Master's thesis

Master's Programme in Data Science

Differentially Private Metropolis-Hastings Algorithms

Ossi Räisä

March 28, 2021

Supervisor(s): Associate Professor Antti Honkela

Examiner(s): Associate Professor Antti Honkela
Doctor Antti Koskela

UNIVERSITY OF HELSINKI
FACULTY OF SCIENCE

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty Faculty of Science	Koulutusohjelma — Utbildningsprogram — Degree programme Master’s Programme in Data Science

Tekijä — Författare — Author

Ossi Räisä

Työn nimi — Arbetets titel — Title

Differentially Private Metropolis-Hastings Algorithms

Työn laji — Arbetets art — Level Master’s thesis	Aika — Datum — Month and year March 28, 2021	Sivumäärä — Sidantal — Number of pages 82

Tiivistelmä — Referat — Abstract

Differential privacy has over the past decade become a widely used framework for privacy-preserving machine learning. At the same time, Markov chain Monte Carlo (MCMC) algorithms, particularly Metropolis-Hastings (MH) algorithms, have become an increasingly popular method of performing Bayesian inference. Surprisingly, their combination has not received much attention in the literature. This thesis introduces the existing research on differentially private MH algorithms, proves tighter privacy bounds for them using recent developments in differential privacy, and develops two new differentially private MH algorithms: an algorithm using subsampling to lower privacy costs, and a differentially private variant of the Hamiltonian Monte Carlo algorithm. The privacy bounds of both new algorithms are proved, and convergence to the exact posterior is proven for the latter. The performance of both the old and the new algorithms is compared on several Bayesian inference problems, revealing that none of the algorithms is clearly better than the others, but subsampling is likely only useful to lower computational costs.

ACM Computing Classification System (CCS):
Mathematics of computing→Probability and statistics→Probabilistic reasoning algorithms
→Markov-chain Monte Carlo methods→Metropolis-Hastings algorithm
Security and privacy→Security Services→Privacy-preserving protocols

Avainsanat — Nyckelord — Keywords

Differential privacy, Metropolis-Hastings, Markov chain Monte Carlo, Hamiltonian Monte Carlo

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Contents

1 Introduction 1

2 Background 5

2.1 Differential Privacy 5

2.2 Bayesian Inference and the Metropolis-Hastings Algorithm 9

2.3 Measure Theory 11

3 Differentially Private MH 23

3.1 Differentially Private MH with the Penalty Algorithm 23

3.2 Applying New Techniques to DP Penalty 25

3.3 The Barker Acceptance Test 27

3.4 The Penalty Algorithm with Subsampling 29

4 Differentially Private Hamiltonian Monte Carlo 35

4.1 MCMC with Hamiltonian Dynamics 35

4.2 Differential Privacy with HMC 39

5 Experimental Setup 49

5.1 The Gaussian Model 49

5.2 The Banana Distribution 49

5.3 Circle Model 50

5.4 Practicalities of Running the Comparison 52

5.5 Maximum Mean Discrepancy 56

6 Experiments 57

6.1 Comparing Privacy Accounting Methods 57

6.2 The Effects of Clipping 57

6.3 Comparison of DP MH Algorithms 59

7 Conclusions 67

Bibliography 71

Appendix A Proof of Theorem 5.2 77

Appendix B Differentiability of the clip**-function** 81

1. Introduction

As the availability of data on private individuals grows ever larger, both the potential gains from analysing the data, and the risks associated with the analysis, grow. This makes analysis techniques that can ensure the privacy of the individuals important. Differential privacy [DMNS06] is a formal definition attempting to capture the notion of a data analysis algorithm that preserves privacy. Metropolis-Hastings (MH) algorithms [MRR+53, Has70] are a particular class of Markov chain Monte Carlo (MCMC) [RC04] algorithms. MCMC is a computational method that enables data analysis using Bayesian inference [GCS+14]. This thesis studies the combination of differential privacy and MH algorithms by introducing existing algorithms, proving tighter privacy bounds for them, and developing two novel algorithms. Finally, both the existing and the new algorithms are compared on a variety of settings.

Traditional approaches in preserving the privacy of data subjects in data analysis, such as simply omitting identifying information, or more advanced techniques such as k-anonymity [SS98, Sam01], can lead to compromises of private information. These leaks can result from the combination of an adversary having access to additional data and linking parts of the additional data and the private data, or the adversary having access to many public results that use the same private dataset [DN03].

Differential privacy [DMNS06] is a notion of an algorithm that preserves privacy by making the data analysis algorithm noisy, thus masking the details of the private input data that are necessary for identifying individuals in the data. The idea of differential privacy is to require that small changes in the input data can only cause small changes in the probability distribution of the output. This makes differential privacy immune to any post-processing and auxiliary information, and allows quantifying the loss in privacy from releasing multiple results based on the same dataset, or for an individual, the privacy loss from participating in multiple studies.

The tradeoff with differential privacy is the noise that must be added to the analysis process, which will decrease the utility of the analysis. For publishing aggregate summaries of large datasets, the amount of noise required is not too large, and differential privacy has been used in such settings in practice by Apple [DPT17], Microsoft [DKY17], the U. S. Census Bureau [Abo18], and others. However, for small datasets, or applications requiring detailed information on individuals, such as COVID-19 tracing apps, the amount of noise is likely to be prohibitively large, so other techniques for maintaining the privacy of individuals are required.

Bayesian inference [GCS+14] is a paradigm of statistical inference where the parameters of a statistical model are considered random variables, and the language of probability is used to describe uncertainty about the values of the parameters. Bayesian inference is based on Bayes' theorem, which allows a data analyst to update his prior knowledge of the parameters based on observed data to obtain a posterior distribution of the parameters. Computing the posterior distribution analytically is often not possible, so numerical methods are needed. Markov chain Monte Carlo (MCMC) is a general method of drawing samples from a given distribution, such as the posterior distribution. Metropolis-Hastings algorithms are a particular class of MCMC algorithms, where the sampling is done by first picking a proposal from a given proposal distribution, and then either choosing to accept or reject the proposal based on an acceptance test. The sample allows the analyst to gain information about the model parameters through the data. MH algorithms only require knowing an unnormalised form of the target distribution, which enables their use on a wide variety of models, as computing the unnormalised posterior is often much easier than computing the normalised posterior.

Combining MH algorithms with differential privacy allows an analyst to conduct the comprehensive Bayesian inference that is possible using MH while preserving the privacy of the data subjects through differential privacy. One differentially private MH algorithm has been developed: the DP penalty algorithm of Yildirim and Ermis [YE19]. A closely related algorithm is the DP Barker algorithm of Heikkilä et al. [HJDH19], which uses the Barker acceptance [Bar65] test instead of the MH acceptance test that MH algorithms use. Although DP Barker is not technically an MH algorithm, it is much close to DP penalty than other DP MCMC algorithms due to the acceptance test it performs.

DP penalty builds on the earlier penalty algorithm [CD99], which was developed for physical simulations where it is difficult to compute even the unnormalised form of the target distribution, so the target must be approximated. DP Barker builds on an earlier algorithm that only uses a part of its input data at a time to reduce computational costs [SPCC17], which itself approximates the Barker algorithm [Bar65] to correct for the error from not using the full data. In DP Barker, not using the full data additionally amplifies the privacy of the algorithm, allowing it to run longer for a given privacy budget. Of these, the DP penalty algorithm is considerably simpler and more extensible, so the further development of differentially private MCMC algorithms in this thesis uses it as the basis.

In 2019, Sommer et al. [SMM19] developed a technique of computing the privacy of iterative algorithms that allows running the algorithms longer than the privacy accounting methods used by Yildirim and Ermis [YE19] and Heikkilä et al. [HJDH19]. This technique is applicable to the DP penalty algorithm, but not the DP Barker algorithm, although new developments of the technique [KJH20] may be applicable to it.

As an MH algorithm, DP penalty is very simple, so developing differentially private variants of more advanced MH algorithms may be fruithful. This thesis develops a differentially private version of Hamiltonian Monte Carlo (HMC) [Nea12, DKPR87], which is able to make better use of the structure of the statistical model being sampled than DP penalty. However, with differential privacy, this comes with a privacy cost, so it is not clear that differentially private HMC is more effective than DP penalty.

Another way to lower the privacy cost of a differentially private algorithm is only using a portion of the data at each iteration, instead of the full dataset. This technique, called subsampling or minibatching, is used by DP Barker, but it is also applicable to DP penalty. As subsampling will decrease the accuracy of the algorithm, it is not guaranteed to improve results over existing algorithms.

There are other frameworks for DP Bayesian inference than DP MH, but none of them offers the full generality and theoretical guarantees that DP MH has. DP variational inference [JHD17] approximates the posterior distribution with a tractable distribution, but the approximating distribution may not be able to match the posterior. For exponential family [BS18] and generalized linear models [KJK+20], differentially private inference is possible by perturbing sufficient statistics or posterior parameterisations, but these sets of models are limited. Simply sampling from the exact posterior is differentially private under suitable conditions [DNZ+17, WFS15, ZRD16], but for many models, exact sampling of the posterior is not possible.

Several non-MH DP MCMC algorithms [WFS15, LCLC19] based on stochastic gradient MCMC algorithms [WT11, CFG14, DFB+14] have been developed. First developed in the non-private setting, stochastic gradient MCMC algorithms use the gradients of the target density to draw efficient proposals. To save computation time, the gradients are only computed over a subsample of the data, and the acceptance test is omitted, which introduces bias to the resulting posterior [Bet15], especially in high dimensions. The differentially private variants of these algorithms [WFS15, LCLC19] greatly benefit from privacy amplification from subsampling and and the lowered privacy cost from omitting the acceptance test, but they inherit the bias from omitting the acceptance test, so all of the algorithm in this thesis include the acceptance test.

This thesis begins with a more detailed, but nevertheless brief, introduction to differential privacy and MH in Chapter 2. Chapter 3 considers the simpler differentially private MH algorithms: the existing algorithms, DP penalty and DP Barker, are introduced in Sections 3.1 and 3.3, the tighter privacy bound for DP penalty is developed in Section 3.2 and the minibatch DP penalty algorithm is developed in Section 3.4. Chapter 4 first introduces HMC in Section 4.1 and develops differentially private HMC in Section 4.2. The performance of the algorithms is compared on a wide variety of models. The models, and technical details concerning the comparison, are discussed in Chapter 5. The results of the experiments are presented in Chapter 6.

2. Background

This chapter covers the basics of differential privacy, Bayesian inference, Markov chain Monte Carlo algorithms and measure theory needed in the later chapters. To keep the introduction brief, only directly needed concepts are covered, and much of the motivation behind the subjects is left out.

2.1 Differential Privacy

Differential privacy (DP) [DMNS06, DR14] is a property of an algorithm that quantifies the amount of information about private data an adversary can gain from the publication of the algorithm's output. The most commonly used definition uses two real numbers, ϵ and δ, to quantify the information gain, or, from the perspective of a data subject, the privacy loss of the algorithm. DP algorithms must necessarily[†] include randomness to mask influence of the private data, so all of the considered algorithms in this thesis are randomised. DP randomised algorithms are also called *mechanisms* in this thesis and in the DP literature.

The most common definition is called (ϵ, δ)-approximate differential privacy (ADP) [DKM+06, DR14]. The case where $\delta = 0$ is called ϵ-DP or pure DP.

Definition 2.1. *A mechanism $\mathcal{M}\colon \mathcal{X} \to \mathcal{U}$ is (ϵ, δ)-ADP if for all neighbouring inputs $x \in \mathcal{X}$ and $x' \in \mathcal{X}$ and all measurable sets $S \subset \mathcal{U}$*

$$P(\mathcal{M}(x) \in S) \leq e^{\epsilon} P(\mathcal{M}(x') \in S) + \delta.$$

The neighbourhood relation in the definition is domain specific. With tabular data the most common definitions are the add/remove neighbourhood and substitute neighbourhood.

Definition 2.2. *Two tabular datasets are said to be add/remove neighbours if they are equal after adding or removing at most one row to or from one of them. The datasets are said to be in substitute neighbours if they are equal after changing at most one row in one of them.*

[†] Unless the algorithm does not actually use the private data.

The neighbourhood relation is denoted by $\sim$. The definitions and theorems of this section are valid for all neighbourhood relations, but later chapters use the substitute neighborhood relation.

There many other definitions of differential privacy that are mostly used to compute (ϵ, δ)-bounds for ADP. This thesis uses two of them: Rényi-DP (RDP) [Mir17] and zero-concentrated differential privacy (zCDP) [BS16]. Both are based on Rényi divergence [Mir17], which is a particular way of measuring the distance* between random variables.

Definition 2.3. *For random variables with density or probability mass functions P and Q, the Rényi divergence of order $1 < \alpha < \infty$ is*

$$D_\alpha(P \,||\, Q) = \frac{1}{\alpha - 1} \ln E_{x \sim Q} \left(\frac{P(x)^\alpha}{Q(x)^\alpha} \right).$$

Orders $\alpha = 1$ and $\alpha = \infty$ are defined by continuity:

$$D_1(P \,||\, Q) = \lim_{\alpha \to 1-} D_\alpha(P \,||\, Q),$$

$$D_\infty(P \,||\, Q) = \lim_{\alpha \to \infty} D_\alpha(P \,||\, Q).$$

Both RDP and zCDP can be expressed as bounds on the Rényi divergence between the outputs of a randomised algorithm with neighbouring inputs:

Definition 2.4. *A mechanism $\mathcal{M}$ is (α, ϵ)-RDP if for all $x \sim x'$*

$$D_\alpha(\mathcal{M}(x) \,||\, \mathcal{M}(x')) \leq \epsilon.$$

$\mathcal{M}$ is ρ-zCDP if for all $\alpha > 1$ and all $x \sim x'$

$$D_\alpha(\mathcal{M}(x) \,||\, \mathcal{M}(x')) \leq \rho\alpha.$$

Rényi-DP and zCDP bounds can be converted to ADP bounds [Mir17, BS16]:

Theorem 2.5. *If a mechanism $\mathcal{M}$ is (α, ϵ)-RDP, $\mathcal{M}$ is also $(\epsilon - \frac{\ln \delta}{\alpha - 1}, \delta)$-ADP for any $0 < \delta < 1$. If $\mathcal{M}$ is ρ-zCDP, $\mathcal{M}$ is also $(\rho + \sqrt{-4\rho \ln \delta}, \delta)$-ADP for any $0 < \delta < 1$.*

A very useful property of all of these definitions is composition [DR14]: if mechanisms $\mathcal{M}$ and $\mathcal{M}'$ are DP, the mechanism first computing $\mathcal{M}$ and then $\mathcal{M}'$, outputting both results, is also DP, although with worse bounds. More precisely:

Definition 2.6. *Let $\mathcal{M} \colon \mathcal{X} \to \mathcal{U}$ and $\mathcal{M}' \colon \mathcal{X} \times \mathcal{U} \to \mathcal{U}'$ be mechanisms. Their composition is the mechanism outputting $(\mathcal{M}(x), \mathcal{M}'(x, \mathcal{M}(x)))$ for input x.*

[*] Statistical divergences are commonly called distances, even though they typically are not metrics.

Theorem 2.7. *Let $\mathcal{M}\colon \mathcal{X} \to \mathcal{U}$ and $\mathcal{M}\colon \mathcal{X} \times \mathcal{U} \to \mathcal{U}'$ be mechanisms. Then*

1. *If $\mathcal{M}$ is (ϵ, δ)-ADP and $\mathcal{M}'$ is (ϵ', δ')-ADP, then their composition is $(\epsilon+\epsilon', \delta+\delta')$-ADP [DKM+06]*

2. *If $\mathcal{M}$ is (α, ϵ)-RDP and $\mathcal{M}'$ is (α, ϵ')-RDP, then their composition is $(\alpha, \epsilon + \epsilon')$-RDP [Mir17]*

3. *If $\mathcal{M}$ is ρ-zCDP and $\mathcal{M}'$ is ρ'-zCDP, then their composition is $(\rho + \rho')$-zCDP [BS16]*

All of the composition results can be extended to any number of compositions by induction. Note that any step of the composition can depend on the results of the previous steps, not only on the private data. There are also other composition theorems for ADP that trade increased δ for decreased ϵ or vice-versa, but this thesis does not apply them directly.

As any randomised algorithm that does not use private data in any way is $(0, 0)$-ADP, 0-zCDP and $(\alpha, 0)$-RDP with all α, Theorem 2.7 has the following corollary, called post-processing immunity:

Theorem 2.8. *Let $\mathcal{M}\colon \mathcal{X} \to \mathcal{U}$ be an (ϵ, δ)-ADP, (α, ϵ)-RDP or ρ-zCDP mechanism. Let $f\colon \mathcal{U} \to \mathcal{U}'$ be any randomised algorithm not using the private data. Then the composition of $\mathcal{M}$ and f is (ϵ, δ)-ADP, (α, ϵ)-RDP or ρ-zCDP.*

There are many different DP mechanisms that are commonly used [DR14]. This thesis only requires one of the most commonly used ones: the Gaussian mechanism [DKM+06].

Definition 2.9. *The Gaussian mechanism with parameter σ^2 is a randomised algorithm that, with input data x and query $f\colon \mathcal{X} \to \mathbb{R}^d$, outputs a sample from $\mathcal{N}(f(x), \sigma^2)$, where $\mathcal{N}$ denotes the normal distribution.*

The privacy bounds of the Gaussian mechanism require that the values of the query f do not vary too much for neighbouring inputs. This requirement is formalised as a bound on the *sensitivity* of f.

Definition 2.10. *The l_p-sensitivity Δ_p, with neighbourhood relation $\sim$, of a function $f\colon \mathcal{X} \to \mathbb{R}^d$ is*

$$\Delta_p f = \sup_{x \sim x'} ||f(x) - f(x')||_p.$$

The RDP and zCDP bounds for the Gaussian mechanism are quite simple. The ADP bound is more complicated:

Theorem 2.11. *If $\Delta_2 f \leq \Delta$, the Gaussian mechanism for query f with noise variance σ is*

1. $(\alpha, \frac{\alpha\Delta^2}{2\sigma^2})$*-RDP [Mir17]*
2. $\frac{\Delta^2}{2\sigma^2}$*-zCDP [BS16]*
3. *k compositions of the Gaussian mechanism, with queries f_i, where $\Delta_2 f_i \leq \Delta$ for $1 \leq i \leq k$, are $(\epsilon, \delta(\epsilon))$-ADP [SMM19] with*

$$\delta(\epsilon) = \frac{1}{2}\left(\operatorname{erfc}\left(\frac{\sigma(\epsilon - k\mu)}{\sqrt{2k}\Delta}\right) - e^{\epsilon}\operatorname{erfc}\left(\frac{\sigma(\epsilon + k\mu)}{\sqrt{2k}\Delta}\right)\right),$$

where $\mu = \frac{\Delta^2}{2\sigma^2}$ and erfc *is the complementary error function.*

Additionally, each of the above privacy bounds is tight, meaning that the Gaussian mechanism with query f is not (α, ϵ')-RDP for any $\epsilon' < \frac{\alpha\Delta^2}{2\sigma^2}$, ρ'-zCDP for any $\rho < \frac{\Delta^2}{2\sigma^2}$, and a composition of k Gaussian mechanisms with queries f_i is not $(\epsilon, \delta'(\epsilon))$-ADP for any $\delta'(\epsilon) < \delta(\epsilon)$.

Theorem 2.11 implies that the value of any function with finite l_2-sensitivity can be privately released using the Gaussian mechanism with appropriate noise variance σ^2. Of course, the utility of the released value depends on the magnitude of σ^2 compared to the actual value. In the ADP bound of Theorem 2.11, as in Definition 2.6, each function f_i can depend on the output of the previous functions f_j, $j < i$.

Note that while the RDP and zCDP bounds of Theorem 2.11 are tight, the conversion from RDP or zCDP to ADP with Theorem 2.5 is not tight. As a result, computing ADP bounds for the Gaussian mechanism through RDP or zCDP, and converting the resulting bounds to ADP with Theorem 2.5 does not result in a tight bound. The difference of the two methods of computing ADP bounds on the MCMC algorithms considered in this thesis is shown in Section 6.1.

When the dataset is a table, the query f of the Gaussian mechanism is typically a sum the values of a function g for each row of the table, specifically

$$f(X) = \sum_{x \in X} g(x)$$

for dataset X. For the substitute neighborhood relation f,

$$\Delta_p f = \sup_{X \sim X'} ||f(X) - f(X')||_p = \sup_{x, x'} ||g(x) - g(x')||_p = \Delta_p^* g,$$

where x and x' are the values of the differing row in X and X', and Δ_p^* denotes sensitivity with the neighborhood relation that considers all values neighboring.

In case $\Delta_p^* g$ is not finite, g can still be used with the Gaussian mechanism if *clipping* is used. Clipping restricts the values of g to a maximum l_p-norm. Specifically, clipping g with the bound $b \in \mathbb{R}$ applies the function $\operatorname{clip}_b^p \colon \mathbb{R}^d \to \mathbb{R}^d$ where

$$\operatorname{clip}_b^p(y) = \begin{cases} \frac{y}{||y||_p} \min\{||y||_p, b\} & \text{if } y \neq 0 \\ 0 & \text{if } y = 0 \end{cases}$$

to the values of g. Then $\Delta_p^*(\operatorname{clip}_b^p \circ g) \leq 2b$ by the triangle inequality, as $|| \operatorname{clip}_b^p(g(x))||_p \leq b$ for all x. As this thesis only uses the Gaussian mechanism and thus only needs to clip the l_2-norm, the superscript p is always 2 and the notation $\operatorname{clip}_b = \operatorname{clip}_b^2$ is used.

2.2 Bayesian Inference and the Metropolis-Hastings Algorithm

In Bayesian inference, the parameters θ of a statistical model are inferred from observed data using Bayes' theorem [GCS+14]. The result is not just a point estimate of θ, but a probability distribution describing the likelihood of different values of θ.

Bayes' theorem relates the *posterior* belief of θ, $p(\theta \mid X)$ to the *prior* belief $p(\theta)$ through the observed data X, and the likelihood of the data $p(X \mid \theta)$ as follows:

$$p(\theta \mid X) = \frac{p(X \mid \theta)p(\theta)}{\int p(X \mid \theta)p(\theta)\mathrm{d}\theta}.$$

It is theoretically possible to compute $p(\theta \mid X)$ given any likelihood, prior and data, but the integral in the denominator is in many cases difficult to compute [GCS+14]. In such cases the posterior cannot be feasibly computed. However, many of the commonly used summary statistics of the posterior, such as the mean, variance and credible intervals, can be approximated from a sample of the posterior. *Markov chain Monte Carlo* (MCMC) is a widely used method to obtain such samples.

MCMC algorithms sequentially sample values of θ with the goal of eventually having the chain of sampled values converge to a given distribution [RC04]. While this can be done in many ways, this thesis focuses on a particular MCMC algorithm: *Metropolis-Hastings* (MH) [MRR+53, Has70].

At each iteration i, the Metropolis-Hastings algorithm samples θ_i from a distribution π of θ by first picking a proposal θ' from a proposal distribution $q(\cdot \mid \theta_{i-1})$ [MRR+53], where θ_{i-1} is the previously sampled value[*]. We shorten θ_{i-1} to θ in the following. The ratio of posterior and proposal densities is calculated

$$r(\theta, \theta') = \frac{\pi(\theta')}{\pi(\theta)} \frac{q(\theta \mid \theta')}{q(\theta' \mid \theta)},$$

[*] The value of θ_0 for the first iteration is given as input to the algorithm.

and the proposal is accepted with probability $\min\{1, r\}$. If the proposal is accepted, $\theta_i = \theta'$, otherwise $\theta_i = \theta$.

It can be shown that, with a suitable proposal distribution, the chain of θ_i values is *ergodic* and has π as its *invariant distribution*, which implies that the chain converges to π [Has70]. The invariance of π follows from the *detailed balance condition* [RC04]:

$$\pi(\theta)q(\theta' \mid \theta)\min\{1, r(\theta, \theta')\} = \pi(\theta')q(\theta \mid \theta')\min\{1, r(\theta', \theta)\}.$$

The ergodicity of the chain depends on the proposal distribution. Proving ergodicity in general can be difficult, but there is an easy to check sufficient condition: if the proposal allows moving to any state from any state with positive probability, the chain is called *strongly irreducible*. Together with the detailed balance condition, strong irreducibility implies ergodicity [RC04]. The Gaussian distribution centered at the current value is a commonly used proposal, which is clearly strongly irreducible.

When MCMC is used in Bayesian inference, the distribution to approximate is

$$\pi(\theta) = p(\theta \mid X) = \frac{p(X \mid \theta)p(\theta)}{\int p(X \mid \theta)p(\theta)\mathrm{d}\theta}.$$

The difficult integral $\int p(X \mid \theta)p(\theta)\mathrm{d}\theta$ in the denominator cancels out when computing r, so only the likelihood and the prior are needed. For numerical stability, r is usually computed in log-space, which makes the acceptance probability $\min\{1, e^{\lambda(\theta,\theta')}\}$ where

$$\lambda(\theta, \theta') = \ln\frac{p(X \mid \theta')}{p(X \mid \theta)} + \ln\frac{p(\theta')}{p(\theta)} + \ln\frac{q(\theta \mid \theta')}{q(\theta' \mid \theta)}. \tag{2.1}$$

The dataset X is typically a table with n independent rows. The likelihood is given as $p(x_j \mid \theta)$ for row x_j. Independence of the rows means that

$$p(X \mid \theta) = \prod_{j=1}^{n} p(x_j \mid \theta),$$

which means that the log-likelihood ratio term of λ is

$$\ln\frac{p(X \mid \theta')}{p(X \mid \theta)} = \sum_{j=1}^{n} \ln\frac{p(x_j \mid \theta')}{p(x_j \mid \theta)}.$$

Algorithm 1 puts all of this together to summarise the MH algorithm used for Bayesian inference.

Algorithm 1: Metropolis-Hastings: number of iterations k, proposal distribution q, initial value θ_0 and dataset X as input.

```
for $1 \leq i \leq k$ do
    denote $\theta = \theta_{i-1}$
    sample $\theta' \sim q(\cdot \mid \theta)$
    $\ln \frac{p(X|\theta')}{p(X|\theta)} = \sum_{j=1}^{n} (\ln p(x_j \mid \theta') - \ln p(x_j \mid \theta))$
    $\lambda = \ln \frac{p(X|\theta')}{p(X|\theta)} + \ln p(\theta') - \ln p(\theta) + \ln q(\theta \mid \theta') - \ln q(\theta' \mid \theta)$
    $\theta_i = \begin{cases} \theta' & \text{with probability } \min\{1, e^{\lambda}\} \\ \theta & \text{otherwise} \end{cases}$
end
return $(\theta_1, \ldots, \theta_k)$
```

2.3 Measure Theory

The study of Hamiltonian Monte Carlo (HMC) in Chapter 4, requires a framework for studying Metropolis-Hastings algorithms that have non-continuous proposal distributions, which is not provided by the theory presented in Section 2.2 that only handles continuous proposals. This section introduces tools from measure theory, in particular the Markov kernels, that do provide a sufficiently general framework for the study of HMC. As with the other sections in this chapter, the introduction is very limited, only covering concepts used in this thesis.

Measure theory is the study of *measures*, which unify the concepts of length, area and volume of subsets of $\mathbb{R}$, $\mathbb{R}^2$ and $\mathbb{R}^3$, the number of elements of a finite set, probabilities of events, and other similar concepts. All of the aforementioned concepts assign a positive real number, or infinity, to a set, that measures the size of the set in some sense. In all cases, the measure of the empty set is zero, and measure is additive: the measure of a union of disjoint sets is the sum of the measures of each set in the union. Ideally, every subset would have a measure, like every finite set has a number of elements, but assigning a length to all subsets of $\mathbb{R}$ in a way that has all the properties expected of lengths is not possible [Çın11]. Similar difficulties occur in higher dimensions and with continuous probability distributions, so the range of a measure is restricted to a collection of where a satisfactory assignment of measure is possible.

Before defining measures, some notation is needed. Denote the set of natural numbers excluding zero by $\mathbb{N}_1$, the set of non-negative real numbers with $\mathbb{R}_+$, and denote $\bar{\mathbb{R}}_+ = \mathbb{R}_+ \cup \{\infty\}$. In sums, the ∞-symbol behaves as expected: $x + \infty = \infty$ for any $x \in \mathbb{R}$ and $\infty + \infty = \infty$. Subtracting infinity is not defined. The set of all subsets of a set E, called the *powerset* of E, is denoted by $\mathcal{P}(E)$. Subsets of $\mathcal{P}(E)$ are also called collections to differentiate them from subsets of E. The indicator function of a set A is denoted by 1_A. The Cartesian product $E_1 \times E_2 \cdots \times E_d$ is denoted by $\times_{i=1}^d E_i$ and when $E_i = E$ for all $i \in \mathbb{N}_1$, $\times_{i=1}^d E = E^d$.

Definition 2.12. *Let E be a set and let $\mathcal{E} \subset \mathcal{P}(E)$. $\mathcal{E}$ is called a σ-algebra on E if*

1. $\emptyset \in \mathcal{E}$
2. $A \in \mathcal{E} \Rightarrow A^C \in \mathcal{E}$
3. $A_1, A_2, \ldots \in \mathcal{E} \Rightarrow \bigcup_{i=1}^{\infty} A_i \in \mathcal{E}$

The pair $(E, \mathcal{E})$ is called a measurable space.

The sets of a σ-algebra $\mathcal{E}$ are called $\mathcal{E}$-measurable, or simply measurable if the corresponding σ-algebra is clear from the context. The smallest σ-algebra on E that contains all the sets of a collection $\mathcal{C} \subset \mathcal{P}(E)$ is called the σ-algebra *generated* by $\mathcal{C}$.

Definition 2.13. *Let $(E, \mathcal{E})$ be a measurable space. A measure on $(E, \mathcal{E})$ is a function $\mu \colon \mathcal{E} \to \bar{\mathbb{R}}_+$ such that*

1. $\mu(\emptyset) = 0$
2. *For any sequence of disjoint sets $A_1, A_2, \ldots \in \mathcal{E}$*

$$\mu\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mu(A_i)$$

The triple $(E, \mathcal{E}, \mu)$ is called a measure space.

The reason for restricting the domain of a measure to a σ-algebra is the impossibility of defining a length for all subsets of $\mathbb{R}$: it is possible to assign a length to all *Lebesgue-measurable* sets, which form a σ-algebra [Çın11]. The measure assigning length in $\mathbb{R}$, area in $\mathbb{R}^2$ and their counterparts in higher dimensions is called the *Lebesgue measure*. Lebesgue measure on $\mathbb{R}^d$ is denoted by m_d, but the subscript d is dropped if the dimension is clear from context.

For this thesis, it is sufficient to work with the *Borel σ-algebra* of $\mathbb{R}^d$, denoted by $\mathcal{B}(\mathbb{R}^d)$, which is the σ-algebra of $\mathbb{R}^d$ generated by the collection of all the open sets of $\mathbb{R}^d$. Members of $\mathcal{B}(\mathbb{R}^d)$ are called the Borel sets, and they include most sets one encounters in mathematics: all open, all closed sets and all sets obtained from them by countable numbers of set-theoretic operations like unions and intersections [Çın11]. As $\mathcal{B}(\mathbb{R}^d)$

is the only σ-algebra used with $\mathbb{R}^d$ in this thesis, the measurable space $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ is shortened to $\mathbb{R}^d$ when this does not cause ambiguity.

A measure μ on a measurable space $(E, \mathcal{E})$ is called a *probability measure* if $\mu(E) = 1$. μ is called *finite* if $\mu(E) < \infty$ and *σ-finite* if there is a countable partition of E into measurable sets $E_1, E_2, \dots$ where $\mu(E_i) < \infty$ for all $i \in \mathbb{N}_1$. Many theorems in measure theory require σ-finiteness from the involved measures to ensure that they are not infinite for too many sets. All probability measures are clearly finite and σ-finite, and the Lebesgue measure is σ-finite but not finite [Çın11].

For measurable spaces $(E, \mathcal{E})$ and $(F, \mathcal{F})$, there is a *product σ-algebra* $\mathcal{E} \otimes \mathcal{F}$ on $E \times F$ that is the σ-algebra generated by the collection of sets of the form $A \times B$ for $A \in \mathcal{E}$ and $B \in \mathcal{F}$. The product extends to products of more than one σ-algebra, denoted by $\bigotimes_i^d \mathcal{E}_i$ for σ-algebrals $\mathcal{E}_i$ for $1 \leq i \leq d$. The measurable space $(E \times F, \mathcal{E} \otimes \mathcal{F})$ is called the *product measurable space* of $(E, \mathcal{E})$ and $(F, \mathcal{F})$. This extends to products of multiple measurable spaces: $(\times_{i=1}^d E_i, \bigotimes_{i=1}^d \mathcal{E}_i)$ is the product of the measurable spaces $(E_i, \mathcal{E}_i)$ for $1 \leq i \leq d$. The measurable space $(\times_{i=1}^d E, \bigotimes_{i=1}^d \mathcal{E})$ is denoted by $(E, \mathcal{E})^d$.

If μ is a measure on $(E, \mathcal{E})$ and ν is a measure on $(F, \mathcal{F})$ and both are σ-finite, the *product measure* of μ and ν is the unique measure $\mu \times \nu$ on $(E \times F, \mathcal{E} \otimes \mathcal{F})$ that has $(\mu \times \nu)(A \times B) = \mu(A)\nu(B)$ for $A \in \mathcal{E}$ and $B \in \mathcal{F}$ [Çın11]. The product $\mu \times \mu \times \cdots \times \mu$ with $d-1$ products is denoted by μ^d. Both product measurable spaces and product measures act on $\mathbb{R}^d$ and the Lebesgue measure as expected: $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d)) = (\mathbb{R}, \mathcal{B}(\mathbb{R}))^d$ and $m_d = m_1^d$ [Çın11].

The equality of two σ-finite measures on a product measurable space comes down to their equality on sets that are cartesian products of sets from the measurable spaces forming the product measurable space, if the measures are σ-finite in a specific way. This is stated precisely in Lemma 2.16, but the main machinery used in its proof is presented first:

Definition 2.14. *Let E be a set. A set $\mathcal{C} \subset \mathcal{P}(E)$ is called a p-system if $A \cap B \in \mathcal{C}$ for all $A, B \in \mathcal{C}$.*

Lemma 2.15. *Let $(E, \mathcal{E})$ be a measurable space and let μ and ν be finite measures on $(E, \mathcal{E})$. Let $\mathcal{C}$ be a p-system generating $\mathcal{E}$. If $\mu(A) = \nu(A)$ for all $A \in \mathcal{C}$, $\mu = \nu$.*

Proof. See [Çın11, Proposition 3.7]. □

Lemma 2.16. *Let μ and ν be measures on $(E, \mathcal{E})^d$ with a countable partition of E^d into sets B_i for $i \in N_1$ of the form $B_i = \times_{j=1}^d C_j^i$ with $C_i^j \in \mathcal{E}$ for all $i, j \in N_1$ such that $\mu(B_i) < \infty$ and $\nu(B_i) < \infty$. Then $\mu = \nu$ if and only if*

$$\mu\left(\bigtimes_{j=1}^{d} A_j\right) = \nu\left(\bigtimes_{j=1}^{d} A_j\right)$$

for all $A_1, \dots, A_d \in \mathcal{E}$.

Proof. Clearly, if $\mu = \nu$,

$$\mu\left(\bigtimes_{j=1}^{d} A_j\right) = \nu\left(\bigtimes_{j=1}^{d} A_j\right).$$

for all $A_1, \ldots, A_d \in \mathcal{E}$.

To prove the other direction, note that sets of the form $\times_{j=1}^{d} A_j$ form a p-system that generates $\bigotimes_{j=1}^{d} \mathcal{E}$. Denote $(\mu|B)(A) = \mu(A \cap B)$ for $A, B \in \mathcal{E}$. Then $\mu|B$ is a measure on $(E, \mathcal{E})^d$ [Çın11]. Now the measures $\mu|B_i$ and $\nu|B_i$ are finite because $(\mu|B_i)(A) = \mu(A \cap B_i) \leq \mu(B_i) < \infty$ for any $A \in \bigotimes_{j=1}^{d} \mathcal{E}^d$, the same holds for $\nu|B_i$ and

$$\begin{aligned}(\mu|B_i)\left(\bigtimes_{j=1}^{d} A_j\right) &= \mu\left(\left(\bigtimes_{j=1}^{d} A_j\right) \cap \left(\bigtimes_{j=1}^{d} C_j^i\right)\right) \\ &= \mu\left(\bigtimes_{j=1}^{d} (A_j \cap C_j^i)\right) \\ &= \nu\left(\bigtimes_{j=1}^{d} (A_j \cap C_j^i)\right) \\ &= (\nu|B_i)\left(\bigtimes_{j=1}^{d} A_j\right)\end{aligned}$$

for all $A_1, \ldots, A_d \in \mathcal{E}$, where the second equality uses the identity $\left(\times_{j=1}^{d} S_j\right) \cap \left(\times_{j=1}^{d} T_j\right) = \times_{j=1}^{d} (S_j \cap T_j)$ for all sets S_j and T_j and the third equality uses the assumption that $\mu\left(\times_{j=1}^{d} D_j\right) = \nu\left(\times_{j=1}^{d} D_j\right)$ for all $D_1, \ldots, D_d \in \mathcal{E}$ with $D_j = A_j \cap C_j^i$. By Lemma 2.15, $\mu|B_i = \nu|B_i$ for all $i \in N_1$. Then

$$\mu(A) = \mu(E^d \cap A) = \mu\left(\bigcup_{i=1}^{\infty} (B_i \cap A)\right) = \sum_{i=1}^{\infty} (\mu|B_i)(A) = \sum_{i=1}^{\infty} (\nu|B_i)(A) = \nu(A)$$

for any $A \in \bigotimes_{i=1}^{d} \mathcal{E}$, so $\mu = \nu$. □

Measure-theoretic concepts that involve functions, particularly the Lebesgue integral discussed later, typically require that the functions preserve the measurability of sets.

Definition 2.17. *Let $(E, \mathcal{E})$ and $(F, \mathcal{F})$ be measurable spaces and let $f\colon E \to F$. f is called $\mathcal{E}$-$\mathcal{F}$-measurable if $B \in \mathcal{F} \Rightarrow f^{-1}(B) \in \mathcal{E}$.*

As with measurable sets, $\mathcal{E}$-$\mathcal{F}$-measurable functions are simply called measurable if $\mathcal{E}$ and $\mathcal{F}$ are clear from the context. Most functions that one comes across are measurable. In particular, the indicators of measurable sets are measurable, continuous functions are Borel-Borel-measurable, and compositions of measurable functions are measurable [Çın11].

One of the hallmarks of measure theory is the Lebesgue integral, which extends the Riemann integral to allow integration over all measurable functions, and allows integrating with respect to an arbitrary measure. The definition [Çın11, Definition 4.3] of the Lebesgue integral is fairly technical and is not given here. Instead, the notation used with the integral is defined, and some properties of the integral are give in Theorem 2.19.

Definition 2.18. *Let $(E, \mathcal{E}, \mu)$ be a measure space and let $f \colon E \to \mathbb{R}_+$ be a measurable function. The Lebesgue integral of f over a set $A \in \mathcal{E}$ with respect to μ is denoted by*

$$\int_A \mu(\mathrm{d}x) f(x).$$

The measure μ to integrate over is placed right next to the integral sign, which makes the nested nested integrals encountered later much more readable than the common approach of placing the function f next to the integral sign. The variable that is integrated over is indicated by the differential symbol $\mathrm{d}x$ inside the parenthesis after the measure μ. Note that the integral was only defined for non-negative functions. It is possible to extend the definition to certain measurable real-valued functions [Çın11], but the integral for non-negative functions is sufficient for this thesis.

Theorem 2.19. *Useful properties of the Lebesgue integral [Çın11]:*

1. *For a measure space $(E, \mathcal{E}, \mu)$ and $A \in \mathcal{E}$, $\int_E \mu(\mathrm{d}x) 1_A(x) = \mu(A)$.*

2. *For a measure space $(E, \mathcal{E}, \mu)$, a measurable function $f \colon E \to \mathbb{R}_+$ and $A \in \mathcal{E}$, $\int_A \mu(\mathrm{d}x) f(x) = \int_E \mu(\mathrm{d}x) 1_A(x) f(x)$.*

3. *For a measure space $(E, \mathcal{E}, \mu)$, measurable $f, g \colon E \to \mathbb{R}_+$, $A \in \mathcal{E}$ and $a, b \in \mathbb{R}_+$, $\int_A \mu(\mathrm{d}x)(af(x) + bg(x)) = a \int_A \mu(\mathrm{d}x) f(x) + b \int_A \mu(\mathrm{d}x) g(x)$.*

4. *For a Riemann-integrable function $f \colon \mathbb{R} \to \mathbb{R}_+$ and $a, b \in \mathbb{R}$ with $a \leq b$, $\int_{[a,b]} m(\mathrm{d}x) f(x) = \int_a^b f(x)\mathrm{d}x$, where the integral on the right is a Riemann integral.*

5. *For a continuous random variable X on $\mathbb{R}^d$ with density π and $A \in \mathcal{B}(\mathbb{R}^d)$, $P(X \in A) = \int_A m(\mathrm{d}x)\pi(x)$. For a measurable function $f \colon \mathbb{R}^d \to \mathbb{R}_+$, $\int_A \pi(\mathrm{d}x) f(x) = \int_A m(\mathrm{d}x)\pi(x) f(x)$.*

6. *Let μ and ν be σ-finite measures on measure spaces $(E, \mathcal{E})$ and $(F, \mathcal{F})$, respectively. Let $f \colon E \times F \to \mathbb{R}_+$ be measurable. Then*

$$\int_A \mu(\mathrm{d}a) \int_B \nu(\mathrm{d}b) f(a, b) = \int_B \nu(\mathrm{d}b) \int_A \mu(\mathrm{d}a) f(a, b).$$

for all $A \in \mathcal{E}$ and $B \in \mathcal{F}$.

Randomised functions come up in several places in the study of MH algorithms. Each sample from an MH algorithm is obtained by evaluating a random function on the previous value, so the entire algorithm can be thought of as repeatedly composing that function, called the *transition kernel*, with itself. Additionally, the proposal is also obtained from a random function on the previous value. The convergence of an MH algorithm can be studied by studying the properties of these functions. Formally, these randomised functions are *Markov kernels*, functions from values to probability measures on values.

Definition 2.20. *Let $(E, \mathcal{E})$ be a measurable space. A Markov kernel on $(E, \mathcal{E})$ is a function $q\colon E \times \mathcal{E} \to [0, 1]$ where*

1. *For all $A \in \mathcal{E}$, the function $q(\cdot, A)$ is measurable.*
2. *For all $a \in E$, the function $q(a, \cdot)$ is a probability measure.*

Lemma 2.21. *The composition of Markov kernels q_1 and q_2 on a measurable space $(E, \mathcal{E})$ is given by*

$$(q_2 \circ q_1)(a, C) = \int_E q_1(a, \mathrm{d}b) q_2(b, C)$$

Proof. See [Çın11, Equation 6.5]. □

A measure μ and a Markov kernel q, both on a measurable space $(E, \mathcal{E})$ can be used to define a measure on $(E, \mathcal{E})^2$. If μ is σ-finite, it suffices to define the measure on sets of the form $A \times B$ for $A, B \in \mathcal{E}$ [Çın11]. This property is central to the following discussion, and is formalised below.

Theorem 2.22. *Let μ be a σ-finite measure and q be a Markov kernel, both on a measurable space $(E, \mathcal{E})$. Then there exists a unique measure ν on $(E, \mathcal{E})^2$ such that*

$$\nu(A \times B) = \int_A \mu(\mathrm{d}a) \int_B q(a, \mathrm{d}b).$$

for all $A, B \in \mathcal{E}$.

Proof. See [Çın11, Theorem 6.11]. □

A central concept for the theory of MH algorithms is the reversibility of Markov kernels with respect to a measure μ, which can be used to express the detailed balance condition for general proposal and target distribution, and is used to simplify the acceptance probability in Lemma 2.29.

Definition 2.23. *Let $(E, \mathcal{E})$ be a measurable space. Let q be a Markov kernel, let μ be a σ-finite measure, both on $(E, \mathcal{E})$ and let*

$$\nu_1(A \times B) = \int_A \mu(\mathrm{d}a) \int_B q(a, \mathrm{d}b),$$

$$\nu_2(A \times B) = \nu_1(B \times A)$$

for $A, B \in \mathcal{E}$ be measures on $(E, \mathcal{E})^2$. q is said to be reversible with respect to μ if $\nu_1 = \nu_2$.

Because of Theorem 2.22, the equations for ν_1 and ν_2 in definition 2.23 define unique measures on $(E, \mathcal{E})^2$. Theorem 2.22 also allows checking that $\nu_1 = \nu_2$ through an equality of integrals:

Lemma 2.24. *Let $(E, \mathcal{E})$ be a measurable space, and let q be a Markov kernel and μ be a σ-finite measure, both on $(E, \mathcal{E})$. Then q is reversible with respect to μ if and only if*

$$\int_A \mu(\mathrm{d}a) \int_B q(a, \mathrm{d}b) = \int_B \mu(\mathrm{d}b) \int_A q(b, \mathrm{d}a)$$

for all $A, B \in \mathcal{E}$.

Proof. Let

$$\nu_1(A \times B) = \int_A \mu(\mathrm{d}a) \int_B q(a, \mathrm{d}b),$$

$$\nu_2(A \times B) = \nu_1(B \times A) = \int_B \mu(\mathrm{d}b) \int_A q(b, \mathrm{d}a).$$

be measures on $(E, \mathcal{E})^2$. If q is reversible with respect to μ, $\nu_1 = \nu_2$, so

$$\int_A \mu(\mathrm{d}a) \int_B q(a, \mathrm{d}b) = \nu_1(A \times B) = \nu_2(A \times B) = \int_B \mu(\mathrm{d}b) \int_A q(b, \mathrm{d}a).$$

If

$$\int_A \mu(\mathrm{d}a) \int_B q(a, \mathrm{d}b) = \int_B \mu(\mathrm{d}b) \int_A q(b, \mathrm{d}a)$$

then

$$\nu_1(A \times B) = \int_A \mu(\mathrm{d}a) \int_B q(a, \mathrm{d}b) = \int_B \mu(\mathrm{d}b) \int_A q(b, \mathrm{d}a) = \nu_2(A \times B).$$

Now the uniqueness in Theorem 2.22 shows that $\nu_1 = \nu_2$. □

If the Markov kernel is defined on a product measurable space, Lemma 2.16 allows factoring the integrals in Lemma 2.24 further:

Lemma 2.25. *Let $(E, \mathcal{E})$ be a measurable space, and let q be a Markov kernel on $(E, \mathcal{E})^2$ and μ be a σ-finite measure on $(E, \mathcal{E})$. Then q reversible with respect to μ^2 if and only if*

$$\int_A \mu(\mathrm{d}a) \int_B \mu(\mathrm{d}b) \int_{C \times D} q((a, b), \mathrm{d}(c, d)) = \int_C \mu(\mathrm{d}c) \int_D \mu(\mathrm{d}d) \int_{A \times B} q((c, d), \mathrm{d}(a, b))$$

Proof. Let

$$\begin{aligned}\nu_1(S \times T) &= \int_S \mu^2(\mathrm{d}s) \int_T q(s, \mathrm{d}t),\\ \nu_2(S \times T) &= \int_T \mu^2(\mathrm{d}t) \int_S q(t, \mathrm{d}s),\end{aligned}$$

where μ^2 is the unique measure defined by

$$\mu^2(A \times B) = \mu(A)\mu(B) = \int_A \mu(\mathrm{d}a) \int_B \mu(\mathrm{d}b).$$

for $A, B \in \mathcal{E}$. Reversibility of q with respect to μ^2 is now equivalent to $\nu_1 = \nu_2$. When $S = A \times B$ and $T = C \times D$,

$$\nu_1(A \times B \times C \times D) = \int_A \mu(\mathrm{d}a) \int_B \mu(\mathrm{d}b) \int_{C \times D} q((a, b), \mathrm{d}(c, d))$$

$$\nu_2(A \times B \times C \times D) = \int_C \mu(\mathrm{d}c) \int_D \mu(\mathrm{d}d) \int_{A \times B} q((c, d), \mathrm{d}(a, b))$$

so $\nu_2(A \times B \times C \times D) = \nu_1(C \times D \times A \times B)$.

If q is reversible with respect to μ, $\nu_1 = \nu_2$, so

$$\begin{aligned}\int_A \mu(\mathrm{d}a) \int_B \mu(\mathrm{d}b) \int_{C \times D} q((a, b), \mathrm{d}(c, d)) &= \nu_1(A \times B \times C \times D)\\ &= \nu_2(A \times B \times C \times D)\\ &= \int_C \mu(\mathrm{d}c) \int_D \mu(\mathrm{d}d) \int_{A \times B} q((c, d), \mathrm{d}(a, b)).\end{aligned}$$

If

$$\int_A \mu(\mathrm{d}a) \int_B \mu(\mathrm{d}b) \int_{C \times D} q((a, b), \mathrm{d}(c, d)) = \int_C \mu(\mathrm{d}c) \int_D \mu(\mathrm{d}d) \int_{A \times B} q((c, d), \mathrm{d}(a, b))$$

then

$$\begin{aligned}\nu_1(A \times B \times C \times D) &= \int_A \mu(\mathrm{d}a) \int_B \mu(\mathrm{d}b) \int_{C \times D} q((a, b), \mathrm{d}(c, d))\\ &= \int_C \mu(\mathrm{d}c) \int_D \mu(\mathrm{d}d) \int_{A \times B} q((c, d), \mathrm{d}(a, b)).\\ &= \nu_2(A \times B \times C \times D).\end{aligned}$$

As μ is σ-finite, there is a countable partition E_i of E such that $\mu(E_i) < \infty$ for all $i \in N_1$. Then $E_i \times E_j \times E_k \times E_l$ for $i, j, k, l \in N_1$ is a countable partition of E^4, and

$$\begin{aligned}\nu_1(E_i \times E_j \times E_k \times E_l) &= \int_{E_i} \mu(\mathrm{d}a) \int_{E_j} \mu(\mathrm{d}b) \int_{E_k \times E_l} q((a, b), \mathrm{d}(c, d))\\ &\leq \int_{E_i} \mu(\mathrm{d}a) \int_{E_j} \mu(\mathrm{d}b)\\ &< \infty,\end{aligned}$$

$$\begin{aligned}\nu_2(E_i \times E_j \times E_k \times E_l) &= \int_{E_k} \mu(\mathrm{d}c) \int_{E_l} \mu(\mathrm{d}d) \int_{E_i \times E_j} q((c,d), \mathrm{d}(a,b)) \\ &\leq \int_{E_k} \mu(\mathrm{d}c) \int_{E_l} \mu(\mathrm{d}d) \\ &< \infty.\end{aligned}$$

Then Lemma 2.16 shows that $\nu_1 = \nu_2$. □

The equation used to check reversibility in Lemma 2.24 also applies if an arbitrary measurable function is included on both sides:

Lemma 2.26. *Let q be a Markov kernel on a measurable space $(E, \mathcal{E})$ reversible with respect to a σ-finite measure μ on $(E, \mathcal{E})$ and let $f\colon E \times E \to \mathbb{R}_+$ be a measurable function. Then*

$$\int_A \mu(\mathrm{d}a) \int_B q(a, \mathrm{d}b) f(a,b) = \int_B \mu(\mathrm{d}b) \int_A q(b, \mathrm{d}a) f(a,b)$$

Proof. Setting

$$\nu_1(A \times B) = \int_A \mu(\mathrm{d}a) \int_B q(a, \mathrm{d}b)$$

$$\nu_2(A \times B) = \int_B \mu(\mathrm{d}b) \int_A q(b, \mathrm{d}a)$$

for $A, B \in \mathcal{E}$ defines unique measures ν_1 and ν_2 on $(E, \mathcal{E})^2$ by Theorem 2.22, which, by Definition 2.23, are equal. Then

$$\begin{aligned}\int_A \mu(\mathrm{d}a) \int_B q(a, \mathrm{d}b) f(a,b) &= \int_A \int_B \nu_1(\mathrm{d}a, \mathrm{d}b) f(a,b) \\ &= \int_A \int_B \nu_2(\mathrm{d}a, \mathrm{d}b) f(a,b) \\ &= \int_B \mu(\mathrm{d}b) \int_A q(b, \mathrm{d}a) f(a,b)\end{aligned}$$ □

For a composition of reversible Markov kernels is not necessarily reversible, but an equality similar to Lemma 2.24 still holds:

Lemma 2.27. *For Markov kernels $q_1 \dots q_k$ on $(E, \mathcal{E})$ reversible with respect to a σ-finite measure μ on $(E, \mathcal{E})$,*

$$\int_A \mu(\mathrm{d}a) \int_C (q_k \circ \dots \circ q_1)(a, \mathrm{d}c) = \int_C \mu(\mathrm{d}c) \int_A (q_1 \circ \dots \circ q_k)(c, \mathrm{d}a).$$

Proof. Lemma 2.21 can be restated as

$$\int_C (q_2 \circ q_1)(a, \mathrm{d}c) = \int_E q_1(a, \mathrm{d}b) \int_C q_2(b, \mathrm{d}c). \tag{2.2}$$

The proof is by induction on k. For $k = 1$, the claim is the definition of reversibility with respect to μ. If the claim holds for $k-1$,

$$\begin{aligned}
&\int_A \mu(\mathrm{d}a) \int_C (q_k \circ \cdots \circ q_1)(a, \mathrm{d}c) \\
&= \int_A \mu(\mathrm{d}a) \int_E (q_{k-1} \circ \cdots \circ q_1)(a, \mathrm{d}b) \int_C q_k(b, \mathrm{d}c) \\
&= \int_E \mu(\mathrm{d}b) \int_A (q_1 \circ \cdots \circ q_{k-1})(b, \mathrm{d}a) \int_C q_k(b, \mathrm{d}c) \\
&= \int_E \mu(\mathrm{d}b) \int_C q_k(b, \mathrm{d}c) \int_A (q_1 \circ \cdots \circ q_{k-1})(b, \mathrm{d}a) \\
&= \int_C \mu(\mathrm{d}c) \int_E q_k(c, \mathrm{d}b) \int_A (q_1 \circ \cdots \circ q_{k-1})(b, \mathrm{d}a) \\
&= \int_C \mu(\mathrm{d}c) \int_A (q_1 \circ \cdots \circ q_k)(c, \mathrm{d}a),
\end{aligned}$$

where the first equality uses Equation 2.2, the second equality uses the induction hypothesis for $k-1$, the third equality uses uses Theorem 2.19 (6) to change the order of integration, the fourth uses Lemma 2.26 to swap the integration variables for two leftmost integrals and the last uses Equation 2.2 again. Thus the claim holds for all $k \in N_1$ by induction. □

If the composition $q_k \circ \cdots \circ q_1$ is symmetric, Lemma 2.27 becomes a statement on the reversibility of the composition. More formally:

Lemma 2.28. *Let $q_1, \ldots, q_k$ be Markov kernels on $(E, \mathcal{E})$ reversible with respect to a σ-finite measure μ on $(E, \mathcal{E})$. If $q_1 \circ \cdots \circ q_k = q_k \circ \cdots \circ q_1$, the composition $q_1 \circ \cdots \circ q_k$ is reversible with respect to μ.*

Proof. Because $q_1 \circ \cdots \circ q_k = q_k \circ \cdots \circ q_1$, by Lemma 2.27

$$\begin{aligned}
\int_A \mu(\mathrm{d}a) \int_C (q_k \circ \cdots \circ q_1)(a, \mathrm{d}c) &= \int_C \mu(\mathrm{d}c) \int_A (q_1 \circ \cdots \circ q_k)(c, \mathrm{d}a) \\
&= \int_C \mu(\mathrm{d}c) \int_A (q_k \circ \cdots \circ q_1)(c, \mathrm{d}a).
\end{aligned}$$

and the claim follows from Lemma 2.24. □

Reversibility is linked to the theory of MH algorithms: the detailed balance condition is equivalent to the reversibility of the transition kernel of the MH algorithm with respect to the target distribution π [Tie98]. Additionally, if target is continuous, and the proposal is reversible with respect to the Lebesgue measure, the acceptance probability is simplified, as shown below.

Lemma 2.29. *For an MH-algorithm, if the proposal Markov kernel q on $\mathbb{R}^n$ is reversible with respect to the Lebesgue measure, the target distribution π is continuous and*

$$\alpha(\theta, \theta') = \min\left\{1, \frac{\pi(\theta')}{\pi(\theta)}\right\}$$

is used as the acceptance probability when moving from θ to θ', the algorithm has π as the invariant distribution.

Proof. With acceptance probability α, the algorithm has π as its invariant distribution if the detailed balance condition,

$$\int_A \pi(\mathrm{d}\theta) \int_B q(\theta, \mathrm{d}\theta')\alpha(\theta, \theta') = \int_B \pi(\mathrm{d}\theta') \int_A q(\theta', \mathrm{d}\theta)\alpha(\theta', \theta)$$

for all $A, B \in \mathcal{B}(\mathbb{R}^d)$, holds [Tie98][*]. For continuous π and q reversible with respect to the Lebesgue measure, for all $A, B \in \mathcal{B}(\mathbb{R}^d)$,

$$\begin{aligned}
\int_A \pi(\mathrm{d}\theta) \int_B q(\theta, \mathrm{d}\theta')\alpha(\theta, \theta') &= \int_A m(\mathrm{d}\theta) \int_B q(\theta, \mathrm{d}\theta')\pi(\theta)\alpha(\theta, \theta') \\
&= \int_A m(\mathrm{d}\theta) \int_B q(\theta, \mathrm{d}\theta') \min\{\pi(\theta), \pi(\theta')\} \\
&= \int_B m(\mathrm{d}\theta') \int_A q(\theta', \mathrm{d}\theta) \min\{\pi(\theta'), \pi(\theta)\} \\
&= \int_A \pi(\mathrm{d}\theta') \int_B q(\theta', \mathrm{d}\theta)\alpha(\theta', \theta).
\end{aligned}$$

□

Deterministic and partially deterministic proposals studied in Chapter 4 require handling Markov kernels that are (partially) deterministic. This is made possible by the *Dirac measure*, which is the probability measure for a degenerate random variable that always takes the same value.

Definition 2.30. *The Dirac measure δ_x for $x \in \mathbb{R}^d$ is a measure on $\mathbb{R}^d$ where*

$$\delta_x(A) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

Based on the definition, δ_x is the probability measure of a random variable that always has the value x. As a Markov kernel, it represents the identity function. The Dirac measure can also represent the Markov kernel of any deterministic function f by $\delta_{f(x)}$. It turns out that with a suitable f, $\delta_{f(x)}$ is reversible with respect to the Lebesgue measure, which is applied together with Lemma 2.29 in Chapter 4.

Lemma 2.31. *For a function $f\colon \mathbb{R}^d \to \mathbb{R}^d$ that preserves Lebesgue measure and has $f^{-1} = f$, the Dirac measure perturbed by f, $\delta_{f(x)}$, is reversible with respect to the Lebesgue measure. As a special case, the Dirac measure δ_x is reversible with respect to the Lebesgue measure.*

[*] Tierney [Tie98] states the detailed balance condition as an equality of measures, which, by Lemma 2.16, can be expressed as an equality of integrals.

Proof. Denote the Lebesgue measure by m. Because f and f^{-1} preserve Lebesgue measure

$$\begin{aligned}
\int_A m(\mathrm{d}x)\delta_{f(x)}(B) &= \int_A m(\mathrm{d}x)1_B(f(x)) \\
&= m(A \cap f^{-1}(B)) \\
&= m(f^{-1}(A \cap f^{-1}(B))) \\
&= m(f^{-1}(A) \cap f^{-1}(f^{-1}(B))) \\
&= m(f^{-1}(A) \cap B) \\
&= \int_B m(\mathrm{d}x)\delta_{f(x)}(A)
\end{aligned}$$

where the first equality uses the identity $\delta_t(B) = 1_B(t)$ for any $t \in \mathbb{R}^d$ the second uses Theorem 2.19 (1, 2) and the identity $1_S(t)1_T(t) = 1_{S\cap T}(t)$ for any $t \in \mathbb{R}^d$ and $S, T \subset \mathbb{R}^d$, the third uses the fact that f^{-1} preserves Lebesgue measure, the fourth uses the identity $f^{-1}(S \cap T) = f^{-1}(S) \cap f^{-1}(T)$, the fifth uses the fact that $f^{-1} = f$, and the last uses the first and second equalities backward. Now the reversibility of $\delta_{f(x)}$ with respect to the Lebesgue measure follows from Lemma 2.24. The special case of δ_x is obtained by setting $f(x) = x$. □

Another useful property of the Dirac measure is the fact that integrating over it evaluates the integrated function [Çın11]: for any measurable $f\colon \mathbb{R}^d \to \mathbb{R}_+$ and $x_0 \in \mathbb{R}^d$,

$$\int_{\mathbb{R}^d} \delta_{x_0}(\mathrm{d}x)f(x) = f(x_0).$$

This property has a simple generalisation when integrating over a set A:

Lemma 2.32. *For any measurable $f\colon \mathbb{R}^d \to \mathbb{R}_+$, $A \in \mathcal{B}(\mathbb{R}^d)$ and $x_0 \in \mathbb{R}^d$,*

$$\int_A \delta_{x_0}(\mathrm{d}x)f(x) = 1_A(x_0)f(x_0).$$

Proof.

$$\int_A \delta_{x_0}(\mathrm{d}x)f(x) = \int_{\mathbb{R}^d} \delta_{x_0}(\mathrm{d}x)\, 1_A(x)f(x) = 1_A(x_0)f(x_0).$$

□

3. Differentially Private MH

As seen in Section 2.1, an algorithm can be made differentially private by adding Gaussian noise its output. The noise could also be added to any intermediate value calculated by the algorithm, and post processing immunity will guarantee that the same DP bounds that hold for releasing the intermediate value also hold for releasing the final result of the algorithm.

In 2019, Yildirim and Ermis [YE19] realised that if Gaussian noise is added to the exact value of λ from Equation 2.1, the noise can be corrected for, yielding a DP MH algorithm, called DP penalty, which converges to the correct distribution. In the same year, Heikkilä et al. [HJDH19] developed another DP MCMC algorithm, called DP Barker, which uses subsampling to amplify privacy. DP Barker is based on the Barker acceptance test [Bar65] instead of the MH test, and uses an approximation of the Barker acceptance test to correct for the noise added for DP.

Also in 2019, Sommer et al. proved the ADP-bound of Theorem 2.11 [SMM19], which gives tight bounds for ADP, unlike the zCDP based privacy accounting Yildirim and Ermis [YE19] used. Section 3.2 applies the new ADP-bound to DP penalty. Section 3.4 combines DP penalty with the subsampling idea from DP Barker, and formulates a subsampled DP penalty algorithm.

3.1 Differentially Private MH with the Penalty Algorithm

In 1999, Ceperley and Dewing [CD99] developed a variant of the MH algorithm called the penalty algorithm, where only a noisy approximation of λ is known. The original algorithm was developed for simulations in physics where computing λ requires computing energies of complex systems, which can only be approximated. The penalty algorithm modifies the acceptance probability to account for the noise added to λ and still converges to the correct distribution if the noise is Gaussian with known variance.

The DP penalty algorithm adds Gaussian noise to the value of λ, and uses the penalty algorithm to correct the acceptance probability so that the algorithm still converges to the correct distribution [YE19]. The corrected acceptance probability for Gaussian noise with variance σ^2 is

$$\min\{1, e^{\lambda(\theta,\theta')-\frac{1}{2}\sigma^2}\}$$

Theorem 3.1 gives the number of iterations DP penalty can be run for when the privacy cost is computed through zCDP, which is what Yildirim and Ermis prove in their paper [YE19].

Theorem 3.1. *Let $\epsilon > 0$, $0 < \delta < 1$, $\alpha > 0$, $\tau > 0$ and $n > 0$. Let*

$$\rho = (\sqrt{\epsilon - \ln\delta} - \sqrt{-\ln\delta})^2$$

$$c(\theta,\theta') = \sup_{x_j,x_j'}(p(x_j \mid \theta') - p(x_j \mid \theta) - (p(x_j' \mid \theta') - p(x_j' \mid \theta)))$$

$$\sigma^2(\theta,\theta') = \tau^2 n^{2\alpha} c^2(\theta,\theta')$$

Then running DP penalty for

$$k = \lfloor 2\tau^2 n^{2\alpha}\rho \rfloor$$

iterations with dataset size n, when using σ^2 as the variance of the Gaussian noise and the substitute neighborhood relation, is (ϵ,δ)-ADP.

The τ-parameter controls the trade-off between the amount of noise and the number of iterations the algorithm is allowed to run for. From the expression for σ^2 and k in Theorem 3.1, the standard deviation of the noise increases linearly with τ, while the number of iterations increases quadratically.

Yildirim and Ermis [YE19] see α as a bound on c of the form

$$c(\theta,\theta') \leq Cn^{-\alpha}$$

for almost all θ and θ' and constant $C > 0$ [YE19, Assumption A1]. They also consider a weaker form where for any ε there is a C such that $P(c(\theta,\theta') > Cn^{-\alpha}) < \varepsilon$ [YE19, Assumption A4]. These bounds imply that σ does not increase with n, which Yildirim and Ermis see as a necessary condition for the DP penalty algorithm to perform well. They also recommend having $\alpha = \frac{1}{2}$, which is used in the experiments in Chapter 6. With clipping, the weaker bound can be achieved by setting the clip bound based on n. For the experiments in Chapter 6, the parameters are optimised separately for each n that is used, so setting any clip bound can be seen as choosing a value of C for each ε.

Theorem 3.1 requires a bound on sensitivity of the log-likelihood ratio. If there is a bound

$$|\ln p(x_j \mid \theta') - \ln p(x_j \mid \theta)| \leq L||\theta - \theta'||_2$$

for all x_j, θ and θ' then

$$c(\theta, \theta') \leq 2L||\theta - \theta'||_2.$$

The former bound is true in some models, such as logistic regression [YE19]. In other models it can be forced by clipping the log-likelihood ratios with the bound $b = L||\theta - \theta'||_2$. This will remove the guarantee of eventually converging to the correct posterior, but if L is chosen to be large enough, the clipping will not affect the acceptance decision frequently. As a tradeoff, picking a large L will increase the variance of the Gaussian noise and slow down convergence through it. The effects of clipping are empirically studied in Section 6.2.

Yildirim and Ermis [YE19] propose two potential variations to improve the performance of the penalty algorithm. The first variation, called *one component updates* (OCU) in this thesis, is only proposing changes in one dimension in a multidimensional problem. This decreases $||\theta - \theta'||_2$, which means that it decreases the noise variance.

The second variation is called *guided walk Metropolis Hastings* (GWMH) [YE19]. In GWMH, proposals change only one dimension, as in OCU. Additionally, a direction is associated with each dimension, and proposals are only made in the current direction of the chosen dimension. After an accepted proposal, the direction is kept the same, but after a rejection it is switched. This means that the chain can move towards areas of higher probability faster because, after some initial proposals are rejected, the directions for each dimension point towards the area of high probability, so all proposals are towards it. Without GRMH, most proposals would move the chain away from the area of high probability, and would likely be rejected.

3.2 Applying New Techniques to DP Penalty

The DP penalty algorithm is a composition of Gaussian mechanisms, so the ADP-bound of Theorem 2.11 is applicable. Unlike Theorem 3.1, Theorem 2.11 gives tight ADP-bounds, so using the latter instead of the former will always give better privacy bounds. Theorem 3.2 formulates the bound of Theorem 2.11 for DP penalty. The iteration numbers the theorems allow are compared in Section 6.1.

Theorem 3.2. *Let $\epsilon > 0$, $\alpha > 0$, $\tau > 0$ and $n > 0$. Define c and σ^2 as in Theorem 3.1. The DP penalty algorithm, after running for k iterations using σ^2 as the noise variance with dataset size n, is $(\epsilon, \delta(\epsilon))$-DP, with the substitute neighborhood relation, for*

$$\delta(\epsilon) = \frac{1}{2}\left(\operatorname{erfc}\left(\frac{\epsilon - k\mu}{2\sqrt{k\mu}}\right) - e^{\epsilon}\operatorname{erfc}\left(\frac{\epsilon + k\mu}{2\sqrt{k\mu}}\right)\right)$$

where $\mu = \frac{1}{2\tau^2 n^{2\alpha}}$. Furthermore, this privacy bound is tight.

Proof. DP penalty is an adaptive composition of Gaussian mechanisms that release noisy values of $\lambda(\theta, \theta')$. With the substitute neighborhood relation, the sensitivity of $\lambda(\theta, \theta')$ is $c(\theta, \theta')$. For the tight ADP bound used here, the sensitivity must be constant in each iteration. This is achieved by releasing $\frac{\lambda(\theta,\theta')}{c(\theta,\theta')}$ instead, which has sensitivity 1. $c(\theta, \theta')$ does not depend on X, so $\lambda(\theta, \theta')$ can be obtained from $\frac{\lambda(\theta,\theta')}{c(\theta,\theta')}$ by post-processing.

Adding Gaussian noise with variance σ_n^2 to $\frac{\lambda(\theta,\theta')}{c(\theta,\theta')}$ is equivalent to adding Gaussian noise with variance $\sigma_n^2 c^2(\theta, \theta')$ to $\lambda(\theta, \theta')$. Setting $\sigma_n^2 = \tau^2 n^{2\alpha}$ and plugging into the ADP bound of Theorem 2.11 proves the claim. □

Theorem 3.2 is harder to use than Theorem 3.1 because the number of iterations DP penalty can be run for given an (ϵ, δ)-bound cannot be computed analytically for the former. However, the maximum number of iterations can be solved for numerically. Algorithm 2 shows a simple procedure that solves the maximum number of iterations using the bisection method.

Algorithm 2: Maximise the number of iterations given ϵ, δ, τ and n. The zcdp-function computes the number of iterations Theorem 3.1 allows, and the adp-function computes $\delta(\epsilon)$ from Theorem 3.2. $\lfloor \cdot \rfloor$ is the floor function that rounds real numbers down. Note that the variables *low*, *high* and *new* are not necessarily integers, as Theorem 3.2 can handle a non-integer number of iterations.

```
$low = \text{zcdp}(\epsilon, \delta, \tau, n)$
$high = \max\{low, 1\}$
while $\text{adp}(\epsilon, high, \tau, n) < \delta$ do
    $high = high \cdot 2$
end
while $\lfloor high \rfloor - \lfloor low \rfloor > 1$ do
    $new = \frac{high+low}{2}$
    if $\text{adp}(\epsilon, new, \tau, n) > \delta$ then
        $high = new$
    end
    else
        $low = new$
    end
end
if $\text{adp}(\epsilon, \lfloor high \rfloor, \tau, n) < \delta$ then
    return $\lfloor high \rfloor$
end
else
    return $\lfloor low \rfloor$
end
```

3.3 The Barker Acceptance Test

The DP Barker algorithm of Heikkilä et al. [HJDH19] is based on the Barker acceptance test [Bar65] instead of the Metropolis-Hastings test. Instead of using the MH acceptance probability, the Barker acceptance test samples $V_{log} \sim \text{Logistic}(0, 1)$ and accepts if $\lambda + V_{log} > 0$.

If Gaussian noise with variance σ^2 is added to λ, as long as σ^2 is not too large, there exists an approximate correction distribution V_{corr} such that $\mathcal{N}(0, \sigma^2) + V_{corr}$ has approximately the same distribution as V_{log}. Because the variance of V_{log} is $\frac{\pi^2}{3}$ [HJDH19], the variance of V_{corr} must be $\frac{\pi^2}{3} - \sigma^2$, which means that there is an upper bound to the noise variance: $\sigma^2 < \frac{\pi^2}{3}$. Testing whether $\lambda + \mathcal{N}(0, \sigma^2) + V_{corr} > 0$ is approximately equivalent to testing whether $\lambda + V_{log} > 0$, which means that it is possible to derive a DP MCMC algorithm based on the Barker acceptance test if the correction distribution can be sampled from.

However, the analytical form of V_{corr} is not known [HJDH19]. Heikkilä et al. approximate the distribution with a Gaussian mixture model. This means that their algorithm only converges to an approximately correct distribution, but the approximation error can be made very small.

By computing the sum in λ only over a subset of the data, the running time of the algorithm is reduced, and the algorithm becomes less sensitive to changes in the input data. The latter property is called *subsampling amplification* of differential privacy [WBK19]. Using the λ computed with subsampling instead of the full data λ introduces an additional error that must be corrected for to have the algorithm converge to the correct distribution.

The *central limit theorem* (CLT) states that the distribution of a sum of random variables approaches a Gaussian distribution as more random variables are summed, if some conditions on the independence and variance of the random variables are met [SPCC17]. With the CLT, it can be argued that the error from using the subsampled λ instead of the full data λ has an approximately Gaussian distribution, if the subsample is large enough [SPCC17].

The variance of the error from subsampling can be estimated by the sample variance of the individual terms in the sum in λ [SPCC17]. This allows combining the errors from subsampling and the Gaussian noise from the Gaussian mechanism to a single Gaussian noise value. The V_{corr} distribution can then be used to approximate the Barker acceptance test as above [HJDH19]. See Algorithm 3 for the DP Barker algorithm*.

Heikkilä et al. [HJDH19] do not directly bound the sensitivity of λ as is done in

*See [HJDH19] for the sampling procedure of V_{corr}.

DP penalty, because the sample variance also depends on input data. Instead they directly bound the Rényi divergence between $\mathcal{N}(0, \sigma^2 - \sigma_b^2)$, where σ_b^2 is the batch sample variance, for two adjacent inputs. Subsampling amplification is accounted for with an amplification theorem for RDP [WBK19].

Theorem 3.3. *If*

$$|\ln p(x_j \mid \theta') - \ln p(x_j \mid \theta)| \leq \frac{\sqrt{b}}{n}$$

$$2 < \alpha < \frac{b}{5}, \alpha \in \mathbb{N}$$

for all θ, θ' and x_j, running k iterations of DP Barker with dataset size n, minibatch size b and noise variance $\sigma^2 = 2$ is $(\alpha, k\epsilon(\alpha))$-RDP for the substitute neighborhood relation, with

$$\epsilon(\alpha) = \frac{1}{\alpha - 1} \ln \left(1 + q^2 \binom{\alpha}{2} \min\{4(e^{\epsilon'(2)} - 1), 2e^{\epsilon'(2)}\} + 2 \sum_{j=3}^{\alpha} q^j \binom{\alpha}{j} e^{(j-1)\epsilon'(j)} \right)$$

and

$$\epsilon'(\alpha) = \frac{5}{2b} + \frac{1}{2(\alpha - 1)} \ln \frac{2b}{b - 5\alpha} + \frac{2\alpha}{b - 5\alpha}$$

where $q = \frac{b}{n}$ [HJDH19].

The RDP bounds given by Theorem 3.3 can be converted to ADP bounds with Theorem 2.5. This requires choosing an optimal α in the given range that maximises the number of iterations the algorithm can run for.

Like DP penalty, DP Barker requires a bound on the log-likelihood ratio for one row of data. The bound can be forced through clipping if the model does not meet it, but because of the n in the denominator of the bound, it can get very tight for large values of n. As a result, clipping may be needed for almost all log-likelihood ratios, which may cause the algorithm to converge to a very different distribution from the posterior.

To alleviate the tight bound on log-likelihood sensitivity, DP Barker is best used with a tempered likelihood [HJDH19]. In tempering, the log-likelihood is multiplied by a number $T = \frac{n_0}{n} < 1$. This increases the variance of the resulting posterior and may lower modeling error in some cases [HJDH19].

Using the tempered likelihood, the log-likelihood bound becomes

$$T|\ln p(x_j \mid \theta') - \ln p(x_j \mid \theta)| \leq \frac{\sqrt{b}}{n},$$

which is equivalent to

$$|\ln p(x_j \mid \theta') - \ln p(x_j \mid \theta)| \leq \frac{\sqrt{b}}{n_0}.$$

Typically $n_0 \ll n$ for large datasets, so using a tempered likelihood requires significantly less clipping than a nontempered likelihood.

Algorithm 3: DP Barker: number of iterations, initial value θ_0, proposal distribution q, noise variance σ^2, data X as input.

```
for $1 \leq i \leq k$ do
    denote $\theta = \theta_{i-1}$
    sample $\theta' \sim q(\cdot \mid \theta)$
    sample $B \subset \{1, \dots, n\}$
    for $j \in B$ do
        $r_j = \mathrm{clip}_{\frac{\sqrt{b}}{n_0}}(\ln p(x_j \mid \theta') - \ln p(x_j \mid \theta))$
    end
    $\sigma_b^2 = \mathrm{Var}\{r_j \mid j \in B\}$
    $\lambda = \frac{n}{b} \sum_{j \in B} r_j + \ln \frac{p(\theta')}{p(\theta)} + \ln \frac{q(\theta|\theta')}{q(\theta'|\theta)}$
    sample $s \sim \mathcal{N}(0, \sigma^2 - \sigma_b^2)$
    sample $c \sim V_{corr}^{\sigma^2}$
    $\theta_i = \begin{cases} \theta' & \text{if } \lambda + s + c > 0 \\ \theta & \text{otherwise} \end{cases}$
end
return $(\theta_1, \dots, \theta_k)$
```

Tuning the parameters of DP Barker is more restrictive than DP penalty, as the former does not have a tunable clip bound, and its noise variance has an upper bound. In addition, choosing a noise variance requires computing the V_{corr} approximation for that variance. For these reasons, only noise variance $\sigma = 2$ from Theorem 3.3 is used, as was done by Heikkilä et al. [HJDH19].

3.4 The Penalty Algorithm with Subsampling

In the DP Barker algorithm, the log-likelihood ratio is computed using only a subsample of the dataset, which amplifies privacy. Subsampling can also be used with the penalty algorithm in the same way, if the acceptance test is corrected for the error from subsampling.

As with DP Barker, the error from subsampling is approximately normally distributed by the central limit theorem. The variance of the subsampling error can be estimated from the sample variance of individual terms of the sum in the log-likelihood ratio. This means that the penalty method can be used to correct for the subsampling error.

The acceptance probability with subsampling is

$$\min\{1, e^{\lambda^*(\theta,\theta')-\frac{1}{2}(\sigma^2+\sigma_b^2)}\},$$

where

$$\lambda^*(\theta,\theta') = \frac{nT}{b}\sum_{j\in B}\ln\frac{p(x_j \mid \theta')}{p(x_j \mid \theta)} + \ln\frac{p(\theta')q(\theta \mid \theta')}{p(\theta)q(\theta' \mid \theta)},$$

T is the tempering multiplier and σ_b^2 is the sample variance of the log-likelihood ratios in batch B. Denote

$$r_j = \ln\frac{p(x_j \mid \theta')}{p(x_j \mid \theta)},$$

$$R = \sum_{x\in B} r_j.$$

Then σ_b^2 can be estimated from the sample variance of r_j:

$$\begin{aligned}\sigma_b^2 &= \mathrm{Var}\left(\frac{nT}{b}\sum_{j\in B} r_j\right) = \frac{nT^2}{b^2}\sum_{j\in B}\mathrm{Var}(r_j) = \frac{nT^2}{b}\mathrm{Var}(r_j)\\ &\approx \frac{(nT)^2}{b^2}\sum_{j\in B}\left(r_j - \frac{R}{b}\right)^2 = \frac{(nT)^2}{b^2}\left(\sum_{j\in B} r_j^2 - \frac{R^2}{b}\right).\end{aligned}$$

Because σ_b^2 depends on the data, releasing λ privately is not enough, $\lambda - \frac{1}{2}\sigma_b^2$ must be released privately. This means that using subsampling requires adding additional noise to account for the sensitivity of $\frac{1}{2}\sigma_b^2$.

The sensitivity of $\lambda - \frac{1}{2}\sigma_b^2$ is

$$\Delta\lambda + \frac{1}{2}\Delta\sigma_b^2.$$

With the bound $r_j \leq L||\theta - \theta'||_2$ used in DP penalty, the bound sensitivity of λ is the same as without subsampling. The sensitivity of σ_b^2 must be bounded separately.

Lemma 3.4. *The sensitivity of $\frac{1}{2}\sigma_b^2$, when $r_j \leq L||\theta - \theta'||_2$, has upper bound*

$$\frac{1}{2}\Delta\sigma_b^2 \leq \left(\frac{nT}{b}\right)^2\left|1 - \frac{1}{b}\right| L^2||\theta - \theta'||_2^2 + \frac{2(b-1)}{b}\left(\frac{nT}{b}\right)^2 L^2||\theta - \theta'||_2^2.$$

Proof. For substitute neighboring datasets $X \sim X'$, denote the common part they have by X^*, and the differing element by $x \in X$ and $x' \in X'$. Then

$$\begin{aligned}
\Delta\sigma_b^2 &= \sup_{D\sim D'} |\sigma_b^2(X) - \sigma_b^2(X')| \\
&= \left(\frac{nT}{b}\right)^2 \sup_{X\sim X'} \left| \sum_{x\in X} r^2(x) - \sum_{x\in X'} r^2(x) + \frac{1}{b}R^2(X') - \frac{1}{b}R^2(X) \right| \\
&= \left(\frac{nT}{b}\right)^2 \sup_{x,x',X^*} \left| r^2(x) - r^2(x') + \frac{1}{b}(R(X^*) + r(x'))^2 - \frac{1}{b}(R(X^*) + r(x))^2 \right| \\
&= \left(\frac{nT}{b}\right)^2 \sup_{x,x',X^*} \Big| r^2(x) - r^2(x') + \frac{1}{b}(R^2(X^*) + 2R(X^*)r(x') + r^2(x')) \\
&- \frac{1}{b}(R^2(X^*) + 2R(X^*)r(x) + r^2(x)) \Big| \\
&= \left(\frac{nT}{b}\right)^2 \sup_{x,x',X^*} \left| \left(1 - \frac{1}{b}\right)(r^2(x) - r^2(x')) + \frac{2}{b}D(X^*)(r(x') - r(x)) \right| \\
&\le \left(\frac{nT}{b}\right)^2 \left|1 - \frac{1}{b}\right| \sup_{x,x'} \left|(r^2(x) - r^2(x'))\right| + \frac{2}{b}\left(\frac{nT}{b}\right)^2 \sup_{x,x',X^*} \left|R(X^*)(r(x') - r(x))\right| \\
&= \left(\frac{nT}{b}\right)^2 \left|1 - \frac{1}{b}\right| \sup_{x,x'} \left|(r^2(x) - r^2(x'))\right| + \frac{2}{b}\left(\frac{nT}{b}\right)^2 \sup_{x,x'} |r(x') - r(x)| \sup_{X^*} |R(X^*)| \\
&\le \left(\frac{nT}{b}\right)^2 \left|1 - \frac{1}{b}\right| \sup_{x,x'} \left|(r^2(x) - r^2(x'))\right| + \frac{2}{b}\left(\frac{nT}{b}\right)^2 \sup_{x,x'} |r(x') - r(x)|(b-1) \sup_{d} |r(x)|.
\end{aligned}$$

Plugging the bound $\sup_x |r(x)| \le L||\theta - \theta'||_2$ into the last expression proves the claim.

□

Theorem 3.5. *Let $\epsilon > 0$, $0 \le \delta < 1$, $\tau > 0$ $T > 0$, $L > 0$, $b > 0$, $n > 0$,*

$$\Delta_\lambda = \frac{2nTL}{b}||\theta - \theta'||_2,$$

$$\Delta_\sigma = \left(\frac{nT}{b}\right)^2 \left|1 - \frac{1}{b}\right| L^2||\theta - \theta'||_2^2, + \frac{2(b-1)}{b}\left(\frac{nT}{b}\right)^2 L^2||\theta - \theta'||_2^2,$$

$$c(\theta, \theta') = \Delta_\lambda + \Delta_\sigma,$$

$$\sigma^2(\theta, \theta') = \tau c^2(\theta, \theta').$$

Then running minibatch DP penalty with subsample size b for a dataset size n for k iterations is $(\alpha, k\epsilon(\alpha))$-RDP for the substitute neighborhood relation, with

$$\epsilon(\alpha) = \frac{1}{\alpha - 1}\ln\left(1 + q^2\binom{\alpha}{2}\min\{4(e^{\epsilon'(2)} - 1), 2e^{\epsilon'(2)}\} + 2\sum_{j=3}^{\alpha} q^j \binom{\alpha}{j} e^{(j-1)\epsilon'(j)}\right),$$

$$\epsilon'(\alpha) = \frac{\alpha}{2\tau},$$

and $q = \frac{b}{n}$.

Proof. By Lemma 3.4, $\Delta_\sigma(\theta, \theta')$ an upper bound to the sensitivity of $\frac{1}{2}\sigma_b^2$, therefore $c(\theta, \theta')$ is an upper bound to the sensitivity of $\lambda - \frac{1}{2}\sigma_b^2$.

This means that a Gaussian mechanism taking a subsample B of the data as input and uses $\sigma(\theta, \theta')$ as the noise variance is $(\alpha, \epsilon'(\alpha))$-RDP with

$$\epsilon'(\alpha) = \frac{\alpha}{2\tau}.$$

By the subsampling amplification theorem [WBK19, Theorem 9] and the composition theorem of RDP (Theorem 2.7), the combination of subsampling and Gaussian mechanism is $(\alpha, k\epsilon(\alpha))$-RDP with

$$\epsilon(\alpha) = \frac{1}{\alpha - 1}\ln\left(1 + q^2\binom{\alpha}{2}\min\{4(e^{\epsilon'(2)} - 1), 2e^{\epsilon'(2)}\} + 2\sum_{j=3}^{\alpha} q^j \binom{\alpha}{j} e^{(j-1)\epsilon'(j)}\right)$$

when run for k iterations for integer $\alpha \geq 2$. □

Theorem 3.5 does not give tight bounds for ADP, as it is based on RDP. Computing tight ADP bounds for the minibatch penalty algorithm requires an analogue of the ADP bound of Theorem 2.11 for the Gaussian mechanism with subsampling. An analytical and tight ADP bound for the subsampled Gaussian mechanism is not known, but the *Fourier accountant* of Koskela et al. [KJH20] can approximate the tight bound with a very small error.

The Fourier accountant takes the subsampling ratio q and noise variance, and computes the ADP bounds of the subsampled Gaussian mechanism with sensitivity 1. For the minibatch penalty algorithm, the released value $\lambda - \frac{1}{2}\sigma_b^2$ has sensitivity $c(\theta, \theta')$, and noise variance $\tau c^2(\theta, \theta')$, but if the released value is normalised to have sensitivity 1, the noise variance becomes τ, which makes the Fourier accountant easy to apply to the minibatch penalty algorithm. As with the DP penalty algorithm, the maximum number of iterations for minibatch DP penalty can be computed using Algorithm 2 by computing values of δ in Algorithm 2 with the Fourier accountant*.

The DP Barker algorithm, like minibatch penalty, uses RDP to compute privacy bounds. Using the Fourier accountant with DP Barker may be possible, but it is not as simple as with minibatch penalty, as DP Barker does not consider releasing the sample variance σ_b^2 directly.

The appearance of n as a multiplier of Δ_λ and n^2 as a multiplier of Δ_σ causes the noise variance to increase with n, without a corresponding decrease in ϵ'. This makes subsampled DP penalty unsuitable for problems with large datasets, unless tempering is used. Like with DP Barker, using tempering $T = \frac{n_0}{n}$ cancels n out of the noise variance.

*The Fourier accountant handles the non-integer iteration numbers used by Algorithm 2 by internally rounding them down.

The minibatch DP penalty algorithm has the same parameters as DP penalty, with the addition of the batch size parameter. The restrictions on the noise variance of DP Barker are not present in minibatch DP penalty. In Theorem 3.5, the expression for σ was simplified by multiplying c^2 by only a single number τ, instead of the $\tau^2 n^{2\alpha}$ used in Theorem 3.1 after Yildirim and Ermis [YE19]. This makes using the Fourier accountant particularly simple, as the noise variance given to it is simply τ.

The proposed variations of the penalty algorithm, one component updates and guided walk MH, are also applicable with subsampling. The sensitivity reduction from OCU may be especially useful, as the variance sensitivity contains the square of the distance between the current and proposed parameter values, so reducing the distance by updating only a single component can greatly reduce the added noise.

4. Differentially Private Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC[†]) [DKPR87, Nea12] is a widely used MCMC algorithm that, like Metropolis-Hastings and the penalty algorithm, was originally developed for simulations in physics, but has since been applied to Bayesian inference.

HMC simulates the trajectory of a moving particle in a way that allows it to draw samples from a target distribution [Nea12]. With perfect simulation, an acceptance test would not be needed, but exact simulation of the moving particle is typically not possible, so a Metropolis-Hastings acceptance test is used for proposals generated by the simulation. Despite imperfections, the simulation can generate long jumps that stay in areas of high probability, giving HMC a higher acceptance rate than Metropolis-Hastings using a Gaussian distribution as the proposal, at the cost of higher computational requirements.

This chapter first introduces HMC in the non-DP setting in Section 4.1. In Section 4.2 HMC is made differentially private using the acceptance test of the penalty algorithm and adding noise to the proposal phase appropriately.

4.1 MCMC with Hamiltonian Dynamics

The motion of a particle on a frictionless surface of varying height is governed by the initial position and velocity of the particle, and the height of the surface at different positions [Nea12]. The kinetic energy of the particle with momentum[‡] $p \in \mathbb{R}^2$ and mass $m \in \mathbb{R}$ is $\frac{p^T p}{2m}$. The potential energy of the particle at position $\theta \in \mathbb{R}^2$ is proportional to the height of the surface, given by a continuously differentiable function U. The sum of potential and kinetic energies, the total energy of the system, is called the Hamiltonian H. Hamilton's equations give the equations of motion for a system with

[†] HMC originally stood for hybrid Monte Carlo, but the name Hamiltonian Monte Carlo has become more common.

[‡] Momentum is velocity times mass.

a given Hamiltonian:

$$\frac{d\theta}{dt} = \frac{\partial H}{\partial p}, \qquad\qquad \frac{dp}{dt} = -\frac{\partial H}{\partial \theta}.$$

Given $H(\theta, p) = U(\theta) + \frac{p^T p}{2m}$, Hamilton's equations become

$$\begin{aligned} \frac{d\theta}{dt} &= \frac{p}{m}, \\ \frac{dp}{dt} &= -\nabla U(\theta). \end{aligned}$$

These equations define a function T_t taking an initial state (θ, p) of the particle to the state $T(\theta, p)$ after time t. Solving T_t analytically is usually not possible, but T_t can be approximately simulated by the *leapfrog method*. The leapfrog method sequentially updates an estimate (θ_i, p_i) of the state in steps L of η as follows:

$$\begin{aligned} p_{i+1/2} &= p_i - \frac{\eta}{2}\nabla U(\theta_i), \\ \theta_{i+1} &= \theta_i + \frac{\eta p_{i+1/2}}{m}, \\ p_{i+1} &= p_{i+1/2} - \frac{\eta}{2}\nabla U(\theta_{i+1}). \end{aligned}$$

The simulation starts with $\theta_0 = \theta$ and $p_0 = p$ and the result is given by $T_{L\eta}(\theta, p) \approx (\theta_L, p_L)$.

The above equations for two-dimensional θ and p generalize to any number of dimensions d [Nea12]. The physical analogue would be a particle moving on a d-dimensional hypersurface with "height" given by $U(\theta)$. The mass of the particle, m, can also be generalized to any positive definite matrix $M \in \mathbb{R}^{d\times d}$, which changes division by m to multiplication by M^{-1}. Specifically, the kinetic energy of the particle becomes $\frac{1}{2}p^T M^{-1} p$ and $\frac{d\theta}{dt} = M^{-1}p$. The generalization to a mass matrix does not have a physical analogue, but it can be useful for MCMC.

Hamiltonian dynamics have three important properties [Nea12] that make them useful as a proposal mechanism for MCMC. They are *invertibilityi*[*], *conservation of the Hamiltonian* and *conservation of the Lebesgue measure*[†].

Invertibility means that the function T_t is injective, meaning that it has an inverse T_{-t} where $T_t(\theta_1, p_1) = (\theta_2, p_2)$ implies that $T_{-t}(\theta_2, p_2) = (\theta_1, p_1)$ [Nea12]. For the Hamiltonian of the particle moving on a surface, the inverse is obtained by $T_t(\theta_2, -p_2) = (\theta_1, -p_1)$.

[*] Invertibility is commonly called reversibility [Nea12], but calling it reversibility in this thesis risks conflating it with the closely related concept of Markov kernel reversibility from Definition 2.23.

[†] Also known as conservation of volume.

Conservation of the Hamiltonian follows from Hamilton's equations and the chain rule of derivatives:

$$\frac{dH}{dt} = \frac{\partial H}{\partial \theta}\frac{d\theta}{dt} + \frac{\partial H}{\partial p}\frac{dp}{dt} = \frac{\partial H}{\partial \theta}\frac{\partial H}{\partial p} - \frac{\partial H}{\partial p}\frac{\partial H}{\partial \theta} = 0.$$

Conservation of the Lebesgue measure in the (θ, p)-space follows from the fact that the absolute value of the Jacobian determinant of T_t is 1 and the injectivity of T_t. A somewhat lengthy proof for the Jacobian determinant of T_t is given by Neal [Nea12].

To see why having $|\det J_f| = 1$, where J_f is the Jacobian matrix of an injective continuously differentiable function f, implies f preserving the Lebesgue measure, recall that the measure of a set A is given by

$$m(A) = \int_A m(\mathrm{d}x)\, 1.$$

The volume of the image of A under f, $f(A)$, is given by

$$m(f(A)) = \int_{f(A)} m(\mathrm{d}x)\, 1 = \int_A m(\mathrm{d}x) |\det J_f(x)| = \int_A m(\mathrm{d}x)\, 1 = m(A).$$

where the first equality follows from the change of variables formula.

The leapfrog method does not conserve the Hamiltonian exactly, but it preserves Lebesgue measure and is invertible exactly [Nea12]. These properties are important for its use as a proposal mechanism for MCMC.

The HMC algorithm samples from the distribution of [Nea12]

$$\pi(\theta, p) \propto \exp(-H(\theta, p)).$$

With $H(\theta, p) = U(\theta) + \frac{1}{2}p^T M^{-1} p$,

$$\pi(\theta, p) \propto \exp(-U(\theta)) \exp\left(\frac{1}{2}p^T M^{-1} p\right)$$

This means that the marginal distributions of θ and p are independent, the marginal distribution of p is a d-dimensional Gaussian distribution with mean 0 and covariance M, and the marginal distribution of θ can have any continuously differentiable log-density $-U$. Because of this, θ is used to represent the variables of interest, while p is a set of auxiliary variables used for the proposals. In Bayesian inference,

$$U(\theta) = -\sum_{x \in X} \ln p(x \mid \theta) - \ln p(\theta)$$

for dataset X. The requirement that U is a continuously differentiable and defined on $\mathbb{R}^d$ means that the log-likelihood $\ln p(x \mid \cdot)$ and the log-prior must also meet these conditions. This means that HMC cannot directly be used with discrete parameters, non-differentiable log-likelihoods or log-priors, or constrained parameters, such as variance parameters that must be positive. Constrained parameters can typically be transformed into unconstrained ones, such as transforming $\sigma = e^s$ for a variance parameter σ^2, and using s in the model instead of σ^2. Discrete parameters or non-differentiable models require other MCMC algorithms, although it may be possible to combine them with HMC if there are continuous parameters.

Generating a (θ, p)-sample is done in two stages [Nea12], which are technically two separate Metropolis-Hastings proposals and acceptance tests. In the first stage, p is proposed from the Gaussian distribution with mean 0 and covariance M. Because the proposal matches the marginal distribution of p, the Metropolis-Hastings acceptance probability is 1, so the proposal is always accepted.

In the second stage, Hamiltonian dynamics for the particle on a surface are simulated with the leapfrog method, starting from the current value of θ and value of p from the previous stage. After the simulation, p is negated, and the resulting (θ, p) pair is used as the proposed value.

Because the proposal is deterministic, the Markov kernel for the proposal is $\delta_{f(\theta,p)}$, where $f(\theta, p)$ denotes the proposal of running the leapfrog simulation starting with (θ, p) and negating p after the simulation. The invertibility of the leapfrog simulation means that $f = f^{-1}$. Together with the measure preservation of the leapfrog simulation, this means that f meets the conditions of Lemma 2.31. Combining this with Lemma 2.29 means that π is the invariant distribution of HMC when the acceptance probability is simply

$$\min\left\{1, \frac{\pi(\theta', p')}{\pi(\theta, p)}\right\} = \min\{1, \exp(H(\theta, p) - H(\theta', p'))\}.$$

Because of the deterministic proposal, the ergodicity of HMC cannot be proved with strong irreducibility, but it has been shown that HMC is ergodic under some conditions on U [DMS19].

If the simulation conserved H exactly, the acceptance probability would always be 1. Because this is usually not possible, there is always some probability of rejecting, which depends on the length of the jump taken by the leapfrog method, η, and the number of leapfrog steps, L. It might be expected that the errors from the leapfrog steps would accumulate during the simulation, but in practice the errors in different steps tend to cancel each other [Nea12], meaning that the acceptance probability mainly depends on η.

4.2 Differential Privacy with HMC

HMC only uses the model log-likelihood, and thus the data, through U. U is used in two ways. Firstly, its value is used in the acceptance test, and secondly, its gradients are used to obtain proposals. Adding appropriate amounts of noise to both accesses would make HMC differentially private, but the addition of noise may break some of the properties required for the algorithm's correctness.

The addition of noise to the log-likelihood ratios can be corrected using the penalty acceptance test, i.e. changing the acceptance probability to

$$\min\left\{\exp\left(H(\theta, p) - H(\theta', p') - \frac{1}{2}\sigma_l^2\right)\right\},$$

where σ_l^2 is the variance of the noise added to the log-likelihood ratios. This is because with $U(\theta) = -\ln p(X \mid \theta) - \ln p(\theta)$, for data X, the difference of the Hamiltonians in the acceptance probability is

$$H(\theta, p) - H(\theta', p') = \ln p(X \mid \theta') - \ln p(X \mid \theta) - \ln p(\theta) + \ln p(\theta') + \frac{1}{2}(p^T M^{-1} p - p'^T M^{-1} p').$$

As with the penalty algorithm, the sensitivity of the log-likelihood ratios must be bounded. If a bound does not exist, clipping must be used, nullifying the asymptotic converge guarantee of the algorithm. However, in Section 6.2 it is shown that clipping low numbers of gradients does not affect convergence in practice.

Releasing the gradients privately requires a trade-off. The leapfrog method remains reversible with noisy and potentially clipped gradients, as is shown in the following, but it no longer simulates the correct Hamiltonian dynamics [CFG14]. This does not affect asymptotic convergence, but it can potentially lower acceptance rates. The dynamics can be corrected by adding a friction term to the equations of motion of the simulation [CFG14], but this removes volume preservation of the simulation, which means that the acceptance probability is no longer valid.

The leapfrog update equations with noisy and clipped gradients become

$$\begin{aligned} p_{i+1/2} &= p_i - \frac{\eta}{2}(g(\theta_i) + \xi) \\ \theta_{i+1} &= \theta_i + \eta M^{-1} p_{i+1/2} \\ p_{i+1} &= p_{i+1/2} - \frac{\eta}{2}(g(\theta_{i+1}) + \xi)), \end{aligned}$$

where $\xi \sim \mathcal{N}_d(0, \sigma_g^2 I)$ and

$$g(\theta) = \sum_{x \in X} \mathrm{clip}_b(\nabla \ln p(x \mid \theta)) + \nabla \ln p(\theta)$$

denotes the clipped gradients summed over the data X.

Using the leapfrog update equations as is results in unnecessary evaluations of the gradients, as the last momentum update of a step and the first momentum update of the next step use the same gradients. Computing the same gradients and incurring their privacy cost twice can be avoided by writing the leapfrog simulation as

$$l_{p_{t_2}} \circ l_\theta \circ l_{p_{t_1}} \circ l_\theta \cdots l_\theta \circ l_{p_{t_1}} \circ l_\theta \circ l_{p_{t_2}}$$

where

$$l_\theta(\theta, p) = (\theta + \eta M^{-1} p, p)$$

and

$$l_{p_t}(\theta, p) = (\theta, p - t(g(\theta) + \xi)).$$

where $\xi \sim \mathcal{N}_d(0, \sigma_g^2 I)$, $t_1 = \eta$ and $t_2 = \frac{\eta}{2}$. Both l_θ and l_{p_t} preserve measure, as l_θ in unchanged from the non-DP case and the Jacobian of l_{p_t} is

$$J_{l_{p_t}}(\theta, p) = \begin{bmatrix} I_{d\times d} & 0_{d\times d} \\ tJ_g(\theta) & I_{d\times d} \end{bmatrix},$$

where $I_{d\times d}$ and $0_{d\times d}$ are the d-by-d identity and zero matrices, respectively. It is possible that $J_g(\theta)$ does not exist, but this can only happen in a null set*, if g is sufficiently well-behaved. The non-existence of $J_g(\theta)$ in a null set does not affect the volume preservation of l_{p_t}. As g is a sum of clipped log-likelihood gradients and the log prior gradient, it suffices to show that the log-likelihood is sufficiently well-behaved, as the prior is already assumed to be differentiable and is not clipped. Appendix B gives very general sufficient conditions for the log-likelihood, and proves that all of the models considered in Chapter 5 meet the conditions.

Because $J_{l_{p_t}}(\theta, p)$ is triangular and the diagonal consists of only ones, $\det J_{l_{p_t}}(\theta, p) = 1$ for all θ and p, which means that l_{p_t} preserves volume. As the step updating θ also preserves volume, the entire leapfrog simulation preserves Lebesgue measure. Showing that the proposal of DP-HMC is reversible with respect to the Lebesgue measure requires more thought, as the individual leapfrog steps are partially deterministic.

With the momentum negation after the leapfrog simulation, the DP-HMC proposal can be written as

$$l_- \circ l_{p_{t_2}} \circ l_\theta \circ l_{p_{t_1}} \circ l_\theta \cdots l_\theta \circ l_{p_{t_1}} \circ l_\theta \circ l_{p_{t_2}}$$

where $l_-(\theta, p) = (\theta, -p)$. Reversibility of the proposal with respect to the Lebesgue measure can then shown by decomposing the sequence into parts that are individually reversible.

*A set of zero Lebesgue measure.

Lemma 4.1. *$l_\theta \circ l_-$ and $l_- \circ l_{pt}$ for any $x \in \mathbb{R}$ are reversible with respect to the Lebesgue measure.*

Proof. Starting with $l_\theta \circ l_l$, denote $f(\theta, p) = (l_\theta \circ l_-)(\theta, p) = (\theta - \eta M^{-1}p, -p)$. Then

$$f(f(\theta, p)) = f(\theta - \eta M^{-1}p, -p) = (\theta, p)$$

which means that $f^{-1} = f$. Because l_θ and l_- preserve Lebesgue measure, f also preserves Lebesgue measure. The Markov kernel for $l_\theta \circ l_-$ is $\delta_{f(\theta,p)}$, so Lemma 2.31 proves the claim for $l_\theta \circ l_l$.

For $l_- \circ l_{p_t}$, recall that

$$(l_- \circ l_{p_t})(\theta, p) = (\theta, -p + t(g(\theta) + \xi))$$

where $\xi \sim \mathcal{N}(0, \sigma_g^2)$. $l_- \circ l_{p_t}$ is a Markov kernel on $\mathbb{R}^{2d}$, so the measures ν_1 and ν_2 in Definition 2.23 are measures on $\mathbb{R}^{4d}$, and the Lebesgue measure in the claim is specifically m_{2d}.

Because $(\mathbb{R}^{2d}, \mathcal{B}(\mathbb{R}^{2d})) = (\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))^2$ and $m_{2d} = m_d^2$, by Lemma 2.25 it is sufficient to show that

$$\begin{aligned}&\int_A m_d(\mathrm{d}\theta) \int_B m_d(\mathrm{d}p) \int_{C\times D} (l_- \circ l_{p_x})(\theta, p, \mathrm{d}(\theta', p')) \\ =&\int_C m_d(\mathrm{d}\theta') \int_D m_d(\mathrm{d}p') \int_{A\times B} (l_- \circ l_{p_x})(\theta', p', \mathrm{d}(\theta, p))\end{aligned}$$

for all $A, B, C, D \in \mathcal{B}(\mathbb{R}^d)$. Denote $f_t(\theta) = tg(\theta)$ and $\sigma^2 = t^2\sigma_g^2$. Denote the density function of the d-dimensional normal distribution with mean μ and covariance Σ by $\mathcal{N}_d(\cdot \mid \mu, \Sigma)$. Then

$$\begin{aligned}\int_{C\times D} (l_- \circ l_{p_x})(\theta, p, \mathrm{d}(\theta', p')) &= \begin{cases} 0 & \text{if } \theta \notin C \\ \int_D m_d(\mathrm{d}p')\mathcal{N}_d(p' \mid -p + f_t(\theta), \sigma^2 I) & \text{if } \theta \in C \end{cases} &(4.1)\\ &= 1_C(\theta) \int_D m_d(\mathrm{d}p')\mathcal{N}_d(p' \mid -p + f_t(\theta), \sigma^2 I) &(4.2)\\ &= \int_C \delta_\theta(\mathrm{d}\theta') \int_D m_d(\mathrm{d}p')\mathcal{N}_d(p' \mid -p + f_t(\theta), \sigma^2 I) &(4.3)\end{aligned}$$

for $C, D \in \mathcal{B}(\mathbb{R}^d)$. The third equality follows from Lemma 2.32. The normal distribution has a shift-invariance property: for any $x, y \in \mathbb{R}^d$

$$\mathcal{N}(x \mid \mu, \Sigma) = \mathcal{N}(x + y \mid \mu + y, \Sigma),$$

which is a consequence of the fact that the density of the normal distribution only depends on x and μ through $x - \mu$. Then, for any $A, B, C, D \in \mathcal{B}(\mathbb{R}^d)$,

$$\begin{aligned}
&\int_A m_d(\mathrm{d}\theta) \int_B m_d(\mathrm{d}p) \int_{C\times D} (l_- \circ l_{p_t})(\theta, p, \mathrm{d}(\theta', p'))\\
&= \int_A m_d(\mathrm{d}\theta) \int_B m_d(\mathrm{d}p) \int_C \delta_\theta(\mathrm{d}\theta') \int_D m_d(\mathrm{d}p')\mathcal{N}_d(p' \mid -p + f_t(\theta), \sigma^2 I)\\
&= \int_A m_d(\mathrm{d}\theta) \int_C \delta_\theta(\mathrm{d}\theta') \int_B m_d(\mathrm{d}p) \int_D m_d(\mathrm{d}p')\mathcal{N}_d(p' \mid -p + f_t(\theta), \sigma^2 I)\\
&= \int_C m_d(\mathrm{d}\theta') \int_A \delta_{\theta'}(\mathrm{d}\theta) \int_B m_d(\mathrm{d}p) \int_D m_d(\mathrm{d}p')\mathcal{N}_d(p' \mid -p + f_t(\theta), \sigma^2 I)\\
&= \int_C m_d(\mathrm{d}\theta') \int_A \delta_{\theta'}(\mathrm{d}\theta) \int_B m_d(\mathrm{d}p) \int_D m_d(\mathrm{d}p')\mathcal{N}_d(p \mid -p' + f_t(\theta), \sigma^2 I)\\
&= \int_C m_d(\mathrm{d}\theta') 1_A(\theta') \int_B m_d(\mathrm{d}p) \int_D m_d(\mathrm{d}p')\mathcal{N}_d(p \mid -p' + f_t(\theta'), \sigma^2 I)\\
&= \int_C m_d(\mathrm{d}\theta') \int_D m_d(\mathrm{d}p') 1_A(\theta') \int_B m_d(\mathrm{d}p)\mathcal{N}_d(p \mid -p' + f_t(\theta'), \sigma^2 I)\\
&= \int_C m_d(\mathrm{d}\theta') \int_D m_d(\mathrm{d}p') \int_{A\times B} (l_- \circ l_{p_t})(\theta', p', \mathrm{d}(\theta, p))
\end{aligned}$$

where the first equality uses Equation 4.3, the second equality uses Theorem 2.19 (6) to change the order of integration, the third uses Lemma 2.26 on the two leftmost integrals, the fourth uses the shift-invariance of the normal distribution, the fifth uses Lemma 2.32, the sixth uses Theorem 2.19 (6) to rearrange the term in the integrals and the last equality uses Equation 4.2. □

Lemma 4.2. *The DP-HMC proposal $l_- \circ l$ is reversible with respect to the Lebesque measure.*

Proof. The DP-HMC proposal

$$l_- \circ l_{p_{1/2}} \circ l_\theta \circ l_{p_1} \circ l_\theta \cdots \circ l_\theta \circ l_{p_{1/2}}$$

can be written as

$$(l_- \circ l_{p_{1/2}}) \circ (l_\theta \circ l_-) \circ (l_- \circ l_{p_1}) \circ (l_\theta \circ l_-) \circ (l_- \circ l_{p_1}) \circ \cdots \circ (l_\theta \circ l_-) \circ (l_- \circ l_{p_{1/2}})$$

which is a composition of $l_- \circ l_{p_{1/2}}$, $l_- \circ l_{p_1}$ and $l_\theta \circ l_-$, which are all reversible with respect to the Lebesgue measure by Lemma 4.1. As the composition is symmetric, Lemma 2.28 extends the reversibility with respect to the Lebesgue measure to the entire composition. □

Reversibility of proposal with respect to the Lebesgue measure can now be used to show that DP-HMC has the correct invariant distribution using Lemma 2.29:

Theorem 4.3. *If*

$$\min\left\{1, \exp\left(H(\theta, p) - H(\theta', p') - \frac{1}{2}\sigma_l^2\right)\right\}$$

is used as the acceptance probability for DP-HMC, the invariant distribution is π.

Proof. Lemmas 2.29 and 4.2 together show that when using

$$\alpha(\theta, p, \theta', p') = \min\left\{1, \frac{\pi(\theta', p')}{\pi(\theta, p)}\right\}$$

as the acceptance probability, where π is any continuous probability distirbution on $\mathbb{R}^{2d}$, the invariant distribution of DP-HMC is π. Recall that in HMC

$$\pi(\theta, p) \propto \exp(-H(\theta, p))$$

so

$$\alpha(\theta, p, \theta', p') = \min\left\{1, \exp\left(H(\theta, p) - H(\theta', p')\right)\right\}.$$

When Gaussian noise with variance σ_l^2 is added to the log-likelihood ratios and the penalty correction is applied,

$$\alpha(\theta, p, \theta', p') = \min\left\{1, \exp\left(H(\theta, p) - H(\theta', p') - \frac{1}{2}\sigma_l^2\right)\right\}. \qquad \square$$

Proving the ergodicity of DP-HMC turns out to be easier than in the non-DP case as a result of adding noise to the proposal, which makes the algorithm strongly irreducible, as shown below.

Theorem 4.4. *DP-HMC is strongly irreducible and ergodic.*

Proof. Consider the last four updates of the proposal,

$$l_- \circ l_{p_{1/2}} \circ l_\theta \circ l_{p_1}.$$

Denote the values of (θ, p) before the last four updates by (θ^*, p^*) and the values after the updates by (θ', p'). The first of these updates adds Gaussian noise to p. In the next l_θ step, the previous value of p is added to θ after multiplication with a non-singular matrix, and the further updates do not change θ, so it is possible to obtain any value for θ', no matter the starting value of (θ^*, p^*). In the next $l_{p_{1/2}}$ update, Gaussian noise is added to the value of p and the l_- update does not change the distribution of p, so it is possible to obtain any value for p' from any (θ^*, p^*). This means that DP-HMC is strongly irreducible, and thus ergodic. □

Together, Theorems 4.3 and 4.4 show that DP-HMC converges to the same target distribution as HMC, though the convergence is slower than in HMC, as clipping and adding noise to gradients gradient, and adding noise to the log-likelihoods all lower the acceptance rate of DP-HMC.

If the friction term that corrects the dynamics is used, the equations of motion become [CFG14]

$$\begin{aligned}
\frac{d\theta}{dt} &= M^{-1}p \\
\frac{dp}{dt} &= -\nabla U(\theta) - \frac{1}{2}\sigma_g^2 M^{-1}p + \sigma_g B_d
\end{aligned}$$

where B_d denotes the d-dimensional Brownian motion. The equation for θ is unchanged, but the equation for p has the additional term $-\frac{1}{2}\sigma_g^2 M^{-1}p$. For the leapfrog updates, l_θ does not change, and l_p changes to

$$l_p(\theta, p) = \left(\theta, p - \frac{\eta}{2}(g(\theta) + \frac{1}{2}\sigma_g^2 M^{-1}p + \mathcal{N}(0, \sigma_g^2))\right).$$

l_p no longer preserves volumes, as the Jacobian of l_p is

$$J_{l_p}(\theta, p) = \begin{bmatrix} I_{d\times d} & 0_{d\times d} \\ J_g(\theta) & I_{d\times d} - \frac{\eta}{4}\sigma_g^2 M^{-1} \end{bmatrix}$$

which means that generally $|\det J_{l_p}(\theta, p)| \neq 1$. This means that the acceptance test is no longer correct, so the friction term in not used in this thesis beyond this brief discussion, but finding a way to correct the acceptance test is a potential avenue for future research.

DP HMC is shown in Algorithm 4. Like the DP penalty algorithm, DP HMC is a composition of Gaussian mechanisms, and its privacy bounds can be computed through Theorems 2.7 and 2.11 for zCDP.

Theorem 4.5. *Let $\epsilon \geq 0$, $0 \leq \delta < 1$, $\tau_g > 0$, $\tau_l > 0$, $b_g > 0$, $b_l > 0$ $n > 0$,*

$$\sigma_l(\theta, \theta') = 2\tau_l\sqrt{n}b_l||\theta - \theta'||_2,$$

$$\sigma_g = 2\tau_g\sqrt{n}b_g,$$
$$\rho = \left(\sqrt{\epsilon - \ln\delta} - \sqrt{-\ln\delta}\right)^2,$$
$$\rho_l = \frac{1}{2\tau_l^2 n},$$
$$\rho_g = \frac{1}{2\tau_g^2 n},$$

Then runnning DP HMC for

$$k = \left\lfloor \frac{\rho - \rho_g}{\rho_l + L\rho_g} \right\rfloor$$

iterations using σ_l^2 as log-likelihood ratio noise variance, σ_g^2 as gradient noise variance, clipping log-likelihood ratios with b_l and clipping gradients by b_g, for a dataset of size n is (ϵ, δ)-ADP for the substitute neighborhood relation.

Proof. Each iteration of the outer for-loop computes the gradient L times and the log-likelihood ratio once. The gradient is also computed once before the outer for-loop. Releasing a single gradient has zCDP privacy cost of

$$\rho_g = \frac{4b_g^2}{2\sigma_g^2} = \frac{1}{2\tau_g^2 n}.$$

Algorithm 4: DP HMC: log likelihood clip bound b_l, gradient clip bound b_g, clipped gradient function g_{b_g}, number of iterations k, number of leapfrog steps L, leapfrog step size η, initial point θ_0, gradient noise variance σ_g^2, log likelihood ratio noise variance σ_l^2, positive-definite mass matrix M and dataset X as input.

```
$G_{1,0} = g_{b_g}(\theta_0) + \mathcal{N}(0, \sigma_g^2)$
for $1 \leq i \leq k$ do
    $p_0 = \mathcal{N}_d(0, M)$
    $\theta_{i,0} = \theta_{i-1}$
    for $1 \leq j \leq L$ do
        $p_{j-\frac{1}{2}} = p_{j-1} + \frac{1}{2}\eta G_{i,j-1}$
        $\theta_{i,j} = \theta_{i,j-1} + M^{-1} p_{j-\frac{1}{2}}$
        $G_{i,j} = g_{b_g}(\theta_{i,j}) + \mathcal{N}(0, \sigma_g^2)$
        $p_j = p_{j-\frac{1}{2}} + \frac{1}{2}\eta G_{i,j}$
    end
    $r_l = \ln \frac{p(\theta_{i,L}|x_l)}{p(\theta_{i-1}|x_l)}$
    $R = \sum_{l=1}^{n} \mathrm{clip}_{b_l ||\theta_{i-1} - \theta_{i,L}||_2}(r_l)$
    $\Delta H = R + \frac{1}{2} p_0^T M^{-1} p_0 - \frac{1}{2} p_L^T M^{-1} p_L + \ln \frac{p(\theta_{i,L})}{p(\theta_{i-1})} + \mathcal{N}(0, \sigma_l^2(\theta_{i-1}, \theta_{i,L}))$
    $u = \mathrm{Unif}(0, 1)$
    if $u < \Delta H - \frac{1}{2}\sigma_l^2(\theta_{i-1}, \theta_{i,L})$ then
        $\theta_i = \theta_{i,L}$
        $G_{i+1,0} = G_{i,L}$
    end
    else
        $\theta_i = \theta_{i-1}$
        $G_{i+1,0} = G_{i,0}$
    end
end
```

Releasing the log-likelihood ratio of θ and θ' has privacy cost

$$\rho_l = \frac{4b_l^2||\theta - \theta'||_2^2}{2\sigma_l(\theta, \theta')^2} = \frac{1}{2\tau_l^2 n}.$$

The total zCDP budget with given (ϵ, δ)-bound is

$$\rho = \left(\sqrt{\epsilon - \ln \delta} - \sqrt{-\ln \delta}\right)^2.$$

This means that the number of iteration that the algorithm is allowed to run for is

$$k = \left\lfloor \frac{\rho - \rho_g}{\rho_l + L\rho_g} \right\rfloor \qquad \square$$

zCDP based privacy accounting for ADP is loose, so the above bound could be improved by using the tight bound for the Gaussian mechanism from Theorem 2.11. Because DP HMC has both different sensitivities and different noise variances between the log-likelihood ratios and gradients, Theorem 2.11 is not directly applicable. However, the bound of Theorem 2.11 does generalise to differing variances between composition.

Theorem 4.6. *Let $\epsilon \geq 0$, $0 \leq \delta < 1$, $\tau_l > 0$, $\tau_g > 0$, $b_l > 0$, $b_g > 0$, $n > 0$,*

$$\sigma_l'^2 = \tau_l^2 n,$$

$$\sigma_g'^2 = \tau_g^2 n.$$

Then running DP HMC for k iterations with L leapfrog steps, using $2b_l\sigma_l'^2||\theta - \theta'||_2$ as the log-likelihood ratio noise variance, $2b_g\sigma_g'^2$ as the gradient noise variance and b_l and b_g as the log-likelihood and gradient clip bounds, with dataset size n, is $(\epsilon, \delta(\epsilon))$-ADP for the substitute neighborhood relation, where

$$\delta(\epsilon) = \frac{1}{2}\left(\operatorname{erfc}\left(\frac{\epsilon - \mu}{2\sqrt{\mu}}\right) - e^{\epsilon}\operatorname{erfc}\left(\frac{\epsilon + \mu}{2\sqrt{\mu}}\right)\right)$$

and

$$\mu = \frac{k}{2\sigma_l'^2} + \frac{kL + 1}{2\sigma_g'^2}.$$

Proof. Sommer et al. [SMM19] prove the ADP bound of Theorem 2.11 with the *privacy loss distribution* (PLD) of the Gaussian mechanism. First, they show that the PLD of the Gaussian mechanism with sensitivity Δ and variance σ^2 is a Gaussian distribution $\mathcal{N}(\mu, 2\mu)$, where $\mu = \frac{\Delta^2}{2\sigma^2}$ [SMM19, Lemma 11]. Next, they show that the ADP bounds $(\epsilon, \delta(\epsilon))$ for a Gaussian PLD $\mathcal{N}(\mu, 2\mu)$ are given by [SMM19, Lemma 12]

$$\delta(\epsilon) = \frac{1}{2}\left(\operatorname{erfc}\left(\frac{\epsilon - \mu}{2\sqrt{\mu}}\right) - e^{\epsilon}\operatorname{erfc}\left(\frac{\epsilon + \mu}{2\sqrt{\mu}}\right)\right).$$

Finally, they show that the PLD of an adaptive composition of Gaussian mechanisms is the convolution of the PLDs of the individual parts of the composition [SMM19, Theorem 1]. As convolution of distributions corresponds to summing random variables, the PLD of a convolution of Gaussian mechanisms with PLDs $\mathcal{N}(\mu_i, 2\mu_i)$ is simply $\mathcal{N}(\sum \mu_i, 2\sum \mu_i)$.

Because the sensitivity of the log-likelihood ratio is $2b_l||\theta - \theta'||_2$ and the sensitivity of the gradient is $2b_g$, both the gradient and log-likelihood ratio are divided by their sensitivities before adding noise, which normalises both to have sensitivity one. Adding noise with variance $\sigma_l^2(\theta, \theta')$ and σ_g^2 to the log-likelihood ratio and gradient respectively is equivalent to adding noise with variance $\sigma_l'^2 = \tau_l^2 n$ and $\sigma_g'^2 = \tau_g^2 n$ to the normalised log-likelihood ratio and gradient.

DP HMC releases the log-likelihood ratio k times and the gradient $kL + 1$ times. This means that the PLD of DP HMC is a Gaussian distribution $\mathcal{N}(\mu, 2\mu)$ with

$$\mu = \frac{k}{2\sigma_l'^2} + \frac{kL + 1}{2\sigma_g'^2}. \qquad \square$$

The maximum number of iterations DP HMC can run for can be computed with Algorithm 2 by using Theorem 4.6 to compute δ, and using Theorem 4.5 to initialise the variable *low*.

The τ_l and b_l parameters of DP HMC are analogous to the τ and clip bound parameters of DP penalty. Additionally, DP HMC releases gradients, which requires a separate parameter τ_g to control the trade-off between noise and privacy cost of a single iteration, and a clip bound b_g for the gradients. The number of leapfrog iterations per sample L also affects the privacy of the algorithm, as it affects the number of gradients released.

Non-DP HMC is known to be hard to tune [Nea12], as changes in η can cause large changes in the acceptance probability, and picking a too large L can cause the leapfrog simulation to turn around and propose small jumps. These issues are only amplified for DP HMC, as 4 new parameters are introduced.

The α value from Theorem 3.1 for DP penalty could be included in Theorem 4.5, which would make Theorem 4.5 the special case where $\alpha = \frac{1}{2}$. Because, as argued for DP penalty in Section 3.1, having $\alpha = \frac{1}{2}$ can be guaranteed by setting the clip bound appropriately, and the clip bound can be chosen to be anything if the parameters are only used with a single value of n, Theorem 4.5 was simplified to not include α.

Figure 4.1 shows an example of a leapfrog trajectory of DP HMC on a circular posterior (Section 5.3). Even with noisy and clipped gradients, the trajectory is able to stay inside the thin, circular area of high probability, while moving a substantial distance on the circle.

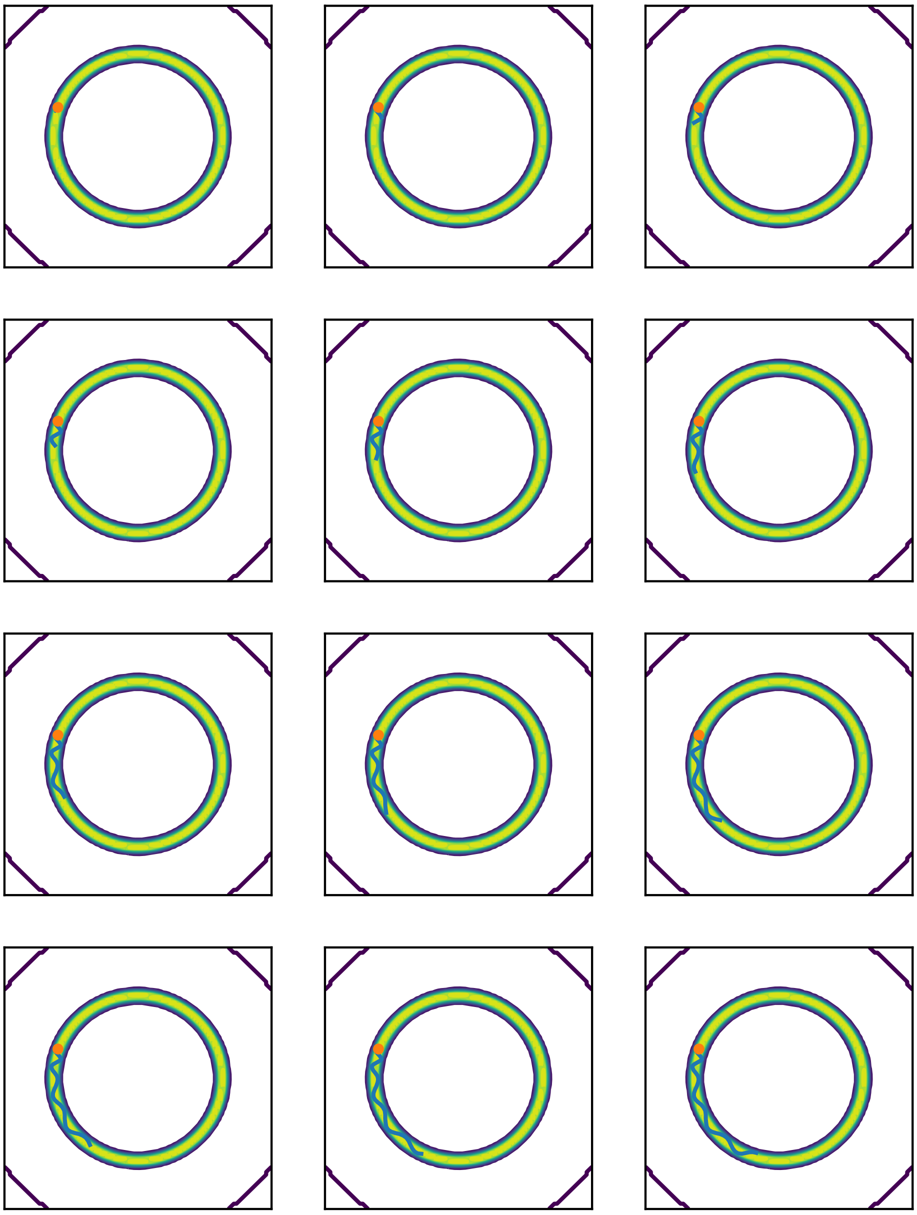

Figure 4.1: DP HMC proposal trajectory on a circular posterior (Section 5.3). The leapfrog simulation progress is shown in successive images from left to right and top to bottom. The orange point is the starting point of the leapfrog simulation and the blue line shows the progress of the leapfrog steps. The background is a contour plot of the circular posterior density.

5. Experimental Setup

To compare the performance of the DP MH algorithms, they were run on several models. This chapter introduces these models in Sections 5.1, 5.2 and 5.3. The practicalities, such as the hyperparameter values of the model and the initialisation of the algorithms, are considered in Section 5.4. Finally, the main performance metric used in the experiments is introduced in Section 5.5. The results of the experiments are discussed in Chapter 6.

5.1 The Gaussian Model

The simplest model used for the experiments is the Gaussian model with known covariance. The likelihood and prior for n points of d-dimensional data $X \in \mathbb{R}^{n\times d}$ and parameters $\theta \in \mathbb{R}^d$ are

$$\begin{aligned} \theta &\sim \mathcal{N}_d(\mu_0, \Sigma_0), \\ x_i &\sim \mathcal{N}_d(\theta, \Sigma), \end{aligned}$$

where x_i is a row of X, Σ is the known covariance, and μ_0 and Σ_0 are the prior hyperparameters. The posterior of this model is another d-dimensional Gaussian distribution with parameters expressible in closed form [GCS+14, Section 3.5], so it is easy to sample from the posterior.

5.2 The Banana Distribution

The banana distribution [TPK14] is a banana-shaped probability distribution that is a challenging target for MCMC algorithms. For this reason it has been used to test MCMC algorithms in the literature [TPK14].

Definition 5.1. *Let x have a d-variate Gaussian distribution with mean μ and covariance matrix Σ. Let*

$$g(x) = (x_1, x_2 - a(x_1 - m)^2 - b, x_3, \dots, x_d),$$

with $a, b, m \in \mathbb{R}$. The banana distribution with parameters μ, Σ, a, b and m is the distribution of $g(x)$. It is denoted by $\mathrm{Ban}(\mu, \Sigma, a, b, m)$.

In the literature, the banana distribution is simply used as the target to sample from, and is not the posterior in a Bayesian inference problem [TPK14]. To test differentially private MCMC algorithms, the target distribution must be the posterior of some inference problem, as otherwise there is no data to protect with differential privacy. Theorem 5.2 gives a suitable inference problem for testing DP MCMC algorithms.

Theorem 5.2. *Let*

$$\begin{aligned}
\theta = (\theta_1, \dots, \theta_d) &\sim \mathrm{Ban}(0, \sigma_0^2 I, a, b, m), \\
x_1 &\sim \mathcal{N}(\theta_1, \sigma_1^2), \\
x_2 &\sim \mathcal{N}(\theta_2 + a(\theta_1 - m)^2 + b, \sigma_2^2), \\
x_3 &\sim \mathcal{N}(\theta_3, \sigma_3^2), \\
&\vdots \\
x_d &\sim \mathcal{N}(\theta_d, \sigma_d^2).
\end{aligned}$$

Given data $x_1, \dots, x_n \in \mathbb{R}^d$ and denoting $\tau_i = \frac{1}{\sigma_i^2}$, the posterior of θ tempered with T is the banana distribution $\mathrm{Ban}(\mu, \Sigma, a, b, m)$ with

$$\bar{x}_i = \frac{1}{n}\sum_{j=1}^{n} x_{ji} \quad i \in \{1, 2\}.$$

$$\begin{aligned}
\mu &= \left(\frac{Tn\tau_1\bar{x}_1}{Tn\tau_1 + \tau_0}, \dots, \frac{Tn\tau_d\bar{x}_d}{Tn\tau_d + \tau_0}\right), \\
\Sigma &= \mathrm{diag}\left(\frac{1}{Tn\tau_1 + \tau_0}, \dots, \frac{1}{Tn\tau_d + \tau_0}\right).
\end{aligned}$$

Proof. See Appendix A. □

The Gaussian distribution is a special case of the banana distribution, as setting $a = 0$ makes g the identity function. Similarly, setting $a = 0$ turns the Bayesian banana model into a Gaussian model.

5.3 Circle Model

Random walk MH algorithms may struggle with posteriors which are concentrated on long, but thin regions. Having a large proposal variance causes the algorithm to frequently jump out of the region of high probability, but lowering the variance to stay in the region causes the chain to take a very long time to move around in the region. An example of such a posterior is the circular posterior described in this section.

The circle posterior is obtained from a model where the log-likelihood is

$$\ln p(r \mid x, y) = -a(x^2 + y^2 - r^2)^2$$

where $r \in \mathbb{R}$ is an observed data point, $a > 0$ is a hyperparameter, and $(x, y) \in \mathbb{R}^2$ are the parameters. The likelihood is circular in the (x, y)-plane, as it only depends on the squared distance of the point (x, y) from the origin. By choosing a flat prior, $p(x, y) = 1$ for all $(x, y) \in \mathbb{R}^2$, the posterior will be proportional to the likelihood, meaning that the posterior will also be circular.

As the prior does not integrate to 1, it is not a proper probability density, so Bayes' theorem does not guarantee that the posterior integrates to a finite value. In this case, for a single data point r and any $a > 0$,

$$\begin{aligned}
\int_{\mathbb{R}^2} p(r \mid x, y)\mathrm{d}x\mathrm{d}y &= \int_{\mathbb{R}^2} e^{-a(x^2+y^2-r^2)^2}\mathrm{d}x\mathrm{d}y \\
&= \int_0^{2\pi}\int_0^{\infty} te^{-a(t^2(\cos^2\phi+\sin^2\phi)-r^2)^2}\mathrm{d}t\mathrm{d}\phi \\
&= \int_0^{2\pi}\int_0^{\infty} te^{-a(t^2-r^2)^2}\mathrm{d}t\mathrm{d}\phi \\
&= \int_0^{2\pi}\int_0^{\infty} e^{-a(u-r^2)^2}\mathrm{d}u\mathrm{d}\phi \\
&< \infty,
\end{aligned}$$

as, on the second to last line, the inner integral is over an unnormalised Gaussian density, and the outer integral is over a finite interval. For multiple data points, the integral of the likelihood is

$$\int_{\mathbb{R}^2}\prod_i^n p(r_i \mid x, y)\mathrm{d}x\mathrm{d}y \leq \prod_i^n \left(\int_{\mathbb{R}^2} p(r_i \mid x, y)^n \mathrm{d}x\mathrm{d}y\right)^{\frac{1}{n}} < \infty,$$

where the first inequality is Hölder's inequality applied $n - 1$ times, and the second follows from the single data point case, as $p(r \mid x, y)^n = e^{-an(x^2+y^2-r^2)^2}$. As the likelihood integrates to a finite value, it can be considered an unnormalised density and sampled from with MCMC.

Obtaining samples from the true posterior with the circle model is not as easy as with the banana and Gaussian models, so using MMD, the metric introduced in Section 5.5 to evaluate the performance of an MCMC algorithm on the circle is nontrivial. However, as the mean of the circle is in its center, and the samples from an MCMC algorithm are likely to lie on the circle, the distance of the mean of the sample from the origin indicates how evenly the samples a distributed throughout the circle.

It remains to show that the mean of the circle is actually the origin. It turns out that this is fairly simple. For the mean of the x-coordinate

$$\begin{aligned} E_x &= \int_{\mathbb{R}} x \int_{\mathbb{R}} p(r \mid x, y) \mathrm{d}x\mathrm{d}y \\ &= \int_{-\infty}^{\infty} x \int_{-\infty}^{\infty} \prod_i e^{-a(x^2+y^2-r_i^2)^2} \mathrm{d}x\mathrm{d}y \\ &= \int_{0}^{\infty} x \int_{-\infty}^{\infty} \prod_i e^{-a(x^2+y^2-r_i^2)^2} \mathrm{d}x\mathrm{d}y + \int_{-\infty}^{0} x \int_{-\infty}^{\infty} \prod_i e^{-a(x^2+y^2-r_i^2)^2} \mathrm{d}x\mathrm{d}y \\ &= \int_{0}^{\infty} x \int_{-\infty}^{\infty} \prod_i e^{-a(x^2+y^2-r_i^2)^2} \mathrm{d}x\mathrm{d}y - \int_{0}^{\infty} x \int_{-\infty}^{\infty} \prod_i e^{-a(x^2+y^2-r_i^2)^2} \mathrm{d}x\mathrm{d}y \\ &= 0. \end{aligned}$$

The mean of the y-coordinate is calculated similarly.

Unlike the banana and Gaussian models, the circle model does not give a method to generate the data points r_i from given values of x and y. The r_i values used in the experiments of Chapter 6 were sampled from a normal distribution with mean 3 and variance 1.

5.4 Practicalities of Running the Comparison

Eight sets of model hyperparameters were used in the experiments. The varied hyperparameters are shown in Table 5.1 and contour plots of the 2-dimensional models are shown in Figure 5.1. All banana models had $b = m = 0$. The likelihood variance of the banana and 30-dimensional Gaussian models was $\sigma_1^2 = 20$, $\sigma_2^2 = 2.5$ and $\sigma_i^2 = 1$ for $i > 2$. The likelihood covariance for the correlated Gaussian model was

$$\begin{bmatrix} 1 & 0.999 \\ 0.999 & 1 \end{bmatrix}.$$

The true values of θ in all experiment except the circle are $\theta_2 = 3$ and $\theta_i = 0$ for $i \neq 2$.

All experiments ran 20 chains for each value of ϵ for DP MH experiments, or clip bound for clipping experiments, for each algorithm included. Each of the 20 chains had a different starting point, but the starting points did not vary across algorithms, or values of ϵ or clip bound. For the banana and Gaussian experiments, the starting points were chosen from a Gaussian distribution centered on the true parameter values. The standard deviation of the Gaussian was the mean of the component-wise standard deviations of the a sample of the true posterior. In the circle experiment, the starting points were chosen from a Gaussian distribution centered on $(0, 1)$ with variance 1. Figure 5.1 shows the starting points for each model as orange points.

The first half of the obtained samples were discarded as they may not represent the true posterior, which is standard practice with MCMC [GCS+14]. The latter half was compared to a reference sample obtained from the true posterior, except in the circle experiment, where the mean of the sample was compared to the true posterior mean of $(0, 0)$.

Publicly available implementations of DP Barker[*] and the Fourier accountant[†] were used, and the rest of the algorithms were implemented by the author. The accuracy and running time of the Fourier acountant is determined by two parameters [KJH20]. They were left at their default values. The code for running the experiments is freely available[‡].

[*] https://github.com/DPBayes/DP-MCMC-NeurIPS2019

[†] https://github.com/DPBayes/PLD-Accountant

[‡] https://github.com/oraisa/masters-thesis

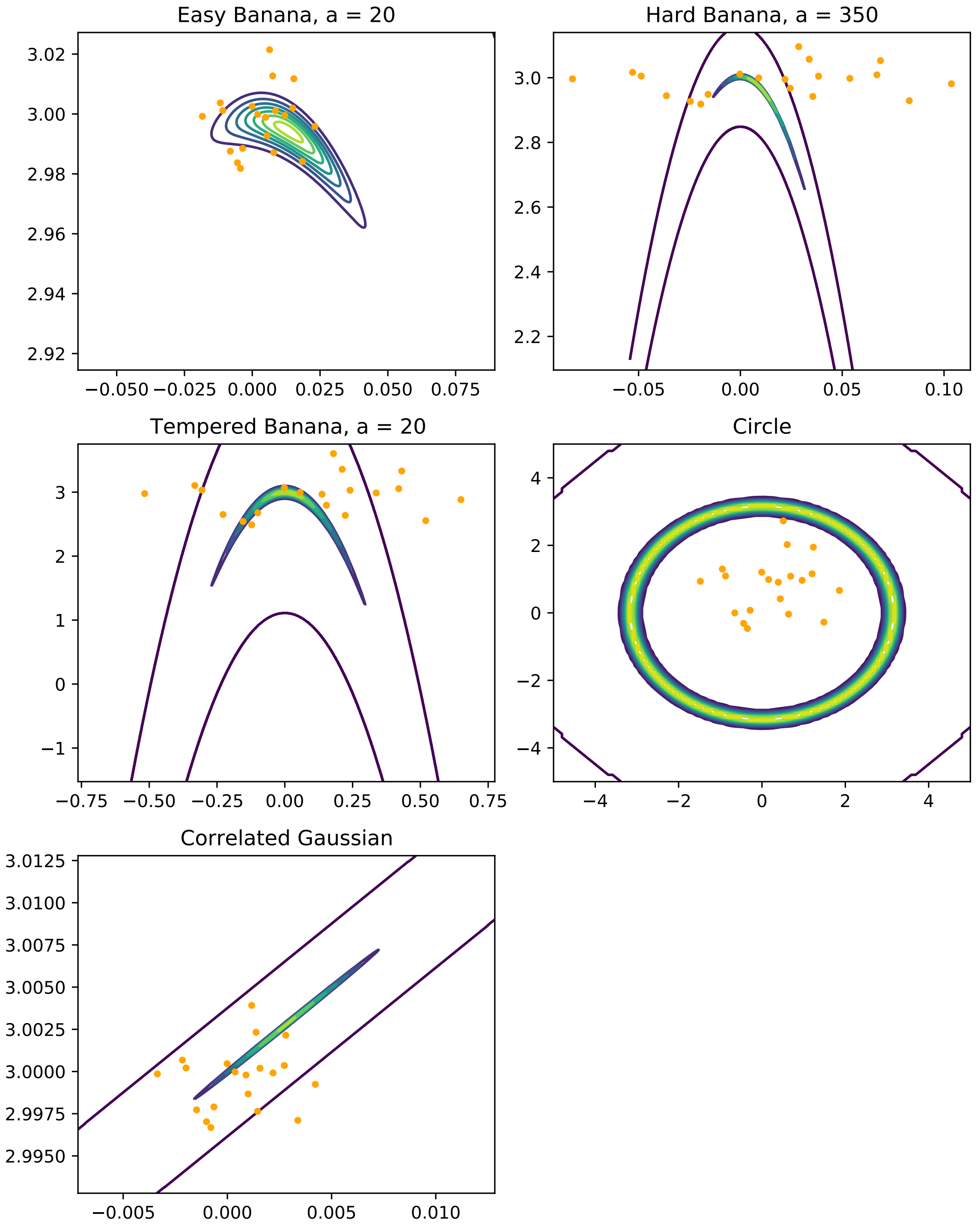

Figure 5.1: Contour plots of the posterior densities of the 2-dimensional models. The orange points are the 20 starting points for each chain.

Table 5.1: Model hyperparameters. n_0 determines tempering by $T = \frac{n_0}{n}$. For missing n_0, $T = 1$.

Name	d	n	n_0	a	σ_0^2
Flat banana, d = 2	2	100000		20	1000
Flat banana, d = 10	10	200000		20	1000
Tempered banana, d = 2	2	100000	1000	20	1000
Tempered banana, d = 10	10	200000	1000	20	1000
High dimensional Gauss	30	200000		0	1000
Narrow banana	2	150000		350	1000
Correlated Gauss	2	200000		0	100
Circle	2	100000		1e-05	

5.5 Maximum Mean Discrepancy

The convergence of non-DP MCMC algorithms is typically assessed using $\hat{R}$ [GCS+14], which measures how well multiple chains started from different points have mixed together. The utility of a sample produced by an MCMC algorithm can be evaluated using effective sample size (ESS) [GCS+14], which is an estimate of the size of an uncorrelated sample of the posterior with the same estimation utility as the MCMC sample.

Both $\hat{R}$ and ESS require that the MCMC algorithm asymptotically targets the true posterior, as they cannot detect an algorithm that has converged to the wrong distribution. Because some of the DP MCMC algorithms use approximations that may cause the algorithms to not converge to the true posterior, such as clipping the log-likelihood ratios, $\hat{R}$ and ESS are not suitable for assessing the performance of the algorithms.

Because the DP MCMC algorithms may not converge to the correct distribution, their performance should be evaluated with a metric that measures how close to the true distribution they are. A very general such metric is maximum mean discrepancy (MMD) [GBR+12]. MMD between distributions p and q is defined as

$$\mathrm{MMD}(p, q) = \sup_{f \in \mathcal{F}} (E_{x \sim p} f(x) - E_{y \sim q} f(y))$$

where $\mathcal{F}$ is some class of functions. By choosing a suitable $\mathcal{F}$, $\mathrm{MMD}(p, q)$ can be estimated from a sample from p and q. The suitable classes $\mathcal{F}$ can be characterised by kernel functions $k\colon P \times Q \to \mathbb{R}$, where P and Q are the supports of p and q, respectively. The Gaussian radial basis function (RBF) kernel

$$k(x, y) = \exp\left(\frac{||x - y||_2^2}{2\sigma^2}\right)$$

is particularly well suited, as it has the property that $\mathrm{MMD}(p, q) = 0$ if and only if $p = q$. After choosing a kernel, $\mathrm{MMD}(p, q)$ may be estimated from finite samples of p and q. To evaluate MCMC algorithms, one of the samples is the output of the algorithm to be evaluated, and the other is a sample from the true posterior.

The choice of the σ parameter of k affects the way MMD evaluates different kinds of differences in p and q. For some preliminary experiments in this thesis, σ was chosen to be 1, which penalised error in the mean much more than errors in higher moments in the experiments. The σ used for the final experiments is chosen by picking 50 subsamples from both samples with replacement and setting σ to be the median between distances of points of the subsamples. This is following the procedure of Gretton et al. [GBR+12], with the addition of the subsampling step to handle samples of different sizes.

6. Experiments

This chapter contains the experiments performed on both non-DP and DP MH algorithms. Section 6.1 compares the zCDP, RDP and ADP based privacy accounting methods that are available for DP penalty, minibatch DP Penalty and DP HMC. Section 6.2 examines the effect clipping log-likelihood ratios, which is necessary for DP MH algorithms, but may adversely affect their convergence. Section 6.3 compares the different DP MH algorithms on several models.

6.1 Comparing Privacy Accounting Methods

DP penalty, minibatch DP penalty and DP HMC have two different methods to compute the number of iterations the algorithm can run for, given a privacy bound and parameters for the algorithm. These privacy accounting methods are given by Theorems 3.1 and 3.2 for DP penalty, Theorems 4.5 and 4.6 for DP HMC, and Theorem 3.5 and the Fourier accountant [KJH20] for minibatch DP penalty.

Figure 6.1 compares the number of iterations each of the algorithms can run for for both accounting methods and different values of ϵ in the 2-dimensional flat banana experiment. The ADP based methods of Theorems 3.2 and 4.6, as well as the Fourier accountant, significantly outperform the other accounting methods. This makes it clear that the tight bounds given by the ADP based methods should be used in favor of loose methods. The Fourier accountant is not easily applicable to DP Barker, as DP Barker does not release the sample variance directly, but it may still be possible to use the Fourier accountant with DP Barker. This is a potential question for future research, as using a better privacy accountant may significantly improve the performance of DP Barker.

6.2 The Effects of Clipping

The this section looks at the effect of clipping log-likelihood ratios, which is necessary for DP MH algorithms, but may affect their convergence to the correct posterior. Both

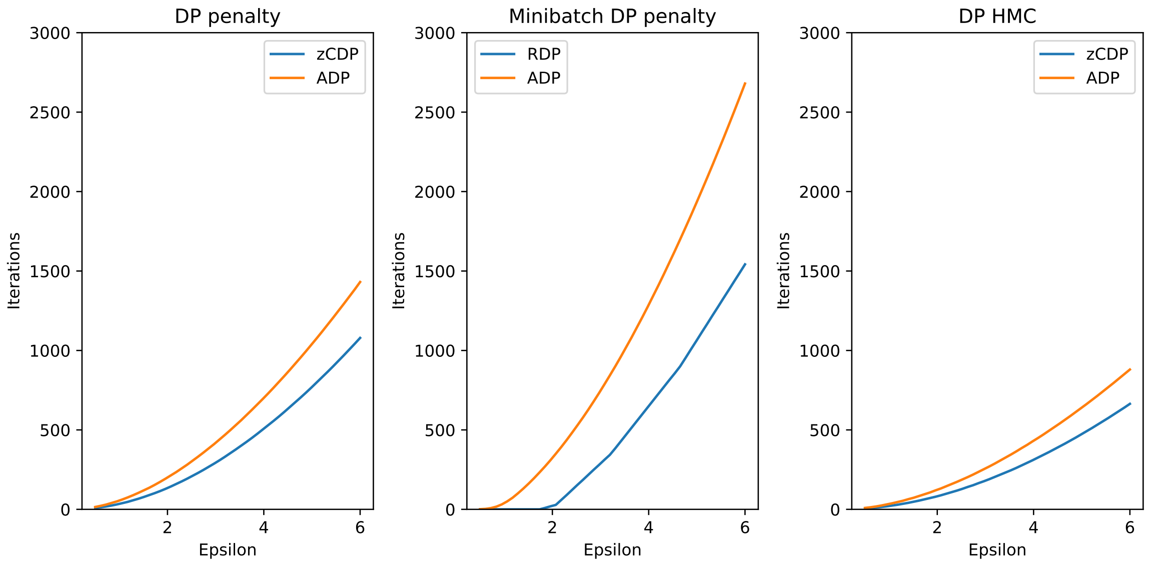

Figure 6.1: Comparing the zCDP or RDP and ADP based privacy accounting methods for the algorithms with multiple accounting methods. The left panel compares zCDP based accounting (Theorem 3.1) and ADP based accounting (Theorem 3.2) for the DP penalty algorithm. The middle panel compares the RDP accounting (Theorem 3.5) and the Fourier accountant. The right panel compares the zCDP (Theorem 4.5) and ADP (Theorem 4.6) based accounting. The ADP based methods significantly outperform the other methods in all cases.

HMC and random walk Metropolis-Hastings (RWMH)* algorithms were run on the 2 and 10 dimensional flat banana models. DP was not used so that error from the extra noise would not affect the results.

For both 2 and 10 dimensions, 500 samples from HMC and 3000 samples from RWMH were taken and the latter half of them were compared to 1000 samples from the true posterior. The reference posterior sample was also compared to other samples from the posterior to obtain a baseline. The sample sizes and other parameters of the algorithms were tuned so that the algorithms converged without clipping.

Figure 6.2 shows the results of the clipping experiment. The top left and bottom left panels show MMD as a function of the clip bound and the fraction of log-likelihoods that was actually clipped for both HMC and RWMH in the 2-dimensional model. The effect of clipping on MMD is nonexistent for all but the lowest clip bounds. The top and bottom right panels show results for the 10-dimensional model. This time there are chains that did not converge correctly with most clip bounds, but the chains with the higher bounds converged. Based on these results, if the clip fraction is less than 20%, clipping is likely undetectable without a large sample. For the other experiments, clip bounds were tuned to achieve less than 10% clipping, to ensure that the error from clipping is minimal. As this experiment only used a single model in fairly low

*Metropolis-Hastings using the Gaussian distribution as the proposal distributions.

dimensions, it is possible that in other settings the clip fraction should be even lower, so future research should look into clipping in other models.

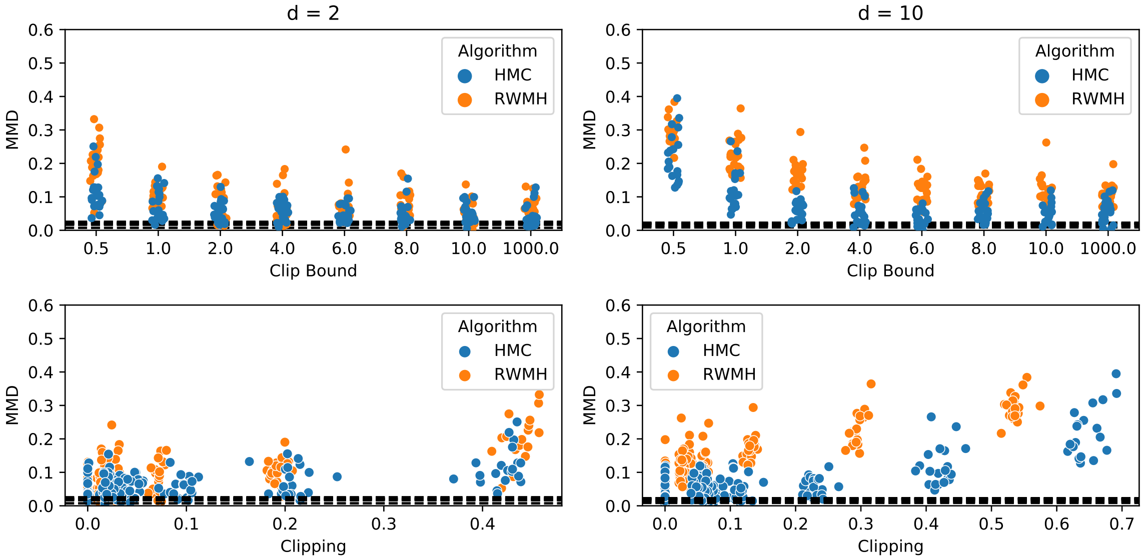

Figure 6.2: The effect of log-likelihood ratio clipping on the posterior of the banana model for random walk Metropolis-Hastings and HMC. The top row shows posterior MMD as a function of the clip bound, and the bottom row shows MMD as a function of the fraction of log-likelihoods that were clipped. The left columns used a 2-dimensional posterior while the right columns had a 10-dimensional posterior. The black lines show the MMDs of ten different samples of the true posterior compared to the reference sample. All of the lines are close the each other and appear as a single line.

6.3 Comparison of DP MH Algorithms

The experiments in this section compare the DP MH algorithms discussed in this thesis. The compared algorithms are the DP penalty algorithm with and without GWMH and OCU [YE19] (Section 3.1), the DP Barker algorithm [HJDH19][*] (Section 3.3), the minibatch DP penalty algorithm, again with and without GWMH and OCU (Section 3.4), and DP HMC (Section 4.2).

The δ for all experiments is $\frac{0.1}{n}$, and ϵ is varied. All algorithms used the best privacy accounting methods discussed in this thesis. For both variants of DP penalty and DP HMC, these are the PLD based Theorems 3.2 and 4.6. DP penalty with subsampling uses the Fourier accountant [KJH20] and DP Barker uses Theorem 3.3. The RDP based theorems for the minibatch algorithms are not tight for the ADP bounds that were used, so their results could be improved by using an accounting method for ADP.

[*] While DP Barker is not technically an MH algorithm, is still uses the Barker acceptance test, and is very close to DP MH algorithms.

The hyperparameters of the algorithms were tuned by running the algorithms with $\epsilon = 4$ and examining the results, trying to find hyperparameters that minimise MMD. Clip bounds were tuned so that less than 10% of the log-likelihood ratios were clipped, as the results of Section 6.2 show minimal effect on MMD at that point.

Figure 6.3 shows the MMD for each algorithm as a function of ϵ for the flat and tempered banana models with 2 and 10 dimensions. In the non-tempered models, the minibatch algorithms are not very useful, with the exception of DP Barker in 10 dimensions. Of the non-minibatch algorithms, with tempering HMC beats the penalty algorithms, but without tempering this is reversed. The black lines at the bottom show MMDs of 10 different samples of the true posterior compared to the reference sample. In 2 dimensions, the best algorithms get close to the MMDs achieved by the non-DP algorithms from the clipping experiment of Section 6.2.

Figure 6.4 shows MMDs for the more complicated models, the 30-dimensional Gaussian, the narrow banana and the highly correlated Gaussian. The minibatch algorithms were not included as the previous experiment indicates that they perform poorly without tempering. In the 30-dimensional Gaussian, DP penalty with OCU and GWMH beats the other two algorithms. In the narrow banana all algorithms are approximately equal. In the highly correlated Gaussian experiment, the penalty algorithms perform equally, and HMC beats both of them. In the 30-dimensional Gaussian, all algorithms have clearly decreasing MMD with increasing ϵ, but in the narrow banana and correlated Gaussian, only HMC on the correlated Gaussian is able to achieve significantly lower MMD with higher ϵ.

Figure 6.5 shows the fraction of log-likelihood ratios that were clipped for each ϵ and algorithm. Almost all runs had less than 10% clipping, with the exception of DP Barker, as adjusting its clip bound is not possible, and the 2-dimensional tempered experiment where some clip fractions are getting closer to 20%.

Figure 6.6 shows clipping for the harder models. Again, almost all runs have less than 10% clipping, with the exception of some runs with on the narrow banana, especially with $\epsilon = 1$.

Figure 6.7 shows the results for the circle model. The distance of the sample mean from the true mean (the origin) was used instead of MMD, as computing MMD requires a sample from the posterior, which is not trivial to obtain for the circle model. Both algorithms perform well on average. Clipping is again low for both algorithms, and HMC has a fairly large variance in clipping between different runs.

Figure 6.8 shows the fraction of clipped gradients for DP HMC. Clipping gradients does not affect the asymptotic convergence of DP HMC, but it may lower the acceptance rate and thus the performance of the algorithm. As raising the clip bound to lower gradient clipping increases the noise added to gradient, thus also lowering the

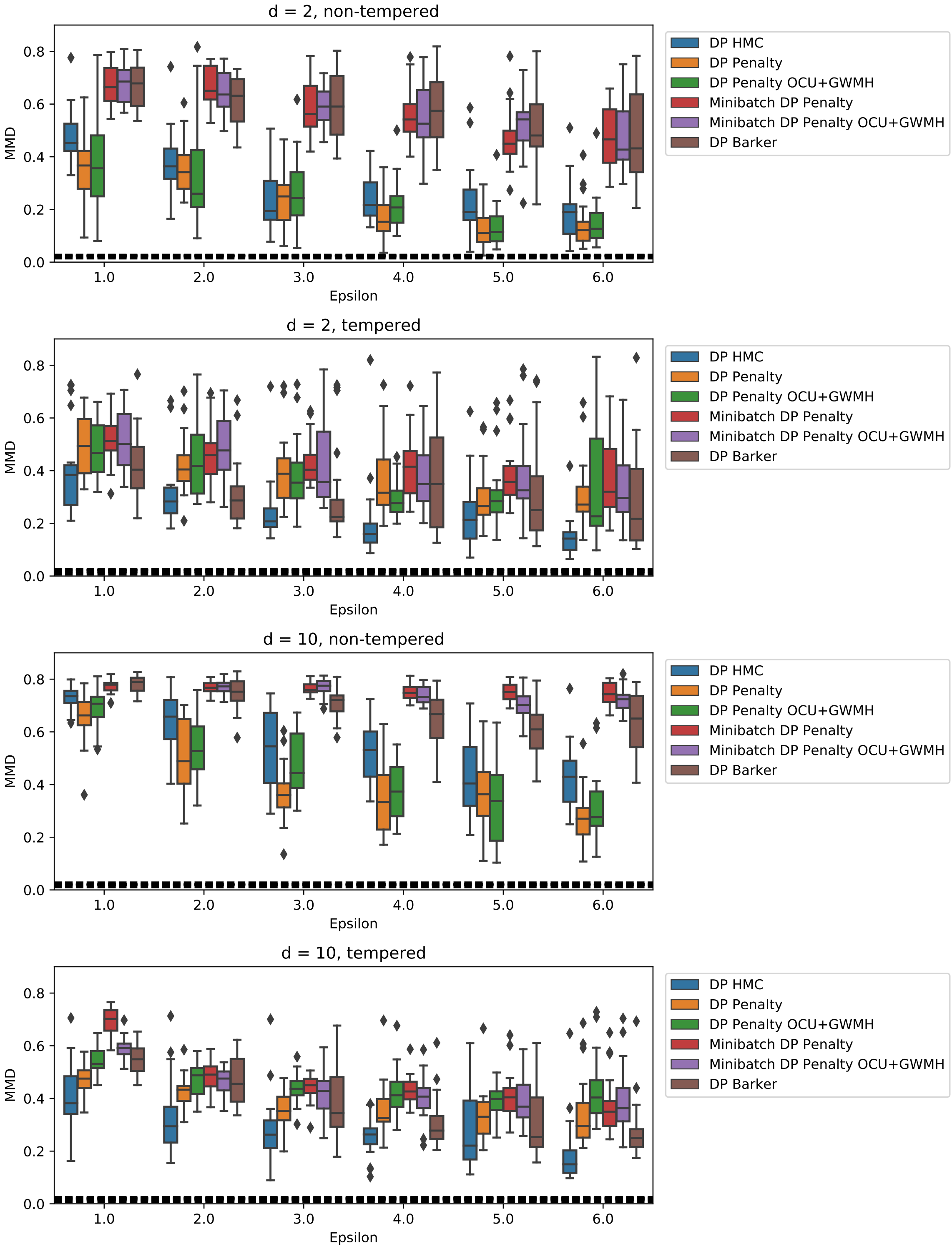

Figure 6.3: MMD as a function of ϵ for the different MCMC algorithms, on flat and tempered banana models with both a low number of dimensions (2) and a high number of dimensions (10). The black lines show the MMD of 10 different samples of the true posterior compared to the reference sample.

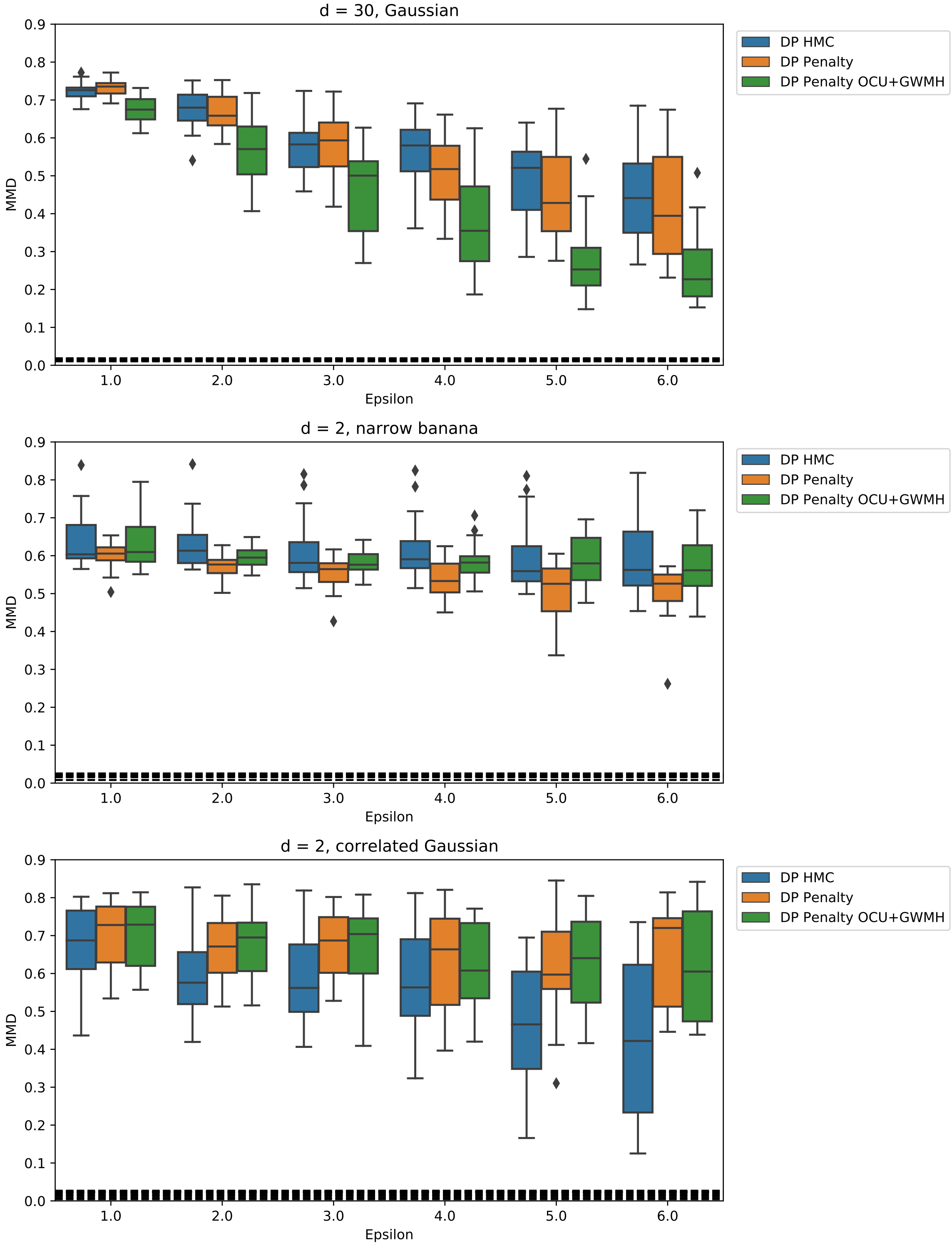

Figure 6.4: MMD for the 30-dimensional Gaussian, hard banana and highly correlated 2-dimensional Gaussian. The black lines show the MMD of 10 samples of the true posterior compared to the reference sample.

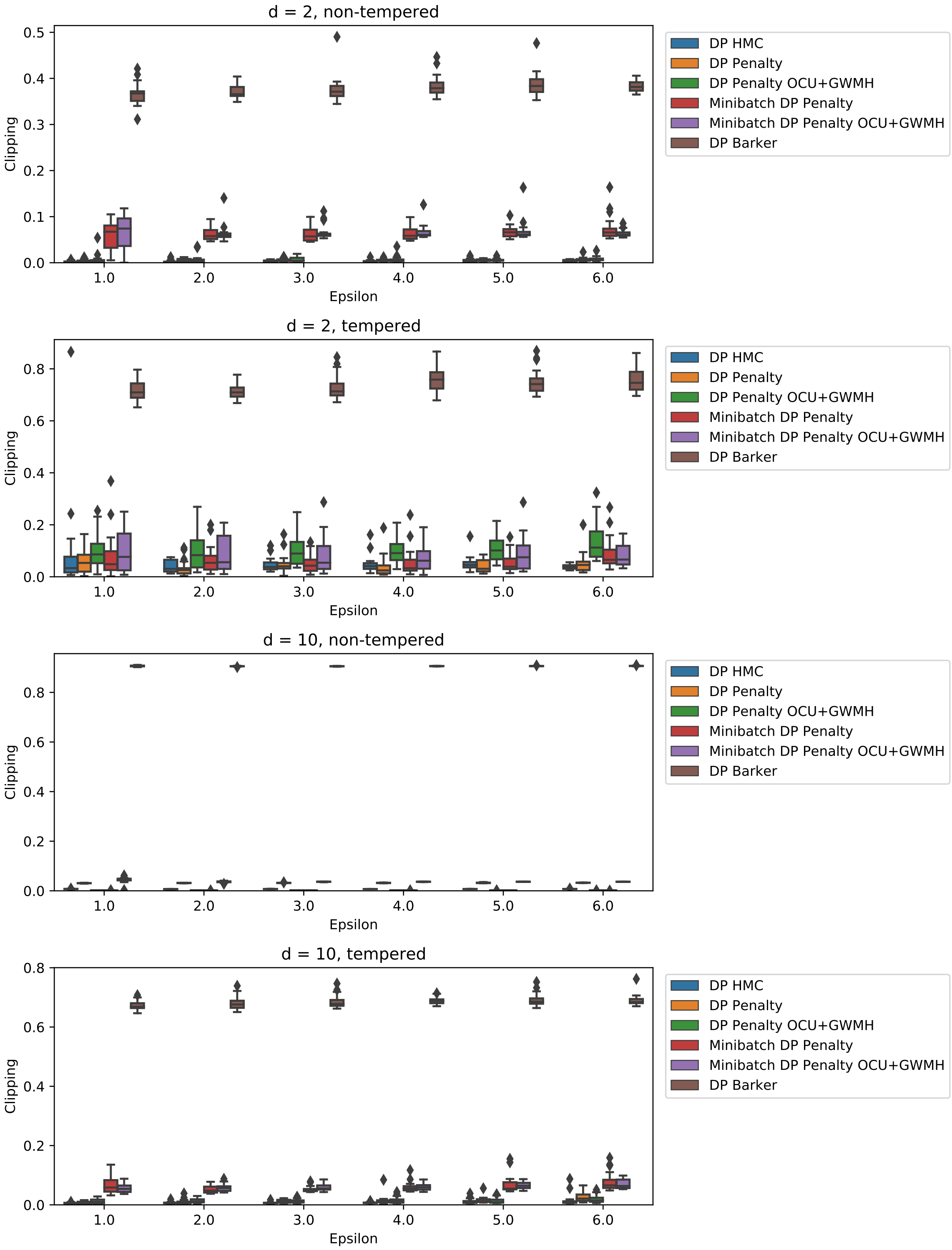

Figure 6.5: Clipping for the easy and tempered banana experiments.

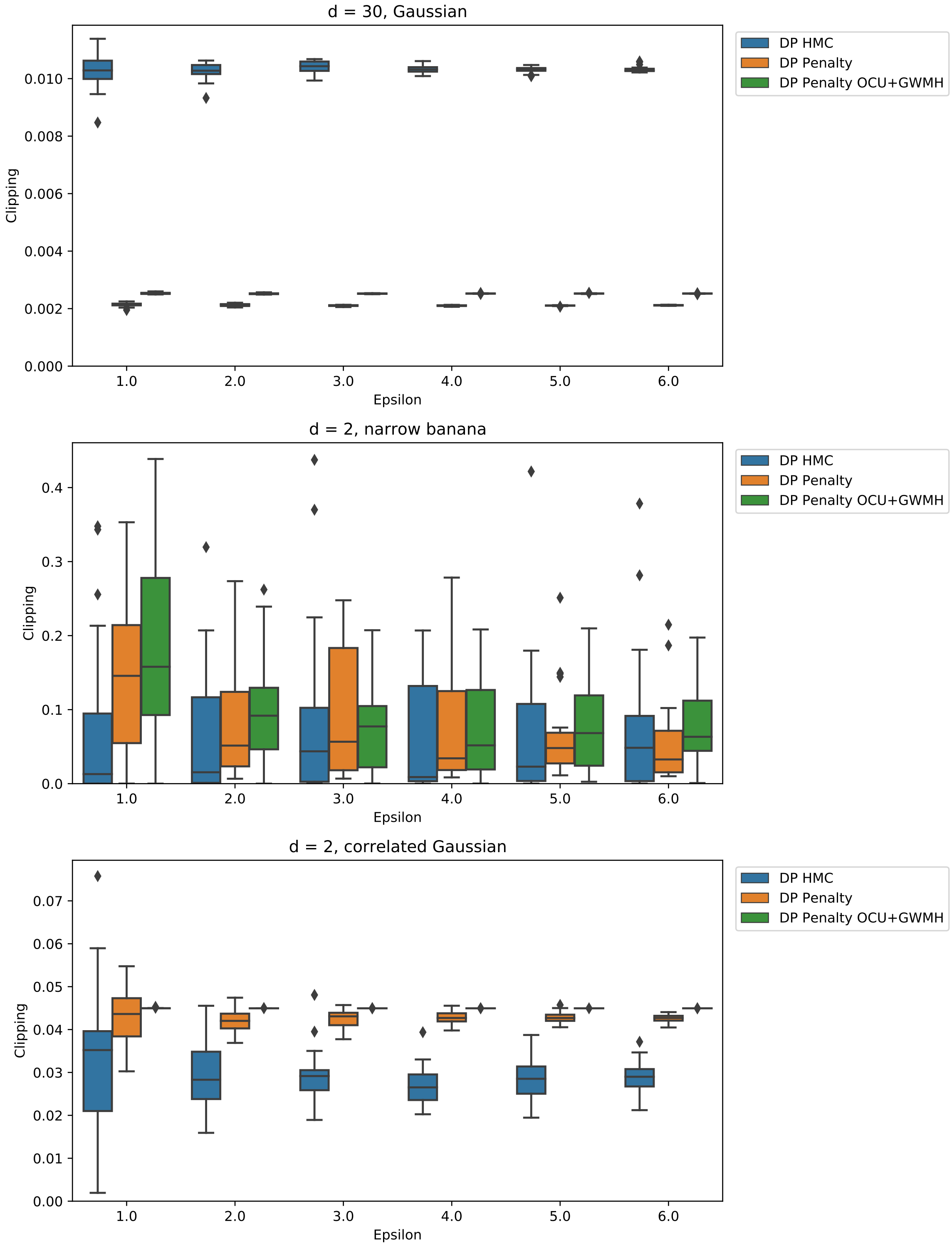

Figure 6.6: Clipping for 30-dimensional Gaussian, hard banana and highly correlated 2-dimensional Gaussian.

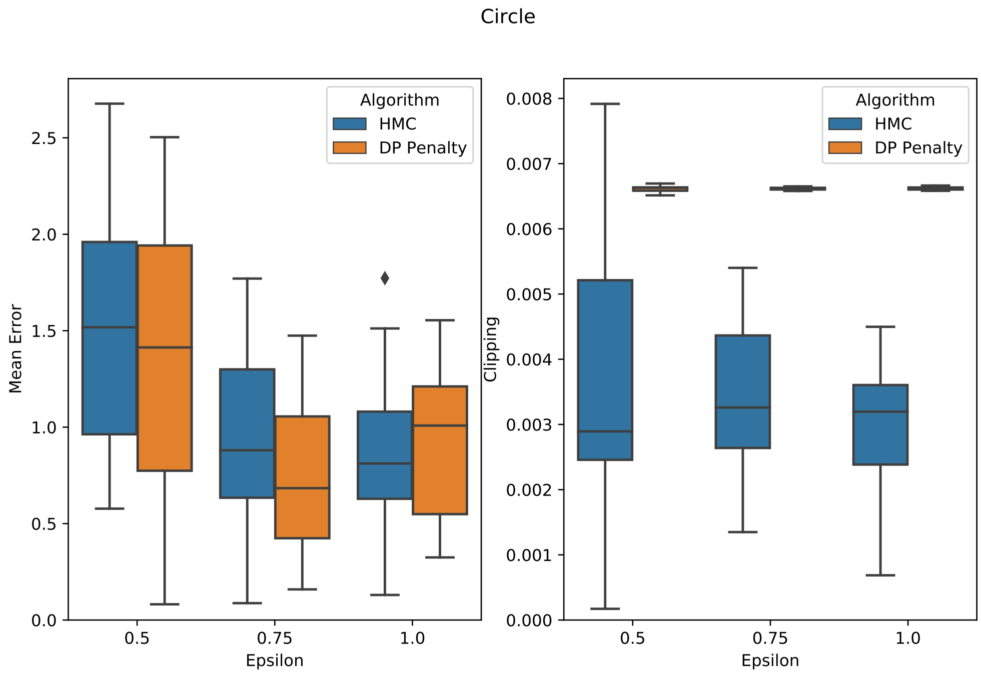

Figure 6.7: Circle mean error on the left and clipping on the right.

acceptance rate, it is not clear what an appropriate level of gradient clipping would be. The observed levels of clipping are fairly constant across values of ϵ, but vary in the models, and in the narrow banana model, there is very large variance in clipping.

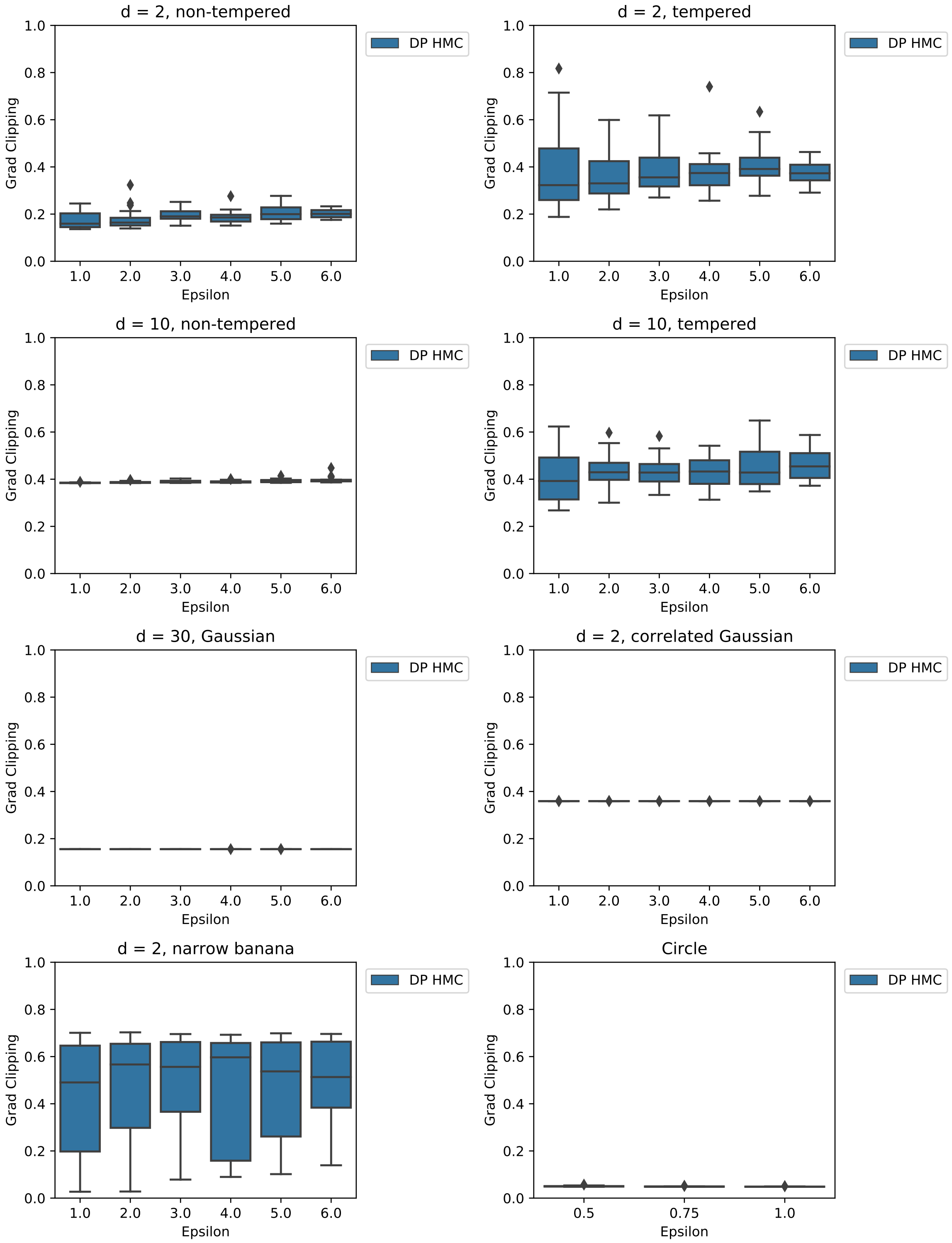

Figure 6.8: Gradient clipping for DP HMC.

7. Conclusions

The results of the experiments of Section 6.3 show that using DP MH for Bayesian inference is possible, but will require large datasets, especially with complex models. With the already fairly large datasets with $10^5 \leq n \leq 2 \cdot 10^5$, the DP MH algorithms were able to achieve significantly lower MMDs with increasing ϵ on most models, but the algorithms only got close to the baseline MMDs from samples of the true posterior on the simplest models, the flat and tempered bananas. This also required a large value of $\epsilon = 6$, while performance with the more reasonable values of ϵ like 1 and 2 was much worse. The circle model is the only exception, where performance with $\epsilon = 1$ was reasonable. The experiments done by Heikkilä et al. [HJDH19] and Yildirim and Ermis [YE19] have similar results: good performance of a DP MH algorithm with a reasonable $\epsilon \leq 2$ requires a large dataset with $n \geq 10^5$.

When comparing the different DP MH algorithms, the results of Section 6.3 show that neither DP HMC or DP penalty is clearly better than the other, as both algorithms beat the other on some models. The usefulness of OCU+GWHM for DP penalty is also debatable, as DP penalty OCU+GWMH only clearly beat DP penalty on the 30-dimensional Gaussian. The only clear conclusion from the comparisons is that the minibatch algorithms are only useful with tempered posteriors, and even there they are not better than the algorithms using the full data, so subsampling the log-likelihood ratios does not seem to be a useful addition to DP MH, which is surprising, as subsampling significantly lowers the privacy cost of releasing the log-likelihood ratio.

The failure of subsampling log-likelihood ratios does not imply that subsampling HMC gradients will not be useful. Gradient subsampling has been done with non-DP HMC [CFG14, DFB$^+$14] to reduce computation time, and using subsampling in the DP setting is a potential direction for future research. However, naive gradient subsampling in HMC has been shown to fail in high dimensions [Bet15], so additional methods to control the error from subsampling are likely needed, such as the friction term [CFG14] briefly discussed in Section 4.2, or the Nosé-Hoover thermostat [DFB$^+$14] that dynamically adjusts the friction.

The large number of parameters DP MH algorithms have compared to non-DP MH makes tuning them particularly difficult. DP HMC is the clearest example of this, where 4 parameters tune the different trade-offs between privacy and accuracy, in addition to the parameters of HMC, which are already known to be difficult to tune in the non-DP case [Nea12].

Several methods and guidelines for tuning the parameters of non-DP MH algorithms have been developed. The simplest of these are heuristic guidelines or in some cases proofs [RGG+97] of optimal acceptance rates. For MH with a Gaussian proposal and a suitable target distribution, it has been shown that the optimal acceptance rate approaches approximately 23% as the dimensionality of the target approaches infinity [RGG+97]. For HMC, a 65% acceptance rate is recommended [Nea12]. These guidelines can be used to tune the proposal variance, or, in the case of HMC, η, until the algorithm has the desired acceptance rate. They can be used when tuning DP MH algorithms, and were used to some extent in tuning the parameters for the experiments of this thesis, but it is not clear how well these guidelines apply under DP. Yildirim and Ermis show that DP penalty has a lower acceptance rate than the non-DP MH algorithm with the same proposal [YE19], so the optimal acceptance rates of DP MH algorithms are likely lower their non-DP counterparts, but the exact optimal acceptance rates will likely depend on the privacy bounds.

Tuning the parameters of an MH algorithm by hand is time consuming, even with a guideline on the optimal acceptance rate, and requires the user of the algorithm to know the guidelines and their applicability. Tuning the parameters automatically alleviates these problems. In the context of HMC, these automatically tuned algorithms are called dynamic HMC algorithms. The most prominent dynamic HMC algorithms is the No-U-turn sampler (NUTS) [HG14], that is able to tune both η and L automatically. The automatic tuning of both parameters made HMC usable in automatic inference libraries such as Stan [Tea20] that are typically used by applied practitioners who are not experts in MCMC algorithms. Extending NUTS and other dynamic HMC algorithms that can tune some parameters automatically to the DP case is a potential avenue of future research. The empirical results on clipping in Section 6.2 may be usable as guidelines for tuning the clip bounds, perhaps even automatically.

A fundamental difficulty in tuning DP MH algorithms is the fact that the metrics used to tune the algorithm, such as acceptance and clipping rates, cannot be released without incurring a privacy cost and adding noise to the metrics. Even tuning the algorithm internally based on the metrics, and releasing only the final parameters incurs some privacy cost, which can be argued to be negligible, but cannot be computed, as formalising the way a human chooses parameters based on the metrics is impossible. This can be sidestepped by tuning the algorithm on synthetic or publicly available data, and only using the private data with the final parameters, but this risks overfitting the parameters to the tuning dataset and having poor performance on the private dataset.

This thesis largely ignores these issues for the sake of comparing the algorithms, but potential users of DP MH on real private data should not ignore them, which makes developing automatic methods for tuning parameters that allow computing the privacy cost of tuning even more important than in the non-DP case.

Other practical issues that the experiments of Section 6.3 ignore are the initialisation of the algorithms, and the fact that computer implementations cannot sample from continuous distributions exactly. In existing work, DP MH algorithms have been initialised using a point estimate obtained from another DP method with very low privacy cost [HJDH19, WFS15]. The initialisation method used Section 6.3 mimics the effects of using a point estimate by choosing random values around the true parameter values. This is unlikely to affect the comparisons of the DP MH algorithms, as all of the algorithms would use the same method to obtain the point estimate in any case.

The issue of sampling continuous distributions on computers for DP were first identified for the Laplace distribution [Mir12]. When sampling the Laplace distribution with finite precision floating point numbers, the bits of the results can contain information about the private data that destroys DP, even when the analysis is theoretically DP with exact sampling of the Laplace distribution. These issues likely exist when sampling the Gaussian distribution, which means that actual implementations of the algorithms in this thesis may not have their theoretical DP guarantees. Recently, the differential privacy of a discrete variant of the Gaussian distribution has been analysed [CKS20]. Using a discrete distribution instead of a continuous one sidesteps floating point issues, but in the context of DP MH, the penalty correction requires adding continuous Gaussian noise. Investigating the effect of using discrete Gaussian noise instead with the penalty algorithm is another potential avenue of future research.

The results presented in this thesis show that DP MH is a viable way to conduct DP Bayesian inference with large enough datasets. There are still issues that should be investigated before deploying DP MH with real private data, but they are likely solvable, and most, like parameter tuning, and inexact sampling, also affect other methods of DP Bayesian inference. The benefits of MH over other Bayesian inference methods are present in the DP setting, so research into the many unknowns of DP MH is likely to be fruitful.

Bibliography

[Abo18] John M. Abowd. The U.S. census bureau adopts differential privacy. In Yike Guo and Faisal Farooq, editors, *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, page 2867. ACM, 2018.

[Bar65] Av A Barker. Monte Carlo calculations of the radial distribution functions for a proton-electron plasma. *Australian Journal of Physics*, 18(2):119–134, 1965.

[Bet15] Michael Betancourt. The fundamental incompatibility of scalable Hamiltonian Monte Carlo and naive data subsampling. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 533–540. JMLR.org, 2015.

[BS16] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part I*, pages 635–658, 2016.

[BS18] Garrett Bernstein and Daniel R. Sheldon. Differentially private bayesian inference for exponential families. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2924–2934, 2018.

[CD99] DM Ceperley and Mark Dewing. The penalty method for random walks with uncertain energies. *The Journal of chemical physics*, 110(20):9812–9820, 1999.

[CFG14] Tianqi Chen, Emily B. Fox, and Carlos Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1683–1691. JMLR.org, 2014.

[Çın11] Erhan Çınlar. *Probability and Stochastics*. Graduate Texts in Mathematics, 261. Springer New York, New York, NY, 1st ed. 2011. edition, 2011.

[CKS20] Clément L. Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[DFB+14] Nan Ding, Youhan Fang, Ryan Babbush, Changyou Chen, Robert D. Skeel, and Hartmut Neven. Bayesian sampling using stochastic gradient thermostats. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3203–3211, 2014.

[DKM+06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 486–503. Springer, 2006.

[DKPR87] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

[DKY17] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 3571–3580, 2017.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006.

[DMS19] Alain Durmus, Eric Moulines, and Eero Saksman. On the convergence of Hamiltonian Monte Carlo, 2019.

[DN03] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In Frank Neven, Catriel Beeri, and Tova Milo, editors, *Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 9-12, 2003, San Diego, CA, USA*, pages 202–210. ACM, 2003.

[DNZ$^+$17] Christos Dimitrakakis, Blaine Nelson, Zuhe Zhang, Aikaterini Mitrokotsa, and Benjamin I. P. Rubinstein. Differential privacy for bayesian inference through posterior sampling. *J. Mach. Learn. Res.*, 18:11:1–11:39, 2017.

[DPT17] Apple Differential Privacy Team. Learning with privacy at scale. Technical report, Apple Inc., 2017.

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[GBR$^+$12] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. A kernel two-sample test. *J. Mach. Learn. Res.*, 13:723–773, 2012.

[GCS$^+$14] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. Chapman & Hall/CRC texts in statistical science series. CRC Press, Boca Raton, third edition, 2014.

[Has70] W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. cited By 7759.

[HG14] Matthew D Hoffman and Andrew Gelman. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.

[HJDH19] Mikko A. Heikkilä, Joonas Jälkö, Onur Dikmen, and Antti Honkela. Differentially private Markov chain Monte Carlo. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 4115–4125, 2019.

[JHD17] Joonas Jälkö, Antti Honkela, and Onur Dikmen. Differentially private variational inference for non-conjugate models. In Gal Elidan, Kristian Kersting, and Alexander T. Ihler, editors, *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press, 2017.

[KJH20] Antti Koskela, Joonas Jälkö, and Antti Honkela. Computing tight differential privacy guarantees using FFT. In Silvia Chiappa and Roberto Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 2560–2569. PMLR, 2020.

[KJK+20] Tejas Kulkarni, Joonas Jälkö, Antti Koskela, Samuel Kaski, and Antti Honkela. Differentially private bayesian inference for generalized linear models. *CoRR*, abs/2011.00467, 2020.

[LCLC19] Bai Li, Changyou Chen, Hao Liu, and Lawrence Carin. On connecting stochastic gradient MCMC and differential privacy. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 557–566. PMLR, 16–18 Apr 2019.

[Mir12] Ilya Mironov. On significance of the least significant bits for differential privacy. In Ting Yu, George Danezis, and Virgil D. Gligor, editors, *the ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, October 16-18, 2012*, pages 650–661. ACM, 2012.

[Mir17] Ilya Mironov. Rényi differential privacy. In *30th IEEE Computer Security Foundations Symposium, CSF 2017, Santa Barbara, CA, USA, August 21-25, 2017*, pages 263–275, 2017.

[MRR+53] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[Nea12] Radford M. Neal. MCMC using Hamiltonian dynamics, 2012.

[RC04] Christian P. Robert and George Casella. Monte carlo statistical methods, 2004.

[RGG$^+$97] Gareth O Roberts, Andrew Gelman, Walter R Gilks, et al. Weak convergence and optimal scaling of random walk metropolis algorithms. *Annals of Applied probability*, 7(1):110–120, 1997.

[Sam01] Pierangela Samarati. Protecting respondents' identities in microdata release. *IEEE Trans. Knowl. Data Eng.*, 13(6):1010–1027, 2001.

[SMM19] David M. Sommer, Sebastian Meiser, and Esfandiar Mohammadi. Privacy loss classes: The central limit theorem in differential privacy. *PoPETs*, 2019(2):245–269, 2019.

[SPCC17] Daniel Seita, Xinlei Pan, Haoyu Chen, and John F. Canny. An efficient minibatch acceptance test for metropolis-hastings. In Gal Elidan, Kristian Kersting, and Alexander T. Ihler, editors, *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017.* AUAI Press, 2017.

[SS98] Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, SRI International, 1998.

[Tea20] Stan Development Team. Stan modeling language users guide and reference manual, 2.26, 2020.

[Tie98] Luke Tierney. A note on metropolis-hastings kernels for general state spaces. *Annals of applied probability*, pages 1–9, 1998.

[TPK14] Minh-Ngoc Tran, Michael K. Pitt, and Robert Kohn. Adaptive Metropolis-Hastings sampling using reversible dependent mixture proposals. *Statistics and Computing*, 26(1-2):361–381, 2014.

[Tu11] Loring W Tu. *Introduction to Manifolds.* Universitext. Springer New York, New York, 2011.

[WBK19] Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled Renyi differential privacy and analytical moments accountant. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, pages 1226–1235, 2019.

[WFS15] Yu-Xiang Wang, Stephen E. Fienberg, and Alexander J. Smola. Privacy for free: Posterior sampling and stochastic gradient monte carlo. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2493–2502. JMLR.org, 2015.

[WT11] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 681–688. Omnipress, 2011.

[YE19] Sinan Yildirim and Beyza Ermis. Exact MCMC with differentially private moves - revisiting the penalty algorithm in a data privacy framework. *Statistics and Computing*, 29(5):947–963, 2019.

[ZRD16] Zuhe Zhang, Benjamin I. P. Rubinstein, and Christos Dimitrakakis. On the differential privacy of bayesian inference. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2365–2371. AAAI Press, 2016.

Appendix A. Proof of Theorem 5.2

Theorem A.2. *Let*

$$\begin{aligned}
\theta = (\theta_1, \dots, \theta_d) &\sim \mathrm{Ban}(0, \sigma_0^2 I, a, b, m) \\
x_1 &\sim \mathcal{N}(\theta_1, \sigma_1^2) \\
x_2 &\sim \mathcal{N}(\theta_2 + a(\theta_1 - m)^2 + b, \sigma_2^2) \\
x_3 &\sim \mathcal{N}(\theta_3, \sigma_3^2) \\
&\vdots \\
x_d &\sim \mathcal{N}(\theta_d, \sigma_d^2)
\end{aligned}$$

Given data $x_1, \dots, x_n \in \mathbb{R}^d$ and denoting $\tau_i = \frac{1}{\sigma_i^2}$, the posterior of θ tempered with T is the banana distribution $\mathrm{Ban}(\mu, \Sigma, a, b, m)$ *with*

$$\bar{x}_i = \frac{1}{n} \sum_{j=1}^{n} x_{ji} \quad i \in \{1, 2\}$$

$$\mu = \left(\frac{T n \tau_1 \bar{x}_1}{T n \tau_1 + \tau_0}, \dots, \frac{T n \tau_d \bar{x}_d}{T n \tau_d + \tau_0} \right),$$

$$\Sigma = \mathrm{diag}\left(\frac{1}{T n \tau_1 + \tau_0}, \dots, \frac{1}{T n \tau_d + \tau_0} \right).$$

Proof. Because

$$g^{-1}(y) = (y_1, y_2 + a(y_1 - m)^2 + b, y_3 \dots, y_d)$$

and the Jacobian determinant of g^{-1} is 1, for a positive-definite Σ the banana distribution has density proportional to

$$\exp\left(-\frac{1}{2} (g^{-1}(x) - \mu)^T \Sigma^{-1} (g^{-1}(x) - \mu) \right)$$

With $\Sigma = \mathrm{diag}(\sigma_1^2, \sigma_2^2)$ the density is proportional to

$$\exp\left(-\frac{1}{2} \left(\left(\frac{x_1 - \mu_1}{\sigma_1} \right)^2 + \left(\frac{x_2 + a(x_1 - m)^2 + b - \mu_2}{\sigma_2} \right)^2 + \sum_{i=3}^{d} \left(\frac{x_i - \mu_i}{\sigma_i} \right)^2 \right) \right)$$

Denote $u = \theta_2 + a(\theta_1 - m)^2 + b$. The tempered posterior of θ is

$$\begin{aligned}
p(\theta \mid X) &\propto p(X \mid \theta)^T p(\theta) \\
&= p(x_1 \mid \theta_1)^T p(x_2 \mid \theta_1, \theta_2)^T \prod_{i=3}^{d} p(x_i \mid \theta_i)^T p(\theta) \\
&= p(x_1 \mid \theta_1)^T p(x_2 \mid \theta_1, \theta_2)^T \prod_{i=3}^{d} p(x_i \mid \theta_i)^T \\
&\cdot \exp\left(-\frac{1}{2}\left(\tau_0\theta_1^2 + \tau_0(\theta_2 + a(\theta_1 - m)^2 + b)^2 \sum_{i=3}^{d} \tau_0\theta_i^2\right)\right) \\
&= p(x_1 \mid \theta_1)^T p(x_2 \mid \theta_1, \theta_2)^T \exp\left(-\frac{1}{2}\left(\tau_0\theta_1^2 + \tau_0(\theta_2 + a(\theta_1 - m)^2 + b)^2\right)\right) \\
&\cdot \prod_{i=3}^{d} p(x_i \mid \theta_i)^T \exp\left(-\frac{1}{2}\sum_{i=3}^{d} \tau_0\theta_i^2\right)
\end{aligned}$$

Considering the upper and lower part of the last expression separately

$$
\begin{aligned}
&p(x_1 \mid \theta_1)^T p(x_2 \mid \theta_1, \theta_2)^T \exp\left(-\frac{1}{2}\left(\tau_0\theta_1^2 + \tau_0(\theta_2 + a(\theta_1 - m)^2 + b)^2\right)\right) \\
&\propto \left(\prod_{i=1}^{n} \exp\left(-\frac{(x_{i1} - \theta_1)^2\tau_1}{2}\right)\right)^T \cdot \left(\prod_{i=1}^{n} \exp\left(-\frac{(x_{i2} - \theta_2 - a(\theta_1 - m)^2 - b)^2\tau_2}{2}\right)\right)^T \\
&\cdot \exp\left(-\frac{1}{2}\left(\tau_0\theta_1^2 + \tau_0(\theta_2 + a(\theta_1 - m)^2 + b)^2\right)\right) \\
&= \exp\left(-\frac{1}{2}\Big(T\tau_1 \sum_{i=1}^{n}(x_{i1} - \theta_1)^2 + T\tau_2 \sum_{i=1}^{n}(x_{i2} - u)^2 + \tau_0\theta_1^2 + \tau_0 u^2\Big)\right) \\
&= \exp\left(-\frac{1}{2}\Big(T\tau_1 \sum_{i=1}^{n}(x_{i1} - \bar{x}_1)^2 + T\tau_1 n(\bar{x}_1 - \theta_1)^2\right. \\
&\left. + T\tau_2 \sum_{i=1}^{n}(x_{i2} - \bar{x}_2)^2 + T\tau_2 n(\bar{x}_2 - u)^2 + \tau_0\theta_1^2 + \tau_0 u^2\Big)\right) \\
&\propto \exp\left(-\frac{1}{2}\Big(T\tau_1 n(\bar{x}_1 - \theta_1)^2 + T\tau_2 n(\bar{x}_2 - u)^2 + \tau_0\theta_1^2 + \tau_0 u^2\Big)\right) \\
&= \exp\left(-\frac{1}{2}\Big(T\tau_1 n\bar{x}_1^2 - 2T\tau_1 n\bar{x}_1\theta_1 + nT\tau_1\theta_1^2 + \tau_0\theta_1^2\right. \\
&\left. + T\tau_2 n\bar{x}_2^2 - 2T\tau_2 n\bar{x}_2 u + nT\tau_2 u^2 + \tau_0 u^2\Big)\right) \\
&\propto \exp\left(-\frac{1}{2}\Big((Tn\tau_1 + \tau_0)\theta_1^2 - 2T\tau_1 n\bar{x}_1\theta_1 + (Tn\tau_2 + \tau_0)u^2 - 2T\tau_2 n\bar{x}_2 u\Big)\right) \\
&= \exp\left(-\frac{1}{2}\Big((Tn\tau_1 + \tau_0)\left(\theta_1^2 - \frac{2T\tau_1 n\bar{x}_1\theta_1}{Tn\tau_1 + \tau_0}\right) + (Tn\tau_2 + \tau_0)\left(u^2 - \frac{2T\tau_2 n\bar{x}_2 u}{Tn\tau_2 + \tau_0}\right)\Big)\right) \\
&\propto \exp\left(-\frac{1}{2}\Big((Tn\tau_1 + \tau_0)\left(\theta_1 - \frac{T\tau_1 n\bar{x}_1}{Tn\tau_1 + \tau_0}\right)^2 + (Tn\tau_2 + \tau_0)\left(u - \frac{T\tau_2 n\bar{x}_2}{Tn\tau_2 + \tau_0}\right)^2\Big)\right)
\end{aligned}
$$

and

$$\begin{aligned}
&\prod_{i=3}^{d} p(x_i \mid \theta_i)^T \cdot \exp\left(-\frac{1}{2}\sum_{i=3}^{d}\tau_0\theta_i^2\right)\\
&\propto \exp\left(-\frac{1}{2}T\sum_{j=3}^{d}\tau_j\sum_{i=1}^{n}(x_{ij}-\theta_j)^2-\frac{1}{2}\sum_{j=3}^{d}\tau_0\theta_j^2\right)\\
&= \exp\left(-\frac{1}{2}\sum_{j=3}^{d}\left(T\tau_j\sum_{i=1}^{n}(x_{ij}-\theta_j)^2+\tau_0\theta_j^2\right)\right)\\
&= \exp\left(-\frac{1}{2}\sum_{j=3}^{d}\left(T\tau_j\sum_{i=1}^{n}(x_{ij}-\bar{x}_j)^2+T\tau_j n(\bar{x}_j-\theta_j)^2+\tau_0\theta_j^2\right)\right)\\
&\propto \exp\left(-\frac{1}{2}\sum_{j=3}^{d}\left(T\tau_j n(\bar{x}_j-\theta_j)^2+\tau_0\theta_j^2\right)\right)\\
&\propto \exp\left(-\frac{1}{2}\sum_{j=3}^{d}\left(-2T\tau_j n\bar{x}_j\theta_j+T\tau_j n\theta_j^2+\tau_0\theta_j^2\right)\right)\\
&\propto \exp\left(-\frac{1}{2}\sum_{j=3}^{d}\left((Tn\tau_j+\tau_0)\theta_j^2-2Tn\tau_j\bar{x}_j\theta_j+\frac{(Tn\tau_j\bar{x}_j)^2}{Tn\tau_j+\tau_0}\right)\right)\\
&= \exp\left(-\frac{1}{2}\sum_{j=3}^{d}(Tn\tau_j+\tau_0)\left(\theta_j-\frac{Tn\tau_1\bar{x}_i}{Tn\tau_j+\tau_0}\right)^2\right)
\end{aligned}$$

Multiplying the resulting expression above gives a density proportional to the banana distribution. As $p(\theta \mid X)$ is proportional to the density of a banana distribution, the posterior is the banana distribution $\mathrm{Ban}(\mu, \Sigma, a, b, m)$ with

$$\mu = \left(\frac{Tn\tau_1\bar{x}_1}{Tn\tau_1+\tau_0}, \dots, \frac{Tn\tau_d\bar{x}_d}{Tn\tau_d+\tau_0}\right),$$

$$\Sigma = \mathrm{diag}\left(\frac{1}{Tn\tau_1+\tau_0}, \dots, \frac{1}{Tn\tau_d+\tau_0}\right). \qquad \square$$

Appendix B. Differentiability of the clip-function

This appendix proves that functions of the form $\mathrm{clip} \circ f$ are almost everywhere differentiable for sufficiently well-behaved f, as required by the proof of volume preservation of DP HMC leapfrog iterations (Section 4.2). Lemma B.2 gives very general sufficient conditions, and Lemma B.3 proves that the models considered in this thesis meet the conditions.

Lemma B.1. *Let $d \geq 2$ and $g \colon U \to \mathbb{R}$, where U is an open subset of $\mathbb{R}^d$, be continuously differentiable. Let $S = \{x \in U \mid f(x) = b\}$, $b \in \mathbb{R}$. If for all $x \in S$, $\nabla f(x) \neq 0$, S is a $(d-1)$-dimensional hypersurface.*

Proof. See [Tu11, Section 9.2]. □

Lemma B.2. *Let $f \colon \mathbb{R}^d \to \mathbb{R}^d$ be continuously differentiable. If the set of saddle points U of $||f||$ is a null set, the set $A \subset \mathbb{R}^d$ where $\mathrm{clip}_b \circ f$ is not differentiable is a null set.*

Proof.

$$\mathrm{clip}_b(x) = \begin{cases} \frac{x}{||x||} \min\{||x||, b\} & \text{if } x \neq 0 \\ 0 & \text{if } x = 0 \end{cases}$$

which means that clip_b is differentiable for all $x \in \mathbb{R}^d$ with $||x|| \neq b$. Consider $\mathrm{clip}_b \circ f$ in a neighbourhood B of point $x_0 \in \mathbb{R}^d$. If $||f(x)|| \leq b$ for all $x \in B$, $\mathrm{clip}_b(f(x)) = f(x)$ in B, which is differentiable. If $||f(x)|| \geq b$ for all $x \in B$, $\mathrm{clip}_b(f(x)) = \frac{f(x)b}{||f(x)||}$ in B, which is also differentiable because $f(x) \neq 0$ when $||f(x)|| \geq b > 0$. In both cases $\mathrm{clip}_b \circ f$ is differentiable at x_0. This means that for all $x_0 \in A$, $||f(x_0)|| = b$, but every neighborhood of x_0 has points x' and x'' such that $||f(x')|| < b$ and $||f(x'')|| > b$. With the differentiability of f, this means that if $\nabla ||f(x_0)|| = 0$, x_0 is a saddle point of $||f||$.

Let $S = \{x \in \mathbb{R}^d \mid ||f(x)|| = b \text{ and } \nabla ||f(x)|| \neq 0\}$. By Lemma B.1, S is a $(d-1)$-dimensional hypersurface, which has zero measure in d-dimensional space. Because $A \cap U^C \subset S$, $A \cap U^C$ is also a null set. Since $A = (A \cap U^C) \cup (A \cap U)$, A is a null set.

□

Lemma B.3. *The log-likelihoods of the Gaussian, banana and circle models meet the conditions of Lemma B.2.*

Proof. For DP HMC, set $f(\theta) = \nabla \ln p(x \mid \theta)$ for some datapoint x. Denote $l(\theta) = \ln(x \mid \theta)$. The conditions of Lemma B.2 for l are met if l is twice continuously differentiable and

$$\nabla ||\nabla l(\theta)|| = \frac{2\nabla l(\theta)}{||\nabla l(\theta)||} J_{\nabla l}(\theta) = \frac{2\nabla l(\theta)}{||\nabla l(\theta)||} H_l(\theta) \neq 0$$

almost everywhere, where H_l is the Hessian matrix of l. Because $\nabla l = 0$ and $||\nabla l|| = 0$ at only one point, having $\det H_l(x) \neq 0$ almost everywhere is sufficient.

For the Gaussian log-likelihood

$$l(\theta) = \frac{1}{2}(x-\theta)^T \Sigma^{-1}(x-\theta),$$

so

$$\nabla l(\theta) = \Sigma^{-1}(x-\theta)$$

and $H_l(\theta) = \Sigma^{-1}$.

For the banana log-likelihood, using the notation from Appendix A,

$$l(\theta) = \frac{1}{2}(g^{-1}(x)-\theta)^T \Sigma^{-1}(g^{-1}(x)-\theta),$$

so

$$\nabla l(\theta) = \Sigma^{-1}(g^{-1}(x)-\theta)$$

and $H_l(\theta) = \Sigma^{-1}$.

The circle log-likelihood is

$$l(x,y) = -a(x^2+y^2-r^2)^2,$$

$$\nabla l(x,y) = -4a(x^3+xy^2-xr^2, x^2y+y^3-yr^2),$$

so

$$H_l(x,y) = -4a\begin{bmatrix} 3x^2+y^2-r^2 & 2xy \\ 2xy & 3y^2+x^2-r^2 \end{bmatrix}$$

and

$$\begin{aligned}\det H_l(x,y) &= -4a((3x^2+y^2-r^2)(3y^2+x^2-r^2)-4x^2y^2) \\ &= -4a(9x^2y^2+3x^4-3x^2r^2+3y^4+y^2x^2-y^2r^2-3y^2r^2-x^2r^2+r^4-4x^2y^2) \\ &= -4a(6x^2y^2+3x^4-4x^2r^2+3y^4-4y^2r^2+r^4) \\ &= -4a(6x^2y^2+3(x^4+y^4)-4r^2(x^2+y^2)+r^4).\end{aligned}$$

$$\nabla \det H_l(x,y) = -4a(12xy^2+12x^3-8r^2x, 12yx^2+12y^3-8r^2y),$$

which is zero in the origin and on the circle

$$x^2+y^2 = \frac{2}{3}r^2.$$

Elsewhere, Lemma B.1 can be applied to $\det H_l$, which means that the set where $\det H_l(x,y) = 0$ is a 1-dimensional curve. □