(\phi)p(y|\phi)$ be the posterior for ϕ given data y. Let $\tilde{\pi}(\theta)$ and $\tilde{p}(y|\theta)$ be the approximate prior and likelihood for parameter $\theta \in \Omega$ with approximate posterior $\tilde{\pi}(\theta|y)$. Let y_{obs} be the data we actually observe. This paper is motivated by problems where we cannot in practice sample $\pi(\phi|y)$ using any known Monte Carlo method. We assume a tractable approximation $\tilde{\pi}(\theta|y)$ is available, and we assume it is possible to sample $\phi \sim \pi(\cdot)$ and $y' \sim p(\cdot|\phi)$ efficiently (just as in ABC).

The estimated credible set is computed for a posterior distribution $\tilde{\pi}(\theta|y)$ which approximates the exact posterior $\pi(\phi|y)$. The exact level α credible set C_y for the exact posterior $\pi(\phi|y)$ satisfies

$$\alpha = \int_\Omega \mathbb{1}(\phi \in C_y)\pi(\phi|y)d\phi.$$

This set C_y has perfect Bayes coverage in the sense that, if $\phi \sim \pi(\cdot)$ is a draw from the prior, and $y \sim p(\cdot|\phi)$ is a draw from the observation model, then $\Pr(\phi \in C_Y|Y = y) = \alpha$. The credible set covers the true parameter ϕ with probability α if nature drew ϕ from the prior, and the data y really was generated using the observation model we are using. This is the definition of Bayesian coverage, not an assumption.

In practice we compute an approximate credible set $\tilde{C}_y$ using the approximate posterior $\tilde{\pi}(\theta|y)$. This is a set $\tilde{C}_y$ satisfying

$$\alpha = \int_\Omega \mathbb{1}(\theta \in \tilde{C}_y)\tilde{\pi}(\theta|y)d\theta.$$

This will not in general have the right coverage for the exact posterior. If $\phi \sim \pi(\cdot)$, and $y \sim p(\cdot|\phi)$, and we let $b(y) = \Pr(\phi \in \tilde{C}_Y|Y = y)$, then

$$b(y) = \int_\Omega \mathbb{1}(\phi \in \tilde{C}_y)\pi(\phi|y)d\phi \tag{3.1}$$

is the operational coverage $\tilde{C}_y$ achieves for ϕ, a draw from the exact posterior. This is not equal α in general. The coverage bias $b(y) - \alpha$ can vary markedly over data space. Cook et al. (2006) observe that coverage may be estimated, but the quantity they mention is equivalent to $\int_\mathcal{Y} p(y)b(y)dy$, an average over the data space which may differ a great deal from $b(y)$, as we see in the example in Chapter 3.4.

In practice we may not be able to compute an exact credible set, even after making the approximation leading to $\tilde{\pi}(\theta|y)$. In this case we would typically simulate $\theta_j \sim \tilde{\pi}(\cdot|y)$ for $j = 1, \ldots, J$, set $\underline{\theta} = \{\theta_1, \ldots, \theta_J\}$ and compute an estimate, $\hat{C}_y(\underline{\theta})$, for

$\tilde{C}_y$ based on the J samples. There is an additional Monte Carlo error in the coverage and so, with ϕ, y and θ distributed as prior, observation model and approximate posterior, we let

$$c(y) = \Pr(\phi \in \hat{C}_Y(\underline{\theta})|Y = y),$$

denote the realised coverage allowing for Monte Carlo error. In this chapter, we give algorithms for estimation of $b(y_{obs})$ and $c(y_{obs})$. We discuss the function $c(y)$ by default, as estimation of $b(y)$ is a simpler special case.

The joint distribution of ϕ, y and $\underline{\theta}$ in the generative model is given by

$$m(\phi, y, \underline{\theta}) = \pi(\phi)p(y|\phi)\tilde{\pi}(\underline{\theta}|y), \tag{3.2}$$

ans the conditional distribution of $\phi, \underline{\theta}$ given y is

$$m(\phi, \underline{\theta}|y) = \pi(\phi|y)\tilde{\pi}(\underline{\theta}|y),$$

where $\tilde{\pi}(\underline{\theta}|y)$ is the joint distribution of $\underline{\theta} \in \Omega^J$ (an abuse of notation). Now $c(y)$ is an expectation over $m(\phi, \underline{\theta}|y)$:

$$c(y) = \int_\Omega \int_{\Omega^J} \mathbb{1}(\phi \in \hat{C}_y(\underline{\theta}))m(\phi, \underline{\theta}|y)d\underline{\theta}d\phi.$$

The coverage is a posterior expectation in the exact distribution $\pi(\phi|y)$. Coverage estimation resembles the original problem of estimating credible sets from the exact posterior, which we have said we cannot do! However we can sample $\pi(\phi)$ and the observation model $p(y|\phi)$, and it proves easier to estimate $c(y)$ than some general expectation in the exact posterior.

If it were possible to simulate $\phi \sim \pi(\cdot|y)$, the exact posterior, then estimation of $b(y)$ and $c(y)$ would be straightforward using an idealized Algorithm 3.1: For $i = 1, \ldots, M$ we simulate $\phi_i \sim \pi(\cdot|y)$ and $\theta_{ij} \sim \tilde{\pi}(\cdot|y)$ for $j = 1, \ldots, J$. We then construct an α level approximate credible set $\hat{C}_i$ based on $\underline{\theta}_i = \{\theta_{i1}, \ldots, \theta_{iJ}\}$, and define

$$c_i = \mathbb{1}(\phi_i \in \hat{C}_i).$$

Now $\hat{c}(y) = \frac{1}{M}\sum_{i=1}^M c_i$ is an unbiased and consistent estimator for $c(y)$. If we replace $\hat{C}_i$ by $\tilde{C}_i$ then the procedure estimates $b(y)$. *We assume that Algorithm 3.1 cannot be implemented.* In the examples we give in Chapter 3.4 we implement Algorithm 3.1 as the ground truth to show we have estimated $c(y_{obs})$ correctly. This will not in general be possible.

This chapter is structured as follows. In Chapter 3.2 we discuss the connection to previous works. We then outline two algorithms for the estimation problem in

Algorithm 3.1 Estimation of operational coverage $c(y)$

Input: Data point y; Exact posterior distribution $\pi(\phi|y)$; Approximate posterior distribution $\tilde{\pi}(\theta|y)$; Number of samples J from the approximate posterior; Number of samples M from the exact posterior.
for i in $1, \ldots, M$ **do**
  Simulate $\phi_i \sim \pi(\phi|y)$ and $\underline{\theta}_i = \{\theta_{i1}, \ldots, \theta_{iJ}\}$ where $\theta_{ij} \sim \tilde{\pi}(\theta|y)$ for $j = 1, \ldots, J$
  Compute the approximate credible set $\hat{C}_i$ based on $\underline{\theta}_i$. Set $c_i = \mathbb{1}(\phi_i \in \hat{C}_i)$
end for
Return: Estimated coverage $\hat{c}(y) = \frac{1}{M}\sum_{i=1}^{M} c_i$

Chapter 3.3. In Chapter 3.4 we apply the algorithms to calibrate a pseudo-likelihood approximation. In Chapter 3.5, we calibrate the approximate posterior of a random partition in a hierarchical model with a Dirichlet-Process prior on the distribution of random effects. We finish with a brief discussion.

3.2 Relation to Previous Works

Lee et al. (2019) introduce the idea of estimating coverage probabilities as a validation procedure, and give two proof-of-concept estimators for $c(y)$ when simulation from the exact posterior $\pi(\phi|y)$ is not possible. Algorithms proposed in this chapter are a qualitative improvement, as those earlier methods completely fail on the examples we give in this paper.

Lee et al. (2019) give an importance sampling procedure which targets an approximation to $c(y)$. Let $\delta(x, y) : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ be a generic distance function such that $\delta(x, y) = 0$ if and only if $x = y$, $\rho > 0$ a tolerance level and let $\Delta(y) = \{y' : \delta(y, y') \leq \rho\}$ be a closed ball in $\mathcal{Y}$ centered at y. Let

$$d(y) = \Pr(\phi \in \hat{C}_Y(\underline{\theta})|Y \in \Delta(y))$$

be an ABC-style approximation to $c(y)$. In terms of the density $m(\phi, y', \underline{\theta})$ in Equation 3.2,

$$d(y) = \int_{\Delta(y)} \int_{\Omega^J} \int_{\Omega} \frac{\mathbb{1}(\phi \in \hat{C}_{y'}(\underline{\theta}))}{z(y, \rho)} m(\phi, y', \underline{\theta}) d\phi d\underline{\theta} dy' \tag{3.3}$$

with $z(y, \rho) = \Pr(Y \in \Delta(y))$ a normalising constant. Lee et al. (2019) use importance sampling to estimate $d(y_{obs})$. Let

$$\tilde{m}(\phi, y, \underline{\theta}) \propto \tilde{\pi}(\phi|y_{obs}) p(y|\phi) \tilde{\pi}(\underline{\theta}|y) \mathbb{1}(y \in \Delta(y_{obs}))$$

be the proposal distribution. Let $\{\phi_i, y_i, \underline{\theta}_i\}_{i=1}^M \overset{i.i.d.}{\sim} \tilde{m}(\phi, y, \underline{\theta})$. Lee et al. (2019) estimate $d(y_{obs})$ with

$$\hat{d} = \sum_{i=1}^{M} w(\phi_i, y_i, \underline{\theta}_i) \mathbb{1}(\phi_i \in \hat{C}_{y_i}(\underline{\theta}_i)),$$

where $w(\phi, y, \underline{\theta}) \propto m(\phi, y, \underline{\theta})/\tilde{m}(\phi, y, \underline{\theta})$ are the importance weights. The authors use $\hat{d}$ as an estimate of $c(y_{obs})$. This approach has two drawbacks: Although $\hat{d} \to d(y_{obs})$ as $M \to \infty$, $d(y_{obs}) \neq c(y_{obs})$ in general so the method is asymptotically biased unless we additionally take ρ to zero. This would be impractical as we will simulate no data $y \in \Delta(y_{obs})$ if ρ is arbitrarily close to zero. Secondly, as the authors observe, this estimator can be unstable due to high weight variance. In Chapter 3.3 we give an improved Algorithm 3.2 based on Annealed Importance Sampling (AIS) (Neal, 2001). Similar to the importance sampling approach proposed by Lee et al. (2019), our Algorithm 3.2 is also asymptotically biased, but AIS is a much more powerful tool. Simulation studies in Chapter 3.4 and 3.7.2 show that the bias can be made very small, and the AIS sampler gives a much higher Effective Sample Size (ESS).

In Algorithm 3.2, we also take advantage of two simple but important simplifications not exploited by Lee et al. (2019). Firstly, we replace $\mathbb{1}(\phi \in \hat{C}_{y'}(\underline{\theta}))$ in Equation 3.3 with $\mathbb{1}(\phi \in \hat{C}_{y_{obs}}(\underline{\theta}))$. This is a sensible thing to do because we will take the limit of $\hat{C}_Y(\underline{\theta})$ as $Y \to y_{obs}$, so we simply substitute the limiting value $\hat{C}_{y_{obs}}(\underline{\theta})$. This avoids the need to compute $\tilde{\pi}(\theta|y)$ and simulate $\underline{\theta}$ from it for each simulated y-value, a big time-saver in some settings. Secondly, we could further simplify Equation 3.3 by dropping the integral with respect to $\underline{\theta}$. That is, instead of treating $\hat{C}_{y_{obs}}(\underline{\theta})$ as a random interval depending on the samples $\underline{\theta} \sim \tilde{\pi}(\theta|y_{obs})$ and marginalising out $\underline{\theta}$, we focus on one realization $\hat{C}_{y_{obs}}$. This is not really an approximation, but a redefinition of the quantity we measure. It is actually more natural to the user: After all we are interested in the coverage of $\hat{C}_{y_{obs}}$ we actually report, and this depends on the specific realisation of $\underline{\theta}$ we used to compute $\hat{C}_{y_{obs}}$.

For comparison with the importance sampling method in Lee et al. (2019), we take their Ising model example and scale it up by a factor of 25 in Chapter 3.4. On this larger problem we find our Algorithm 3.2 far out-performs the importance sampling approach in Lee et al. (2019), yielding an accurate estimate of the coverage, and an ESS some 10 times larger in a comparable run time.

Lee et al. (2019) suggest an alternative regression procedure for estimating $c(y)$. If we simulate $\{\phi_i, y_i, \underline{\theta}_i\}_{i=1}^M \overset{i.i.d.}{\sim} m(\phi, y, \underline{\theta})$, then at each y_i, we have $(\phi_i, \underline{\theta}_i|y_i) \sim \pi(\phi|y_i)\tilde{\pi}(\underline{\theta}|y_i)$. It follows that if we compute $\hat{C}_i = \hat{C}_{y_i}(\underline{\theta}_i)$ and a binary "response"

$c_i = \mathbb{1}(\phi_i \in \hat{C}_i)$ then we have $c_i \sim Bernoulli(c(y_i))$ for $i = 1, \ldots, M$. Lee et al. (2019) suggest using a Generalised Additive Model (GAM) for logistic regression to estimate the function $c(y)$. The authors take a p-dimensional summary statistic $s(y_i) = (s_1(y_i), \ldots, s_p(y_i))$ with $p \ll \dim(\mathcal{Y})$ as the regression covariates, and the binary c_i as the response. This works well when $s(y)$ is sufficient, as is the case in our Ising model example in Chapter 3.4. Choosing $s(y)$ is difficult in general since a low dimensional sufficient statistics of y is not always available. However this is the sort of problem, in which random forest regression and Bayesian Additive Regression Trees (BART) (Chipman et al., 2010; Kapelner and Bleich, 2016) are very effective. In Chapter 3.3, we also give Algorithm 3.3 based on a Probit BART regression model. For comparison with the previous work, we take an example where the regression approach in Lee et al. (2019) fail in Chapter 3.5, and show that estimation via BART gives reproducible results.

3.2.1 Symmetry in approximation

We note a symmetry of the approximation that will hold more widely: Suppose our observed data is associated with some categorical labels, then the approximation error is invariant under permutation of labels of levels of categorical variables. Let $y \in \mathbb{R}^n$ be a data vector and let $\mathcal{L} = \{\ell_1, \ldots, \ell_N\}$ be the levels of a single categorical variable $x \in \mathcal{L}^n$. Suppose our data are simply $\{y, x\}$ pairs with "response" y and "covariate" x. Let $T(x) = \{T_1, \ldots, T_N\}$ be a collection of subsets of $\{1, \ldots, n\}$ partitioning the indices of x into groups of observations in the same level, so that $i \in T_j \Leftrightarrow x_i = \ell_j$ for each $i = 1, \ldots, n$ and each $j = 1, \ldots, N$. Let $\mathcal{P}_R = \{\sigma \in \mathcal{P} : T(x_\sigma) = T(x)\}$ be the set of permutations corresponding to level-relabeling. If the approximation does not distinguish labels of the levels, we have $c(y) = c(y_\sigma)$ for each $\sigma \in \mathcal{P}_R$ (permuting $y's$ with $x's$ being fixed gives the same data set).

This can be extended to multiple categorical variables. In a complete balanced design, every level of every covariate co-occurs with every level of every other covariate the same number of times. Levels are then exchangeable within each variable simultaneously. In this setting we can compute $c(y)$ by computing it on one special quadrant $\mathcal{Y}_0$ of $\mathcal{Y}$ and then mapping identical copies of $c(y)$ out over $\mathcal{Y}$ by permutation. For example, we take $\mathcal{Y}_0 = \{y \in \mathcal{Y} : \bar{y}_{T_1} \leq \bar{y}_{T_2} \leq \ldots \bar{y}_{T_N}\}$ (i.e. ordered on the averages of y's associated with each level $i = 1, \ldots, N$, with additional order constraints for each variable if there are multiple categorical variables). In our random effect model example in Chapter 3.5, we simulate $\{\phi, y, \underline{\theta}\}$ pairs from $m(\phi, y, \underline{\theta})$, map y into $\mathcal{Y}_0$ and regress over this smaller space where y-values are more densely packed and regression

(using e.g. BART) is easier. We then map the function $\hat{c}(y)$ back to the quadrant containing y_{obs} by permutation.

3.3 Estimating the Operational Coverage

3.3.1 A Weighted-Sample Estimate for Coverage

Here we demonstrate how to estimate $c(y_{obs})$ using Annealed Importance Sampling (AIS) (Neal, 2001). This leverages our ability to draw samples from the approximate posterior $\tilde{\pi}(\phi|y_{obs})$ and use them as a starting point for the AIS iteration.

Let N_{AIS} be the number of AIS steps. Let $\gamma_0 = 0, \beta_0 = 0$. Let $\{\gamma_j\}_{j=1}^{N_{AIS}}$ and $\{\beta_j\}_{j=1}^{N_{AIS}}$ be positive and increasing sequences with $\gamma_{N_{AIS}} = 1$. Define an initial distribution

$$p_0(\phi, y) = \tilde{\pi}(\phi|y_{obs}) \times p(y|\phi),$$

and, for $j = 1, \ldots, N_{AIS}$, intermediate distributions

$$\begin{aligned} p_j(\phi, y) &\propto \pi(\phi)^{\gamma_j} \tilde{\pi}(\phi|y_{obs})^{1-\gamma_j} p(y|\phi) \exp\left(-\beta_j \delta(y, y_{obs})\right) \\ &\propto \pi(\phi) \tilde{p}(y_{obs}|\phi)^{1-\gamma_j} p(y|\phi) \exp\left(-\beta_j \delta(y, y_{obs})\right). \end{aligned}$$

If $\gamma_{N_{AIS}} = 1$, then $p_{N_{AIS}}(\phi, y)$ converges to $\pi(\phi)p(y_{obs}|\phi)$ as $\beta_{N_{AIS}} \to \infty$ and so the marginal $p_{N_{AIS}}(\phi)$ to $\pi(\phi|y_{obs})$, the true posterior (See also Theorem 3.3.1). The approximate posterior $\tilde{\pi}(\phi|y_{obs})$ is a useful part of the initial distribution p_0: If $\tilde{\pi}(\phi|y_{obs})$ is a reasonable approximation of the exact posterior, then it should also support ϕ values for which typical synthetic data $y \sim p(y|\phi)$ are relatively close to the observed y_{obs} from the start.

We now describe an update scheme for ϕ and y that propagates samples $\{\phi, y\}$ from $p_j(\phi, y)$. For each $j = 1, \ldots, N_{AIS}$, let

$$Q_j((\phi, y), (\phi', y')) = q_j((\phi, y) \to (\phi', y'))\alpha_j((\phi, y) \to (\phi', y'))$$

be a Metropolis-Hasting transition kernel with proposal distribution $q_j(\{\phi_j, y_j\} \to \{\phi', y'\}) = f_j(\phi_j, \phi')p(y'|\phi')$ based on a simple local proposal distribution $f_j(\phi, \phi')$ for ϕ, and an acceptance probability

$$\begin{aligned} \alpha_j &= 1 \wedge \frac{p_j(\phi', y')q_j(\{\phi', y'\} \to \{\phi_j, y_j\})}{p_j(\phi_j, y_j)q_j(\{\phi_j, y_j\} \to \{\phi', y'\})} \\ &= 1 \wedge \frac{\pi(\phi')\tilde{p}(y_{obs}|\phi')^{1-\gamma_j} f_j(\phi', \phi_j)}{\pi(\phi_j)\tilde{p}(y_{obs}|\phi_j)^{1-\gamma_j} f_j(\phi_j, \phi')} \times \frac{\exp(-\beta_j \delta(y', y_{obs}))}{\exp(-\beta_j \delta(y_j, y_{obs}))} \end{aligned} \tag{3.4}$$

It is straightforward to verify that $Q_j((\phi, y), (\phi', y'))$ admits $p_j(\phi, y)$ as an invariant distribution.

Let $d_{N_{AIS}} = E_{p_{N_{AIS}}}(\mathbb{1}(\phi \in \hat{C}_{y_{obs}}))$. Let $\{w_i, \phi_i\}_{i=1}^M$ be a set of self-normalised weighted samples generated by the AIS algorithm described in Algorithm 3.2. These pairs are approximately weighted samples from the target $p_{N_{AIS}}(\phi, y)$. Let

$$\hat{c}(y_{obs}) = \sum_{i=1}^{M} w_i \mathbb{1}(\phi_k \in \hat{C}_{y_{obs}}) \tag{3.5}$$

be an estimator for $d_{N_{AIS}}$.

Theorem 3.3.1. *If $\gamma_{N_{AIS}} = 1$, $\delta(y, y_{obs})$ is twice differentiable with respect to $y \in \mathcal{Y}$ and there exists $L > 0$ such that $p(y_{obs}|\phi) < L$ for all $\phi \in \Omega$, then $\hat{c}(y_{obs})$ in (3.5) is a consistent estimator of the true realized coverage achieved by $\hat{C}_{y_{obs}}$ as $M \to \infty$ and $\beta_{N_{AIS}} \to \infty$.*

Proof. See Chapter 3.7.1. □

Algorithm 3.2 summarizes the procedure. In contrast to the importance sampling approach in Lee et al. (2019), we do not need to compute the approximate credible set for each synthetic data vector $y_i^{(N_{AIS})}$ in Algorithm 3.2. This can speed up computation a great deal.

3.3.2 A Regression Estimate for Coverage

In Lee et al. (2019), the authors also suggest estimating $c(y)$ via regression. Let $\{\phi_i, y_i\}_{i=1}^M$ be samples from the generative model $\pi(\phi)p(y|\phi)$, let $\hat{C}_{y_i}$ be an approximate credible set for y_i, and $c_i = \mathbb{1}(\phi_i \in \hat{C}_{y_i})$. Conditioned on y_i, we have

$$c_i \sim Bernoulli(c(y_i)), \quad c(y_i) = \Pr(\phi_i \in \hat{C}_{Y_i} | Y_i = y_i).$$

Let $s(y_i) \in \mathbb{R}^p$ be a vector of summary statistics of y_i. Lee et al. (2019) use a GAM logistic regression model with response c_i and covariates $s(y_i)$ to learn the map from y to $c(y)$. Raynal et al. (2019) and Marin et al. (2018) observe that, for ABC work, Random Forests allow us to handle a potentially high dimensional summary statistics (even if some or many of the dimensions are poorly informative) without preliminary selection. Inspired by their ideas, we applied a Probit Bayesian Additive Regression Tree (BART) model (Chipman et al., 2010) to estimate $c(y)$ when low-dimensional sufficient statistics are not available.

Algorithm 3.2 AIS Estimation of operational coverage $c(y)$

Input: Observed data y_{obs}; Number of AIS iterations N_{AIS}; Summary statistics $s : \mathcal{Y} \to \mathbb{R}^p$; Number of samples J from the approximate posterior; Number of samples M from the generative model; Distance metric $\delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$; positive and increasing sequences $\{\gamma_j\}_{j=1}^{N_{AIS}}$ and $\{\beta_j\}_{j=1}^{N_{AIS}}$ with $\gamma_{N_{AIS}} = 1$.
Simulate $\underline{\theta} = \{\theta_1, \ldots, \theta_J\}$ where $\theta_1, \ldots, \theta_J \sim \tilde{\pi}(\cdot|y_{obs})$; Compute the approximate credible set $\hat{C}_{y_{obs}}(\underline{\theta})$.
for i in $1, \ldots, M$ **do**
  Sample $(\phi_i^{(1)}, y_i^{(1)}) \sim p_0(\phi, y)$.
  Compute $w_i^{(1)} \propto \frac{p_1(\phi_i^{(1)}, y_i^{(1)})}{p_0(\phi_i^{(1)}, y_i^{(1)})}$.
  for j in $2, \ldots, N_{AIS}$ **do**
    Sample $\phi_i' \sim f_{j-1}(\cdot|\phi_i^{(j-1)})$, $y_i' \sim p(y|\phi_i')$.
    Set $\phi_i^{(j)} = \phi_i'$, $y_i^{(j)} = y_i'$ with probability α_j defined in (3.4) and set $\phi_i^{(j)} = \phi_i^{(j-1)}$, $y_i^{(j)} = y_i^{(j-1)}$ otherwise.
    Compute $w_i^{(j)} \propto \frac{p_j(\phi_i^{(j)}, y_i^{(j)})}{p_{j-1}(\phi_i^{(j)}, y_i^{(j)})}$
  end for
  Compute $c_i = \mathbb{1}(\phi_i^{(N_{AIS})} \in \hat{C}_{y_{obs}})$
  Compute $w_i = \prod_{j=1}^{N_{AIS}} w_i^{(j)}$
end for
Compute $W_i = w_i / \sum_{i=1}^{M} w_i$
Return: Estimated coverage $\hat{c}(y_{obs}) = \sum_{i=1}^{M} W_i c_i$

BART is a sum-of-trees model where each tree is regularized by a prior to be a weak learner. Let $Y \in \{0, 1\}$ be a generic binary output and $x \in \mathbb{R}^p$ be a generic input. In the classification setting we wish to infer an unknown function f such that $\Pr(Y = 1|x) = \Phi(f(x))$, with $\Phi(\cdot)$ being the standard normal CDF. The Probit BART model approximates the function $f(x)$ with $h(x)$, a sum of N_T trees, that is

$$f(x) \approx h(x) = \sum_{m=1}^{N_T} g_i(x),$$

where each $g_i(x)$ is given by a separate regression tree. A regularizing prior is imposed on the trees to keep individual tree effects small and prevent overfitting. The posterior distribution over trees is sampled using a Bayesian backfitting procedure. Details can be found in Chipman et al. (2010) and Kapelner and Bleich (2016).

Let $s(y)$ be a potentially high dimensional vector of summary statistics for y. We fit a Probit BART model with $\{c_i, s(y_i)\}_{i=1}^{M}$ as the training data. For $s_{obs} = s(y_{obs})$, we obtain $\pi_B(h(s_{obs})|\{c_i, s(y_i)\}_{i=1}^{M})$, the posterior distribution of $h(s_{obs})$ based on the

fitted model, and we can estimate $c(y_{obs})$ using

$$\hat{c}(y_{obs}) = E_{\pi_B}(\Phi(h(s_{obs})|\{c_i, s(y_i)\}_{i=1}^M)),$$

the (sample) posterior mean of $\Phi(h(s_{obs}))$.

We chose BART for two reasons. First, the tree structure is capable of handling potentially high dimensional input $s(y)$. This is crucial when low dimensional sufficient statistics for $y \in \mathcal{Y}$ are unavailable. Also, since BART is Bayesian, we have a natural way to assess the uncertainty of our estimate. We fit Probit BART models using the R package `bartMachine` (Kapelner and Bleich, 2016).

Algorithm 3.3 Estimation of operational coverage $c(y)$ via regression

Input: Observed data y_{obs}; Summary statistics $s : \mathcal{Y} \to \mathbb{R}^p$; Number of samples J from the approximate posterior; Number of samples M from the generative model; A regression model $\mathcal{M}$.
for i in $1, \ldots, M$ **do**
  Simulate $\phi_i \sim \pi(\phi)$, $y_i \sim p(y|\phi_i)$ and $\underline{\theta}_i = \{\theta_{i1}, \ldots, \theta_{iJ}\}$ where $\theta_{ij} \sim \tilde{\pi}(\theta|y_i)$ for $j = 1, \ldots, J$
  Compute the approximate credible set $\hat{C}_i$ based on $\underline{\theta}_{(i)}$. Set $c_i = \mathbb{1}(\phi_i \in \hat{C}_i)$ and compute the p-dimensional summary statistics $s(y_i)$.
end for
Fit the regression model $\mathcal{M} : c \sim h(s(y))$ to learn the relation between coverage c_i and summary statistics $s(y_i)$ using $\{c_i, s(y_i)\}_{i=1}^M$ as training data.
Return: $\hat{c}(y_{obs})$, the fitted value at $s(y_{obs})$ based on the regression model $\mathcal{M}$.

3.4 2-D Ising Model

Figure 3.1 is a 200 by 200 binary image obtained by thresholding a grey-level image of ice floes from Banfield and Raftery (1992). We illustrate our method on the problem of fitting an Ising model with smoothing parameter $\phi > 0$ and free boundary conditions to these data. The model with free boundary conditions has an intractable likelihood so we approximate it using a solveable model with toroidial boundary conditions. We then calibrate the approximate credible interval for ϕ.

The Ising model is a Markov model on a binary lattice. Let $G = (V, E)$ be a graph with edge set E and vertices V. For each $v \in V$, let $y_v \in \{0, 1\}$ be binary data at v and $y = \{y_v\}_{v \in V} \in \{0, 1\}^{|V|}$ the collection of all y_v. Let $\langle u, v \rangle \in E$ be an edge in G between vertices u, v. Let

$$f(y, E) = \sum_{\langle u,v \rangle \in E} \mathbb{1}(y_u \neq y_v)$$

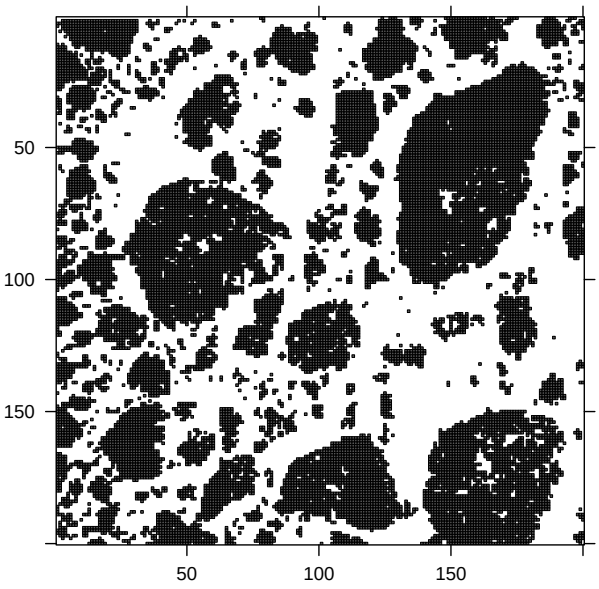


Figure 3.1: Ice floe image from Banfield and Raftery (1992)

be the number of pairs of vertices with disagreeing neighbours on G. In this example, G is a rectangular $N_I \times N_I$ lattice with $N_I = 200$ and free boundary conditions. We denote the graph by $G_F = (E_F, V)$. Free boundary conditions imply that the interior vertices on G have degree 4, edge vertices have degree 3 and corner vertices have degree 2. We consider also the toroidal boundary condition. This means the lattice $G_T = (E_T, V)$ is wrapped onto a torus so all vertices in G have degree 4.

Let $\phi > 0$ be a scalar parameter. The likelihood under free boundary conditions is

$$p_F(y|\phi) = Z_F(\phi)^{-1} \exp(-\phi f(y, E_F))$$

where

$$Z_F(\phi) = \sum_{x \in \{0,1\}^{|V|}} \exp(-\phi f(x, E_F))$$

is the normalizing constant. Similarly, let $Z_P(\phi)$ be the normalizing constant under toroidal boundary conditions. When N_I is large, $Z_F(\phi)$ is computationally intractable. However, $Z_P(\phi)$ is given in closed form in Kaufman (1949).

Following Lee et al. (2019) we impose a uniform prior $\pi(\phi) \propto \mathbb{1}(\phi \in (0, 2))$ on ϕ. The posterior is then

$$\pi(\phi|y) \propto Z_F(\phi)^{-1} \exp(-\phi f(y, E_F)) \mathbb{1}(\phi \in (0, 2)).$$

Although $\pi(\phi|y)$ is doubly intractable, we can use an exchange algorithm (Murray et al., 2006) to perform asymptotically exact inference. Alternatively, we can approximate $Z_F(\phi)$ by $Z_P(\phi)$. Let

$$\tilde{\pi}(\theta|y) \propto Z_P(\theta)^{-1} \exp(-\theta f(y, E_F)) \mathbb{1}(\theta \in (0, 2))$$

denote the approximate posterior. The univariate approximate posterior density $\tilde{\pi}(\theta|y)$ can be evaluated pointwise up to a normalising constant, and the corresponding CDF can be evaluated using numerical integration methods. We compute the equal-tail, 95% approximate credible set $\tilde{C}_{y_{obs}}$ based on $\tilde{\pi}(\theta|y_{obs})$. We find $\tilde{C}_{y_{obs}} = (0.899, 0.913)$. We used an exchange algorithm (Murray et al., 2006) and Algorithm 3.1 to estimate the coverage $c(y_{obs})$ as the ground truth. This Monte Carlo estimate, which we treat as the truth, is $c^{(1)}(y_{obs}) = 0.518$, which is far from the nominal level of 95%. Clearly we should reject this approximation scheme.

We run Algorithm 3.2 to estimate the operational coverage. We initialise the AIS sampler with $M = 1000$ samples $\{\phi_i, y_i\}_{i=1}^M$ with $\phi_i \sim \tilde{\pi}(\phi|y_{obs})$, the *approximate* posterior, and $y_i \sim p_F(y_i|\phi_i)$, the *true* likelihood. We set the number of AIS iterations $N_{AIS} = 60$ with cooling schedule $\beta_j = 1.05^j$ for $j = 1, \ldots, N_{AIS}$, $\gamma_j = 0.02j$ for $j = 1, \ldots 50$ and $\gamma_j = 1$ for $j = 51, \ldots, N_{AIS}$. We use the K-S distance $\delta(y_i, y_{obs}) = |G_{y_i} - G_{y_{obs}}|_\infty$, where G_{y_i} is the CDF of the approximate posterior $\tilde{\pi}(\phi|y_i)$, as the distance metric. Algorithm 3.2 gives $\hat{c}(y_{obs}) = 0.545$ with standard error 0.037 and an effective sample size (ESS) equal 180. In Figure 3.2 we see the estimated operational coverage at iteration $N_{AIS} = 60$ is close to the true value, indicating the effectiveness of AIS here. We repeated the whole experiment 15 times with excellent consistency within uncertainty. The importance sampler in Lee et al. (2019) gives a similar estimate $\hat{c}_{IS}(y_{obs}) = 0.529$ with the threshold $\rho = 0.5$ (varying ρ gave no improvement). However, the ESS is just 22 for a similar runtime.

We also estimate $c(y_{obs})$ using Algorithm 3.3. We initialise Algorithm 3.3 with $M = 1000$ samples $\{\phi_i, y_i\}_{i=1}^M \sim \pi(\phi)p_F(y|\phi)$, the joint prior distribution. Figure 3.2 shows a BART estimate of $c(y)$ as a function of the "covariate" $S(y) = f(y, E_F)$. This gives $\hat{c}(y_{obs}) = 0.565$ with 95% credible interval $(0.420, 0.702)$, in good agreement with the AIS estimate. From Figure 3.4 we see that $S(y_{obs})$ is a relatively "common" sample from the prior predictive distribution of $S(y)$. It confirms that we are not extrapolating the fitted regression model. This is an easy problem for the regression approaches as there is a scalar sufficient statistic. In order to demonstrate that BART is capable of handling high dimensional summary statistics, we also use the same dataset to fit a Probit BART model using the raw binary image, which can be viewed

as a $(N_I)^2 \times 1$ binary vector, as the input. In Chapter 3.7.3 we show the estimated $\hat{c}(y)$ based on the raw binary image is in good agreement with the fitted curve in Figure 3.2 (Right), which is estimated based on the scalar sufficient statistics $S(y)$. This demonstrates the effectiveness of BART when the dimension of summary statistics is high.

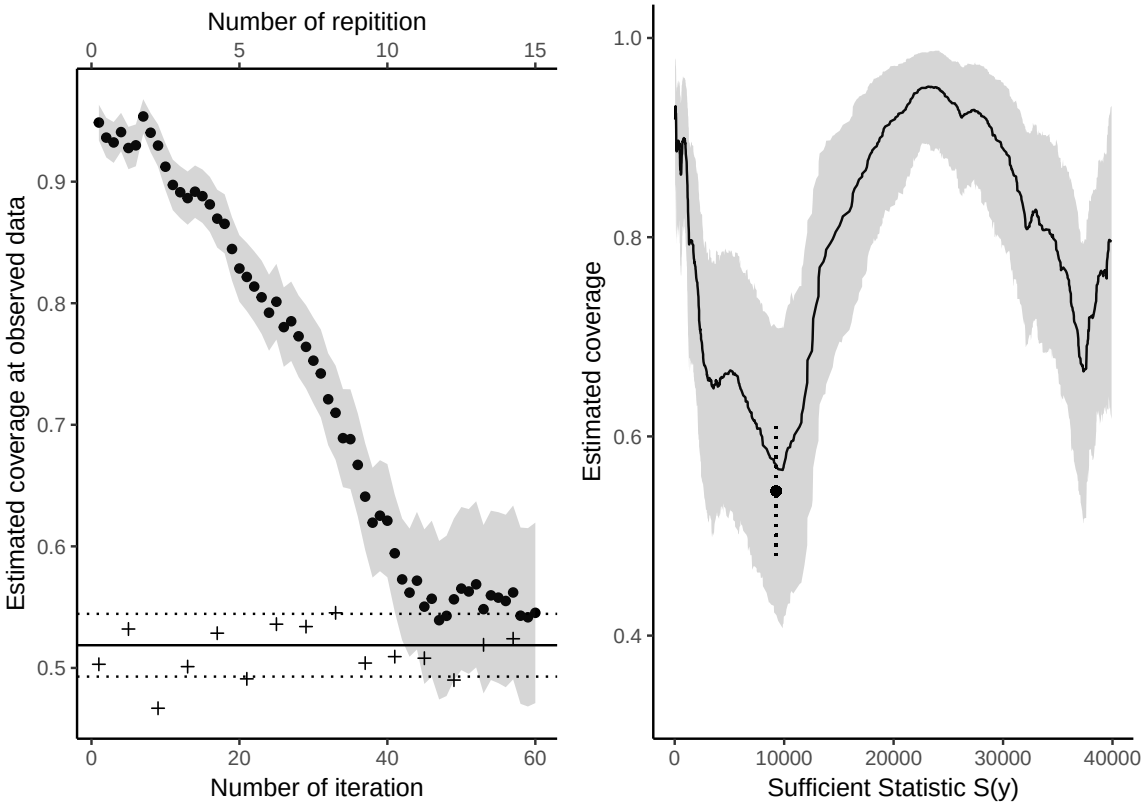


Figure 3.2: Left: Algorithm 3.2; Dots are the estimated $\hat{c}(y_{obs})$ based on the intermediate distribution p_j at each iteration j of the AIS sampler, shaded area is the corresponding 2σ error band. Dashed lines represent the 2σ error bar for the true value $c^{(1)}(y_{obs})$. We see that $\hat{c}(y_{obs})$ converges to $c^{(1)}(y_{obs})$ while the standard error of $\hat{c}(y_{obs})$ increases due to decreasing effective sample size. Crosses correspond to final results for 15 repeats of the algorithm (with arbitrary x-values). Right: Algorithm 3.3; the estimated $c(y)$ as a function of the natural sufficient statistics $s(y)$. Shaded area is the 95% credible band of the estimated values. Vertical dotted segment is the 2σ error bar of $\hat{c}(y_{obs})$ based on Algorithm 3.2.

3.5 Dirichlet Process Random Effect Model

Lee et al. (2019) show that their approach works well for calibration of a completely collapsed MCMC algorithm for partition structure in a Dirichlet process. However it

is easy to find more challenging problems on which their approach performs poorly. In this section, we use Algorithm 3.3 to estimate the realised coverage $c(y_{obs})$ for a dataset and an approximation procedure on which the methods of Lee et al. (2019) fail. We show that credible sets based on the Laplace approximation of the marginal likelihood are unreliable in this example.

Our dataset has the classical format of a complete design, with five categorical variables, including `Treatment` ($N = 12$ levels), and four block variables, `B1` (with $q = 3$ levels), `B2` ($r = 2$ levels), `B3` (two levels) and `B4` (seven levels) so that we have $n = 1008$ observations. Let the data vector be $y = (y_1, \ldots, y_n) \in \mathbb{R}^n$. In our example we fit a hierarchical model with known fixed effects `B1` $*$ `B2`. Let X be the $n \times p$ design matrix for the fixed effects ($p = 6$ here).

We take a Dirichlet process prior for the hierarchy of random effects in our hierarchical model. Our aim here is not to develop new models but to illustrate calibration. The model is similar in structure to the model considered in Malsiner-Walli et al. (2018), differing mainly in the choice of partition model, a Chinese Restaurant Process (CRP) in our case. We suppose the scientist wants to cluster the treatments into groups with similar effects. Each treatment has a vector of random effects, so the object here is to estimate a partition of the $N = 12$ random effect vectors for treatments. The output of the uncalibrated analysis is an approximate HPD credible set for the unknown partition of treatment effects. We calibrate this credible *set* of partitions here instead of a credible interval, reflecting the ease of application of our methods in more general settings.

Let $\mathcal{A} = \{1, \ldots, N\}$ give the distinct levels of `Treatment` and let A be the $n \times 1$ covariate vector with $\mathrm{A}_i \in \mathcal{A}$ giving the level of `Treatment` in the i'th observation. Similarly, let B1, B2 be two $n \times 1$ covariate vectors giving the levels of `B1`, `B2`. Let $S = \{S_1, \ldots, S_K\}$ be a partition of $\mathcal{A}$ and let S be a $n \times 1$ unobserved categorical covariate giving the grouping, so that $\mathrm{S}_i = k$ means $\mathrm{A}_i \in S_k$. These are the levels of cluster. The interaction between `B1` and `Treatment` is a random effect so we have a vector of random effects $\eta_j^A \in \mathbb{R}^q$, $\eta_j^A \overset{i.i.d.}{\sim} N(0, \Sigma_A)$ for $j = 1, \ldots, N$ for the different levels of A and another offset vector of random effects $\eta_k^S \in \mathbb{R}^q$, $\eta_k^S \overset{i.i.d.}{\sim} N(0, \Sigma_S)$ for $k = 1, \ldots, K$. Let Z be a $n \times q$ matrix of indicators for the levels of `B1`. Denote by x_i, z_i the ith row of X and Z, let $\beta = (\beta_1, \ldots, \beta_p)$ be the vector of fixed effects, and let $\epsilon_i \overset{i.i.d.}{\sim} N(0, \sigma^2)$ for $i = 1, \ldots, n$. The observation model is

$$y_i = x_i\beta + z_i\eta^S_{\mathrm{S}_i} + z_i\eta^A_{\mathrm{A}_i} + \epsilon_i, \quad i = 1, \ldots, n$$

with likelihood $p(y|\psi)$ for parameter $\psi = (\beta, \eta^A, \eta^S, \Sigma_A, \Sigma_S, \sigma^2), \psi \in \Omega_S$. The parameter space Ω_S of ψ depends on the partition S, as the dimension of η^S is determined by S.

The partition S is an unknown parameter in the posterior with a Chinese Restaurant Process (CRP) prior

$$\pi(S) = \frac{\alpha^K \Gamma(\alpha) \prod_{k=1}^K \Gamma(|S_k|)}{\Gamma(\alpha + n)}$$

for $S \in \mathcal{P}$, where α is a model parameter, $|S_k|$ is the number of elements in the set S_k and $\mathcal{P}$ is the space of partitions. We took $\alpha = 1$. Our setup is equivalent to taking a Dirichlet process prior $G_S \sim \mathrm{DP}(\alpha, H)$ with base distribution $H = N(0, \Sigma_S)$ on the random effects due to the partition of N levels of `Treatment`. The joint prior $\pi(\psi, S)$ is

$$\pi(\psi, S) = \pi(\eta^S | S, \Sigma_S)\pi(\beta, \eta^A, \Sigma_A, \Sigma_S, \sigma^2)\pi(S)$$

with $\pi(\eta^S | S, \Sigma_S) = \prod_{k=1}^K N(\eta_k^S; 0, \Sigma_S)$ and

$$\begin{aligned}\pi(\beta, \eta^A, \Sigma_A, \Sigma_S, \sigma^2) = N(\beta; 0, \sigma_\beta^2 I_p) \prod_{j=1}^N N(\eta_j^A; 0, \Sigma_A) \\ \times \mathrm{IW}(\Sigma_A; \nu_A, V_A)\mathrm{IW}(\Sigma_S; \nu_S, V_S)\mathrm{IG}(\sigma^2; \alpha_\sigma, \beta_\sigma).\end{aligned}$$

Here $\sigma_\beta, \nu_A, V_A, \nu_S, V_S, \alpha_\sigma, \beta_\sigma$ are prior hyperparameters. The joint posterior distribution is then

$$\pi(\psi, S|y) \propto p(y|\psi)\pi(\psi, S). \tag{3.6}$$

Estimation of S by sampling the joint posterior $\pi(\psi, S|y)$ using MCMC is a variable dimension problem. It is convenient to work with the marginal posterior $\pi(S|y) \propto p(y|S)\pi(S)$, where

$$p(y|S) = \int p(y|\psi)\pi(\psi|S)d\psi \tag{3.7}$$

is the marginal likelihood. However, $p(y|S)$ is computationally intractable. Suppose we approximate it with a Laplace approximation. How much harm does this do? Let b_S be the Bayesian Information Criterion (BIC) of the hierarchical model with a given partition S. Recall that BIC can be interpreted as a Laplace approximation of the log marginal likelihood that only uses terms depending on the sample size. If the partition is S, then

$$\tilde{p}_{BIC}(y|S) = \exp(-b_S/2)$$

approximates the marginal likelihood $p(y|S)$. This also corresponds to a choice of unit information priors on model parameters ψ (Kass and Raftery, 1995; Raftery,

1999) and can be seen as part of the approximation we are calibrating. Packages for computing the BIC for complex models are available. We use the R-package `lme4` (Bates et al., 2015). The approximate posterior for S is then

$$\tilde{\pi}(S|y) \propto \tilde{p}_{BIC}(y|S)\pi(S).$$

We sample $\tilde{\pi}(S|y_{obs})$ and construct an approximate credible set for S using standard Metropolis-Hasting MCMC. Can we trust this credible set?

We apply Algorithm 3.3 to the problem. For $i = 1, \ldots, M$ we sample partitions $S^{(i)} \sim \pi(S)$, $\psi^{(i)} \sim \pi(\cdot|S^{(i)}), \psi^{(i)} \in \Omega_S$ and $y^{(i)} \sim p(\cdot|\psi^{(i)}), y^{(i)} \in \mathbb{R}^n$. Low dimensional sufficient statistics are not available, so we try two sets of relatively high dimensional summary statistics. Covariates `B3` and `B4` do not appear in the model, so we average the observations with the same `Treatment`, `B1` and `B2` levels. Recall that we denote y_i the ith entry of y. Denote by $\bar{y}^{(i)}_{jkl}$ the mean of observations $\{y^{(i)}_{i'} : i' = 1, \ldots, n; \mathrm{A}_{i'} = j; \mathrm{B1}_{i'} = k; \mathrm{B2}_{i'} = l\}$ and let

$$T(y^{(i)}) = \{\bar{y}^{(i)}_{jkl}\}, \quad j = 1, \ldots, N; \quad k = 1, \ldots, r; \quad l = 1, \ldots, q$$

denote these summary statistics, with $N = 12, r = 2$, $q = 3$ and dimension $Nrq = 72$. Tree-based BART has no difficulty with summary statistics of this dimension.

As noted at the end of Chapter 3.2, level-labels are exchangeable so we can permute each of the data vectors $y^{(i)}, i = 1, \ldots, M$ and map them into a "tighter" subregion of $\mathbb{R}^n$. Regression is easier on the subregion where the $y^{(i)}$-values are more densely packed. Let $\sigma \in \mathcal{P}_R$ be the set of relabeling permutations for which $c(y_\sigma) = c(y)$. In our setting with three categorical covariates `Treatment`, `B1` and `B2` with $N = 12$, $q = 3$ and $r = 2$ levels respectively, and a complete design, the number of "legal" permutations of the collapsed data T is $|\mathcal{P}_R| = N!q!r!$.

We use this permutation invariance to define a second coarser set of summary statistics. Consider the $N = 12$ treatment levels. For $i = 1, \ldots, M$, let $H_N(y^{(i)}) = \{\bar{y}^{(i)}_j\}_{j=1}^N$, where $\bar{y}^{(i)}_j$ is the sample mean of $\{y^{(i)}_{i'} : i' = 1, \ldots, N; \mathrm{A}_{i'} = j\}$. Take the permutation $\sigma_N \in \mathcal{P}_R$ such that $H_N(\sigma_N(y^{(i)})) = \{\bar{y}^{(i)}_{(1)}, \ldots, \bar{y}^{(i)}_{(N)}\}$ matches the order statistics of $\{\bar{y}^{(i)}_j\}_{j=1}^N$. Let $H_q(\sigma_q(y^{(i)}))$ and $H_r(\sigma_r(y^{(i)}))$ give the corresponding sorted averages for `B1` and `B2`. Let

$$H(y^{(i)}) = (H_N(\sigma_N(y^{(i)})), H_q(\sigma_q(y^{(i)})), H_r(\sigma_r(y^{(i)})))$$

denote this collection of the $p = 17$ order statistics. We take $H(y)$ as a second set of summary statistics.

Table 3.1: Estimates of $c(y_{obs})$ and the corresponding 95% credible interval based on Probit BART models.

Model	$\hat{c}(y_{obs})$	95% Credible Interval
$M1$	0.262	(0.065,0.549)
$M2$	0.308	(0.124,0.552)
$M2_1$	0.285	(0.067,0.564)
$M2_2$	0.322	(0.086,0.618)
$M2_3$	0.347	(0.095,0.673)
$M2_4$	0.270	(0.056,0.565)

We simulate $M = 810$ pairs $\{c^{(i)}, y^{(i)}\}_{i=1}^{M}$ of training data following Algorithm 3.3. We fit two probit BART models $M1 : c^{(i)} \sim T(y^{(i)})$ and $M2 : c^{(i)} \sim H(y^{(i)})$ i.e. we fit two models using $T_i = T(y^{(i)})$ and $H_i = H(y^{(i)})$ as summary statistics. The estimated values $\hat{c}(y_{obs})$ at y_{obs} are given in Table 2.1. Estimates based on the different summary statistics $M1$ and $M2$ agree. In order to further test the robustness of our method, we partition the full synthetic dataset $\{c^{(i)}, y^{(i)}\}_{i=1}^{M}$ into four equal-size subsets and fit a Probit BART model using formula $M2$ to each subset. Let $M2_1, M2_2, M2_3$ and $M2_4$ label these models. Fitted values at y_{obs} of the four smaller models are all in line with fitted values determined on the full training set, with wider credible intervals as their training sets are smaller. This suggests that our estimate of coverage is robust. The estimated coverage $\hat{c}(y_{obs})$ is far lower than the nominal level $\alpha = 0.95$, so we conclude that the approximate marginal likelihood $\tilde{p}_{BIC}(y|S)$ is a poor approximation here, and the approximate credible set should not be trusted.

3.6 Conclusion and further discussions

In this chapter we give a computational framework for estimating the error in coverage due to approximations made in carrying out Bayesian inference. We provide improved estimators for the calibration problem defined in Lee et al. (2019). We demonstrate their effectiveness by diagnosing poor approximate coverage in two examples. We note that the quality of the approximate coverage may depend on the data, so one approximation scheme may work well for some data sets and not others.

Our assumptions in this chapter are similar to those of ABC (we can simulate the prior and observation model efficiently). A vanilla application of ABC would also give a natural though in general inefficient estimator for $b(y)$ in Equation 3.1. In our setup, we have help from the approximate posterior $\tilde{\pi}(\theta|y_{obs})$ and so our AIS method for

estimating coverage can be seen as a hybrid of the two approximation schemes, where $\tilde{\pi}(\theta|y_{obs})$ is used as a sensible proposal distribution of the AIS sampler. Our BART based regression uses the same simulation stage as ABC, but a regression model is used to estimate a probability function over data space $\mathcal{Y}$, in a manner similar to the model selection procedure in Raynal et al. (2019).

The two coverage estimators we suggest, Algorithm 3.2 based on AIS and Algorithm 3.3 based on BART, have complimentary strengths. Algorithm 3.2 with the AIS sampler uses "local" information provided by $\tilde{\pi}(\theta|y_{obs})$. Algorithm 3.3 with BART can more easily estimate the global performance of an approximation scheme over the data space $\mathcal{Y}$, and does not require careful specification of summary statistics or related distance measures. Finally we stress that what we are offering is a consistency check: a good outcome (i.e. an estimated coverage close to α) is a necessary but not a sufficient condition for us to trust the original estimated credible set.

Our methods rely on the assumption that the model is correctly specified. We would like to stress that the whole setup is well defined irrespective of model misspecification. Estimating $b(y)$ or $c(y)$ is a problem in probability – it measures the difference between the exact posterior and the approximate posterior. However, model misspecification does have an impact on the problem: if the model is misspecified, then the true data would be like an outlier, because the true data would likely to be in a part of data-space we don't often visit with our simulated $\{\phi_i, y_i\}$ pairs from the joint prior distribution $\pi(\phi)p(y|\phi)$. This makes estimation harder. In the regression approach, such wild extrapolation may also lead to unstable or unreliable estimates.

We allow no model misspecification in definition of coverage. This is how the level of a credible set is defined in all Bayes inference. It would be interesting to know if we are covering nature's true parameter. But we don't address this in our setup: We estimate the change in coverage as we move from exact to approximate posterior, not change as we move from nature's generative model to the misspecified model.

3.7 Appendix of Chapter 3

3.7.1 Proof of Theorem 3.3.1

Here we give proof of Theorem 3.3.1.

Theorem 3.3.1. *If $\gamma_{N_{AIS}} = 1$, $\delta(y, y_{obs})$ is twice differentiable with respect to $y \in \mathcal{Y}$ and there exists $L > 0$ such that $p(y_{obs}|\phi) < L$ for all $\phi \in \Omega$, then $\hat{c}(y_{obs})$ is a consistent estimator of the true realized coverage achieved by $\hat{C}_{y_{obs}}$ as $K \to \infty$ and $\beta_{N_{AIS}} \to \infty$.*

Proof. Since $\hat{c}(y_{obs})$ is a self-normalised importance sampling estimator for the quantity

$$\Pr_{p_{N_{AIS}}}(\phi \in \hat{C}_{y_{obs}}) = \int \mathbb{1}(\phi \in \hat{C}_{y_{obs}}) p_{N_{AIS}}(\phi) d\phi,$$

where $p_{N_{AIS}}(\phi)$ is the marginal distribution of $p_{N_{AIS}}(\phi, y)$, we have that as $K \to \infty$,

$$\hat{c}(y_{obs}) \xrightarrow{p} \Pr_{p_{N_{AIS}}}(\phi \in \hat{C}_{y_{obs}}). \tag{3.8}$$

Under the assumptions that $\gamma_{N_{AIS}} = 1$, $\delta(y, y_{obs})$ is twice differentiable w.r.t. $y \in \mathcal{Y}$ and $p(y_{obs}|\phi)$ is bounded w.r.t. ϕ, it is straightforward to show the (marginal) target density function $p_{N_{AIS}}(\phi)$ converges pointwisely to the true posterior density $\pi(\phi|y_{obs})$ as $\beta_{N_{AIS}} \to \infty$ using Laplace approximation. Hence by Scheffé's lemma, $p_{N_{AIS}}(\phi)$ converges in distribution to $\pi(\phi|y_{obs})$. So we must have

$$\Pr_{p_{N_{AIS}}}(\phi \in \hat{C}_{y_{obs}}) \longrightarrow \Pr(\phi \in \hat{C}_{y_{obs}}|Y = y_{obs}), \tag{3.9}$$

as $\beta_{N_{AIS}} \to \infty$, where $\Pr(\phi \in \hat{C}_{y_{obs}}|Y = y_{obs})$ is the true posterior probability (i.e. the true realized coverage achieved by $\hat{C}_{y_{obs}}$). Combining (3.8) and (3.9), we conclude that $\hat{c}(y_{obs})$ is a consistent estimator for the realised coverage achieved by $\hat{C}_{y_{obs}}$. □

3.7.2 Comparing efficiency of Algorithm 3.2 and the Importance sampling method in Lee et al. (2019)

We use a toy example to compare the performance of the Algorithm 3.2 in our paper (we refer to as the AIS sampler) and the Algorithm 2 in Lee et al. (2019) (we refer to as the IS sampler) as the dimension of the observation y_{obs} varies. We will show the performance of the IS sampler deteriorates quickly as the dimension of y_{obs} grows.

Suppose we draw the parameter ϕ form the prior $\phi \sim N(0, 1)$, and i.i.d. observations $\{y_1, ..., y_d\} \sim N(y; \phi, 1)$. Then the exact posterior is

$$\pi(\phi|y_1, ..., y_d) \sim N\left(\frac{\sum_{i=1}^{d} y_i}{d+1}, \frac{1}{d+1}\right)$$

Suppose we replace the exact likelihood by a tempered likelihood $\tilde{p}(y|\phi) = N(y; \phi, 1)^v$ for some $v > 0$, then the corresponding approximated posterior

$$\tilde{\pi}(\phi|y_1, ..., y_d) \propto N(\phi; 0, 1) \prod_{i=1}^{d} N(y_i; \phi, 1)^v,$$

that is

$$\tilde{\pi}(\phi|y_1, ..., y_d) \sim N\left(\frac{v\sum_{i=1}^{d} y_i}{dv+1}, \frac{1}{dv+1}\right)$$

Note that when $v = 1$, the approximate posterior is exact. Since both the exact and approximate posterior are Normally distributed, the true value of of the coverage $b(y) = \Pr(\phi \in \tilde{C}_Y | Y = y)$, where $\tilde{C}_Y$ is an approximate level α credible set, can be exactly computed. In this section we follow the convention and set $\alpha = 0.95$

We estimate $b(y)$ using IS and AIS samplers. For both algorithms, we use the the Euclidean distance as the distance function. For AIS sampler, we set $N_{AIS} = 20$, $\alpha_j = 0.05j, \beta_j = 0.2j$ for $j = 1, \ldots, N_{AIS}$. For IS sampler, we set tolerance level $\rho = 0.5$. We start from a scalar observation y (i.e. $d = 1$). Both algorithms work well when $0 < v < 1$. However, the IS sampler performs poorly when $v > 1$, i.e. the approximate likelihood $\tilde{p}(y|\phi)$ is more concentrated than the true likelihood. We set $v = 2$ and estimate $b(y)$ at 100 equidistant points ys over the interval $(-3, 3)$ using both IS and AIS samplers with $M = 300$ samples. We report the estimated values in Figure 3.3. We can see that $b_{IS}(y)$, the IS sampler estimate of the true coverage, has large variability for almost all y values, while $b_{AIS}(y)$ performs much better and is much closer to the true values.

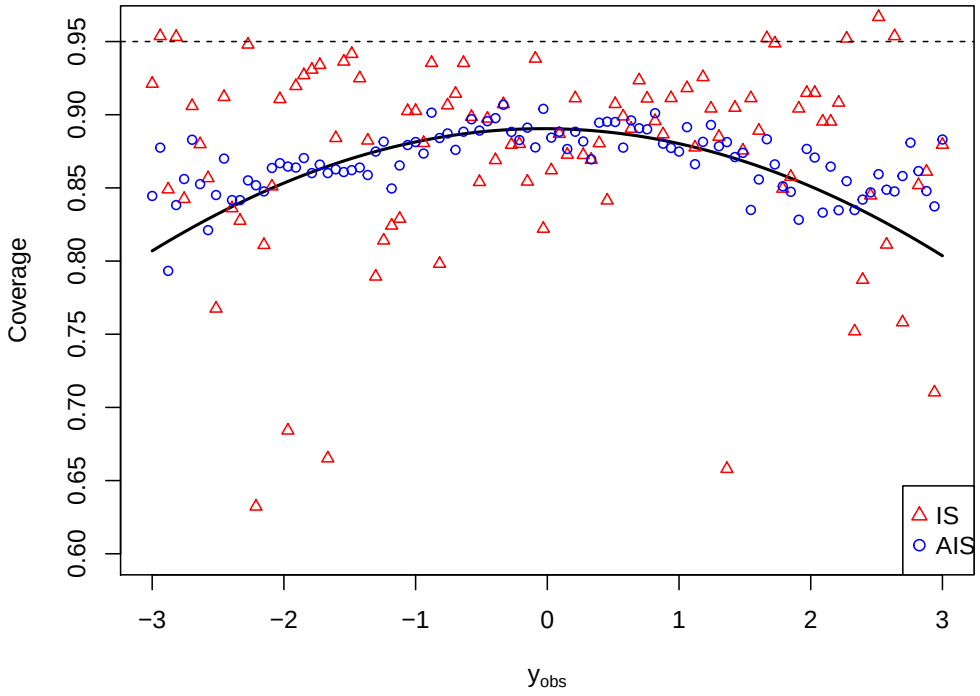


Figure 3.3: Coverage $b(y)$ as a function of y. The solid line corresponds to the true coverage $b(y)$, the IS and AIS estimates are represented by red and blue points. The dashed line is the nominal coverage $\alpha = 0.95$

We also extend the comparison study to higher dimensions ($d > 1$). For each $d = 3, 6, 9, .., 30$, we sample $M = 100$ synthetic parameters $\{\phi_i\}_{i=1}^M$ from the prior $N(\phi; 0, 1)$ and $y^{(i)} = \{y_1^{(i)}, ..., y_d^{(i)}\} \overset{i.i.d.}{\sim} N(y; \phi_i, 1)$ for $i = 1, ..., M$ as synthetic observations. We compute the corresponding coverage estimates $b_{IS}(y^{(i)})$ and $b_{AIS}(y^{(i)})$ using the two algorithms for each $i = 1, ..., M$, then report the MSE $\bar{R}_{IS} = \frac{1}{M}\sum_{i=1}^M (b_{IS}(y^{(i)}) - b(y^{(i)}))^2$ and $\bar{R}_{AIS} = \frac{1}{M}\sum_{i=1}^M (b_{AIS}(y^{(i)}) - b(y^{(i)}))^2$, and the average effective sample size (ESS) for both algorithms. We set $v = 5$ (i.e. the approximate posterior is heavily under-dispersed with respect to the true posterior) and initiate both algorithms with $M = 1000$ samples. The results are reported in Table 3.2. We can see that as d increases, AIS sampler outperforms the IS sampler in both MSE and average ESS. This supports the effectiveness of our algorithm.

d	$\bar{R}_{IS}$	$\bar{R}_{AIS}$	ESS_{IS}	ESS_{AIS}
3	3.77e-02	**9.96e-03**	3.59e+01	**4.74e+02**
6	4.10e-02	**3.11e-03**	3.64e+01	**3.38e+02**
9	4.62e-02	**9.41e-04**	3.62e+01	**5.53e+02**
12	3.72e-02	**1.69e-03**	3.26e+01	**4.98e+02**
15	4.21e-02	**2.13e-03**	3.42e+01	**5.81e+02**
18	3.77e-02	**3.39e-03**	2.92e+01	**5.49e+02**
21	3.88e-02	**4.29e-03**	3.54e+01	**6.73e+02**
24	4.12e-02	**5.05e-03**	2.90e+01	**7.26e+02**
27	4.54e-02	**6.44e-03**	3.60e+01	**6.41e+02**
30	4.50e-02	**7.03e-03**	3.38e+01	**6.87e+02**

Table 3.2: The average MSE and ESS of both Algorithms over $N = 100$ repetitions under different dimensions d. Lower MSE and higher ESS are in boldface.

3.7.3 Handling high dimensional summary statistics using BART

In Chapter 3.4, we fit a BART model using the natural sufficient statistics $S(y) = f(y, E_F)$ as the input. We plot the KDE estimate of the prior predictive distribution of $S(y)$ in Figure 3.4. To demonstrate that BART is able to handle high dimensional summary statistics, we use the same dataset $\{\phi_i, y_i\}_{i=1}^M$ in Chapter 3.4 to fit a BART model using the raw 200×200 binary image (which can be viewed as a 40000×1 binary vector) as the input. In Figure 3.5, we report $\hat{c}(y)$ estimated by the BART model with $S(y)$ as the input (left) and the BART model with the raw image as the input (right). We plot the fitted $\hat{c}(y)$ as a function of $S(y)$ for both models. Comparing to the BART trained by $S(y)$, the BART trained by the raw image learned a similar pattern and

reproduces a curve similar to the left with greater uncertainty. This demonstrates that BART is capable of handling high dimensional summary statistics.

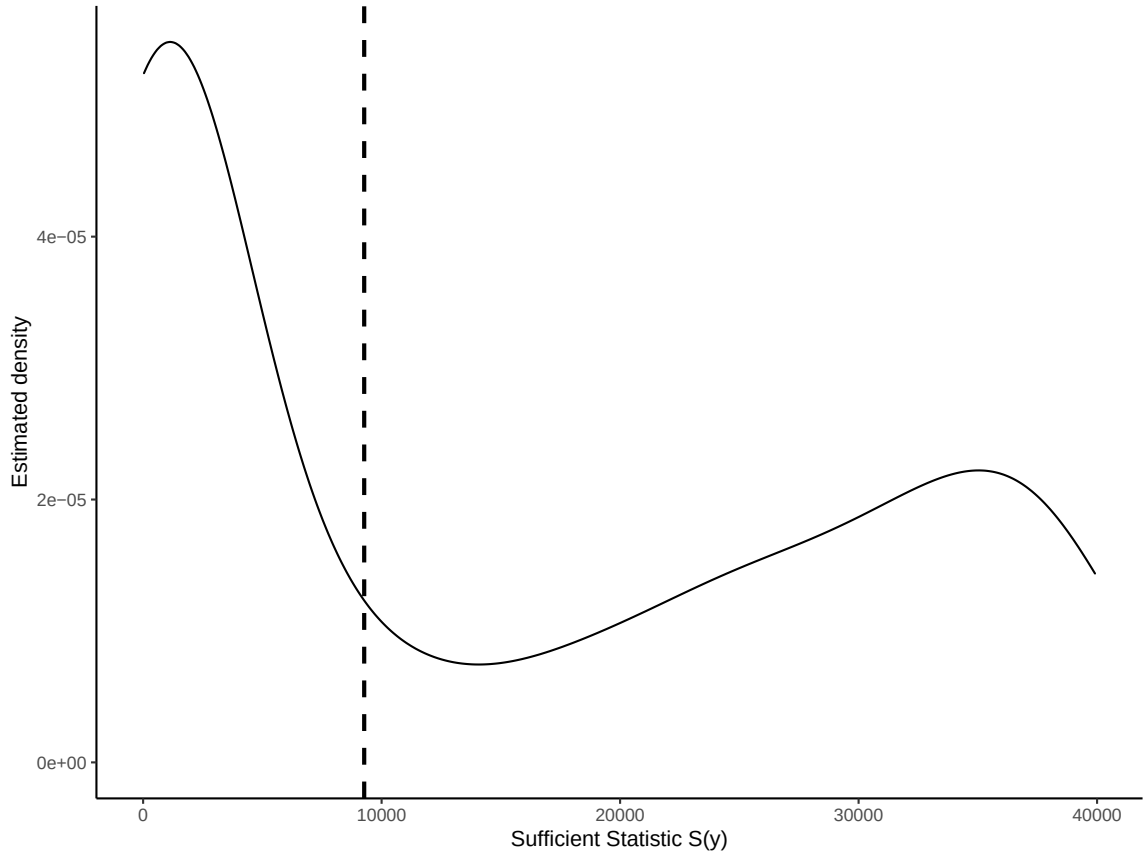


Figure 3.4: Prior predictive of $S(y)$ estimated using KDE with $N = 1000$ samples. The dashed line indicates the location of $S(y_{obs})$.

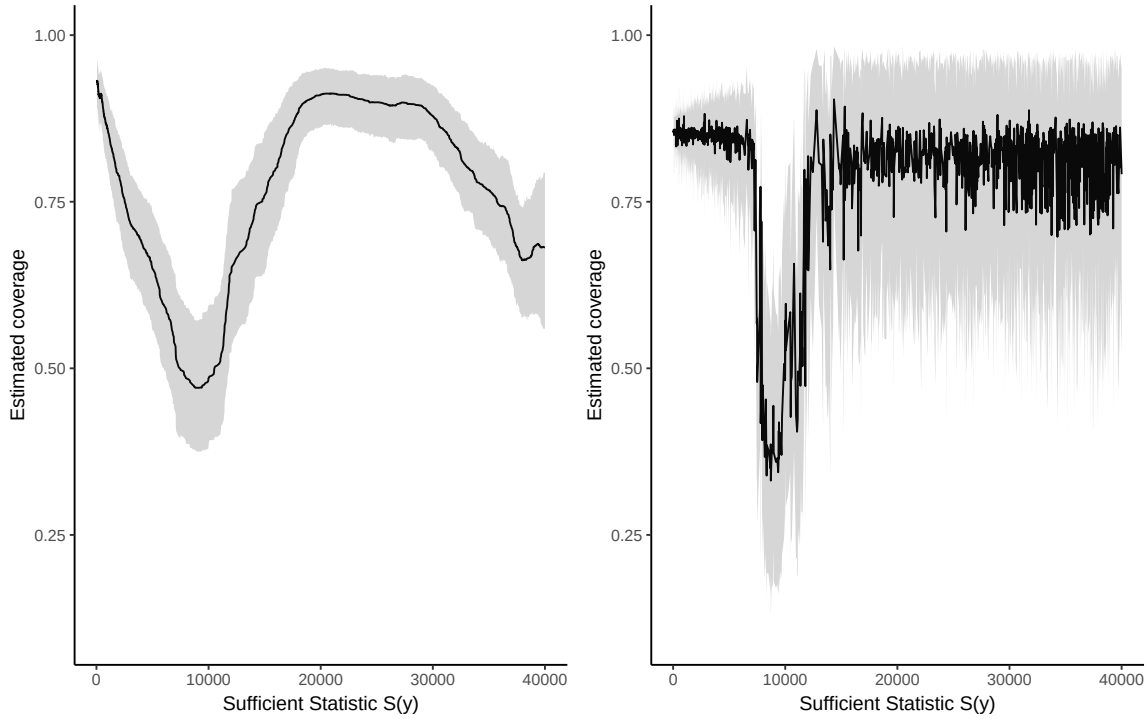


Figure 3.5: Estimated $\hat{c}(y)$ as a function of the sufficient statistics $S(y)$. Left: BART trained by the sufficient statistics. Right: BART trained by the full 200×200 image. Grey band indicates the 95% credible interval.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

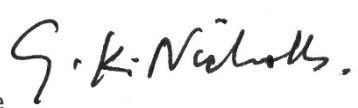
Title of Paper	Calibrated approximate Bayesian inference
Publication Status	√ Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Xing, Hanwen, Geoff Nicholls, and Jeong Lee. "Calibrated approximate Bayesian inference." *International Conference on Machine Learning*. PMLR, 2019.

Student Confirmation

Student Name:	Hanwen Xing		
Contribution to the Paper	I contributed to the design and implementation of the proposed methods and led the drafting of the paper. In addition, I also conducted all the simulated and real-world experiments.		
Signature 		Date	Sept 29 2022

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Prof Geoff Nicholls		
Supervisor comments		
Signature 	Date	29-09-22

This completed form should be included in the thesis, at the end of the relevant chapter.

Chapter 4

Distortion estimate for approximate Bayesian inference

Coverage estimation in the last chapter essentially estimates the posterior expectation (conditioned on y_{obs}) of a scalar function that depends on the approximate posterior. This scalar summary may not contain enough diagnostic details of the approximation error in the approximate posterior. For example, both underdispersion and bias in the approximate posterior mode can result in the operational coverage of an approximate credible set being lower than its nominal coverage. In this chapter, we give a visual diagnostic tool for diagnosing marginals of approximate posteriors conditioned on the observed data y_{obs}. We estimate "distortion maps" that act on *univariate marginals* of the approximate posterior to move them closer to the exact posterior marginals, without actually recourse to the exact posterior. The proposed distortion map contains detailed and easily-interpreted diagnostic information about the approximation error in the approximate marginal CDF. Intuitively speaking, it reflects how the exact marginal posterior is distorted by the approximate. In addition, we show that our proposed distortion map is able to identify approximation error that is overlooked by existing diagnostic tools using a real world example.

In the last chapter, we do not make any assumption on the support of the approximate and exact posterior distributions. However, in this chapter, we focus on diagnostics for univariate parameters, or scalar functions of multivariate parameters (We will discuss bivariate and multivariate extensions of the proposed method in Chapter 4.3.2 and 4.9). In order to emphasis this distinction, our notation in this chapter takes the parameter $x \in \mathbb{R}$. We still use y to denote generic observed values. We assume the parameter is continuous, but this is not essential. Similar to the previous chapter, we make the assumptions that 1) we can efficiently sample from both the prior distribution and the observation model, and 2) the approximation

scheme we are testing is itself reasonably computationally efficient, as we may need to call the approximation algorithm repeatedly.

4.1 Introduction and background

In this chapter, we introduce and estimate a family of "distortion maps"

$$D_y : [0,1] \longrightarrow [0,1],\ y \in \mathcal{Y}$$

which act on an univariate marginal of the multivariate approximate posterior conditioned on a generic data point y. The exact distortion map D_y transports an approximate marginal posterior CDF $G_y(x)$ onto the corresponding exact marginal posterior CDF $F_y(x)$. The distortion map D_y is a function of $G_y(x)$ defined for each $y \in \mathcal{Y}$ by the relation $F_y = D_y \circ G_y$ given in Eqn. 4.1 below. The distortion map $D_{y_{obs}}$ at the observed data y_{obs} contains easily-interpreted diagnostic information about the approximation error in the approximate marginal CDF $G_{y_{obs}}$. If the distortion map $D_{y_{obs}}$ differs substantially from the identity map, then the magnitude and location of any distortion is of interest.

A reliable estimate of $D_{y_{obs}}$ must be hard to achieve, as it maps to the exact posterior CDF $F_{y_{obs}}$. Our goal is to estimate a map $\hat{D}_{y_{obs}}$ such that the "corrected" univariate CDF $\hat{F}_{y_{obs}} = \hat{D}_{y_{obs}} \circ G_{y_{obs}}$ is *asymptotically closer in KL-divergence* to $F_{y_{obs}}$, but not necessarily equal to it. If $G_{y_{obs}}$ is far from $F_{y_{obs}}$ in KL divergence then it must be easy to find a CDF $\hat{F}_{y_{obs}}$ which is closer to $F_{y_{obs}}$ than $G_{y_{obs}}$ was. It follows that if $\hat{D}_{y_{obs}}$ differs significantly from the identity map then the approximation defining $G_{y_{obs}}$ is poor. In this chapter, we aim to estimate this distortion map, which contains diagnostically useful information, without sampling or otherwise constructing the exact posterior.

The map $D_y,\ y \in \mathcal{Y}$ may be represented in several equivalent ways, with varying convenience depending on the setting. We can parameterize it as a transport map from the approximate density to the exact density, or a mapping between the CDF's, or a function of the approximate random variable itself. Since we are not interested in approximating the true posterior, but in checking an existing approximation for quality, we choose to parameterize D_y as a mapping between CDF's, and estimate how the exact CDF is distorted by approximation errors for each marginal of the joint posterior distribution. This has some benefits and some disadvantages.

On the plus side, the mapping from the CDF of the approximate posterior to the CDF of the exact posterior is an invertible mapping between functions of domain

and range $[0, 1]$. This resembles a copula-like construction (see in particular Eqn. 4.8) and doesn't change from one problem to another, making it easier to write generic code. There is also a simple simulation based fitting scheme, Algorithm 4.1, to estimate the map. On the downside, we restrict ourselves to diagnostics for low-dimensional marginal distributions. However, multivariate posterior distributions are in practise almost always summarised by point estimates, credible intervals and univariate marginal densities, and the best tools we have seen, Prangle et al. (2014) and Talts et al. (2020), also focus on univariate marginals. We extend our diagnostics to bivariate marginal distributions in Chapter 4.3.2 and give examples of the bivariate "distortion surfaces" in examples below. We also discuss an alternative multivariate extension strategy in Chapter 4.9.

4.2 Distortion map

Let $\pi(\cdot)$ be the prior distribution of a scalar parameter $x \in \mathcal{X} \subseteq \mathbb{R}$ and let $p(\cdot|x)$ be the likelihood function of generic data $y \in \mathcal{Y}$. Let y_{obs} be the observed data value. Given generic data y, let $F_y(x)$ be the CDF of the exact posterior $\pi(x|y) \propto \pi(x)p(y|x)$. In practice these densities will the marginals of some multivariate parameter of interest. For $X \sim \pi(\cdot)$ and $Y|(X = x) \sim p(\cdot|x)$, we have $X|(Y = y) \sim \pi(\cdot|y)$. We assume $X|(Y = y)$ is continuous, so that $F_y(x)$ is continuously differentiable and strictly increasing with x at every $y \in \mathcal{Y}$. The case of discrete X is a straightforward extension. Let $\tilde{\pi}(x|y)$ be a generic approximate posterior on $\mathcal{X}$ with CDF $G_y(x)$. We define a distortion map $D_y : [0, 1] \longrightarrow [0, 1]$ such that for each $x \in \mathcal{X}$ and each $y \in \mathcal{Y}$

$$D_y(G_y(x)) = F_y(x). \tag{4.1}$$

The distortion map D_y is a strictly increasing function mapping the unit interval to itself and, as Prangle et al. (2014) point out, is itself the CDF of $Q = G_y(X)$ when $X \sim F_y$. To see this, observe that since $F_y(X) \sim U(0, 1)$ we have $D_y(Q) \sim U(0, 1)$ from Eqn. 4.1, and this is necessary and sufficient for $Q \sim D_y$.

Denote by

$$d_y(q) = \frac{d}{dq} D_y(q)$$

the density associated with the CDF D_y so that $Q \in [0, 1]$ is a random variable with probability density $d_y(q)$ for $q \in [0, 1]$. Since $\pi(x|y) = \frac{d}{dx} F_y(x)$, we have from Eqn. 4.1,

$$\pi(x|y) = d_y(G_y(x))\tilde{\pi}(x|y), \tag{4.2}$$

connecting the two posterior densities.

We seek an estimate, $\hat{D}_{y_{obs}}$, of the true distortion map at the data y_{obs}, or equivalently an estimate, $\hat{d}_{y_{obs}}$, of its density. Other authors, focusing on constructing new posterior approximations, have considered related problems, either without the distortion-map representation, or in an ABC setting. However, since we seek a diagnostic map, not a new approximate posterior, it is not necessary to estimate D_y exactly, but simply to find an approximate $\hat{D}_y$ that moves G_y towards F_y as measured by KL-divergence. The recalibrated CDF

$$\hat{F}_{y_{obs}}(x) = \hat{D}_{y_{obs}}(G_{y_{obs}}(x)) \tag{4.3}$$

should be a better approximation (in KL-divergence) to $F_{y_{obs}}$ than $G_{y_{obs}}$ *even if both are bad*. The same argument applies at the level of densities. From Eqn. 4.1, the recalibrated density

$$\hat{\pi}(x|y) = \hat{d}_y(G_y(x))\tilde{\pi}(x|y)$$

must improve the original approximation $\tilde{\pi}(x|y)$. If our original approximation $\tilde{\pi}(x|y)$ is bad, then we should be able to improve it easily.

Working with the distortion map $D_y(q)$ is very convenient for building generic code: our diagnostic wrapper, Algorithm 4.1 below, is always based on a probability density on the unit interval $[0, 1]$. In practice users will have a multivariate approximation $\tilde{\pi}(x^{(1)}, ..., x^{(p)}|y_{obs})$ and will get diagnostics by simulating or otherwise computing the marginals $\tilde{\pi}(x^{(i)}|y_{obs})$, $i = 1, ..., p$. These marginal distributions are computationally tractable, in contrast to the true marginals $\pi(x^{(i)}|y_{obs})$.

4.3 Estimating a Distortion map

We now explain how we estimate the distortion map without simulating the exact posterior. The distortion map D_y we would like to approximate is a continuous distribution on $[0, 1]$ so one approach is to sample from it and use the samples to estimate D_y. The difficulty is that $D_y(x)$ is a function of x which varies from one y-value to another. We can proceed as in Algorithm 4.1 below which we now outline.

We start by explaining how to simulate $Q \sim D_y$. If we simulate the generative model, $\{x, y\} \sim \pi(x)p(y|x)$, then by Bayes rule $\{x, y\} \sim p(y)\pi(x|y)$ with $p(y) = \int_{\mathcal{X}} \pi(x)p(y|x)dx$ the marginal likelihood, so a simulation from the generative model gives us a draw X from the exact posterior at the data $Y = y$. This observation is just the starting point for ABC. Now, from our discussion below Equation 4.1, if $Q = G_y(X)$ then the pair $\{Q, Y\}$ have a joint distribution with density $d_y(q)p(y)$ and

conditional distribution $Q|(Y=y) \sim D_y$. This is a recipe to simulate $\{q_i, y_i\}_{i=1}^N$ pairs which are realisations of $\{Q, Y\}$: Simulate $\{x_i, y_i\}_{i=1}^N$ with $x_i \sim \pi(\cdot)$ and $y_i \sim p(\cdot|x_i)$ and then set $q_i = G_{y_i}(x_i)$ (the subscript $i = 1, ..., N$ runs over samples, not multivariate components). If $\tilde{\pi}(x|y)$ admits a closed form CDF $G_y(x)$ then q_i can be evaluated directly. If $G_y(x)$ is not tractable (as in our examples below) then we estimate it using samples from the approximate posterior. We form the empirical CDF $\hat{G}_y(x)$ and set $q_i = \hat{G}_{y_i}(x_i)$. The samples $\{q_i, y_i\}_{i=1}^N$ are our "data" for learning about D_y.

We next define a semi-parametric model for $D_y(q)$ and a log-likelihood for our new "data" $\{q_i, y_i\}_{i=1}^N$. For $q \in [0,1]$ and $w \in \mathbb{R}^m$, let $\mathcal{D}_m = \{D_y(\cdot; w); w \in \mathbb{R}^m\}$ be a family of continuously differentiable and strictly increasing CDF's parameterized by $w \in \mathbb{R}^m$. We assume $D_y(\cdot; w)$ is continuous with respect to y and w. We also require that $\mathcal{D}_m$ includes the identity map, i.e. there exists some $w_I \in \mathbb{R}^m$ such that $D_y(q; w_I) = q$ for all $q \in [0,1]$ and $y \in \mathcal{Y}$. Because we are targeting the distortion map, we are working with a probability distribution on $[0,1]$, so we simply model $d_y(q; w)$, the corresponding density of $D_y(q; w)$, using

$$d_y(q; w) = \text{Beta}(q; a(y; w), b(y; w)), \tag{4.4}$$

a Beta density with parameters $a = a(y; w)$ and $b = b(y; w)$ which vary over $\mathcal{Y}$. The Beta parameters $a(y; w), b(y; w) : \mathcal{Y} \to (0, \infty)$ are smooth functions parameterized by $w \in \mathbb{R}^m$. We also tried to parameterize $d_y(q; w)$ as a mixture of Beta-distributions but found no real gain from taking more than one mixture component.

We now use our model $D_y(\cdot; w)$ to estimate $D_{y_{obs}}$ at the observed data y_{obs}. The log likelihood for our parameters given our model $D_y(q; w)$ and simulations $\{q_i, y_i\}_{i=1}^N$ is

$$\ell(w; \{q_i, y_i\}_{i=1}^N) = \frac{1}{N} \sum_{i=1}^{N} \log d_{y_i}(q_i; w). \tag{4.5}$$

Let $\hat{w}_N$ maximise this log-likelihood. The estimated distortion map at y_{obs} takes the form $\hat{D}_{y_{obs}}(q) = D_{y_{obs}}(q; \hat{w}_N),\ q \in [0,1]$. Let

$$W = \{w^* \in \mathbb{R}^m : D_y(q) = D_y(q; w^*)\ \forall q \in [0,1], y \in \mathcal{Y}\} \tag{4.6}$$

be the set of parameter values giving the true distortion map. This set is empty unless $D_y \in \mathcal{D}_m$ for all $y \in \mathcal{Y}$.

We show below that, if $\mathcal{D}_m$ is sufficiently expressive, so that W is non-empty, then $D_y(q; \hat{w}_N) \xrightarrow{p} D_y(q)$ for any fixed $\{y, q\}$. This is not straightforward as w^* in Eqn. 4.6 may not be identifiable so standard regularity conditions for MLE-consistency are not satisfied. Our result compliments that of Papamakarios and Murray (2016) and

Greenberg et al. (2019). Interpreting their results in our setup, those authors show that the *maximiser of the limit* of the scaled log-likelihood gives the true distortion map (if $\mathcal{D}_m$ is sufficiently expressive). Our consistency proof shows that the *limit of the maximiser* $\hat{w}_N$ converges to the set W of parameter values that recover the true distortion map.

Proposition 4.3.1 translates the result of Papamakarios and Murray (2016) to our setting. At fixed $y \in \mathcal{Y}$ and $w \in \mathbb{R}^m$, suppose the exact and approximate distortion maps, $D_y(q)$ and $D_y(q; w)$ have associated densities $d_y(q)$ and $d_y(q; w)$. Their KL-divergence is

$$\mathrm{KL}(D_y(\cdot), D_y(\cdot; w)) \equiv \int_0^1 d_y(q) \log\left(\frac{d_y(q)}{d_y(q; w)}\right) dq.$$

Here, as in Papamakarios and Murray (2016), the KL-divergence of interest is the complement of that used in variational inference. We choose the approximating distribution $D_y(\cdot; w)$ to fit samples drawn from the true distribution $D_y(\cdot)$. This is possible using ABC-style joint sampling from the generative model. By contrast, in variational inference $D_y(\cdot; w)$ is varied so that *its* samples match $D_y(\cdot)$.

Proposition 4.3.1. *Suppose the set W in (4.6) is non-empty. Let $\{q_i, y_i\} \sim p(y_i)d_{y_i}(q_i)$ independently for $i = 1, ..., N$. Then $N^{-1}\ell(w, \{q_i, y_i\}_{i=1}^N)$ in (4.5) converges in probability to*

$$-E_Y(KL(D_Y(\cdot), D_Y(\cdot; w))) + E_{Q,Y}(\log(d_Y(Q)).$$

This limit function is maximized at $w \in W$.

Proof. See Chapter 4.10. □

Proposition 4.3.1 tells us that we are maximising the right function, since the limiting KL divergence is minimised at $D_Y(\cdot; w) = D_Y(\cdot)$, the true distortion map. However, it does not show consistency for $D_y(\cdot; \hat{w}_N)$. In Lemma 4.3.1 we show that $D_y(q; \hat{w}_N)$ is a consistent estimate of $D_y(q)$.

Lemma 4.3.1. *Under the conditions of Proposition 4.3.1, and the regularity conditions given by Redner et al. (1981), the estimated $D_y(q; \hat{w}_N)$ is consistent, that is*

$$\lim_{N\to\infty} \Pr(|D_y(q; \hat{w}_N) - D_y(q)| > \epsilon) = 0.$$

for any $\epsilon > 0$, $q \in [0, 1]$ and $y \in \mathcal{Y}$.

Proof. See Chapter 4.10. □

Our main result, Theorem 4.3.1, follows from Lemma 4.3.1. It states that, asymptotically, and in KL divergence, the "improved" CDF $\hat{F}_y(x) = D_y(G_y(x); \hat{w}_N)$ is closer to the true posterior CDF $F_y(x)$ than the original approximation $G_y(x)$.

Theorem 4.3.1. *Under the conditions of Lemma 4.3.1 and assuming* $KL(F_y, G_y) > 0$,

$$\Pr(KL(F_y, \hat{F}_y) < KL(F_y, G_y)) \to 1$$

as $N \to \infty$ *for every fixed* y.

Proof. See Chapter 4.10. □

We now give Algorithm 4.1, a practical implementation of the estimation procedure we described above. In Algorithm 4.1, we parameterize $a(y; w), b(y; w)$ as a neural net with input y, parameter w and two positive outputs a and b. We choose to use a neural net because its flexibility, and the capability of handling relatively high dimensional input y. Even though this parameterization may not satisfy all the regularity conditions we assumed previously, we find it works well in practice, and is able to accurately capture approximation errors in both synthetic and real world examples (see Chapter 4.5, 4.6 and 4.7).

The fitted distortion map at the data, $\hat{D}_{y_{obs}}(q) = D_{y_{obs}}(q; \hat{w}_N)$ is of interest as a diagnostic tool , as it relates the approximate and exact posterior. The improved posterior CDF, $\hat{F}_y(x)$ in Equation 4.3, or the corresponding PDF $\hat{\pi}(x|y)$, is of only indirect interest to us. The point here is that $\hat{D}_{y_{obs}}$ may be a useful diagnostic for the approximate posterior even if $\hat{F}_y(x)$ is a poor approximation to F_y, as $\hat{F}_y(x)$ is at least asymptotically closer in KL-divergence to F_y than G_y is. If we can improve on the approximation G_y substantially in KL-divergence to the true posterior, then it is not a good approximation.

Plots of $d_{y_{obs}}(q; \hat{w}_N)$ give an easily interpreted visual check on the approximate posterior $\tilde{\pi}(x|y_{obs})$. A check of this sort is not a formal test, but such a test would be helpful as we *know* $\tilde{\pi}(\cdot|y_{obs})$ is an approximation and want to know where it deviates from the truth and how badly. Since D_y is a quantile map, if $d_{y_{obs}}(q; \hat{w}_N)$ is a cup shaped function of $q \in [0, 1]$ then G_y is under-dispersed, cap-shaped is over-dispersed, and if say $D_{y_{obs}}(1/2; \hat{w}_N) \gg 1/2$ then the median of G_y lies above the median of F_y and so this is evidence that G_y is skewed to the right.

When we apply Algorithm 4.1 we need good neural net regression estimates $\hat{D}_y$ for y in the neighborhood of y_{obs} only. Fitting the neural net may be quite costly, and since the distortion-map estimate at y_{obs} is in any case dominated by information

Algorithm 4.1 Estimating the distortion map $D_{y_{obs}}$

Input: Observed data y_{obs}; Functions evaluating summary statistics $s(y), y \in \mathcal{Y}$ and the approximate CDF $G_y(x)$; Subset $\Delta \subset \mathcal{Y}$ centered at y_{obs}; Functions simulating the prior $\pi(x)$ and observation model $p(y|x)$.
for i in $1, \ldots, N$ **do**
  Sample $\{x_i, y_i\} \sim \pi(x)p(y|x)$ until $y_i \in \Delta$
  Compute $q_i = G_{y_i}(x_i)$
end for
Fit a neural net with weights $w \in \mathbb{R}^m$, input vector $s(y_i) \in \mathbb{R}^p$ and two positive scalar outputs $a(s(y_i); w), b(s(y_i); w)$ by minimising the loss function $-\ell(w; \{q_i, y_i\}_{i=1}^N)$ given by Eqns. 4.4 and 4.5 using e.g. gradient descend.
Return: the fitted distortion map $\hat{D}_{y_{obs}}(q) = D_{y_{obs}}(q; \hat{w}_N), q \in [0, 1]$ where $\hat{w}_N$ are the fitted weights.

from pairs $\{q, y\}$ at y-values close to y_{obs}, we regress on pairs $\{q_i, y_i\}$ such that $y_i \in \Delta$, where $\Delta \subseteq \mathcal{Y}$ is a neighbourhood of y_{obs}. The neural net adapts to the dependence around y_{obs} and would not be helped by data from further afield. This is not "an additional approximation" and quite different to the windowing used in ABC, since we are regressing, not "averaging" over this neighbourhood. Extending the regression to the whole of $\mathcal{Y}$ space would be straightforward but pointless.

Note that in Algorithm 4.1 we have introduced summary statistics $s(y)$ on the data. This may be useful if the data are high dimensional, or where there is a low dimensional sufficient statistic. In the examples which follow we found we were either able to train the neural network with $s(y) = y$, the raw data, or $s(y)$ being a sufficient statistics in an exponential family model for random networks.

4.3.1 Validation checks on $\hat{D}_y$

In this section we discuss the choice of N and the sample variation of $\hat{D}_y$. In order to check we have taken N large enough so that taking it larger will not lead to significant change, we run Algorithm 4.1 and estimate $D_{y_{obs}}(\cdot; \hat{w}_{N_j})$ using different samples sizes $N_1, ..., N_J$ where $\{N_j\}_{j=0}^J$ is an increasing, equally spaced sequence with $N_0 = 0$ and $N_J = N$. We check that $D_{y_{obs}}(q; \hat{w}_{N_j})$ converges numerically at each $q \in [0, 1]$ with increasing $j = 1, ..., J$ and is stable. In order to check the sample dependence, we can also break up our sample $\{q_i, y_i\}_{i=1}^N$ into blocks $\{q_i, y_i\}_{i=N_j}^{N_{j+1}}$ and, for $j = 0, ..., J-1$, form separate estimates $\hat{D}_{y_{obs}}^{(j)}$ and check the variation between function estimates is small.

4.3.2 Extending to higher dimensions

In this section we show how to estimate distortion maps and the corresponding densities for the approximate posterior density $\tilde{\pi}(x_1, x_2|y)$ of a continuous bivariate parameter $(x_1, x_2) \in \mathbb{R}^2$. The extension to higher dimensions is straightforward but not obviously useful for diagnostics.

Let $G_{x_1,y}(x_2)$ and $F_{x_1,y}(x_2)$ be the CDF's of the approximate and exact conditional posteriors, respectively $\tilde{\pi}(x_2|x_1, y)$ and $\pi(x_2|x_1, y)$, and let $G_y(x_1)$ and $F_y(x_1)$ be the CDF's of the approximate and exact marginal posteriors, respectively $\tilde{\pi}(x_1|y)$ and $\pi(x_1|y)$. Let $D_{x_1,y}$ be the distortion map defined by

$$D_{x_1,y}(G_{x_1,y}(x_2)) = F_{x_1,y}(x_2), \tag{4.7}$$

with $D_y(G_y(x_1)) = F_y(x_1)$ as before. Then the transformation of the joint density is

$$\pi(x_1, x_2|y) = d_{x_1,y}(G_{x_1,y}(x_2))d_y(G_y(x_1))\tilde{\pi}(x_1, x_2|y) \tag{4.8}$$

If the approximation is good at $y \in \mathcal{Y}$, then the densities $\pi(x_1, x_2|y)$ and $\tilde{\pi}(x_1, x_2|y)$ are near equal, which holds if the "distortion surface", $d_y(q_1, q_2)$ defined by

$$d_y(q_1, q_2) \equiv d_{G_y^{-1}(q_1),y}(q_2)d_y(q_1), \tag{4.9}$$

is close to one for all arguments $(q_1, q_2) \in [0, 1]^2$.

We estimate $D_y(q_1)$ as before. We estimate $D_{x_1,y}(q_2)$ by treating x_1 as data alongside y. We apply Algorithm 4.1, but now we simulate $\{x_{1,i}, x_{2,i}, y_i\}$ from the generative model in the for-loop, and create two datasets. The first dataset, $\{q_{1,i}, y_i\}_{i=1}^N$ with $q_{1,i} = G_{y_i}(x_{1,i})$, is the same as before. The second, $\{q_{2,i}, (x_{1,i}, y_i)\}_{i=1}^N$ with $q_{2,i} = G_{x_{1,i},y_i}(x_{2,i})$, is used to estimate the conditional $D_{x_1,y}$. We fit two neural network models for the Beta-density parameters, one fitting the Beta-CDF $D_y(q_1; w)$ using inputs $s(y_i)$ and choosing weights $w \in \mathbb{R}^{m_1}$ to maximise the likelihood

$$\ell(w; \{q_{1,i}, y_i\}_{i=1}^N) = \sum_{i=1}^N \log d_{y_i}(q_{1,i}; w) \tag{4.10}$$

and the other fitting the Beta-CDF $D_{x_1,y}(q_2; v)$ using inputs $(x_{1,i}, s(y_i))$ and choosing weights $v \in \mathbb{R}^{m_2}$ to maximise the likelihood

$$\ell(v; \{q_{2,i}, (x_{1,i}, y_i)\}_{i=1}^N) = \sum_{i=1}^N \log d_{x_{1,i},y_i}(q_{2,i}; v). \tag{4.11}$$

The run-time is approximately doubled. If $\hat{w}_N$ and $\hat{v}_N$ are the MLE's then the estimates are $\hat{D}_y(q_1) = D_y(q_1; \hat{w}_N)$ and $\hat{D}_{x_1,y}(q_2) = D_{x_1,y}(q_2; \hat{v}_N)$.

Finally, we plot the estimated distortion surface

$$\hat{d}_{y_{obs}}(q_1, q_2) \equiv \hat{d}_{G^{-1}_{y_{obs}}(q_1), y_{obs}}(q_2)\hat{d}_{y_{obs}}(q_1) \tag{4.12}$$

as a diagnostic plot. Both components are simply Beta-densities and straightforward to evaluate.

4.4 Further related works

Prangle et al. (2014) show that $\tilde{\pi}(x|y_{obs}) = \pi(x|y_{obs})$ for all x iff $G_{y_{obs}}(X) \sim U(0,1)$ for $X \sim \pi(\cdot|y_{obs})$. The authors give a diagnostic tool based on this idea for an ABC posterior using the simulated Q's as test statistics. They sample $\{x_i, y_i\}$ from the truncated generative distribution $\pi(x)p(y|x)\mathbb{1}(y \in \Delta)$, where $\Delta \subset \mathcal{Y}$ is a subset containing y_{obs}, and compute $q_i = G_{y_i}(x_i)$ for $i = 1, ..., N$. Then they check that the simulated $\{q_i\}_{i=1}^N$ are uniformly distributed over $[0, 1]$. This corresponds to studying the distribution of the marginalized random variable $Q = E_{Y\in\Delta}(G_Y(X)|Y)$ rather than the conditional random variable $[Q|(Y = y_{obs})] = G_{y_{obs}}(X)$ which we study. The diagnostic histogram plotted by Prangle et al. (2014) estimates the marginal density $d_\Delta(\cdot)$ of Q,

$$Q \sim d_\Delta(\cdot), \quad d_\Delta(Q) \propto \int_{y\in\Delta} d_y(Q)p(y)dy. \tag{4.13}$$

Since Δ is typically rather large, $d_y(\cdot)$ may vary over $y \in \Delta$. In this case the marginal distribution of Q may be flat when the conditional distribution of $Q|(Y = y_{obs})$ is far from flat (or the converse). We give an example in which this is the case. Similar ideas are explored in Talts et al. (2020) and Yao et al. (2018). Notice that when we window our data $\{q, y\}, y \in \Delta$ for neural net regression estimation of $\hat{w}_N$ in Algorithm 4.1, there is no integration over data y. We regress the distribution of $Q|(Y = y)$ at each $y \in \Delta$ (i.e. close y_{obs}), so we explicitly model variation in $d_y(\cdot)$ with y within Δ.

Rodrigues et al. (2018) give a post-processing recalibration scheme for the ABC posterior developing Prangle et al. (2014). The setup is a multivariate version of Equation 4.1. Ignoring the intrinsic ABC approximation, the "approximation" they correct is due to the fact that they have posterior samples at one y-value and they want to transport or recalibrate them so that they are samples from the posterior at a different y-value. The main difference is that these authors are approximating the true posterior, whilst we are trying to avoid doing that.

Greenberg et al. (2019) propose Automatic Posterior Transformation (APT) to construct an approximate posterior. Our Algorithm 4.1 can be seen as the first

loop of their Algorithm 4. In their notation, let $q_{F(y,w)}(x)$ be an approximation to $\pi(x|y)$ where w are parameters of the fitted approximation. Let $p_r(x)$ be a proposal distribution for x. Define

$$\tilde{q}_{F(y,w)}(x) = q_{F(y,w)}(x)\frac{p_r(x)}{\pi(x)Z(y,w)}, \tag{4.14}$$

with Z a normalizing constant, and

$$\tilde{\mathcal{L}}(w) = \sum_{i=1}^{N} \log \tilde{q}_{F(y_i,w)}(x_i), \tag{4.15}$$

where $\{x_i, y_i\} \sim p_r(x)p(y|x)$ i.i.d. for $i = 1, ..., N$. Appealing to Papamakarios and Murray (2016), the authors show that the w-values maximising the scaled limit of $\mathcal{L}(w)$, w^* say, satisfy $q_{F(y,w^*)}(x) = \pi(x|y)$ (if the representation is sufficiently expressive) and this leads to a novel algorithm for approximating the posterior. Our approach is a special case obtained by taking $x \in \mathbb{R}$, $p_r(x) = \pi(x)$ (so $Z(y,w) = 1$) and the special parameterization

$$q_{F(y,w)}(x) = d_y(G(x); w)\tilde{\pi}(x|y). \tag{4.16}$$

In further contrast, we are concerned with diagnosing an approximation, not targeting a posterior.

Some previous work on diagnostics has also avoided forming a good approximation to F_Y by focusing on estimating the approximation errors for posterior expectations of special functions only. Works on calibration of credible sets by Xing et al. (2019) and Lee et al. (2019) fall in this category. Instead of estimating the distortion over the whole CDF G_Y, these authors estimate the distortion in the value of one quantile. This lacks diagnostic detail compared to our distortion map. They consider a level α approximate credible set $\tilde{C}_y(\alpha) \subseteq \mathcal{X}$ computed from the approximate posterior. Xing et al. (2019) estimate how well this approximate credible set covers the true posterior, that is they estimate

$$c_{y_{obs}}(\alpha) = E_{X|Y=y_{obs}}(\mathbb{1}(X \in \tilde{C}_{y_{obs}}(\alpha))), \tag{4.17}$$

using regression and methods related to importance sampling, and then compare $c_{y_{obs}}(\alpha)$ to the nominal level α. In contrast to Xing et al. (2019), we estimate $D_{y_{obs}}(q)$ as a function of q, so we estimate the distortion in the univariate CDF, not just the distortion in the mass it puts on one set.

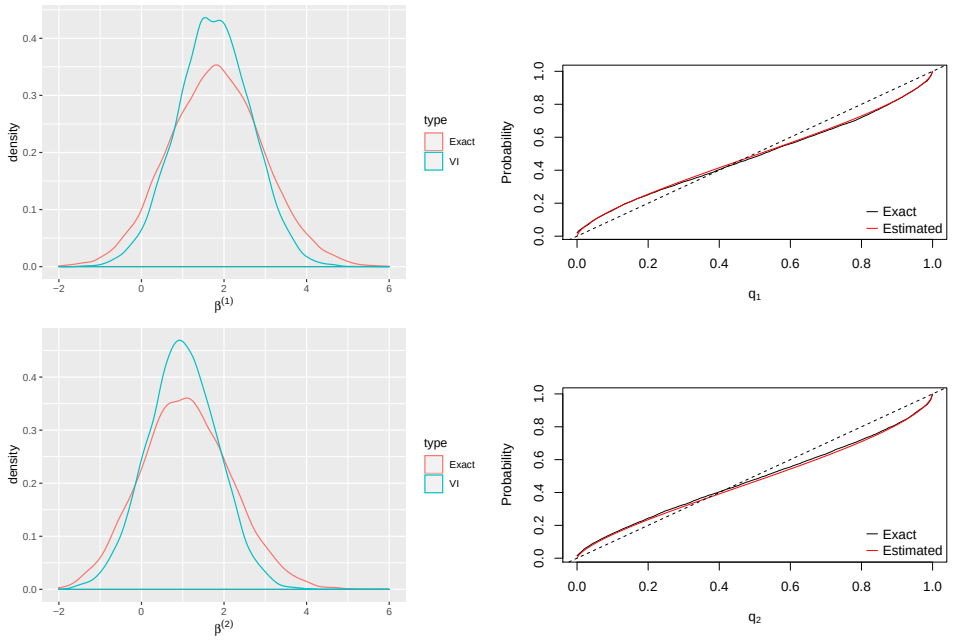


Figure 4.1: Left: Exact and approximate posterior for $\beta^{(i)}$, $i = 1, 2$. Right: Exact $D^{(i)}_{y_{obs}}(\cdot)$ and fitted $\hat{D}^{(i)}_{y_{obs}}(\cdot)$ for $\beta^{(i)}$, $i = 1, 2$. Dashed line is the identity map.

4.5 A Toy example

We apply Algorithm 4.1 to Bayesian logistic regression. Let X be a $n \times p$ design matrix, let $\beta \in \mathbb{R}^p$ be regression coefficients and $y = (y_{(1)}, ..., y_{(n)}) \in \{0, 1\}^n$ be binary response data. For each $j = 1, ..., n$, $y_{(j)} \sim \text{Bernoulli}(p_j)$ where $\text{logit}(p_j) = x_{(j)}^T \beta$ and $x_{(j)}$ is the jth row of X. The likelihood is

$$p(y|\beta) = \prod_{j=1}^{n} p_j{}^{y_{(j)}} (1 - p_j)^{1-y_{(j)}}, \quad p_j = \frac{\exp(x_{(j)}^T \beta)}{\exp(x_{(j)}^T \beta) + 1}$$

We take a prior distribution $\pi(\beta) = \text{Normal}(0, 2I_p)$ with I_p the $p \times p$ identity matrix. Let the observed value y_{obs} be a random draw from the marginal likelihood $p(y) = \int \pi(\beta) p(y|\beta) d\beta$. We are interested in the posterior distribution $\pi(\beta|y_{obs}) \propto \pi(\beta) p(y_{obs}|\beta)$.

The exact posterior can be sampled via standard MCMC. It is also possible to approximate the exact $\pi(\beta|y_{obs})$ using computationally cheaper Variational Inference (VI) with approximate posterior $\tilde{\pi}(\beta|y_{obs})$ (Jaakkola and Jordan, 1997). In this example, we set $p = 8, n = 50$, and we would like to diagnose the performance of the variational posterior $\tilde{\pi}(\beta|y_{obs})$ using Algorithm 4.1. In our example, each entry in the design matrix X is sampled independently from $U(0, 1)$. We simulate 10^6 synthetic

$\{\beta, y\}$-pairs from the generative model $\pi(\beta)p(y|\beta)$, randomly pick one synthetic data point as our observed y_{obs}, and keep the 1% of pairs $\{\beta_i, y_i\}_{i=1}^N$ closest in Euclidean distance to y_{obs} as our training data (this corresponds to a particular choice of Δ in Algorithm 4.1). Since there is no low dimensional sufficient statistic for this model, we simply use $s(y) = y$, the $n = 50$ dimensional binary response vector, as the summary statistic. We then apply Algorithm 4.1 using a feed froward neural net with two hidden layers of 80 nodes to estimate the distortion map $\hat{D}^{(j)}_{y_{obs}}(\cdot)$ for each dimension $j = 1, ..., p$ of β (recall $p = 8$), and compare the estimated map $\hat{D}^{(j)}_{y_{obs}}(q)$ to the exact $D^{(j)}_{y_{obs}}(q)$ as a function of $q \in [0, 1]$. The exact distortion map we report here is in the form $D_y(q) = \hat{F}_y \circ \hat{G}_y^{-1}(q)$, where $\hat{F}_y(\cdot)$ and $\hat{G}_y(\cdot)$ are empirical estimate of $F_y(\cdot)$ and $G_y(\cdot)$.

We plot the marginal posteriors and the corresponding exact and fitted distortion maps for the first two dimensions $\beta^{(1)}$, $\beta^{(2)}$ of the regression parameter β in Fig. 4.1. The fitted distortion map $\hat{D}^{(1)}_{y_{obs}}(\cdot)$ and $\hat{D}^{(2)}_{y_{obs}}(\cdot)$ in the right column accurately recover the exact map. Both $\hat{D}^{(1)}_{y_{obs}}(\cdot)$ and $\hat{D}^{(2)}_{y_{obs}}(\cdot)$ slightly deviate from the identity map, correctly showing that the marginal VI posteriors for $\beta^{(1)}$ and $\beta^{(2)}$ are slightly under-dispersed compared to the exact posterior. This simple example shows our method is able to handle moderately high dimensional ($n = 50$) summary statistics $s(y)$.

4.6 Karate club network

In this section we estimate distortion maps measuring the quality of three distinct approximate schemes for network models. The data we choose are relatively simple, but happen to illustrate several points neatly. We repeat the analysis on a larger data set in the following section. The small size of the network data in this example is not an essential point. Conclusions from the larger data set are similar though in some respects less interesting.

The Zachary's Karate Club network (Zachary, 1977) is a social network with 34 vertices (representing club members) and 78 undirected edges (representing friendship). The data is available at UCINET IV Datasets. See Fig. 4.2.

We fit an Exponential Random Graph Model (ERGM) (Robins et al., 2007) to these data. Let $\mathcal{Y}$ be the set of all graphs with n nodes. Given $y \in \mathcal{Y}$, let $s(y) \in \mathbb{R}^p$ be a p-dimensional graphical summary statistic computed on y and let $x \in \mathbb{R}^p$ be the corresponding ERGM parameter. In this example $p = 3$. In an ERGM, the likelihood of the graph y is

$$p(y|x) = \exp\{x^T s(y)\}/z(x) \tag{4.18}$$

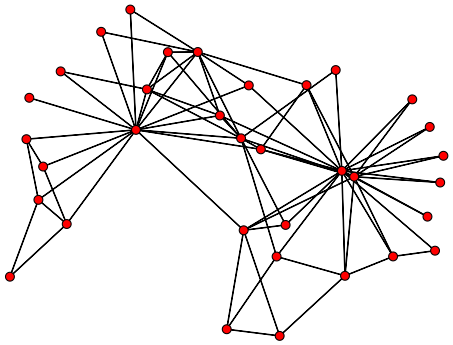


Figure 4.2: Zachary's Karate Club network (Zachary, 1977), consists of 34 vertices and 78 undirected edges.

where the normalizing constant $z(x) = \sum_{y \in \mathcal{Y}} \exp\{x^T s(y)\}$ is intractable even for relatively small networks.

Our example come from Caimo and Friel (2014) and Bouranis et al. (2018). Let $s_1(y)$ be the number of edges in y. Following Hunter and Handcock (2006), let $\mathrm{EP}_l(y)$ be the number of connected dyads in y that have l common neighbors, and let $D_l(y)$ equal the number of nodes in y that have l neighbors. Let

$$v(y, \phi_v) = e^{\phi_v} \sum_{l=1}^{n-2} \{1 - (1 - e^{-\phi_v})^l\} \mathrm{EP}_l(y)$$

be the geometrically weighted edgewise shared partners (gwesp) statistic and

$$u(y, \phi_u) = e^{\phi_u} \sum_{l=1}^{n-1} \{1 - (1 - e^{-\phi_u})^l\} \mathrm{D}_l(y)$$

be the geometrically weighted degree (gwd) statistic. Following Caimo and Friel (2014) let $\phi_v = 0.2$ and $\phi_u = 0.8$, $s(y) = (s_1(y), v(y, \phi_v), u(y, \phi_u))$ and $x = \{x^{(1)}, x^{(2)}, x^{(3)}\} \in \mathbb{R}^3$. Our observation model is given by Eqn 4.18. The prior distribution $\pi(\cdot)$ for x is multivariate Normal with $\mu = (-2, 0, 0)$ and $\Sigma = 5I_3$.

The exact $\pi(x|y) = \pi(x)p(y|x)/p(y)$ is doubly intractable. We consider three approximation schemes yielding different approximations $\tilde{\pi}(x|y)$:

- Approximate Bayesian Computation with ABC acceptance fraction $\rho = 0.5\%$ and local linear regression adjustment ("ABC-reg", Pritchard et al. (1999); Beaumont (2010)).

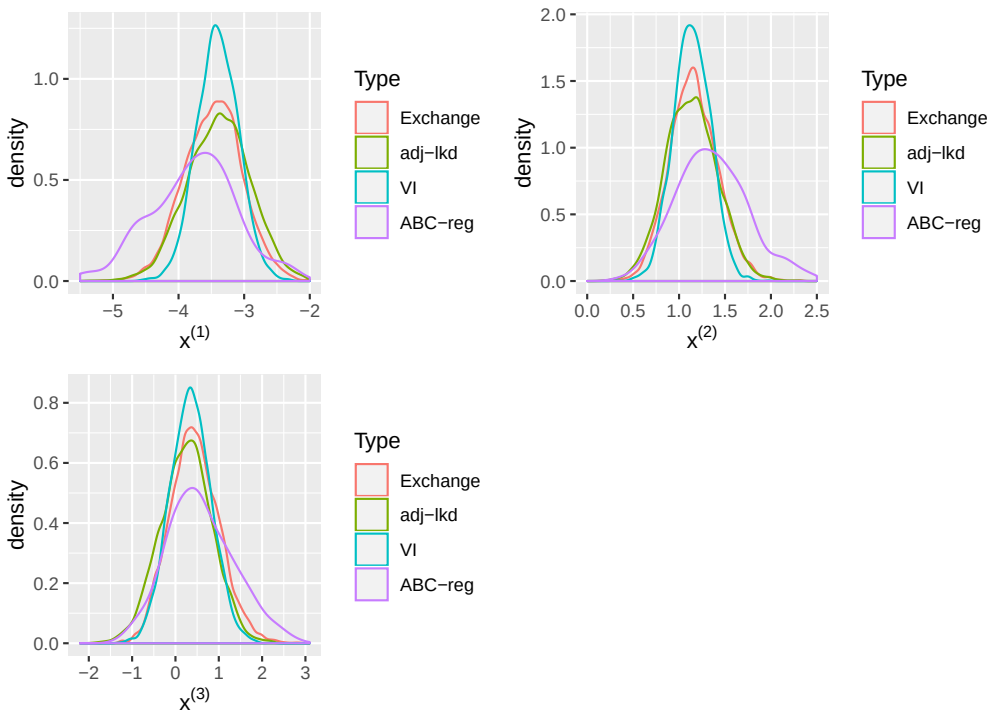


Figure 4.3: Approximate and exact posteriors for the Karate club data

- Fully adjusted pseudolikelihood ("adj-lkd") (Bouranis et al., 2017, 2018);
- Variational inference ("VI") (Tan and Friel, 2020)

We have ground truth in this example. We can approximately sample from $\pi(x|y)$ using an exchange algorithm (Murray et al., 2006). This is still approximate but very accurate. For each approximation scheme and dimension $x^{(p)}$, $p = 1, ..., 3$, we fit the distortion map $\hat{D}^{(p)}_{y_{obs}}$ using Algorithm 4.1 and compare our $\hat{d}^{(p)}_{y_{obs}}$-diagnostic plot with diagnostic plots obtained using the methods of Prangle et al. (2014) and Talts et al. (2020).

We simulated $N = 3 \times 10^5$ pairs $\{x_i, y_i\}_{i=1}^N$ from the generative model $\pi(x)p(y|x)$, taking pairs $\{x_i, y_i\}$ pairs in the top 15% by least Euclidean distance to $s(y_{obs})$ as our training data. We first report the approximate posteriors themselves. In Fig. 4.3 (left column) we see that the adj-lkd approach (top row) gives the best approximate posterior for all dimensions. In comparison, the VI approach (bottom row) gives an under-dispersed approximation while the ABC-reg posterior (middle row) is over-dispersed and slightly biased. In a real application we would not have this ground truth.

We now run Algorithm 4.1 using a feed froward neural net with two hidden layers of 80 nodes and estimate $\hat{D}^{(p)}_{y_{obs}}$ for all approximation schemes and dimensions $x^{(p)}$. For

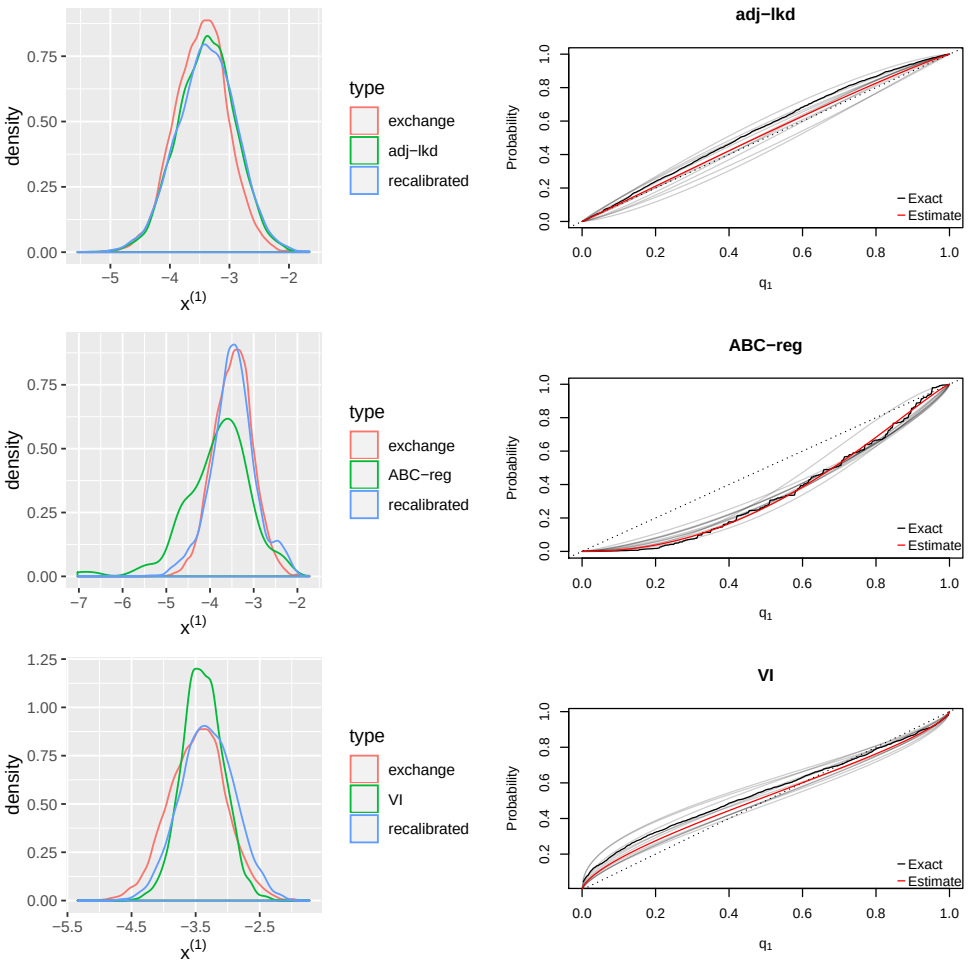


Figure 4.4: Left: Recalibrated posterior $\hat{F}_{y_{obs}}$ for $x^{(1)}$ for each approximation scheme Right: Exact $D^{(1)}_{y_{obs}}(\cdot)$ and fitted $\hat{D}^{(1)}_{y_{obs}}(\cdot)$ for $x^{(1)}$, Dashed line represents the identity map. Grey lines are $\hat{D}^{(1)}_{y_{obs}}(\cdot)$ fitted repeatedly using 70% random subset of the training data.

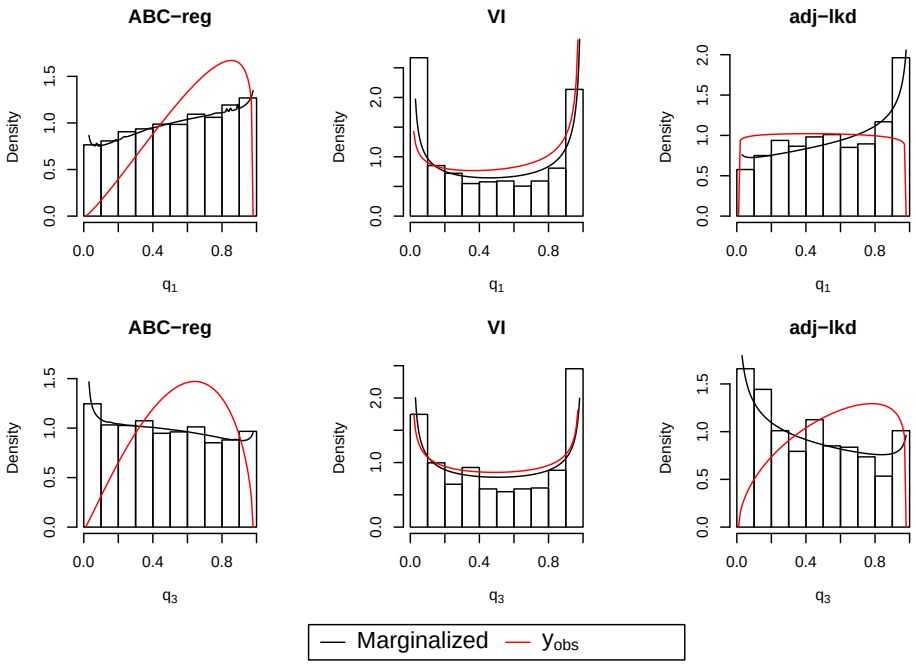


Figure 4.5: Diagnostic plot (Prangle et al., 2014) for each approximation scheme for $x^{(1)}$ (upper) and $x^{(3)}$ (lower). Black curve: marginalized (averaged) $\hat{d}_{\Delta}(\cdot)$ over y s.t. $s(y) \in \Delta_{s(y_{obs})}$. Red curve: fitted $\hat{d}_{y_{obs}}(\cdot)$ at y_{obs}. Recall that $\hat{d}(\cdot)$ represents the corresponding PDF of $\hat{D}(\cdot)$

brevity we now focus on the marginal distribution of $x^{(1)}$. In Fig. 4.4 (right column) we show the exact $D^{(1)}_{y_{obs}}(\cdot)$ (not available in real applications, but useful to show the method is working) and the fitted $\hat{D}^{(1)}_{y_{obs}}(\cdot)$ for $x^{(1)}$ for all three approximation schemes with the corresponding recalibrated posteriors $\hat{\pi}(x^{(1)}|y)$ (left column). For all approximation schemes the estimated $\hat{D}^{(1)}_{y_{obs}}(\cdot)$ is close to the exact $D^{(1)}_{y_{obs}}(\cdot)$, and are stable under repeated runs (which were fitted using 70% of the training data). The approximate posterior (with CDF G_y) matches the true posterior (in the graphs at left in Fig. 4.4) when $\hat{D}^{(1)}_{y_{obs}}(\cdot)$ is close to an identity map (in the graphs at right in the same figure). The recalibrated posteriors (with CDF $\hat{F}_y$) are closer to the exact, again indicating that our fitted $\hat{D}^{(1)}_{y_{obs}}(\cdot)$ is accurate.

Plots of the distortion density d_y allow direct comparison with the diagnostic histograms of Prangle et al. (2014) and Talts et al. (2020). Adopting those methods in our setting, we average d_y over an open ball centered at $s(y_{obs})$ containing the top 2.5% of $s(y_i)$'s closest to $s(y_{obs})$ in Euclidean distance. We see in Fig. 4.5, where we plot diagnostics for $x^{(1)}$ (top row) and $x^{(3)}$ (bottom row), that these diagnostic histograms successfully identify the under-dispersion of VI posteriors (a U-shape in the corresponding histograms (Talts et al., 2020) in the middle column). However,

the histogram of ABC-reg is reasonably flat for $x^{(3)}$, which seems healthy. This is misleading as the ABC-reg posterior for $x^{(3)}$ is in fact over-dispersed at y_{obs} as the d_y-graph in red shows. In contrast, the non-uniformity in the histogram of adj-lkd posterior of $x^{(1)}$ (top right) suggests that that approximation is poor, when we see from d_y-graph in red that the approximation is excellent (with ground truth in the top row of Fig. 4.3 agreeing). The diagnostic histograms of Prangle et al. (2014) and Talts et al. (2020) give both false-positive and false-negative alerts in this example.

To further illustrate this behavior on the adj-lkd example for $x^{(1)}$, we sampled $K = 200$ pairs $\{x_k, y_k\}_{k=1}^K \sim \pi(x)p(y|x)\mathbb{1}(s(y) \in \Delta_{s(y_{obs})})$, so that the $s(y_k)$'s are all close to $s(y_{obs})$. For each data set y_k, we compute an equal-tail approximate credible set with level $\alpha = 0.8$ for $x^{(1)}$ using the adj-lkd posterior. Following Xing et al. (2019) we can ask, what is the true (i.e. "operational") coverage $\tilde{c}_{y_k}^{(1)}(\alpha)$ achieved by this approximate set in the exact posterior? Does the approximate credible set at the data have the stated coverage in the true posterior? The exchange algorithm gives (fairly accurate) samples from the true posterior so the expectation in Eqn. 4.17 is easily estimated.

In Fig. 4.6 we plot the points $s(y_k) \in \mathbb{R}^3$ colored by their coverage. Red points points correspond to data where we are getting the right coverage. However there is an orange-colored plane region in the top right part of the plot where $\tilde{c}_{y_k}^{(1)}(\alpha)$ is much lower than the nominal level of 80%. The data y_{obs} is a red point so the coverage from the adj-lkd approximation is fine (as we would expect from the healthy diagnostics in Fig 4.5). However when we average we include data where the approximation is poor and reach the wrong conclusion. This illustrates how the quality of approximation can vary over a subset of data space $\mathcal{Y}$.

Finally, we estimate and report the bivariate distortion surface $d_{y_{obs}}$ for VI and adj-lkd approximations $\tilde{\pi}(x^{(1)}, x^{(2)}|y_{obs})$ to the posterior for the first two parameters $x^{(1)}$ and $x^{(2)}$. From Chapter 4.3.2, taking $q_1 = G_{y_{obs}}(x^{(1)})$ and $q_2 = G_{x^{(1)},y_{obs}}(x^{(2)})$, the distortion surface $d_{y_{obs}}(q_1, q_2)$ is

$$d_{y_{obs}}(q_1, q_2) \equiv d_{G^{-1}_{y_{obs}}(q_1),y_{obs}}(q_2) d_{y_{obs}}(q_1).$$

Fig. 4.7 shows that for the VI posterior, the distortion surface peaks on the boundary and corners of the $[0, 1]^2$ square, and is below 1 at the center (recall that it is a normalized bivariate probability density). This is the 2-D equivalent of the U shaped diagnostic plots for scalars described in Prangle et al. (2014) and Talts et al. (2020), reflecting the under-dispersed VI posterior approximation. In contrast, the distortion

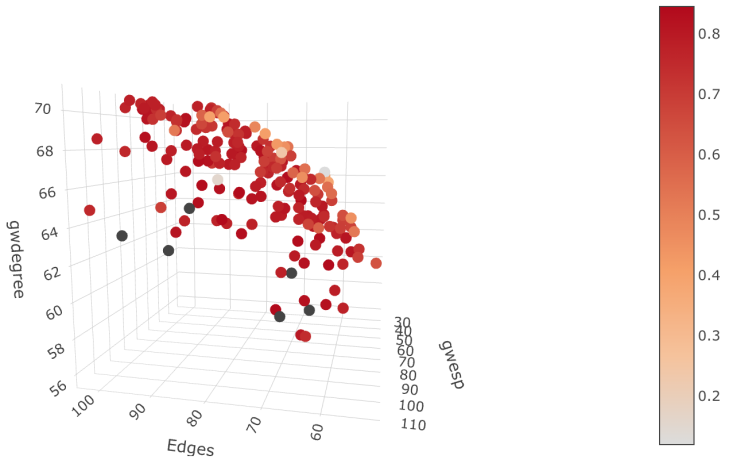


Figure 4.6: The estimated operational coverage of adj-lkd posterior of $x^{(1)}$ at each $s(y)$, magnitude of operational coverage is represented by colour, nominal level $\alpha = 0.8$

surface of adj-lkd posterior is between $0.9 \sim 1.2$ and relatively flat over much of the $[0,1]^2$ square: there is no evidence here for a problem with the adj-lkd approximation.

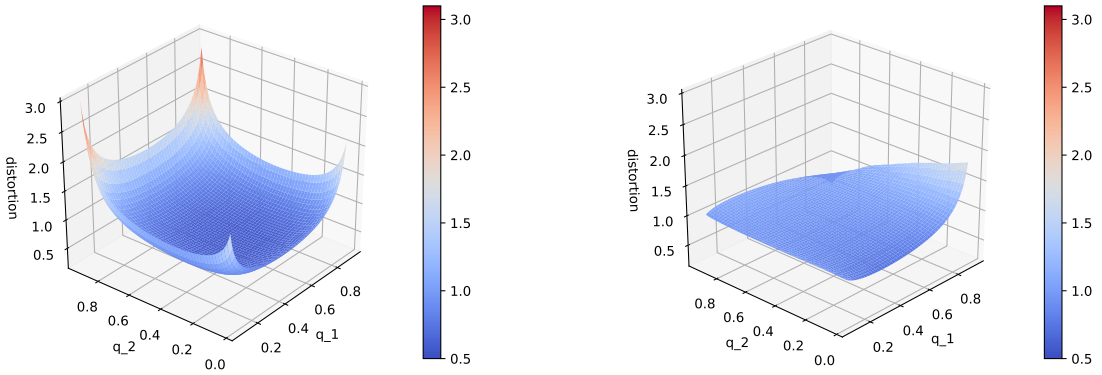


Figure 4.7: Left: Distortion surface of VI posterior with respect to q_1, q_2. Right: Distortion surface of adj-lkd posterior with respect to q_1, q_2.

4.7 Gene Fusion network

We also apply our approach on the larger Gene Fusion network (Höglund et al., 2006; Kunegis, 2013) with 291 nodes and 279 edges. Nodes represent genes and an edge is

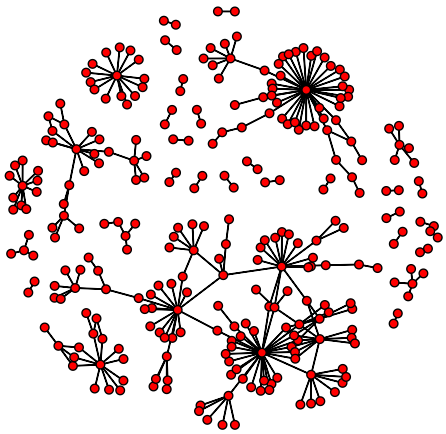


Figure 4.8: Gene Fusion network (Höglund et al., 2006; Kunegis, 2013), consists of 291 nodes and 279 edges.

present if fusion of the two genes is observed during the emergence of cancer. The same ERGM given in Chapter 4.6 is used.

In this example we report the ABC-reg and adj-lkd posteriors only, as the VI posterior behaves in the same way as in the Karate club network example (accurate mode, under-dispersed tails). Again, we report the fitted distortion map $\hat{D}_{y_{obs}}$ and the recalibrated $\hat{\pi}(x^{(p)}|y_{obs})$ for each $p = 1, 2, 3$ and for both approximation schemes in Fig. 4.9 and 4.10. Fig. 4.9 and 4.10 show that the estimated distortion maps $\hat{D}^{(p)}_{y_{obs}}$ are close to exact maps $D^{(p)}_{y_{obs}}$ for each dimension of the two approximation schemes. The estimated distortion map deviates from the identity map when the approximate marginals $\tilde{\pi}(x^{(p)}|y_{obs})$ deviate from the exact $\pi(x^{(p)}|y_{obs})$ substantially, and is close to the identity map when $\tilde{\pi}(x^{(p)}|y_{obs}) \approx \pi(x^{(p)}|y_{obs})$.

As in Chapter 4.6 we plot the distortion surfaces for the ABC-reg and adj-lkd posteriors for $\{x^{(1)}, x^{(3)}\}$. In this example there is little interesting bivariate structure as the joint distortion map is essentially the product of the univariate maps. From Fig. 4.11 we see the distortion surface of the ABC-reg posterior is far from 1, indicating that the ABC-reg approximation of the bivariate marginal posterior $\pi(x^{(1)}, x^{(3)}|y_{obs})$ is unreliable. The distortion surface of adj-lkd posterior at the data is reasonably close to 1, though somewhat barrel-shaped, reflecting the fact that the approximation to $x^{(3)}$ is (fairly slightly) overdispersed.

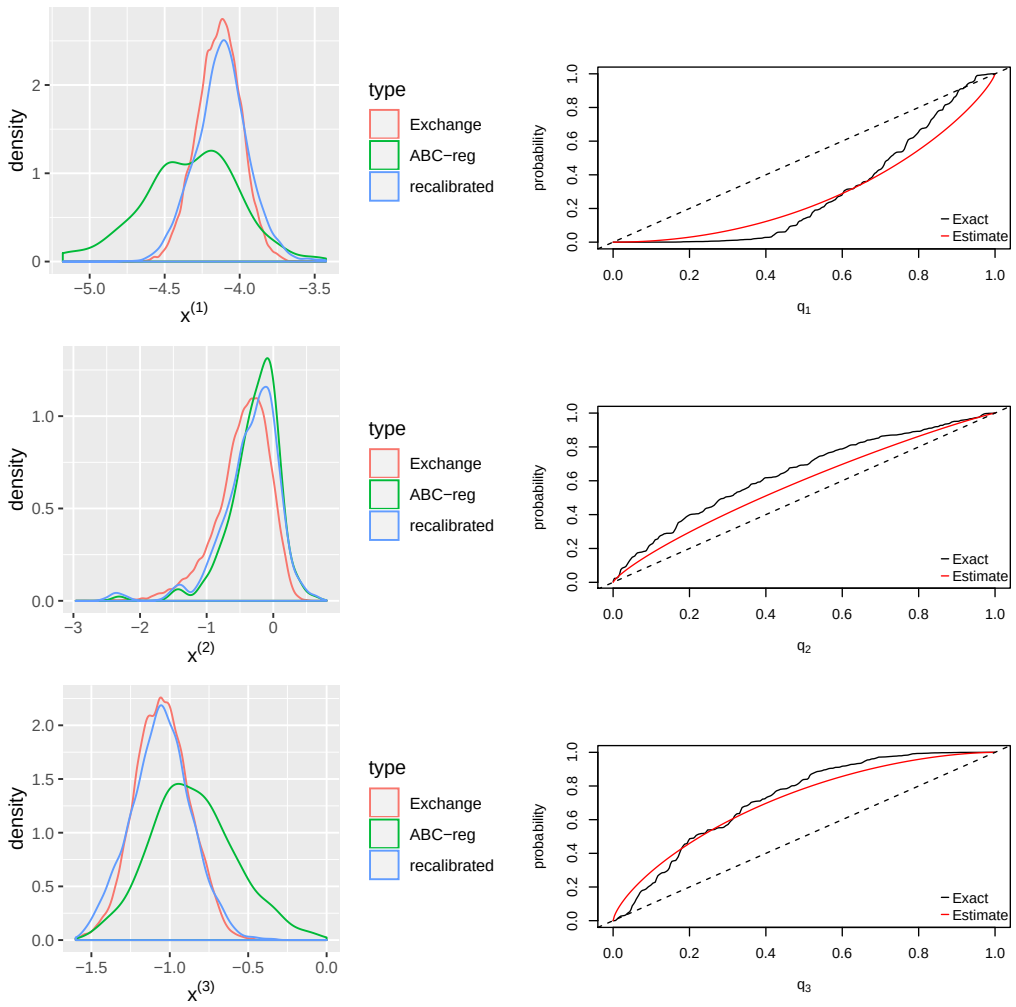


Figure 4.9: Left: Recalibrated posterior of $x^{(p)},\ p = 1, ...3$ for ABC-reg scheme Right: Exact $D^{(p)}_{y_{obs}}(\cdot)$ and fitted $\hat{D}^{(p)}_{y_{obs}}(\cdot)$ for $x^{(p)}$, Dashed line represents the identity map.

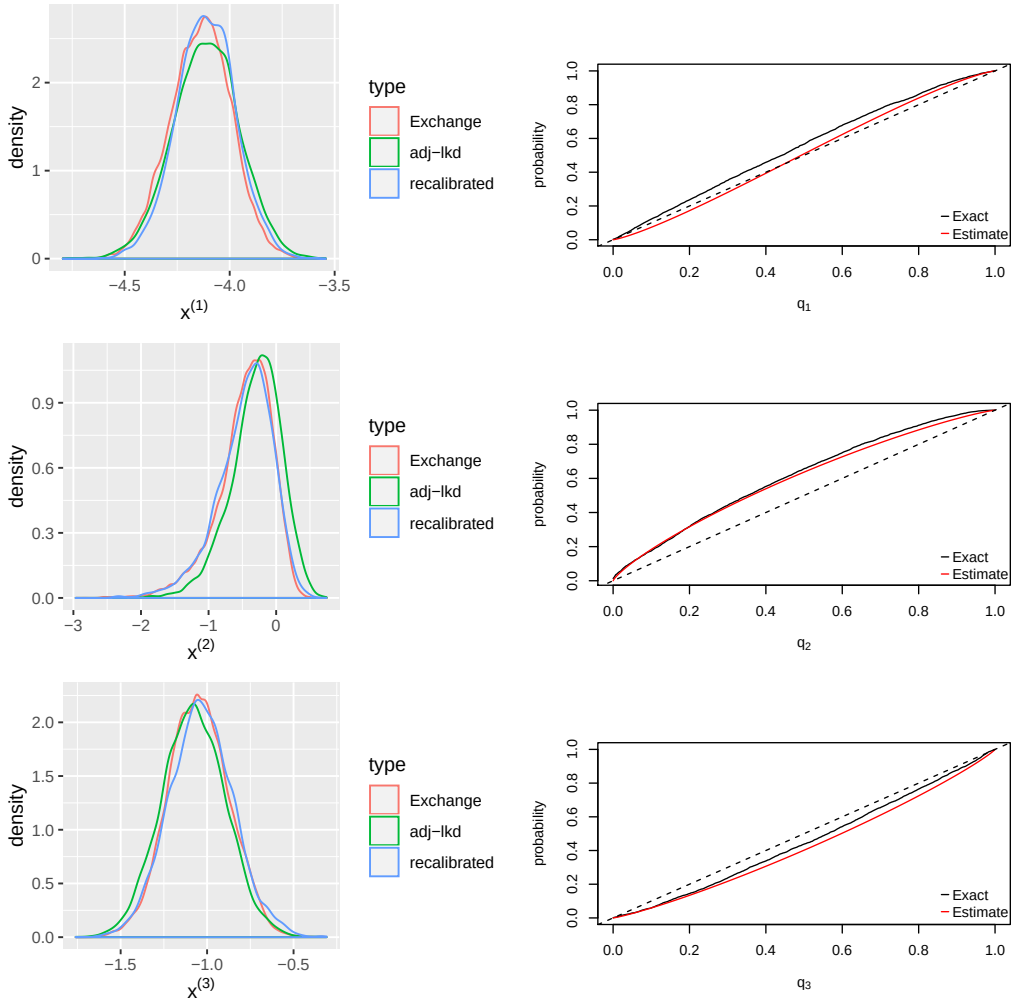


Figure 4.10: Left: Recalibrated posterior of $x^{(p)}$, $p = 1, ...3$ for adj-lkd scheme Right: Exact $D^{(p)}_{y_{obs}}(\cdot)$ and fitted $\hat{D}^{(p)}_{y_{obs}}(\cdot)$ for $x^{(p)}$, Dashed line represents the identity map.

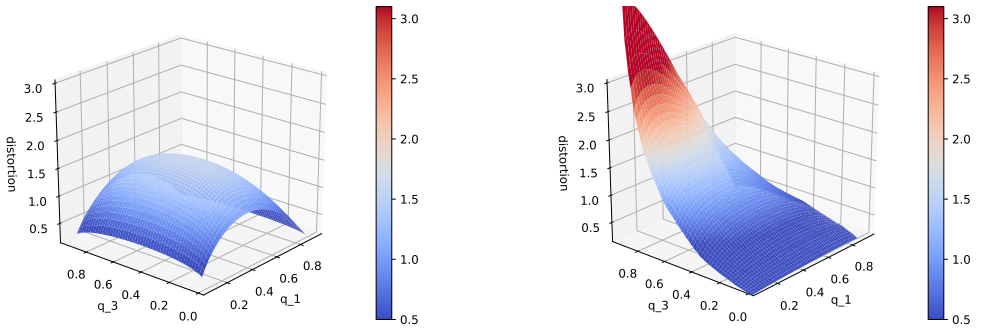


Figure 4.11: Left: Distortion surface of adj-lkd posterior with respect to q_1, q_3. Right: Distortion surface of abc-reg posterior with respect to $x^{(1)}, x^{(3)}$.

4.8 Conclusion and further discussion

In this chapter we give new diagnostic tools for approximate Bayesian inference. The distortion map $D_{y_{obs}}$ is a visual diagnostic tool for approximate marginal posteriors, which gives us diagnostic details about the approximation error. It is computationally demanding to estimate. Estimating the distortion map $D_{y_{obs}}$ requires sampling synthetic data from the generative model and calling the approximation scheme at each synthetic data point. In contrast to existing methods it checks the quality of approximation *at the observed data* y_{obs}, instead of estimating "averaged performance" over data space. Much of the code-base (simulation outline, fitting the Beta-density conditioned on y-values in a neighborhood of y_{obs}) carries over from one problem to another, so the user only needs to provide simulators for the generative model and the approximate posterior. This approach can be extended from diagnosing uni- or bivariate marginals to higher dimensions. We discuss possible strategies in Chapter 4.9.

4.8.1 Comparison to Approximate Bayesian Computation

Distortion map estimation shares a number of features with Approximate Bayesian Computation (ABC). These include simulation of the generative model and the presence of windowing on data $y \in \Delta$. The window plays different roles, as a marginalizing window in ABC and a conditioning window in distortion map seem superficially similar. How do the methods compare?

We would like to stress that ABC and distortion maps are designed for different purposes. ABC sets out to approximate either the marginal or the joint posterior of the multivariate parameter, while a distortion map is a diagnostic tool that aims to assess the approximation quality of the scalar or at most bivariate marginals of a given approximation scheme.

Compared to ABC, estimation of the distortion map also has the additional computational cost of (a) repeatedly applying the approximation scheme on each synthetic data points (easy when G_y is available in closed form, as is sometimes the case, as in mean-field VI). Existing methods such as Talts et al. (2020); Rodrigues et al. (2018) pay the same price. Another cost is (b) fitting the distortion map based on the simulated data set (we do this just once). In our experience (b) requires much less time than (a), so the method presented in this paper works best when the approximation scheme is computationally cheap.

4.8.2 Parameterization of D_y

In this chapter we parameterize D_y as a Beta CDF. This may seem an arbitrary and restrictive choice. However we are partly benefiting from the Normalizing flow type parameterization we have set up, as the distortion map is a CDF on $(0, 1)$. More fundamentally we feel that a parametric restriction or "regularization" of this sort is the price we pay for estimating a bias (i.e. the discrepancy between the approximate posterior and the truth) without knowing the truth. We use the neural net to regress on a space of (scalar) functions $D_y(x)$. By restricting this function space we regularize the fit in a helpful way. Other (possibly more flexible) parameterizations of D_y are available. For example, we tried parameterizing D_y with a mixture of Beta CDFs (up to 4 components) but found no improvement, just longer run times.

One drawback of our setup is that the single component Beta can be fooled: for example, if the true distortion density d_y was trimodal with peaks at 0.01, 0.5 and 0.99, then the estimated $\hat{d}_y$ based on a single Beta component would be close to uniform over (0,1) so our estimated diagnostic would seem to be good when the truth was bad (far from uniform). We can spot this by fitting a mixture of Beta CDFs as a diagnostic.

4.8.3 Windowing in Algorithm 4.1

In Algorithm 4.1, we only retain the simulated samples $\{x_i, y_i\}$ satisfying $y_i \in \Delta$, where Δ is a subset of $\mathcal{Y}$ centered at y_{obs}. This is because we are only interested in the approximation quality at y_{obs}. However, such rejection sampling strategy may not be computationally efficient if y_{obs} is not in the region of the data space $\mathcal{Y}$ where the marginal $p(y)$ puts most of its probability mass on. This is also a limitation of Algorithm 3.3, the regression estimator of operational coverage in Chapter 3. One possible strategy to address this limitation is to first generate synthetic data close to y_{obs} from some proposal that is *different* from the generative process $\pi(x)p(y|x)$, then reweight them appropriately so that they also lead to an unbiased estimate of the distortion map.

One straightforward way is to use the approximate posterior $\tilde{\pi}(x|y_{obs})$, instead of the prior $\pi(x)$, as the proposal of parameter x, as samples from a sensible approximate posterior are more likely to be associated with synthetic data close to y_{obs} than samples from the prior. In other words, we first sample synthetic pairs $\{x_i, y_i\}_{i=1}^N \sim \tilde{\pi}(x|y_{obs})p(y|x)$, then compute the importance weights $w_i = \frac{\pi(x)}{\tilde{\pi}(x|y_{obs})}$. In this case, the weighted samples $\{w_i, (x_i, y_i)\}_{i=1}^N$ can also be viewed as samples from

the generative process $\pi(x)p(y|x)$. However, this is not practical in general since the density function $\tilde{\pi}(x|y_{obs})$ is not always computationally tractable (e.g. the approximate posterior density of an ABC-type approximation is usually computationally intractable). Additionally, there is no guarantee that $\tilde{\pi}(x|y_{obs})$ is a sensible approximation. Hence the resulting synthetic data may not resemble the observed y_{obs}, rendering this strategy useless. One way to address this problem is to construct an alternative proposal $\bar{\pi}(x|y_{obs})$ using e.g. sequential approximation schemes such as Lueckmann et al. (2017); Greenberg et al. (2019); Durkan et al. (2020). Intuitively speaking, these methods are sequential "refinement" procedures that iteratively push an initial distribution toward the true posterior conditioned on y_{obs}. In our case, one may construct $\bar{\pi}(x|y_{obs})$ using e.g. Automatic Posterior Transformation (APT) (Greenberg et al., 2019) with either the prior $\pi(x)$ or the approximate posterior $\tilde{\pi}(x|y_{obs})$ being the initial distribution. We would like to stress that in our case, we do not aim to use e.g. APT to construct another potentially more accurate approximation of the exact posterior. Instead, we are only interested in constructing an alternative proposal $\bar{\pi}(x|y_{obs})$ for the parameter x such that samples from $\bar{\pi}(x|y_{obs})$ are more likely to be associated with synthetic data close to y_{obs}. This does not require $\bar{\pi}(x|y_{obs})$ to capture all the details in the true posterior. Hence constructing an alternative proposal $\bar{\pi}(x|y_{obs})$ using approximation methods such as APT is a much easier task and requires less computational cost than constructing a good approximation of the exact joint posterior. Once we have obtained $\bar{\pi}(x|y_{obs})$, we can then construct IS weighted samples in a similar fashion as before, using $\bar{\pi}(x|y_{obs})p(y|x)$ as the joint proposal.

However, since $\tilde{\pi}(x|y_{obs})$ and $\bar{\pi}(x|y_{obs})$ tend to be much more concentrated than the prior (i.e. having much lighter tail than the prior), the resulting IS weighted samples may exhibit high variability. Hence the weighted samples obtained in this fashion may not always be stable.

4.8.4 Diagnostic for our diagnostic

We cannot guarantee our estimate $\hat{D}_y$ based on $\hat{w}_N$ is reliable for any given sample size N. So some diagnostics are needed to check our diagnostic tool. Chapter 4.3.1 lists two obvious validation checks on $\hat{D}_y$. Another possible validation check is to run Algorithm 4.1 multiple times under different parameterizations of $\hat{D}_y$ and see if they yield similar results. For example, in the previous discussion we considered using a mixture of Beta instead of a single Beta CDF to parameterize D_y. Running Algorithm 4.1 again using a mixture of Beta could help us spot multimodality in the true distortion map.

In principle we have access to an unlimited amount of data to learn D_y, if we can efficiently simulate the generative model. However, this type of check can be time consuming, as it requires repeated calls of the approximation scheme for each synthetic data point. This also means our method is most effective when the computational cost of the evaluating the approximation $G_y(x)$ is manageable.

4.9 Multivariate extensions of distortion map

Even though the distortion map is able to assess the approximation quality of an approximation scheme at a specific data point y_{obs} and provides easy-to-interpret diagnostic details, diagnostics and re-calibration based on the distortion map are only practical for uni- or bivariate marginals of the approximate posterior. Here we discuss possible multivariate extensions of the distortion map. From Chapter 4.2 we see the distortion map D_y essentially estimates $F_y \circ G_y^{-1}$, which is the identity map if and only if $F_y = G_y$. However, it is not the only way to summarize the discrepancy between F_y and G_y. For example, we also have $F_y^{-1} \circ G_y$ is the identity map if and only if $F_y = G_y$. Hence we can also view $F_y^{-1} \circ G_y$ as a discrepancy summary between the true and approximate posterior: Note that if $x \sim \tilde{\pi}(\cdot|y)$ then we necessarily have $F_y^{-1}(G_y(x)) \sim \pi(\cdot|y)$, the exact posterior. By definition, $F_y^{-1} \circ G_y$ is a bijective function. Therefore this transformation can be interpreted as a Normalizing flow model that maps samples from the approximate posterior to the exact using "base" distribution $\tilde{\pi}(\cdot|y)$ and bijective transformation $F_y^{-1} \circ G_y$, and the difference between the true and approximate posterior is reflected by how the transformation $F_y^{-1} \circ G_y$ differs form the identity map. This observation can be generalized to the multivariate case: Suppose a multivariate approximation scheme is given. One can first estimate a conditional Normalizing flow model (e.g. Real-NVP (Dinh et al., 2016) or Neural spline flow (Durkan et al., 2019)) that maps samples from the approximate posterior $\tilde{\pi}(\cdot|y)$ to the exact $\pi(\cdot|y)$ conditioned on any $y \sim p(y)$ using e.g. the armortized training strategy given in Rothfuss et al. (2019) and Trippe and Turner (2018). Then for any generic data point $y \sim p(y)$, one can assess the approximation quality of $\tilde{\pi}(\cdot|y)$ by comparing the estimated bijective transformation conditioned on the chosen y with the multivariate identity map.

However, this idea is not practical in general, since the approximate posterior densities, i.e. the "base" densities in the Normalizing flow model, are not always available. For example, ABC-type approximation schemes in general only return a collection of approximate posterior samples, and the corresponding approximate posterior densities

are computationally intractable. In this case, estimating a Normalizing flow model with intractable base densities is not straightforward. In the following sections, we consider an alternative approach that does not require the approximate posterior densities to be tractable, and propose a diagnostic procedure that is able to diagnose both the joint approximate posterior and any uni- or multivariate marginals of it.

Similar to Chapter 3, we let ϕ, θ be generic samples from the exact posterior $\pi(\phi|y)$ and the approximate $\tilde{\pi}(\theta|y)$ respectively. Inspired by noise contrastive learning (Gutmann and Hyvärinen, 2010), we formulate the diagnostic problem as a a binary classification problem that aims to distinguish a "positive" sample $\phi \sim \pi(\cdot|y)$ from a "negative" sample $\theta \sim \tilde{\pi}(\cdot|y)$ at any given $y \in \mathcal{Y}$. If the classifier is able to separate ϕ from θ easily and confidently at a given $y \in \mathcal{Y}$, then the approximate $\tilde{\pi}(\cdot|y)$ must be very different from the true $\pi(\cdot|y)$, indicating poor approximation quality at the given y. In contrast, if a sensible classifier can not do better than a random guess at a given $y \in \mathcal{Y}$, then we know $\pi(\cdot|y)$ and $\tilde{\pi}(\cdot|y)$ must be reasonably close to each other. This suggests that the approximation is reliable at y. This binary classification setup is closely related to the approximate inference schemes in Greenberg et al. (2019); Durkan et al. (2020) and Thomas et al. (2022). We will discuss the connection between these works and our proposed diagnostic approach in later sections.

4.9.1 Proposed method

In this section we describe the proposed diagnostic method. We start by estimating a binary classifier that distinguishes samples from the true and approximate posterior at a generic data $y \in \mathcal{Y}$. We will then summarize the discrepancy between the true and approximate posterior based on the estimated classifier, and show how it is connected to the f-divergence between the two distributions. Let $J \in \mathbb{N}^+$. Consider the following generative process:

$$\phi \sim \pi(\cdot); \quad y \sim p(\cdot|\phi); \quad \theta_1, ..., \theta_J \overset{i.i.d.}{\sim} \tilde{\pi}(\cdot|y). \tag{4.19}$$

Let $\underline{\theta} = \{\theta_1, ..., \theta_J\}$. Then the joint density of the triplet $\{\phi, \underline{\theta}, y\}$ can be written as

$$p(\phi, \underline{\theta}, y) = \pi(\phi|y) \prod_{j=1}^{J} \tilde{\pi}(\theta_j|y) p(y) \tag{4.20}$$

Let $D(\cdot, \cdot) : \Theta \times \mathcal{Y} \to (0, 1)$ be a discriminator function. Consider the loss functional

$$L(D(\cdot, \cdot)) = -E_p \left(\log D(\phi, y) + \frac{1}{J} \sum_{j=1}^{J} \log \left(1 - D(\theta_j, y)\right) \right) \tag{4.21}$$

where E_p means taking expectation with respect to the joint density $p(\phi, \underline{\theta}, y)$.

Proposition 4.9.1 (Loss function for classification)**.** *Suppose* $J \in \mathbb{N}^+$, $p(\phi, \underline{\theta}, y) = \pi(\phi|y) \prod_{j=1}^{J} \tilde{\pi}(\theta_j|y)p(y)$ *defined in* (4.20)*. The loss functional* $L(D(\cdot,\cdot))$ *is minimized at*

$$D^*(\theta, y) = \frac{\pi(\theta|y)p(y)}{\pi(\theta|y)p(y) + \tilde{\pi}(\theta|y)p(y)} \tag{4.22}$$

$$= \sigma\left(r(\theta, y)\right), \tag{4.23}$$

where $\sigma(\cdot)$ *is the sigmoid function and* $r(\theta, y) = \log \frac{\pi(\theta|y)}{\tilde{\pi}(\theta|y)}$ *is the log density ratio.*

Proof. See Chapter 4.10. □

Proposition 4.9.1 justifies that $L(D(\cdot,\cdot))$ is a sensible loss if our goal is to distinguish samples from the true and approximate posterior. Note that we can estimate $D^*(\theta, y)$ by equivalently estimating the density ratio $r(\theta, y)$. We focus on the later. Let $J, N \in \mathbb{N}^+$. We sample $\{\phi_i, \underline{\theta}_i, y_i\}_{i=1}^N \sim p(\phi, \underline{\theta}, y)$ independently, where $\underline{\theta}_i = \{\theta_{ij}\}_{j=1}^J$. Let $g(\cdot,\cdot;\omega) : \Theta \times \mathcal{Y} \to \mathbb{R}$ be a function parameterized by $\omega \in \mathbb{R}^l$. Let

$$\hat{L}(\omega; \{\phi_i, \underline{\theta}_i, y_i\}_{i=1}^N) = -\frac{1}{N} \sum_{i=1}^{N} \left(\log \sigma(g(\phi_i, y_i; \omega)) + \frac{1}{J} \sum_{j=1}^{J} \log\left(1 - \sigma(g(\theta_{ij}, y_i; \omega))\right) \right) \tag{4.24}$$

be an empirical estimate of the loss $L(\sigma(g(\cdot,\cdot;\omega)))$. Let

$$\hat{\omega}_N = \arg\min_{\omega \in \mathbb{R}^l} \hat{L}(\omega; \{\phi_i, \underline{\theta}_i, y_i\}_{i=1}^N). \tag{4.25}$$

Then $g(\theta, y; \hat{\omega}_N)$ can be viewed as an estimator of the log density ratio $r(\theta, y) = \log \pi(\theta|y) - \log \tilde{\pi}(\theta|y)$, and $\sigma(g(\theta, y; \hat{\omega}_N))$ can be viewed as an estimator of $\frac{\pi(\theta|y)}{\pi(\theta|y)+\tilde{\pi}(\theta|y)}$.

Once we have obtained an estimator $g(\theta, y; \hat{\omega}_N)$ of the log density ratio, we can then approximately estimate the f-divergence between the exact and approximate posterior, and use it as a discrepancy summary between them. Suppose the estimated density ratio $g(\theta, y; \hat{\omega}_N)$ is given. Let y be a generic observed value, $K \in \mathbb{N}^+$, $\underline{\theta} = \{\theta_k\}_{k=1}^K \sim \tilde{\pi}(\cdot|y)$ be a collection of samples from the approximate posterior conditioned on y. Let an f-divergence with its associated generator function f be given. By the definition of f-divergence in Equation 2.8, we can view

$$\hat{D}_f(\pi(\cdot|y), \tilde{\pi}(\cdot|y)) = \frac{1}{K} \sum_{k=1}^{K} f(g(\theta_k, y; \hat{\omega}_N)) \tag{4.26}$$

as an approximate estimate of the f-divergence $D_f(\pi(\cdot|y), \tilde{\pi}(\cdot|y))$ between the true $\pi(\cdot|y)$ and the approximate $\tilde{\pi}(\cdot|y)$. This means $\hat{D}_f(\pi(\cdot|y), \tilde{\pi}(\cdot|y))$ is a natural scalar

summary of the approximation error in $\tilde{\pi}(\cdot|y)$. In addition to diagnosing the approximation error at a specific observation y, we can also assess the average approximation quality of an approximation scheme by averaging $\hat{D}_f(\pi(\cdot|y), \tilde{\pi}(\cdot|y))$ over the data space $\mathcal{Y}$. Suppose $M \in \mathbb{N}^+$. Define

$$\hat{D}_f = \frac{1}{M} \sum_{m=1}^{M} \hat{D}_f(\pi(\cdot|y_i), \tilde{\pi}(\cdot|y_i)), \quad y_1, ..., y_M \overset{i.i.d.}{\sim} p(y) \tag{4.27}$$

Then $\hat{D}_f$ can be viewed as an approximate estimate of the *averaged* f-divergence $\int_{\mathcal{Y}} D_f(\pi(\cdot|y), \tilde{\pi}(\cdot|y))p(y)dy$ between the exact and approximate posteriors. We can then use this quantity as a scalar summary of the *average* approximation quality of an approximation scheme. Note that unlike existing "global" diagnostic approaches such as Yao et al. (2018) and Talts et al. (2020), our approach in principle will not be fooled by any approximation error since the averaged f-divergence $\int_{\mathcal{Y}} D_f(\pi(\cdot|y), \tilde{\pi}(\cdot|y))p(y)dy$ is zero if and only if $D_f(\pi(\cdot|y), \tilde{\pi}(\cdot|y)) = 0$ almost everywhere. We summarize the above procedure in Algorithm 4.2. We discuss its connection to existing methods in the next section.

Algorithm 4.2 Diagnostic via f-divergence

Require: Samplers for the prior distribution $\pi(\phi)$, observation model $p(y|\phi)$ and the approximate posterior $\tilde{\pi}(\theta|y)$; Observed data y_{obs}; Discriminator function $g(\theta, y; \omega)$, $\omega \in \mathbb{R}^l$; Sample size $J, N, M \in \mathbb{N}^+$; Generator function f of the chosen f-divergence.
Sample $\{\phi_i, \underline{\theta}_i, y_i\}_{i=1}^N \sim p(\phi, \underline{\theta}, y)$ in 4.20
Compute $\hat{\omega}_N = \arg\min_{\omega \in \mathbb{R}^l} \hat{L}(\omega; \{\phi_i, \underline{\theta}_i, y_i\}_{i=1}^N)$ in (4.25)
Compute $\hat{D}_f(\pi(\cdot|y_{obs}), \tilde{\pi}(\cdot|y_{obs}))$ in 4.26 and/or $\hat{D}_f$ in 4.27
return The estimated discrepancy summary $\hat{D}_f(\pi(\cdot|y_{obs}), \tilde{\pi}(\cdot|y_{obs}))$ at y_{obs} and/or the global summary $\hat{D}_f$.

Note that Algorithm 4.2 is not only applicable to the joint approximate posterior. It also applies to any uni- or multivariate marginals of the approximation posterior as well. The only modification we need is that once we have obtained samples ϕ or $\underline{\theta}$ from $p(\phi, \underline{\theta}, y)$, we only retain the entries in the multivariate ϕ and $\underline{\theta}$ that we are interested in.

4.9.2 Connections to existing works

In this section we discuss the connection between the proposed approach and some existing works.

4.9.2.1 Pareto smoothed importance sampling

In Algorithm 4.2 and Equation 4.26, the estimated density ratios $g(\theta_k, y; \hat{\omega}_N)$ can be viewed as (approximate) Importance weights with the approximate $\tilde{\pi}(\cdot|y)$ being the proposal distribution and the exact $\pi(\cdot|y)$ being the target. Therefore behaviours of these (approximate) Importance weights such as their sample mean or variability can also be used to assess the discrepancy between the exact and approximate posterior conditioned on y. The PSIS diagnostic given in Yao et al. (2018) and Vehtari et al. (2022) utilizes this idea in the context of variational inference. In their setup, both the true and approximate posterior densities can be evaluated up to multiplicative constants, and the unnormalized Importance weights are the density ratios between the variational posterior (the proposal) and the unnormalized true posterior (the target). The main idea of the PSIS diagnostic in Yao et al. (2018) is to first fit a Generalized Pareto distribution (GPD) to a subset of the largest Importance weights, then use the estimated shape parameter $\hat{k}$, which controls the tail of the fitted GPD, as a diagnostic tool. The shape parameter k in a GPD is connected to the $\alpha-$divergence (Rényi, 1961), a special class of f-divergences, between the approximate and the true posterior. In particular, the authors show that $k > 0.5$ implies that the χ^2 divergence $\chi(\pi(\cdot|y), \tilde{\pi}(\cdot|y)) = \infty$ (This also means the variance of the Importance weights with the approximate posterior $\tilde{\pi}(\cdot|y)$ being the proposal, is unbounded), and $k > 1$ implies the KL-divergence $KL(\pi(\cdot|y), \tilde{\pi}(\cdot|y)) = \infty$. However, the diagnostic summary $\hat{k}$ can be fooled easily. As discussed in Yao et al. (2018), if the approximation is slightly over-dispersed compared with the truth, then the corresponding diagnostic summary $\hat{k}$ will not detect this deviation, as the Importance weights are bounded from above. In addition, the PSIS diagnostic $\hat{k}$ is also not able to detect if the approximate posterior misses any modes in the exact posterior, as it is invariant to the scale of the Importance weights. To see this, suppose the true posterior consists of two modes far away from each other while the approximate captures one of them perfectly. In this case, the Importance weights with the approximate being the proposal will have very low variability, leading to a low $\hat{k}$. This will fool the diagnostic procedure. Compared with their approach, our diagnostics $\hat{D}_f(\pi(\cdot|y), \tilde{\pi}(\cdot|y))$ in principle will not be fooled by any approximation error because the f-divergence between the true and approximate posterior is zero if and only if the two distribution are identical almost everywhere.

In addition to diagnostics, Yao et al. (2018) also suggest using PSIS to recalibrate the approximate posterior samples when $\hat{k}$ is reasonably small (Vehtari et al., 2022). It is straightforward to apply this recalibration procedure to our approach using the estimated density ratio $g(\theta, y; \hat{\omega}_N)$ as the approximate IS weights. Such a recalibration

step can also be viewed as one iteration of the Sequential Contrastive Likelihood-free Inference in Durkan et al. (2020) (We will further discuss it in later sections). However, we do not incorporate this step in our Algorithm 4.2 because there is no guarantee that a generic approximation $\tilde{\pi}(\cdot|y)$ and the exact posterior $\pi(\cdot|y)$ will put most of their probability mass on a common region of the parameter space (In contrast, variational inference usually tends to capture the posterior mode reasonably accurately). This assumption is key for an Important sampling "style" correction to be reliable. For example, similar to the last paragraph, suppose the exact posterior conditioned on y is an equally weighted mixture distribution with two components that are far away from each other, and the approximate posterior captures one of the components perfectly. In this case, if the estimated density ratio is reliable, then we expect the approximate Importance weight $g(\theta, y; \hat{\omega}_N) \approx \frac{1}{2}$ for any $\theta \sim \tilde{\pi}(\cdot|y)$. This means the PSIS diagnostic $\hat{k}$ will be small as the variability in the approximate Importance weights is low. In this case, PSIS will recalibrate the approximate posterior samples by giving them almost identical approximate Importance weights, which is pointless and can be misleading. Therefore we only focus on diagnostics in Algorithm 4.2, and choose not to include a recalibration step in it.

4.9.2.2 Distortion map

From Equation 4.2 we see $d_y \circ G_y(\cdot)$ can be viewed as a particular parameterization for estimating the density ratio between the true and approximate posterior (Recall that d_y is the distortion density). In the 1-d case, $\theta \sim \tilde{\pi}(\cdot|y)$ implies $G_y(\theta) \sim UNIF(0, 1)$. Hence if we write $\frac{\pi(\theta|y)}{\tilde{\pi}(\theta|y)} = d_y \circ G_y(\theta), \theta \in \mathbb{R}$, then the f-divergence $D_f(\pi(\cdot|y), \tilde{\pi}(\cdot|y))$ can be written as

$$D_f(\pi(\cdot|y), \tilde{\pi}(\cdot|y)) = \int_0^1 f(d_y(u))du. \tag{4.28}$$

By definition we have $f(1) = 0$, so the f-divergence between the two distributions is solely determined by the deviation between d_y and the constant function 1 over the unit interval. Therefore in this 1-d setup, estimating the f-divergence between the true and approximate distribution is equivalent to checking the discrepancy between the distortion density d_y and the constant function 1.

Compared with distortion map, which is only applicable to uni- or bivariate marginals of the approximation, the proposed approach estimates the density ratio between the true and approximate posterior using a more general parameterization $g(\theta, y; \omega)$, and therefore is applicable to both the marginal and the joint approximate posterior. However, one drawback of the proposed approach is the lack of diagnostic

details. Recall that distortion map is a visual diagnostic, and we are able to read off approximation errors such as positive/negative bias or over-/under-dispersion from the distortion map directly. However, we can not in general infer such details of the approximation error using the proposed approach.

4.9.2.3 Coverage estimate

In Algorithm 4.2, users have the freedom to choose the f-divergence they want to estimate. If we choose to estimate the total variation distance between $\pi(\cdot|y_{obs})$ and $\tilde{\pi}(\cdot|y_{obs})$ (i.e. setting the generator function $f(t) = \frac{1}{2}|t-1|$), then the diagnostic summary $\hat{D}_f(\pi(\cdot|y_{obs}), \tilde{\pi}(\cdot|y_{obs}))$ in Algorithm 4.2 is closely related to the coverage estimation discussed in Chapter 3: Suppose an approximate credible set $\tilde{C}_{y_{obs}}$ with nominal coverage α is given. In Chapter 3 we showed how to estimate the operational coverage achieved by $\tilde{C}_{y_{obs}}$, i.e. the probability of a sample from the *true* posterior $\pi(\theta|y_{obs})$ falls inside the *approximate* credible set $\tilde{C}_{y_{obs}}$. If $\hat{D}_f(\pi(\cdot|y_{obs}), \tilde{\pi}(\cdot|y_{obs}))$ in Algorithm 4.2 is an estimate of the total variation distance $d_{TV}(\pi(\cdot|y_{obs}), \tilde{\pi}(\cdot|y_{obs}))$, then by the definition of the total variation distance, $\hat{D}_f(\pi(\cdot|y_{obs}), \tilde{\pi}(\cdot|y_{obs}))$ can be approximately viewed as the upper bound of the absolute difference between the nominal and the operational coverage achieved by *any* approximate credible set $\tilde{C}_{y_{obs}}$. Hence one may use Algorithm 4.2 as a substitute of the coverage estimation procedures given in Chapter 3.

4.9.2.4 Approximate inference via contrastive learning

The idea of estimating the density ratio using binary classification is closely related to approximate inference schemes based on contrastive learning (Hermans et al., 2020; Durkan et al., 2020; Thomas et al., 2022). Our Algorithm 4.2 can be viewed as one iteration of the Sequential Contrastive Likelihood-free Inference in Durkan et al. (2020) with the approximate $\tilde{\pi}(\theta|y)$ being the proposal distribution. However, the main objective here is not to improve the approximation quality of an approximation scheme, but is to provide a diagnostic tool that checks the performance of a given approximation. Therefore we do not adopt their sequential approach. Hermans et al. (2020) also use the performance of a binary classifier to assess the quality of an approximation scheme, and summarize the diagnostic results using the ROC curve of the fitted classifier. However, their approach only checks the average performance of the approximation scheme over the entire data space $\mathcal{Y}$, and does not provide diagnostic information *at any specific data* y.

4.10 Appendix of Chapter 4

Here we give proofs of Proposition 4.3.1, Lemma 4.3.1, Theorem 4.3.1 and Proposition 4.9.1. The following proposition reproduces a result given in Papamakarios and Murray (2016).

Proposition 4.3.1. *Suppose the set W in (4.6) is non-empty. Let $\{q_i, y_i\} \sim p(y_i)d_{y_i}(q_i)$ independently for $i = 1, ..., N$. Then $N^{-1}\ell(w, \{q_i, y_i\}_{i=1}^N)$ in (4.5) converges in probability to*

$$-E_Y(KL(D_Y(\cdot), D_Y(\cdot; w))) + E_{Q,Y}(\log(d_Y(Q)).$$

This limit function is maximized at $w \in W$.

Proof. Our presentation here is very brief as this result is known. We include this proof outline in order to make the meaning of the proposition clear.

By the WLLN,

$$N^{-1}\ell(w, \{q_i, y_i\}_{i=1}^N) \xrightarrow{P} E_{Q,Y}(\log(d_Y(Q; w))$$

and the first statement follows as

$$\begin{aligned} E_{Q,Y}(\log(d_Y(Q; w)) \;=\; & -E_Y(\mathrm{KL}(D_Y(\cdot), D_Y(\cdot; w))) \\ & +E_{Q,Y}(\log(d_Y(Q)). \end{aligned}$$

The second term does not depend on w so we maximise the scaled limit of the log-likelihood by minimising the KL-divergence. Since $\mathrm{KL}(D_y(\cdot), D_y(\cdot; w^*)) = 0$ for all $y \in \mathcal{Y}$ iff $D_y(q; w^*) = D_Y(q)$ at each q, y, and is otherwise continuous and positive, the limit function is maximised at $w^* \in W$ whenever this set is non-empty. □

The result above shows that the *maximiser of the limit* of the scaled log-likelihood gives the true distortion map. However a proof of consistency must show that the *limit of the maximiser* of the scaled log-likelihood converges in probability to the set of parameter values that express the true distortion map. Standard theory for the MLE does not apply as the true parameter may not be identifiable. The corresponding result for the non-identifiable case was given in Redner et al. (1981). The following proof of consistency is based on that paper.

Lemma 4.3.1. *Under the conditions of Proposition 4.3.1, and the regularity conditions given by Redner et al. (1981), the estimated $D_y(q; \hat{w}_N)$ is consistent, that is*

$$\lim_{N\to\infty} \Pr(|D_y(q; \hat{w}_N) - D_y(q)| > \epsilon) = 0.$$

for any $\epsilon > 0$, $q \in [0, 1]$ and $y \in \mathcal{Y}$.

Proof. Let $W = \{w^* : D_y(\cdot; w^*) = D_y(\cdot),\ y \in \mathcal{Y}\}$. We assume it is non-empty. Let $\tau(\mathbb{R}^m)$ be the quotient topological space defined by taking $\mathbb{R}^m$, choosing a point $W^* \in W$, and identifying all points in W in the original space $\mathbb{R}^m$ with the single point W^* in $\tau(\mathbb{R}^m)$.

The proof is an application of Theorem 4 in Redner et al. (1981) and the continuous mapping theorem. This is not an immediate consequence of standard regularity conditions for the convergence of the MLE, as we do not assume that there is a unique w^* satisfying $D_y(q) = D_y(q; w^*)$, so w^* is not necessarily identifiable. However, the MLE convergence results for a non-identifiable parameter are given in Redner et al. (1981). Recall that W^* is the point in the $\tau(\mathbb{R}^m)$ corresponding to the set W in the original space $\mathbb{R}^m$. By Theorem 4 of Redner et al. (1981), assume all regularity conditions hold, we have $\hat{w}_N \overset{a.s.}{\to} W^*$ as $N \to \infty$.

It then follows from the continuity of $d_y(q; w)$ (and therefore $D_y(q; w)$) and the continuous mapping theorem that, for each pair $\{q, y\} \in [0, 1] \times \mathcal{Y}$,

$$D_y(q; \hat{w}_N) \overset{P}{\to} D_y(q; W^*)$$

and then since $D_y(q; W^*) = D_y(q)$ we have

$$D_y(\cdot; \hat{w}_N) \overset{P}{\to} D_y(\cdot).$$

□

Theorem 4.3.1. *Under the conditions of Lemma 4.3.1 and assuming* $KL(F_y, G_y) > 0$,

$$\Pr(KL(F_y, \hat{F}_y) < KL(F_y, G_y)) \to 1$$

as $N \to \infty$ *for every fixed* y.

Proof. $\hat{F}_y(x) = D_y(G_y(x); \hat{w}_N)$ so the density of $\hat{F}_y$ is

$$\hat{\pi}(x|y) = \tilde{\pi}(x|y) d_y(G_y(x); \hat{w}_N).$$

Recalling $\pi(x|y) = \tilde{\pi}(x|y) d_y(G_y(x))$, we have

$$\begin{aligned}
KL(F_y, \hat{F}_y) &\equiv \int_{-\infty}^{\infty} \pi(x|y) \log\left(\frac{\pi(x|y)}{\hat{\pi}(x|y)}\right) dx \\
&= \int_{-\infty}^{\infty} \tilde{\pi}(x|y) d_y(G_y(x)) \log\left(\frac{d_y(G_y(x))}{d_y(G_y(x); \hat{w}_N)}\right) dx \\
&= \int_0^1 d_y(q) \log\left(\frac{d_y(q)}{d_y(q; \hat{w}_N)}\right) dq \\
&= KL(D_y(\cdot), D_y(\cdot; \hat{w}_N)),
\end{aligned}$$

where we made the change of variables $q = G_y(x)$ to get from the second to third lines. Taking $KL(F_y, G_y) = \epsilon$ with $\epsilon > 0$ we have

$$\Pr(KL(F_y, G_y) > KL(F_y, \hat{F}_y)) = \Pr(\epsilon > KL(D_y(\cdot), D_y(\cdot; \hat{w}_N))).$$

By Lemma 4.3.1, we have $\hat{w}_N \xrightarrow{a.s.} W^*$ and $D_y(\cdot; \hat{w}_N) \xrightarrow{P} D_y(\cdot)$. It is straightforward to verify that the KL-divergence $KL(D_y(\cdot), D_y(\cdot; w))$ is a continuous function w.r.t. w, so $KL(D_y(\cdot), D_y(\cdot; \hat{w}_N)) \to 0$ in probability by the continuous mapping theorem. It follows that the limit as $N \to \infty$ of the quantity on the RHS of the last equality is equal one. □

Theorem 4.3.1 is a fairly natural consequence of Lemma 4.3.1: the procedure is Maximum-Likelihood, satisfies (some rather special) regularity conditions, and is therefore consistent. However we state the result in this form in order to emphasise that Algorithm 4.1 returns a distortion map that moves $\hat{F}_y$ closer to F_y, with high probability for all sufficiently large N, so that the map contains information about the distorting effects of the approximation, without actually sampling F_y, or even making it possible to sample F_y.

Proposition 4.9.1. (Loss function for classification) *Suppose* $J \in \mathbb{N}^+$, $p(\phi, \underline{\theta}, y) = \pi(\phi|y) \prod_{j=1}^{J} \tilde{\pi}(\theta_j|y)p(y)$ *defined in* (4.20). *The loss functional* $L(D(\cdot,\cdot))$ *is minimized at*

$$D^*(\theta, y) = \frac{\pi(\theta|y)p(y)}{\pi(\theta|y)p(y) + \tilde{\pi}(\theta|y)p(y)} \tag{4.29}$$

$$= \sigma\left(r(\theta, y)\right), \tag{4.30}$$

where $\sigma(\cdot)$ *is the sigmoid function and* $r(\theta, y) = \log \frac{\pi(\theta|y)}{\tilde{\pi}(\theta|y)}$ *is the log density ratio.*

Proof. We can rewrite

$$L(D(\cdot,\cdot)) = -E_p\left(\log D(\phi, y) + \frac{1}{J}\sum_{j=1}^{J} \log\left(1 - D(\theta_j, y)\right)\right) \tag{4.31}$$

$$= -\int_{\Theta}\int_{\Theta^J}\int_{\mathcal{Y}}\left[\log D(\phi, y) + \frac{1}{J}\sum_{j=1}^{J} \log\left(1 - D(\theta_j, y)\right)\right] \pi(\phi|y) \prod_{j=1}^{J} \tilde{\pi}(\theta_j|y)p(y) d\theta_{1:J} d\phi dy \tag{4.32}$$

$$= -\int_{\Theta}\int_{\mathcal{Y}} \log D(\phi, y)\pi(\phi|y)p(y) d\phi dy - \frac{1}{J}\sum_{j=1}^{J}\int_{\Theta}\int_{\mathcal{Y}} \log\left(1 - D(\theta_j, y)\right) \tilde{\pi}(\theta_j|y)p(y) d\theta_j dy \tag{4.33}$$

$$= -\int_{\Theta}\int_{\mathcal{Y}} \left[\log D(\theta, y)\pi(\theta|y)p(y) + \log\left(1 - D(\theta, y)\right) \tilde{\pi}(\theta|y)p(y)\right] d\theta dy, \tag{4.34}$$

where E_p means taking expectation with respect to the joint density $p(\phi, \underline{\theta}, y)$. By taking functional derivative w.r.t. D and set it to zero, we see the optimal $D^*(\cdot, \cdot)$ that minimizes the loss functional $L(D(\cdot, \cdot))$ must satisfy

$$0 = \frac{\partial L}{\partial D}\Big|_{D=D^*} \tag{4.35}$$

$$= \frac{\pi(\theta|y)p(y)}{D^*(\theta, y)} - \frac{\tilde{\pi}(\theta|y)p(y)}{1 - D^*(\theta, y)} \tag{4.36}$$

This implies $D^*(\cdot, \cdot)$ that minimizes the loss functional $L(D(\cdot, \cdot))$ takes the from

$$D^*(\theta, y) = \frac{\pi(\theta|y)p(y)}{\pi(\theta|y)p(y) + \tilde{\pi}(\theta|y)p(y)} \tag{4.37}$$

$$= \frac{\pi(\theta|y)}{\pi(\theta|y) + \tilde{\pi}(\theta|y)} \tag{4.38}$$

$$= \sigma\left(r(\theta, y)\right), \tag{4.39}$$

where $\sigma(\cdot)$ is the sigmoid function and $r(\theta, y) = \log \frac{\pi(\theta|y)}{\tilde{\pi}(\theta|y)}$ is the log density ratio. $\square$

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

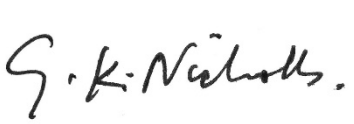
Title of Paper	Distortion estimates for approximate Bayesian inference
Publication Status	√ Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Xing, Hanwen, Geoff Nicholls, and Jeong Kate Lee. "Distortion estimates for approximate Bayesian inference." *Conference on Uncertainty in Artificial Intelligence*. PMLR, 2020.

Student Confirmation

Student Name:	Hanwen Xing		
Contribution to the Paper	I proposed the method and led the design and implementation of it. I also designed and conducted all the simulated and real-world experiments, and contributed to the drafting of the paper.		
Signature 		Date	Sept 29 2022

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Geoff Nicholls		
Supervisor comments		
Signature 	Date	29-09-22

This completed form should be included in the thesis, at the end of the relevant chapter.

Chapter 5

Conclusion and Discussion

In this thesis, we consider the problem of facilitating Bayesian inference using machine learning methods. We give novel computational frameworks for 1) constructing statistically more efficient estimator of Bayes factor and 2) diagnosing approximate error in approximate Bayesian inference. The proposed methods achieve state-of-the-art performance on various simulated and real world examples.

In Chapter 2, we use deep generative models to construct efficient estimators of the ratio of normalizing constants between two unnormalized densities. This is closely related to Bayes factor estimation and Bayesian model selection. We improve the statistical efficiency of the Bridge estimator using a flexible transformation that maps the samples from one probability distribution to the other. This transformation is parameterized as a Normalizing flow and is estimated using the min-max training strategy given in Nowozin et al. (2016). By exploring the connection between Bridge sampling and f-divergence estimation, we show how the variational lower bound of f-divergence in Nguyen et al. (2010) can be used to find Bridge estimators with minimal relative mean square error. The resulting methodology provides not only an accurate estimate of the ratio of normalizing constants, but also a reliable uncertainty quantification of the resulting estimate. Empirically, we show that our approach outperforms existing improvement strategies significantly, and is able to reach a similar level of accuracy using far less samples in both simulated and real world examples. This is especially appealing when the user only have a limited amount of samples from the distributions of interest.

In Chapter 3, we switch the topic to diagnosing approximate error in approximate Bayesian inference. In particular, we focus on calibrating the coverage of approximate credible sets obtained from approximate posteriors: Given an approximate credible set with nominal coverage level $\alpha \in (0, 1)$, we consider two computational strategies to estimate the operational coverage it actually achieves. In both simulated and real

world examples, we demonstrate that our proposed methods are able to accurately estimate the operational coverage of approximate credible sets obtained from various approximation schemes. Given the important role credible sets play in quantifying uncertainty in the parameters of interest, we recommend that wherever an approximate credible set is reported, the estimated operational coverage should be given alongside with its nominal coverage.

In Chapter 4, we change our focus from approximate credible sets to the marginals of a potentially multivariate approximate posterior. We introduce distortion map as a visual diagnostic tool containing easy-to-interpret diagnostic details about the approximation error in the approximate marginals. Distortion map uses its deviation from the identity map to summarize the discrepancy between the true and approximate marginals, so users can read off the type of approximation error directly from the shape of the estimated distortion map. In practice, we show the proposed method is able to accurately identify the approximation error in various approximation schemes such as variational inference and approximate Bayesian computation in both simulated and real world examples. We also show that distortion map is able to identify approximation error that is overlooked by existing diagnostic tools using a real world example. In addition to identifying approximation error in the marginals, we also give strategies that extend distortion map to multivariate cases, and discuss their connection to existing methods.

5.1 Criticism of the proposed methods

In this section we respond to some natural criticism of the methods proposed in the previous chapters.

5.1.1 Diagnostic tools, or alternative approximations?

In Chapter 3 and 4, we focus on diagnostic tools for approximate Bayesian inference. Identifying approximation error in the approximate posteriors necessarily requires some information of the exact posterior. Hence the distinction between diagnosing approximation error and constructing an alternative, potentially more accurate posterior with more computational resource is not always clear. For example, Algorithm 3.2 in Chapter 3 estimates the operational coverage by first generating a collection of weighted samples coming from another ABC-style approximate posterior using Annealed Importance sampling (Neal, 2001). In this case, our diagnostic results for the original approximation is effectively built on how the original approximation differs

from the this new ABC-style approximation, which can be viewed as a "refinement" of the current approximate posterior. The Importance sampling estimator of the operational coverage in Lee et al. (2019) is also based on the same idea. However, this type of diagnostic tools are not ideal: if we are capable of constructing a more reliable approximate posterior, then there is no need to retain the original approximation! To address this issue, we proposed the regression approach for operational coverage estimation (Algorithm 3.3). Here, instead of constructing an alternative, potentially more reliable approximate posterior, we cast the problem of estimating the operational coverage as estimating the posterior expectation of a *scalar* function of the parameter using logistic regression. By doing so, we are able to obtain the diagnostic quantity of interest without reconstruct any alternative joint approximate posteriors. This is a far less challenging task (note that here we summarize the information in the exact posterior as a scalar), especially when the dimension of parameter is high or the true posterior has complicated structures.

The distortion map in Chapter 4 also avoids constructing alternative joint approximate posteriors. Even though we discuss a recalibration step in Chapter 4 that aims to improve the existing approximation, such recalibration or reconstruction step is only a byproduct of our diagnostic procedure, and only acts on the approximate marginals instead of the joint. Additionally, in Algorithm 4.1, we use a very restrictive parameterization to estimate the distortion map. Such restrictive parameterization is sufficient for our diagnostic purpose since we are interested in checking if the distortion map is drastically different from the identity, but not in trying to recover all the details in it. Therefore this choice of parameterization also makes our proposed diagnostic procedure a much simpler task than constructing better approximations of the exact joint posterior.

5.1.2 Computational cost

The methods proposed in this thesis are computationally intensive. For example, in Chapter 2, our purposed method relies on estimating a Normalizing flow model. As we have discussed in Chapter 2.7.1, estimating such a model can be computationally challenging, and the practical implementation of the proposed method heavily relies on software platforms such as Torch (Paszke et al., 2017) and GPU acceleration architecture such as CUDA (NVIDIA et al., 2020). Similarly, in Chapter 3 and 4, the diagnostic tools require calling the approximation scheme repeatedly and estimating separate regression models based on the outcomes of the approximation scheme. This maybe computationally demanding or even infeasible. However, we believe the

computational cost should not stop us from applying and further extending these methods. Firstly, these methods are able to achieve goals that existing methods are not able to. For example, in Chapter 4.6 we show our proposed distortion map is able to identify approximation error that is ignored by existing diagnostic tools using a real world social network model. Secondly, even though our methods are computationally more costly in term of e.g. number of floating number operations or total number of function evaluations, with the help of carefully optimized software platforms and sophisticated parallelism on both CPU and GPU, these methods are able to attain wall-clock running time comparable to existing methods, while outperforming the existing methods in term of accuracy or statistical efficiency. For example, in Chapter 2, even though our proposed f-GB takes roughly 2 to 3 times longer to run compared with e.g. Warp-U (Wang et al., 2022) and GBS (Jia and Seljak, 2020), it is orders-of-magnitude more accurate than these existing methods in both simulated and real world examples. Since computational resources are getting more and more accessible, we believe such gain is attractive to practitioners in fields such as applied Bayesian modelling.

5.2 Future works

In this section, we present some potential directions for future research based on the methods proposed in this thesis.

5.2.1 Extending f-GB to a wider range of estimators

In Chapter 2, we propose an improvement strategy for the optimal Bridge estimator using the connection between the RMSE of the optimal Bridge estimator and a particular f-divergence between the two distributions of interest. This idea can be further extended to other Monte Carlo estimators of the ratio of normalizing constants between two unnormalized densities. For example, Ratio Importance sampling estimator (RIS) (Chen et al., 1997) is an appealing alternative to Bridge estimators. Chen et al. (1997) show that RIS with the optimal proposal distribution has lower asymptotic error than its competitors such as Bridge estimators (Meng and Wong, 1996) and path sampling (Gelman and Meng, 1998). The authors also show that the RMSE of RIS with the optimal proposal distribution can be written as a monotonic function of the total variation distance (recall that it is also an f-divergence) between the two distributions of interest. Therefore in principle, one can improve the efficiency of RIS by finding a transformation T that minimizes the total variation

distance $d_{TV}(q_1^{(T)}, q_2)$ using the same min-max framework given in Chapter 2. However, one main challenge for this extension is that, unlike Bridge estimators, RIS requires sampling from a third proposal distribution which differs from the two distributions of interest. The probability density of the optimal proposal distribution, which leads to the RIS estimator with minimal RMSE, is proportional to the absolute difference between the *normalized* densities of the two distributions of interest, which is in general unavailable. Therefore we are also interested in the problem of approximately sampling form this optimal proposal distribution using deep generative models such as a Normalizing flow.

5.2.2 Diagnostics via classification

In Chapter 4.9 we discuss approximately estimating the density ratio between the true and approximate posterior using binary classification. Intuitively speaking, we expect the classifier to do no better than a random guess when the true and approximate posterior are reasonably close to each other, and to be highly accurate if the true and approximate posterior are far away from each other. Nielsen (2014) formalizes this intuition and show that the misclassification rate of the Bayes classifier can be used to construct an unbiased estimator of the total variation distance between the true and approximate posterior. Therefore we are also interested in constructing diagnostic tools based on the performance of a binary classifier. For example, we may first train a binary classifier that aims to separate the samples from the true and approximate posterior conditioned on any observed value y in a similar fashion to Chapter 4.9, then estimate a separate regression model that smooths the misclassification rate as a diagnostic quantity over the entire data space.

5.3 Concluding remarks

To sum up, we tackle computational challenges arising from Bayesian statistical modelling and inference using machine learning methods. In particular, we consider two topics: Bayes factor estimation and approximation error diagnostics. We show how to utilize methods such as deep generative models and flexible regression models taken from machine learning literature to address these computational challenges. For example, in Chapter 2, we use deep generative models such as Normalizing flow to guide us finding statistically more efficient estimators of Bayes factor. In Chapter 3, we use flexible regression models such as Bayesian additive regression trees to estimate the bias in the coverage of an approximate credible set. We also take special notice to

the applicability of the proposed methods. As a result, the computational tools given in this thesis can also be viewed as generic computational "wrappers" which can be integrated seamlessly into existing inferential pipelines.

We believe the synergy between Bayesian statistical modelling and machine learning will open up vast possibilities for future research directions. Bayesian statistical modelling provides a conceptually principled and natural, but computationally challenging framework for extracting information from observational data and combining it with any prior knowledge. Using novel machine learning techniques and architectures to address the computational challenges in the inferential procedure allows practitioners to tackle more challenging modelling problems, and hence further extend the applicability of Bayesian statistical modelling in different fields. In this thesis, we make contributions to facilitating the Bayesian inferential framework using machine learning techniques and architectures, and we hope it will serve as a stepping stone to a better understanding of the connection between Bayesian statistical modelling and machine learning methods.

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Ali, S. M. and Silvey, S. D. (1966). A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society: Series B (Methodological)*, 28(1):131–142.

Andrieu, C., De Freitas, N., Doucet, A., and Jordan, M. I. (2003). An introduction to mcmc for machine learning. *Machine learning*, 50(1):5–43.

Arjovsky, M. and Bottou, L. (2017). Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*.

Banfield, J. D. and Raftery, A. E. (1992). Ice floe identification in satellite images using mathematical morphology and clustering about principal curves. *Journal of the American Statistical Association*, 87(417):7–16.

Bates, D., Mächler, M., Bolker, B., and Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48.

Beaumont, M. A. (2010). Approximate Bayesian computation in evolution and ecology. *Annual Review of Ecology, Evolution, and Systematics*, 41:379–406.

Bennett, C. H. (1976). Efficient estimation of free energy differences from monte carlo data. *Journal of Computational Physics*, 22(2):245–268.

Besag, J. (1975). Statistical analysis of non-lattice data. *The Statistician*, pages 179–195.

Bouranis, L., Friel, N., and Maire, F. (2017). Efficient Bayesian inference for exponential random graph models by correcting the pseudo-posterior distribution. *Social Networks*, 50:98–108.

Bouranis, L., Friel, N., and Maire, F. (2018). Bayesian model selection for exponential random graph models via adjusted pseudolikelihoods. *Journal of Computational and Graphical Statistics*, 27(3):516–528.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2018). JAX: composable transformations of Python+NumPy programs.

Brehmer, J., Kling, F., Espejo, I., and Cranmer, K. (2020). Madminer: Machine learning-based inference for particle physics. *Computing and Software for Big Science*, 4(1):1–25.

Bridges, M., Feroz, F., Hobson, M., and Lasenby, A. (2009). Bayesian optimal reconstruction of the primordial power spectrum. *Monthly Notices of the Royal Astronomical Society*, 400(2):1075–1084.

Burda, Y., Grosse, R., and Salakhutdinov, R. (2015). Accurate and conservative estimates of mrf log-likelihood using reverse annealing. In *Artificial Intelligence and Statistics*, pages 102–110. PMLR.

Caimo, A. and Friel, N. (2014). Bergm: Bayesian exponential random graphs in R. *Journal of Statistical Software*, 61(2):1–25.

Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., Vogt-Maranto, L., and Zdeborová, L. (2019). Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4):045002.

Chen, M.-H. and Shao, Q.-M. (1997). Estimating ratios of normalizing constants for densities with different dimensions. *Statistica Sinica*, pages 607–630.

Chen, M.-H., Shao, Q.-M., et al. (1997). On monte carlo methods for estimating ratios of normalizing constants. *The Annals of Statistics*, 25(4):1563–1594.

Chib, S. (1995). Marginal likelihood from the gibbs output. *Journal of the american statistical association*, 90(432):1313–1321.

Chipman, H. A., George, E. I., McCulloch, R. E., et al. (2010). BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298.

Chopin, N., Papaspiliopoulos, O., et al. (2020). *An introduction to sequential Monte Carlo*. Springer.

Cook, S. R., Gelman, A., and Rubin, D. B. (2006). Validation of software for Bayesian models using posterior quantiles. *Journal of Computational and Graphical Statistics*, 15(3):675–692.

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., and Bharath, A. A. (2018). Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65.

Dai, C. and Liu, J. S. (2022). Monte carlo approximation of bayes factors via mixing with surrogate distributions. *Journal of the American Statistical Association*, 117(538):765–780.

Dhariwal, P. and Nichol, A. (2021). Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2016). Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*.

Doucet, A., De Freitas, N., Gordon, N. J., et al. (2001). *Sequential Monte Carlo methods in practice*, volume 1. Springer.

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. (2019). Neural spline flows. *arXiv preprint arXiv:1906.04032*.

Durkan, C., Murray, I., and Papamakarios, G. (2020). On contrastive learning for likelihood-free inference. In *International Conference on Machine Learning*, pages 2771–2781. PMLR.

Fitzmaurice, G. M. and Laird, N. M. (1993). A likelihood-based method for analysing longitudinal binary responses. *Biometrika*, 80(1):141–151.

Fourment, M., Magee, A. F., Whidden, C., Bilge, A., Matsen IV, F. A., and Minin, V. N. (2020). 19 dubious ways to compute the marginal likelihood of a phylogenetic tree topology. *Systematic biology*, 69(2):209–220.

Friel, N. and Pettitt, A. N. (2008). Marginal likelihood estimation via power posteriors. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(3):589–607.

Friel, N. and Wyse, J. (2012). Estimating the evidence–a review. *Statistica Neerlandica*, 66(3):288–308.

Frühwirth-Schnatter, S. (2004). Estimating marginal likelihoods for mixture and markov switching models using bridge sampling techniques. *The Econometrics Journal*, 7(1):143–167.

Gelman, A. and Meng, X.-L. (1998). Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical science*, pages 163–185.

Geweke, J. (1999). Using simulation methods for bayesian econometric models: inference, development, and communication. *Econometric reviews*, 18(1):1–73.

Geweke, J. (2004). Getting it right: Joint distribution tests of posterior simulators. *Journal of the American Statistical Association*, 99(467):799–804.

Geyer, C. J. (1994). Estimating normalizing constants and reweighting mixtures. *Technical Report 568, School of Statistics, University of Minnesota*.

Goodfellow, I. (2017). Nips 2016 tutorial: Generative adversarial networks.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Grathwohl, W., Chen, R. T., Bettencourt, J., Sutskever, I., and Duvenaud, D. (2018). Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*.

Greenberg, D., Nonnenmacher, M., and Macke, J. (2019). Automatic posterior transformation for likelihood-free inference. In *International Conference on Machine Learning*, pages 2404–2414. PMLR.

Grover, A., Dhar, M., and Ermon, S. (2018). Flow-gan: Combining maximum likelihood and adversarial learning in generative models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Gui, J., Sun, Z., Wen, Y., Tao, D., and Ye, J. (2021). A review on generative adversarial networks: Algorithms, theory, and applications. *IEEE Transactions on Knowledge and Data Engineering*.

Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304.

Hermans, J., Begy, V., and Louppe, G. (2020). Likelihood-free mcmc with amortized approximate ratio estimators. In *International Conference on Machine Learning*, pages 4239–4248. PMLR.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347.

Höglund, M., Frigyesi, A., and Mitelman, F. (2006). A gene fusion network in human neoplasia. *Oncogene*, 25(18):2674.

Hunter, D. R. and Handcock, M. S. (2006). Inference in curved exponential family models for networks. *Journal of Computational and Graphical Statistics*, 15(3):565–583.

Jaakkola, T. S. and Jordan, M. I. (1997). A variational approach to Bayesian logistic regression models and their extensions. In *Sixth International Workshop on Artificial Intelligence and Statistics*, pages 283–294. PMLR.

Jennrich, R. I. (1969). Asymptotic properties of non-linear least squares estimators. *The Annals of Mathematical Statistics*, 40(2):633–643.

Jia, H. and Seljak, U. (2020). Normalizing constant estimation with gaussianized bridge sampling. In *Symposium on Advances in Approximate Bayesian Inference*, pages 1–14. PMLR.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233.

Kahn, H. (1950). Random sampling (monte carlo) techniques in neutron attenuation problems. *Nucleonics*, 6(5):27–passim.

Kang, B., Hughes, J., and Haran, M. (2021). Diagnostics for monte carlo algorithms for models with intractable normalizing functions. *arXiv preprint arXiv:2109.05121*.

Kapelner, A. and Bleich, J. (2016). bartMachine: Machine learning with Bayesian additive regression trees. *Journal of Statistical Software*, 70(4):1–40.

Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795.

Kaufman, B. (1949). Crystal statistics. II. partition function evaluated by spinor analysis. *Physical Review*, 76(8):1232.

Kingma, D., Salimans, T., Poole, B., and Ho, J. (2021). Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. (2016). Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29:4743–4751.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kong, A., McCullagh, P., Meng, X.-L., Nicolae, D., and Tan, Z. (2003). A theory of statistical models for monte carlo integration. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(3):585–604.

Kunegis, J. (2013). KONECT: The Koblenz network collection. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13 Companion, pages 1343–1350, New York, NY, USA. ACM.

Kuśmierczyk, T., Sakaya, J., and Klami, A. (2019). Variational Bayesian decision-making for continuous utilities. In *Advances in Neural Information Processing Systems*, pages 6392–6402.

Lacoste-Julien, S., Huszár, F., and Ghahramani, Z. (2011). Approximate inference for the loss-calibrated Bayesian. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 416–424.

Landau, D. and Binder, K. (2021). *A guide to Monte Carlo simulations in statistical physics*. Cambridge university press.

Lartillot, N. and Philippe, H. (2006). Computing bayes factors using thermodynamic integration. *Systematic biology*, 55(2):195–207.

Le Cam, L. M. (1969). Théorie asymptotique de la décision statistique. *Presses de l'Université de Montréal*.

Lee, J. E., Nicholls, G. K., and Ryder, R. J. (2019). Calibration procedures for approximate bayesian credible sets. *Bayesian Analysis*, 14(4):1245–1269.

Lin, J. (1991). Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151.

Lueckmann, J.-M., Goncalves, P. J., Bassetto, G., Öcal, K., Nonnenmacher, M., and Macke, J. H. (2017). Flexible statistical inference for mechanistic models of neural dynamics. In *Advances in Neural Information Processing Systems*, pages 1289–1299.

Lunn, D. J., Thomas, A., Best, N., and Spiegelhalter, D. (2000). Winbugs-a bayesian modelling framework: concepts, structure, and extensibility. *Statistics and computing*, 10(4):325–337.

Malsiner-Walli, G., Pauger, D., and Wagner, H. (2018). Effect fusion using model-based clustering. *Statistical Modelling*, 18(2):175–196.

Marin, J.-M., Pudlo, P., Estoup, A., and Robert, C. (2018). Likelihood-free model choice. *Handbook of Approximate Bayesian Computation*, page 153.

Menendez, P., Fan, Y., Garthwaite, P., and Sisson, S. (2014). Simultaneous adjustment of bias and coverage probabilities for confidence intervals. *Computational Statistics & Data Analysis*, 70:35 – 44.

Meng, X.-L. and Schilling, S. (1996). Fitting full-information item factor models and an empirical investigation of bridge sampling. *Journal of the American Statistical Association*, 91(435):1254–1267.

Meng, X.-L. and Schilling, S. (2002). Warp bridge sampling. *Journal of Computational and Graphical Statistics*, 11(3):552–586.

Meng, X.-L. and Wong, W. H. (1996). Simulating ratios of normalizing constants via a simple identity: a theoretical exploration. *Statistica Sinica*, pages 831–860.

Metz, L., Poole, B., Pfau, D., and Sohl-Dickstein, J. (2017). Unrolled generative adversarial networks. In *5th International Conference on Learning Representations, ICLR, Toulon, France*.

Monahan, J. F. and Boos, D. D. (1992). Proper likelihoods for Bayesian analysis. *Biometrika*, 79(2):271–278.

Murray, I., Ghahramani, Z., and MacKay, D. J. C. (2006). MCMC for doubly-intractable distributions. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, pages 359–366. AUAI Press.

Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing*, 11(2):125–139.

Neal, R. M. et al. (2011). Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2.

Newey, W. K. and McFadden, D. (1994). Large sample estimation and hypothesis testing. *Handbook of econometrics*, 4:2111–2245.

Nguyen, X., Wainwright, M. J., and Jordan, M. I. (2010). Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861.

Nielsen, F. (2014). Generalized bhattacharyya and chernoff upper bounds on bayes error using quasi-arithmetic means. *Pattern Recognition Letters*, 42:25–34.

Nowozin, S., Cseke, B., and Tomioka, R. (2016). f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, pages 271–279.

NVIDIA, Vingelmann, P., and Fitzek, F. H. (2020). Cuda, release: 10.2.89.

Onken, D., Fung, S. W., Li, X., and Ruthotto, L. (2021). Ot-flow: Fast and accurate continuous normalizing flows via optimal transport. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9223–9232.

Overstall, A. M. and Forster, J. J. (2010). Default bayesian model determination methods for generalised linear mixed models. *Computational Statistics & Data Analysis*, 54(12):3269–3288.

Papamakarios, G. and Murray, I. (2016). Fast ε-free inference of simulation models with Bayesian conditional density estimation. In *Advances in Neural Information Processing Systems*, pages 1028–1036.

Papamakarios, G., Nalisnick, E. T., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.*, 22(57):1–64.

Papamakarios, G., Pavlakou, T., and Murray, I. (2017). Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pages 2338–2347.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch. In *NIPS 2017 Workshop on Autodiff*.

Pinsker, M. S. (1964). Information and information stability of random variables and processes. *Holden-Day*.

Prangle, D., Blum, M. G., Popovic, G., and Sisson, S. (2014). Diagnostic tools for approximate Bayesian computation using the coverage property. *Australian & New Zealand Journal of Statistics*, 56(4):309–329.

Price, L. F., Drovandi, C. C., Lee, A., and Nott, D. J. (2018). Bayesian synthetic likelihood. *Journal of Computational and Graphical Statistics*, 27(1):1–11.

Pritchard, J. K., Seielstad, M. T., Perez-Lezaun, A., and Feldman, M. W. (1999). Population growth of human Y chromosomes: a study of Y chromosome microsatellites. *Molecular Biology and Evolution*, 16(12):1791–1798.

Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Raftery, A. E. (1999). Bayes factors and BIC: Comment on "A critique of the Bayesian information criterion for model selection". *Sociological Methods & Research*, 27(3):411–427.

Ranganath, R., Gerrish, S., and Blei, D. (2014). Black box variational inference. In *Artificial intelligence and statistics*, pages 814–822. PMLR.

Raynal, L., Marin, J.-M., Pudlo, P., Ribatet, M., Robert, C. P., and Estoup, A. (2019). ABC random forests for Bayesian parameter inference. *Bioinformatics*, 35(10):1720–1728.

Redner, R. et al. (1981). Note on the consistency of the maximum likelihood estimate for nonidentifiable distributions. *The Annals of Statistics*, 9(1):225–228.

Rényi, A. (1961). On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 547–561. University of California Press.

Rezende, D. J. and Mohamed, S. (2015). Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR.

Robins, G., Pattison, P., Kalish, Y., and Lusher, D. (2007). An introduction to exponential random graph (p^*) models for social networks. *Social networks*, 29(2):173–191.

Rodrigues, G., Prangle, D., and Sisson, S. A. (2018). Recalibration: A post-processing method for approximate Bayesian computation. *Computational Statistics & Data Analysis*, 126:53–66.

Rothfuss, J., Ferreira, F., Walther, S., and Ulrich, M. (2019). Conditional density estimation with neural networks: Best practices and benchmarks. *arXiv preprint arXiv:1903.00954*.

Skilling, J. et al. (2006). Nested sampling for general bayesian computation. *Bayesian analysis*, 1(4):833–859.

Sturtz, S., Ligges, U., and Gelman, A. (2005). R2winbugs: a package for running winbugs from r. *Journal of Statistical software*, 12:1–16.

Talts, S., Betancourt, M., Simpson, D., Vehtari, A., and Gelman, A. (2020). Validating Bayesian inference algorithms with simulation-based calibration. *arXiv preprint arXiv:1804.06788v2*.

Tan, L. S. and Friel, N. (2020). Bayesian variational inference for exponential random graph models. *Journal of Computational and Graphical Statistics*, 29(4):910–928.

Thomas, O., Dutta, R., Corander, J., Kaski, S., and Gutmann, M. U. (2022). Likelihood-free inference by ratio estimation. *Bayesian Analysis*, 17(1):1–31.

Tran, D., Vafa, K., Agrawal, K., Dinh, L., and Poole, B. (2019). Discrete flows: Invertible generative models of discrete data. In *Advances in Neural Information Processing Systems*, pages 14719–14728.

Trippe, B. L. and Turner, R. E. (2018). Conditional density estimation with bayesian normalising flows. *arXiv preprint arXiv:1802.04908*.

Uehara, M., Sato, I., Suzuki, M., Nakayama, K., and Matsuo, Y. (2016). Generative adversarial nets from a density ratio estimation perspective. *arXiv preprint arXiv:1610.02920*.

Vehtari, A., Simpson, D., Gelman, A., Yao, Y., and Gabry, J. (2022). Pareto smoothed importance sampling. *arXiv preprint arXiv:1507.02646*.

Voter, A. F. (1985). A monte carlo method for determining free-energy differences and transition state theory rate constants. *The Journal of chemical physics*, 82(4):1890–1899.

Wang, L., Jones, D. E., and Meng, X.-L. (2022). Warp bridge sampling: The next generation. *Journal of the American Statistical Association*, 117(538):835–851.

Wong, J. S., Forster, J. J., and Smith, P. W. (2020). Properties of the bridge sampler with a focus on splitting the mcmc sample. *Statistics and Computing*, pages 1–18.

Wood, S. N. (2010). Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466(7310):1102.

Xing, H. (2022). Improving bridge estimators via f-gan. *Statistics and Computing*, 32(72).

Xing, H., Nicholls, G., and Lee, J. (2019). Calibrated approximate bayesian inference. In *International Conference on Machine Learning*, pages 6912–6920. PMLR.

Xing, H., Nicholls, G., and Lee, J. K. (2020). Distortion estimates for approximate bayesian inference. In *Conference on Uncertainty in Artificial Intelligence*, pages 1208–1217. PMLR.

Yao, Y., Vehtari, A., Simpson, D., and Gelman, A. (2018). Yes, but did it work?: Evaluating variational inference. In *International Conference on Machine Learning*, pages 5581–5590. PMLR.

Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):452–473.

Zhang, J., Mercado, R., Engkvist, O., and Chen, H. (2021). Comparative study of deep generative models on chemical space coverage. *Journal of Chemical Information and Modeling*, 61(6):2572–2581.

Zhao, D., Dalmasso, N., Izbicki, R., and Lee, A. B. (2021). Diagnostics for conditional density models and bayesian inference algorithms. In *Uncertainty in Artificial Intelligence*, pages 1830–1840. PMLR.