# Differentially Private Hamiltonian Monte Carlo

**Ossi Räisä** **Antti Koskela** **Antti Honkela**
Helsinki Institute for Information technology HIIT
Department of Computer Science, University of Helsinki

## Abstract

We present DP-HMC, a variant of Hamiltonian Monte Carlo (HMC) that is differentially private (DP). We use the penalty algorithm of Yildirim and Ermis [18] to make the acceptance test private, and add Gaussian noise to the gradients of the target distribution to make the HMC proposal private. Our main contribution is showing that DP-HMC has the correct invariant distribution, and is ergodic. We also compare DP-HMC with the existing penalty algorithm, as well as DP-SGLD [12, 17] and DP-SGNHT [17].

## 1 Introduction

Bayesian inference is one of the widely used approaches for analysis of potentially sensitive data. We present the first differentially private (DP) variant of one of the most common modern Bayesian inference algorithms, Hamiltonian Monte Carlo (HMC).

While there are many existing methods for DP Bayesian inference, most require strong assumptions on the statistical model, such as ability to sample from the exact posterior [4, 17] or specific model structure [1, 2, 11, 19]. Of the more general algorithms, DP variational inference [10] is not able to exactly sample the posterior, and DP stochastic gradient Markov Chain Monte Carlo (MCMC) algorithms like DP-SGLD [12, 17] and DP-SGNHT [17] only have a weak convergence guarantee requiring decreasing their step size to 0.

Conventional MCMC algorithms avoid these issues by including a Metropolis–Hastings (MH) acceptance test. Currently, only two implementations of DP MH algorithms exist [9, 18]. We build on the penalty algorithm [18] to ensure both privacy and exact convergence for HMC that is able to make use of the gradients of the posterior to speed up convergence.

## 2 Background

### 2.1 Differential Privacy

We exclusively use the Gaussian mechanism, and thus use the *approximate DP* (ADP) [6] definition, also known as $(\epsilon, \delta)$-DP:

**Definition 2.1.** *A mechanism $\mathcal{M}\colon \mathcal{X} \to \mathbb{R}^d$ is $(\epsilon, \delta)$-ADP for neighbourhood relation $\sim$ if for all measurable $S \subset \mathbb{R}^d$ and all $X, X' \in \mathcal{X}$ with $X \sim X'$, $P(\mathcal{M}(X) \in S) \leq e^\epsilon P(\mathcal{M}(X') \in S) + \delta$.*

We exclusively focus on tabular data and the substitute neighbourhood relation $\sim_S$ which means that datasets $X, X' \in \mathbb{R}^{n \times d_x}$ are neighbors in $\sim_S$-relation, $X \sim_S X'$, if they differ in at most one row. We use $x \in X$ to denote that $x$ is a row of $X$.

To make HMC DP, we use the *Gaussian mechanism* [6], together with post-processing immunity and composition. To compute the privacy bounds for compositions of several Gaussian mechanisms, we use the tight ADP bound of Sommer et al. [16, Lemma 12].

### 2.2 Bayesian Inference, Metropolis-Hastings and Hamiltonian Monte Carlo

The goal of Bayesian inference is inferring information on the parameters $\theta$ as a statistical model that is assumed to generate observed data $X$. This is done through the posterior $p(\theta|X)$ that is given by Bayes' theorem:

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{p(X)}.$$

We focus on the case where $X \in \mathbb{R}^{n \times d_x}$ is a table, and each row $x$ of $X$ is an i.i.d observation from the distribution $p(x|\theta)$. Then $p(X|\theta)$ factors: $p(X|\theta) = \prod_{x \in X} p(x \mid \theta)$.

Computing $p(X)$ is often intractable, so posterior inference is done through sampling $p(\theta|X)$ using an *Markov Chain Monte Carlo* (MCMC) algorithm. The *Metropolis-Hastings* (MH) [8, 13] algorithm is a particular family of MCMC algorithms that sample a given distribution $\pi$ by forming a Markov chain by starting from a given value $\theta_0$, picking a proposal $\theta'$ from a give proposal distribution $q(\theta \mid \theta')$ and accepting the proposal with probability

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

where $\theta$ denotes the current value of the chain. If the proposal is accepted, the chain moves to the proposal, otherwise it stays with the current value. For posterior sampling, we simply set $\pi(\theta) = p(\theta|X)$. The intractable $p(X)$ term does not need to be evaluated as it cancels out in the expression for $\alpha(\theta, \theta')$.

To correctly sample from the target, an MH algorithm must have $\pi$ as its invariant distribution, and must be ergodic. Using the acceptance probability $\alpha(\theta, \theta')$ guarantees the correct invariant distribution, but ergodicity depends on the proposal and must be proven separately. A simple sufficient condition for ergodicity is *strong irreducibility*: if the proposal can propose any state from any state, the algorithm is said to be strongly irreducible, which implies ergodicity [15].

The *Hamiltonian Monte Carlo* (HMC) algorithm is an MH algorithm that generates proposals deterministically by simulating Hamiltonian dynamics. The dynamics are given by the *Hamiltonian* $H(\theta, p) = U(\theta) + \frac{1}{2} p^T M^{-1} p$, where $p \in \mathbb{R}^d$ is an auxiliary momentum variable, $M \in \mathbb{R}^{d \times d}$ is a positive-definite mass matrix, and $U(\theta) = -\ln \pi(\theta)$. The simulation is done using *leapfrog simulation*, given for a step-size $\eta > 0$ by

$$l = l_{p_{\eta/2}} \circ l_\theta \circ l_{p_\eta} \cdots \circ l_{p_\eta} \circ l_\theta \circ l_{p_{\eta/2}},$$

where

$$l_{p_s}(\theta, p) = (\theta, p - s\nabla U(\theta)), \quad l_\theta(\theta, p) = (\theta + \eta M^{-1} p, p).$$

With the auxiliary variable $p$, HMC targets the distribution

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

so the marginal distributions of $\theta$ and $p$ are independent, the marginal of $\theta$ is $\pi$, and the marginal of $p$ is a $d$-dimensional Gaussian with mean 0 and covariance $M$.

Proposing a new sample is done in two steps, both of which having a separate MH acceptance test. First, $p$ is sampled from its marginal distribution, which is always accepted. Second, the leapfrog simulation is run and the final value of $p$ is negated, which gives a proposal for $(\theta, p)$. The acceptance probability for the second step is $\alpha(\theta, p, \theta', p') = \min\{1, \exp(H(\theta, p) - H(\theta', p'))\}$.

## 3 DP-HMC

Our goal is making HMC DP, which requires adding noise to every point where the data $X$ is used. With HMC, $X$ is used in the leapfrog simulation through $\nabla U(\theta)$, and in the acceptance test to compute $\alpha(\theta, \theta')$.

The DP-penalty algorithm of Yildirim and Ermis [18] makes the MH acceptance test private by adding Gaussian noise to the log-likelihood ratio $\lambda(\theta, \theta') = \ln \frac{p(X|\theta')p(\theta')}{p(X|\theta)p(\theta)}$. They correct the MH

acceptance probability with the penalty algorithm [3], that changes the acceptance probability to

$$\alpha(\theta, \theta') = \min\left\{1, \exp\left(\lambda(\theta, \theta') + \xi + \ln\frac{q(\theta \mid \theta')}{q(\theta' \mid \theta)} - \frac{1}{2}\sigma_l^2(\theta, \theta')\right)\right\},$$

where $\xi \sim \mathcal{N}(0, \sigma_l^2(\theta, \theta'))$ is the Gaussian noise added to the log likelihood ratio. For the DP-penalty algorithm, $\sigma_l(\theta, \theta') = 2\tau b_l||\theta - \theta'||_2$, $b_l||\theta - \theta'||_2$ is the log-likelihood ratio clip bound and $\tau > 0$ controls the amount of noise.

The privacy bounds for the algorithm are then given by [16, Lemma 12]. The convergence of the penalty algorithm requires that the log-likelihood ratios are not actually clipped, However, in our experiments small amounts of clipping did not affect the resulting posterior.

In non-DP HMC, the proposal is the deterministic leapfrog simulation, which can be made DP by simply clipping the gradients of the log-likelihood and adding Gaussian noise. In Theorem 3.1, we show that applying the penalty correction the HMC acceptance probability results in the correct invariant distribution when using noisy and clipped gradients in the leapfrog simulation. We also prove the ergodicity of DP-HMC, in Theorem 3.2.

In the noisy and clipped leapfrog simulation, the momentum update changes to $l_{p_s}(\theta, p) = (\theta, p - s(g(\theta) + \xi))$, where $g(\theta) = \sum_{x \in X} \text{clip}_b(\nabla \ln p(x \mid \theta)) + \nabla \ln p(\theta)$ and $\xi \sim \mathcal{N}(0, \sigma_g^2)$. The noisy and clipped leapfrog is then

$$l = l_{p_{\eta/2}} \circ l_\theta \circ l_{p_\eta} \cdots \circ l_{p_\eta} \circ l_\theta \circ l_{p_{\eta/2}}. \tag{1}$$

As $l_-$ is an involution, $l_- \circ l$ can be decomposed as

$$l_- \circ l = (l_- \circ p_{\eta/2}) \circ (l_\theta \circ l_-) \circ (l_- \circ l_{p_\eta}) \cdots \circ (l_- \circ l_{p_\eta}) \circ (l_\theta \circ l_-) \circ (l_- \circ l_{p_{\eta/2}}).$$

Denoting $l_{p_s}^- = l_- \circ l_{p_s}$ and $l_\theta^- = l_\theta \circ l_-$, the decomposition can be written as

$$l_- \circ l = l_{p_{\eta/2}}^- \circ l_\theta^- \circ l_{p_\eta}^- \circ \cdots \circ l_{p_\eta}^- \circ l_\theta^- \circ l_{p_{\eta/2}}^-.$$

This form makes showing that DP-HMC has the correct invariant distribution convenient.

**Theorem 3.1.** *For a continuous distribution $\pi$ that is supported on $\mathbb{R}^d$ and has a differentiable log-likelihood, if*

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

*where $\xi \sim \mathcal{N}(0, \sigma_l^2(\theta, \theta'))$, is used as the acceptance probability of DP-HMC and log-likelihood ratios are not clipped, the invariant distribution is $\pi^*(\theta, p) \propto \exp(-H(\theta, p))$.*

*Proof.* We sketch the proof. The invariance of the target for HMC is proven using the fact that the leapfrog simulation is an involution, and it preserves Lebesgue measure [14]. As the DP-HMC leapfrog is a random algorithm, it is not an involution, but it meets a related notion of *reversibility* for a *Markov kernel*, which is the measure-theoretic formulation of a random algorithm. We show the DP-HMC leapfrog reversibility by proving that both $l_{p_s}^-$ and $l_\theta^-$ are themselves reversible and use the fact that a symmetric composition of reversible Markov kernels is itself reversible.

Using the reversibility of the DP-HMC leapfrog, we show that DP-HMC without adding noise to the log-likelihood ratios has the correct invariant distribution. Adding noise to the log-likelihood ratios is handled with the penalty algorithm [18]. □

**Theorem 3.2.** *DP-HMC is strongly irreducible, and thus ergodic.*

*Proof.* Each of the $l_p$ steps adds Gaussian noise to $p$, and $l_\theta$ adds $p$ multiplied by an invertible matrix to $\theta$, so after all of the steps, the noisy leapfrog can get to any $(\theta', p')$ when starting from any $(\theta, p)$, so DP-HMC is strongly irreducible and thus ergodic. □

In the noisy leapfrog simulation, the gradient $\nabla U$ is evaluated $L + 1$ times per iteration of the outer for-loop, for a total of $k(L + 1)$ times, and the log-likelihood ratio is evaluated $k$ times in total. The privacy cost can then be computed from the tight bound for the Gaussian mechanism [16]. We control the noise variances for log-likelihood ratios by $\tau_l$ and gradients by $\tau_g$.

**Theorem 3.3.** *DP-HMC is $(\epsilon, \delta(\epsilon))$-ADP for substitute neighbourhood for*

$$\delta(\epsilon) = \frac{1}{2}\left(\operatorname{erfc}\left(\frac{\epsilon-\mu}{2\sqrt{\mu}}\right) - e^{\epsilon}\operatorname{erfc}\left(\frac{\epsilon+\mu}{2\sqrt{\mu}}\right)\right), \quad \textit{where} \quad \mu = \frac{k}{2\tau_l^2} + \frac{k(L+1)}{2\tau_g^2}.$$

*Proof.* The sensitivity of the log-likelihood ratio is $2b_g||\theta - \theta'||_2$ and the sensitivity of the gradient is $2b_g$. Thus, adding noise with variance $\sigma_l^2(\theta, \theta')$ to the log-likelihood ratio gives a sensitivity-variance ratio $\mu_l = \frac{1}{2\tau_l^2}$. Adding noise with variance $\sigma_g^2$ to the gradients has sensitivity-variance ratio $\mu_g = \frac{1}{2\tau_l^2}$. As the log-likelihood ratio is evaluated $k$ times and the gradients are evaluated $k(L+1)$ times, the total $\mu$ in [16, Lemma 12] is $\mu = k\mu_l + k(L+1)\mu_g = \frac{k}{2\tau_l^2} + \frac{k(L+1)}{2\tau_g^2}$. □

Theorem 3.3 implies a tradeoff between the number of iterations, and the noise variance: running more iterations requires increasing noise variance, which decreases the acceptance rate and with it the quality of the obtained samples. There is also a tradeoff between log-likelihood ratio and gradient noise: both kinds of noise decrease the acceptance rate, and reducing one requires increasing the other, if privacy bounds are held constant.

## 4 Experiments

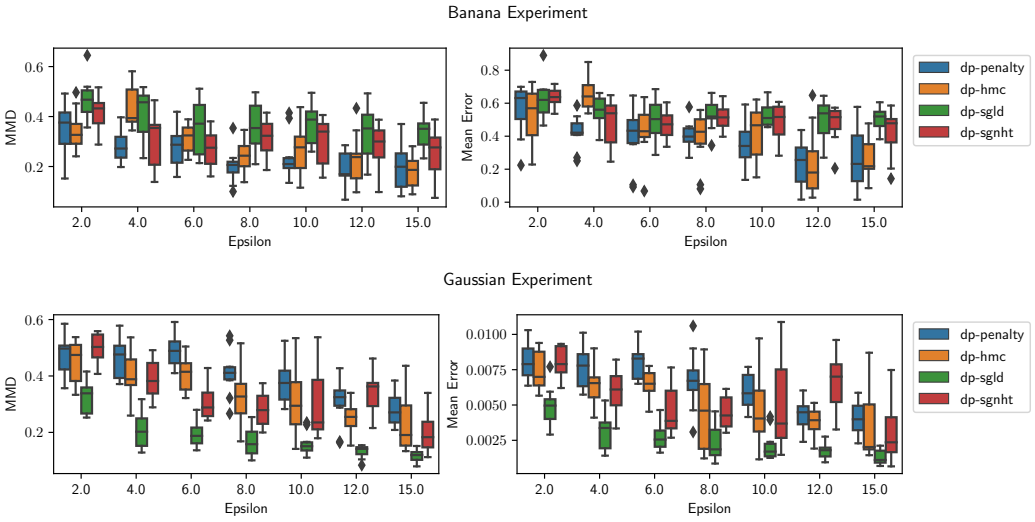


Figure 1: MMD and mean error for the banana and Gaussian models.

We ran experiments on two synthetic posterior distributions: a 10-dimensional correlated Gaussian model and a 2-dimensional banana distribution model that results in a non-convex banana shaped posterior. We compare DP-HMC with DP-penalty [18], DP-SGLD [12, 17] and DP-SGNHT [5, 17] by running 4 chains, started from separate points chosen randomly around the posterior and repeat each run 10 times. The 4 chains are combined and compared with a reference sample from the true posterior. As evaluation metrics, we used *maximum mean discrepancy* (MMD) [7], which is a distance measure of distributions that is only 0 for identical distributions. We also used the distance of the means of the resulting sample and the reference sample as more interpratable metric.

The top row of Figure 1 shows the result of running each algorithm on the banana model. DP-HMC and DP-penalty have roughly equal performance on both MMD and mean error, while DP-SGLD and DP-SGNHT have significantly worse performance, especially with the higher values of $\epsilon$. The bottom row shows the results with the Gaussian model. The best performer was DP-SGLD, DP-HMC and DP-SGNHT were mostly equal, and DP-penalty performed the worst.

## References


[1] Garrett Bernstein and Daniel R. Sheldon. Differentially private Bayesian inference for exponential families. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*, pages 2924–2934, 2018.

[2] Garrett Bernstein and Daniel R. Sheldon. Differentially private bayesian linear regression. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 523–533, 2019.

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

[4] 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.

[5] Nan Ding, Youhan Fang, Ryan Babbush, Changyou Chen, Robert D. Skeel, and Hartmut Neven. Bayesian sampling using stochastic gradient thermostats. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems*, pages 3203–3211, 2014.

[6] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 4004 of *Lecture Notes in Computer Science*, pages 486–503. 2006.

[7] 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.

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

[9] 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*, pages 4115–4125, 2019.

[10] Joonas Jälkö, Onur Dikmen, and Antti Honkela. Differentially private variational inference for non-conjugate models. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI*. 2017.

[11] Tejas Kulkarni, Joonas Jälkö, Antti Koskela, Samuel Kaski, and Antti Honkela. Differentially private bayesian inference for generalized linear models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5838–5849. 18–24 Jul 2021.

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

[13] 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.

[14] Radford M. Neal. MCMC using Hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo*. Chapman & Hall / CRC Press, 2011.

[15] Christian P. Robert and George Casella. *Monte Carlo statistical methods*. Springer texts in statistics. Springer, New York, 2nd ed. edition, 2004.

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

[17] Yu-Xiang Wang, Stephen E. Fienberg, and Alexander J. Smola. Privacy for free: Posterior sampling and stochastic gradient Monte Carlo. In *Proceedings of the 32nd International Conference on Machine Learning, ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2493–2502. 2015.

[18] 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.

[19] Zuhe Zhang, Benjamin I. P. Rubinstein, and Christos Dimitrakakis. On the differential privacy of Bayesian inference. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2365–2371. 2016.