Quantum-Inspired Hamiltonian Monte Carlo for Bayesian Sampling

Ziming Liu†,⋆ and Zheng Zhang†
†Department of Electrical & Computer Engineering, University of California, Santa Barbara, CA, USA
⋆Department of Physics, Peking University, Beijing, China
E-mails: liu_zi_ming@pku.edu.cn, zhengzhang@ece.ucsb.edu

Abstract
Hamiltonian Monte Carlo (HMC) is an efficient Bayesian sampling method that can make distant proposals in the parameter space by simulating a Hamiltonian dynamical system. Despite its popularity in machine learning and data science, HMC is inefficient to sample from spiky and multimodal distributions. Motivated by the energy-time uncertainty relation from quantum mechanics, we propose a Quantum-Inspired Hamiltonian Monte Carlo algorithm (QHMC). This algorithm allows a particle to have a random mass with a probability distribution rather than a fixed mass. We prove the convergence property of QHMC in the spatial domain and in the time sequence. We further show why such a random mass can improve the performance when we sample a broad class of distributions. In order to handle the big training data sets in large-scale machine learning, we develop a stochastic gradient version of QHMC using Nosé-Hoover thermostat called QSGNHT, and we also provide theoretical justifications about its steady-state distributions. Finally in the experiments, we demonstrate the effectiveness of QHMC and QSGNHT on synthetic examples, bridge regression, image denoising and neural network pruning. The proposed QHMC and QSGNHT can indeed achieve much more stable and accurate sampling results on the test cases.

1 Introduction
Hamiltonian Monte Carlo (HMC) [1, 2] improves the traditional Markov Chain Monte Carlo (MCMC) method [3] by introducing an extra momentum vector \( q \) conjugate to a state vector \( x \). In MCMC, a particle is usually only allowed to randomly walk in the state space. However, in HMC, a particle can flow quickly along the velocity direction and make distant proposals in the state space based on Hamiltonian dynamics. The energy-conservation property in continuous Hamiltonian dynamics can largely increase the acceptance rate, resulting in much faster mixing rate than standard MCMC. In numerical simulations where continuous time is quantized by discrete steps, the Metropolis-Hastings (MH) correction step can guarantee the correctness of the invariant distribution. HMC plays an increasingly important role in machine learning [4–6].

HMC requires computation of the gradients of the energy function, which is the negative logarithm of the posterior probability in a Bayesian model. Therefore, HMC is born for smooth and differentiable functions but not immediate for non-smooth functions. In the case where functions have discontinuities, extra reflection and refraction steps [7, 8] need to be involved in order to enhance the sampling efficiency. In the case characterized by \( \ell_p \) prior where \( 1 \leq p < 2 \), proximity operator methods [9, 10] can be used to increase the accuracy for simulating the Hamiltonian dynamics.

This paper aims to develop a new type of HMC method in order to sample efficiently from a possibly spiky or multimodal posterior distribution. A representative example is Bayesian model of sparse or low-rank modeling using an \( \ell_p \) (0 < \( p \) ≤ 1) norm as the penalty. It is shown that sparser
models can be obtained because $\ell_p$ ($0 < p < 1$) prior [11, 12] is a better, although non-convex, approximation for $\ell_0$ norm [13] than $\ell_1$ norm that is widely used in compressed sensing [14,15] and model reductions [16].

Leveraging the energy-time uncertainty relation in quantum mechanics, this paper proposes a quantum-inspired HMC (QHMC) method. In quantum mechanics, a particle can have a mass distribution other than a fixed mass value. Although being a pedagogical argument from physics, we will show from both theories and experiments that this principle can actually help achieve better sampling efficiency for non-smooth functions than standard HMC with fixed mass. The main idea of QHMC is illustrated in Fig. 1 with a one-dimensional harmonic oscillator. Assume that we have a ball with mass $m$ attached to a spring at the origin. The restoring force exerted on the ball pulls the ball back to the origin and the magnitude of the force is proportional to the displacement $x$, i.e., $F = -kx$. The ball oscillates around the origin with the time period $T = 2\pi \sqrt{m/k}$. In standard HMC, $m$ is fixed and usually set as 1. In contrast, QHMC allows $m$ to be time-varying, meaning that the particle is sometimes moving faster and sometimes moving slower. This is equivalent to employing a varying time scale. Consequently, different distribution landscapes can be explored more effectively with different time scales. In a flat but broad region, QHMC can quickly scan the whole region with a small time period $T$ (or small $m$). In a spiky region, we need to carefully explore every corner of the landscape, therefore a large $T$ (or large $m$) is preferred. In standard HMC, it is hard to choose a fixed time scale or mass to work well for both cases.

This paper is organized as follows. In Section 2, we review the standard HMC and summarize several HMC variants that are related to the physics literature. In Section 3, we propose the novel QHMC algorithm that treats mass as a random variable rather than a fixed value. We investigate the convergence properties of QHMC and demonstrate the advantages of QHMC with toy examples where HMC fails. Section 4 explains why treating the mass as a random variable improves the sampling performance for a large class of distributions. In Section 5, we propose quantum stochastic gradient Nosé-Hoover thermostat (QSGNHT) to implement QHMC with massive training data, and prove its convergence based on stochastic differential equations (SDE). In Section 6 we use both synthetic and realistic examples to demonstrate the effectiveness of QHMC and QSGNHT. They can relieve the pain of parameter tuning and achieve superior sampling accuracy for a wide range of distributions, especially for non-smooth and spiky functions which are common in Bayesian learning.
Algorithm 1: Hamiltonian Monte Carlo

**Input:** starting point $x_0$, step size $\epsilon$; simulation steps $L$, mass $M = mI$; initialization.

for $j = 1, 2, \cdots$ do
  Resample $q \sim \mathcal{N}(0, M)$;
  $(x_0, q_0) = (x^{(i)}, q^{(i)})$;
  Simulate dynamics based on Eq. (1);
  $q_0 \leftarrow q_0 - \frac{\epsilon}{2} \nabla U(x_0)$;
  for $i = 1, \cdots, L - 1$ do
    $x_i \leftarrow x_{i-1} + \epsilon M^{-1} q_{i-1}$;
    $q_i \leftarrow q_{i-1} - \epsilon \nabla U(x_i)$
  end
  $x_L \leftarrow x_{L-1} + \epsilon M^{-1} q_{L-1}$;
  $q_L \leftarrow q_{L-1} - \frac{\epsilon}{2} \nabla U(x_L)$;
  $(\dot{x}, \dot{q}) = (x_L, q_L)$;
  M-H step: $u \sim \text{Uniform}[0, 1]$;
  $\rho = e^{-H(x, q) + H(x^{(i)}, q^{(i)})}$;
  if $u < \min(1, \rho)$ then
    $(x^{(i+1)}, q^{(i+1)}) = (\dot{x}, \dot{q})$
  else
    $(x^{(i+1)}, q^{(i+1)}) = (x^{(i)}, q^{(i)})$
  end
end

**Output:** $\{x^{(1)}, x^{(2)}, \cdots\}$

2 Hamiltonian Monte Carlo

In Bayesian inference, we want to sample the parameters $x$ from the posterior distribution $p(x|D)$ given the dataset $D$. Instead of directly sampling $x$, HMC introduces an extra momentum variable $q$, and samples in the joint space of $(x, q)$. By defining a potential function $U(x) = -\log p(x|D)$ and a kinetic energy $K(q) = \frac{1}{2} q^T M_0^{-1} q$ where $M_0$ is a time-independent positive-definite mass matrix, HMC [1][2] samples from the joint density of $(x, q)$ by simulating the following Hamiltonian dynamics:

$$d \begin{pmatrix} x \\ q \end{pmatrix} = dt \begin{pmatrix} M_0^{-1} q \\ -\nabla U(x) \end{pmatrix}$$

The resulting steady-state distribution is

$$\pi(x, q) \propto \exp (-\beta U(x) - \beta K(q)),$$

where $\beta = 1/k_B T$1 and is set as 1 in standard HMC. Because $x$ and $q$ are independent with each other, one can easily marginalize the joint density over $q$ to obtain the invariant distribution in the parameter space $\pi(x) \propto \exp(-U(x)) = p(x|D)$.

The algorithm flow of a standard HMC is summarized in Alg. [1]. The HMC sampler switches between two steps: 1) travel on a constant energy surface according to Hamiltonian dynamics in Eq. (1) with step size $\epsilon$ and number of simulation steps $L$, and 2) maintain state $x$ but resample momentum $q$ to transit to another energy surface. In order to guarantee volume preservation and accuracy of the simulations [17][2], the “leapfrog” integrator is commonly used in HMC. In practice, one needs to choose a step size $\epsilon$ to discretize the continuous time $t$, and the number of simulation steps $L$ to decide how many steps the dynamics runs before resampling the momentum.

However, the performance of a standard HMC can degrade due to the following limitations:

---

1In statistical physics, $k_B$ is the Boltzmann constant, $T$ is the temperature.
Table 1: Summary of various HMC methods.

<table>
<thead>
<tr>
<th>Challenges to address</th>
<th>HMC variants</th>
<th>Physic theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ill-conditioned</td>
<td>Riemannian HMC [22]</td>
<td>General relativity</td>
</tr>
<tr>
<td>distribution</td>
<td>Magnetic HMC [18]</td>
<td>Electromagnetism</td>
</tr>
<tr>
<td></td>
<td>Wormhole HMC [21]</td>
<td>General relativity</td>
</tr>
<tr>
<td></td>
<td>Tempered HMC [23]</td>
<td>Thermodynamics</td>
</tr>
<tr>
<td>Multimodal</td>
<td>Stochastic Gradient HMC [19]</td>
<td>Langevin dynamics</td>
</tr>
<tr>
<td></td>
<td>Thermostat HMC [24]</td>
<td>Thermodynamics</td>
</tr>
<tr>
<td></td>
<td>Relativistic HMC [20]</td>
<td>Special relativity</td>
</tr>
<tr>
<td>Large training data</td>
<td>Quantum-inspired HMC (this work)</td>
<td>Quantum mechanics</td>
</tr>
<tr>
<td>set</td>
<td>Options [17]</td>
<td></td>
</tr>
<tr>
<td>Discontinuous</td>
<td>Options HMC [7]</td>
<td>Optics</td>
</tr>
<tr>
<td>energy function</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spiky &amp; multimodal</td>
<td>Quantum-inspired HMC (this work)</td>
<td>Quantum mechanics</td>
</tr>
<tr>
<td>distributions</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Ill-conditioned distribution: the isotropic mass in HMC assumes the target distribution has the same scale in all directions, so HMC is poor at exploring ill-conditioned distributions, e.g., a Gaussian distribution with a covariance matrix whose eigenvalues have a wide spread.
- Multimodal distribution: although the momentum variable can help the particle reach higher energy levels than MCMC, it is still hard for HMC to explore multimodal distributions because of the fixed temperature \( (k_B T = 1) \).
- Stochastic implementation: standard HMC involves full gradient computations over the whole data set. This can be very expensive when dealing with big-data problems.
- Non-smooth distribution: HMC needs computing gradients of the energy function, therefore it can behave badly or even fail for non-smooth (discontinuous or spiky) functions.

Interestingly, some physical principles have been employed to overcome some aforementioned limitations of HMC. Specifically, magnetic HMC [18] can efficiently explore multimodal distributions by introducing magnetic fields; Stochastic gradient HMC improves the efficiency of handling large data set by simulating the second-order Langevin dynamics [19]. The relativistic HMC [20] achieves better convergence rate due to analogy between speed of light and gradient clipping; The reflection and refraction HMC [7] leverages optics theory to efficiently sample from discontinuous distributions; the wormhole HMC [21] applies Einstein’s general relativity to build shortcuts between two modes; Inspired by general relativity, the Riemannian HMC [22] can adapt mass matrix according to geometries of the Riemannian manifold, in this way attacking the ill-conditioned distributions. Table 1 has summarized some representative HMC algorithms and their corresponding physical models.

In this paper, inspired by quantum mechanics, we construct a stochastic process with time-varying mass matrix with the goal to sample efficiently from a possibly spiky distribution. Our method is readily combined with other methods due to its efficiency (nearly no extra cost) and flexibility.

3 Quantum-Inspired Hamiltonian Monte Carlo

In this section, we propose the physical model and numerical implementation of a quantum-inspired Hamiltonian Monte Carlo (QHMC). In Section 3.2 and 3.3, we prove that its resulting steady-state distribution and time-averaged distribution are indeed the targeted posterior distribution \( p(x|D) \). In Section 3.4, we discuss a few possible extensions of QHMC.

3.1 QHMC Model and Algorithm

Different from HMC that simulates the Hamiltonian dynamics of a constant mass \( M_0 \), our QHMC allows \( M(t) \) to be time-varying and random. Specifically, we construct a stochastic process and let each realization be a mass \( M(t) \):

\[
M(t) = m(t)I, \quad m(t) = 10^{\omega(t)}, \quad \text{with} \quad \omega(t) \sim \mathcal{N}(\mu_m, \sigma_m^2).
\]

In general, \( M(t) \) can be sampled from any general distribution. In Section 3.1, we will explain why we choose the mass as in (3). After obtaining a realization of time-varying mass \( M(t) \), we simulate the following dynamical system:

\[
d \begin{pmatrix} x \\ q \end{pmatrix} = dt \begin{pmatrix} M(t)^{-1} q \\ -\nabla U(x) \end{pmatrix}
\]

(4)
where we assume that \( U(x) \) a differentiable potential energy function with a well-defined gradient \( \nabla U(x) \) everywhere, except for some points with zero measure. The discretized version of Eq. (3) and (4) is

\[
\begin{align*}
\omega_i & \sim \mathcal{N}(\mu_m, \sigma_m^2), m_i = 10^{\omega_i}, M_i = m_i I \\
q_{i+\frac{1}{2}} & \leftarrow q_i - \frac{\epsilon}{2} \nabla U(x_i) \\
x_{i+1} & \leftarrow x_i + \epsilon M_i^{-1} q_{i+\frac{1}{2}} \\
q_{i+1} & \leftarrow q_{i+\frac{1}{2}} - \frac{\epsilon}{2} \nabla U(x_{i+1}).
\end{align*}
\]

The resulting numerical algorithm flow is shown in Alg. [2]. The implementation of QHMC is rather simple: compared with HMC, we only need an extra step of resampling the mass matrix. It is worth noting that HMC can be regarded as a special case of QHMC: QHMC becomes a standard HMC when \( \sigma_m = 0 \).

**Physical Intuition:** In standard HMC [1, 2] described by Eq. (1), the mass matrix \( M \) is usually chosen as diagonal such that \( M = m I \) and further the scalar mass \( m \) is commonly set as 1. Such a choice corresponds to the classical physics where the mass is scalar and invariant. In the quantum physics, however, the energy and time obeys the following uncertainty relationship:

\[ \Delta E \Delta t \approx \hbar. \]  

Here \( \Delta E \) is the uncertainty of the static energy for a particle, and \( \Delta t \) is seen as length of time the particle can survive (lifetime), and \( \hbar \approx 1.05 \times 10^{-34} \) is the Planck constant. Moreover, based on the well-known mass-energy relation discovered by Albert Einstein

\[ E = mc^2, \]

we know that the mass of a particle is proportional to its static energy. Combining Eq. (6) and (7), we conclude \( \Delta m \approx \hbar / (c^2 \Delta t) \), indicating that a quantum particle can have a mass distribution \( P_m(m) \) in the time domain, or equivalently, each realization of the stochastic process can produce a time-varying mass \( m = m(t) \). In quantum mechanics, the mass distribution \( P_m(m) \) is only dependent on the types of the quantum particles, therefore we assume that \( P_m(m) \) is independent of \( x \) and \( q \) in this manuscript. Without this assumption, a wrong distribution may be produced, which will be shown in Section [3.3]

### 3.2 Steady-State Distribution of QHMC (Space Domain)

Now we show that both the continuous-time stochastic process in Eq. (4) and the discrete version in Eq. (5) and Alg. [2] can produce a correct steady distribution that describes the desired posterior density \( p(x|D) \sim \exp(-U(x)) \). In Lemma [1], we first prove that any time-modifier \( A(t) \) (the meaning of which will be clear later) in Eq. (8) preserves the correct steady distribution.

**Lemma 1.** Consider a time-dependent continuous Markov process

\[
d\begin{pmatrix} x \\ q \end{pmatrix} = dt \begin{pmatrix} A(t) & 0 \\ 0 & A(t) \end{pmatrix} \begin{pmatrix} q \\ -\nabla U(x) \end{pmatrix} 
\]

where \( A(t) \) is a deterministic time-varying symmetric matrix. For general symmetric \( A(t) \), we have a steady distribution for Eq. (8) as \( p_s(x, q) \propto \exp(-U(x) - q^T q/2) \). Further if the momentum is resampled every time interval \( t_0 \) as \( p(q) \propto \exp(-q^T q/2) \), then the steady distribution is unique.

\[ ^{2} \]

A set of points on the \( x \)-axis is said to have measure zero if the sum of the lengths of intervals enclosing all the points can be made arbitrarily small. The extension to higher dimensions is straightforward.

\[ ^{3} \]

A classical particle differs from a quantum particle in many aspect. For instance, a classical particle has an infinite lifetime, deterministic position and momentum, and continuous energy; in contrast, a quantum particle has a finite lifetime, uncertain position and momentum, and discrete energy levels (for bound states). In this paper we only focus on the mass-changing nature of a quantum particle.
We consider the specific density function $\rho A$.

**Algorithm 2: Quantum Hamiltonian Monte Carlo**

**Input:** starting point $x_0$, step size $\epsilon$,
simulation steps $L$, and mass distribution
parameters $\mu_m$ and $\sigma_m$
initialization;

for $j = 1, 2, \ldots$ do
  Resample $\omega \sim \mathcal{N}(\mu_m, \sigma_m^2)$, $m = 10^2$, $M = mI$;
  Resample $q \sim \mathcal{N}(0, M)$;
  $(x_0, q_0) = (x^{(i)}, q^{(i)})$;
  Simulate dynamics based on Eq. (2);
  $q_0 \leftarrow q_0 - \frac{1}{2} \nabla U(x_0)$;
  for $i = 1, \ldots, L - 1$ do
    $x_i \leftarrow x_{i-1} + \epsilon M^{-1} q_{i-1}$;
    $q_i \leftarrow q_{i-1} - \epsilon \nabla U(x_i)$;
  end
  $x_L \leftarrow x_{L-1} + \epsilon M^{-1} q_{L-1}$;
  $q_L \leftarrow q_{L-1} - \frac{1}{2} \nabla U(x_L)$;
  $(\bar{x}, \bar{q}) = (x_L, q_L)$;
  M-H step: $u \sim \text{Uniform}[0, 1]$;
  $\rho = e^{-H(\bar{x}, \bar{q}) + H(x^{(t)}, q^{(t)})}$;
  if $u < \min(1, \rho)$ then
    $(x^{(t+1)}, q^{(t+1)}) = (\bar{x}, \bar{q})$
  else
    $(x^{(t+1)}, q^{(t+1)}) = (x^{(t)}, q^{(t)})$
  end
end

**Output:** $\{x^{(1)}, x^{(2)}, \ldots \}$

**Proof.** The evolution of probability density $p(x, q)$ for the particles $(x, q)$ in Eq. (8) can be described by the following Fokker-Planck equation:

$$\partial_t p + \frac{dx}{dt} \cdot \nabla_x p + \frac{dq}{dt} \cdot \nabla_q p = 0. \tag{9}$$

We consider the specific density function $p_u(x, q) \propto \exp(-U(x) - q^T q/2)$. By setting $dx/dt = A(t)q$, $dq/dt = -A(t)\nabla U(x)$ according to Eq. (8) and noticing that $\nabla_x p_s = -p_s \nabla U(x)$ and $\nabla_q p_s = -p_s q$, we have

$$\partial_t p_s = 0. \tag{10}$$

Therefore $p_u(x, q) \propto \exp(-U(x) - q^T q/2)$ is a stationary distribution of the time-dependent process described by Eq. (8). If the momentum is resampled for some time interval, the steady distribution will not change and the Markov process will be guaranteed as ergodic, because the sufficient conditions for ergodicity are (1) irreducibility (2) aperiodicity and (3) positive recurrence, the first two of which are guaranteed by the resampling step as diffusion noise. Condition (3) is met when $U(x) \rightarrow \infty$ for $|x| \rightarrow \infty$ which is a reasonable assumption for energy functions. Once the Markov process is ergodic, the steady distribution should be unique.

The physical interpretation of $A(t)$ can be understood by looking at a special case $A(t) = a(t)I$. In this case $a(t)$ can be absorbed into $dt$ as $dt' = a(t)dt$. So the role of $a(t)$ is to increase or shrink the time scale (hence the name “time-modifier”). The steady distribution is by definition independent of time, thus $a(t)$ does not change the steady distribution. Note that Eq. (8) becomes a standard time-independent Hamiltonian dynamics with $M = I$ when $A(t) = I$.

---

The Fokker-Planck equation is a generic tool to analyze the time evolution of the density function for the unknown state variables in a stochastic differential equations (SDE). Although Eq. (8) is a deterministic process instead of a stochastic one, it falls into the framework of SDE as a special case with a zero diffusion term. In statistical mechanics, the special deterministic case is referred to as the Liouville’s theorem.

---
In the following theorem, we show that employing a (deterministic) time-varying mass matrix $M(t)$ is equivalent to employing a (deterministic) time-modifier $A(t)$.

**Theorem 1.** Consider a continuous-time Hamiltonian dynamics with a deterministic time-varying positive-definite matrix $M(t)$ in Eq. (1). The time-dependent distribution $p(x, q, t) \propto \exp(-U(x) - \frac{1}{2}q^T M(t)^{-1} q)$ satisfies the Fokker-Planck equation of Eq. (4). Furthermore, the marginal density $p_s(x) \propto \exp(-U(x))$ is a unique steady-state distribution in the $x$ space if momentum resampling steps $p(q) \propto \exp(-\frac{1}{2}q^T M(t)^{-1} q)$ are included.

**Proof.** We change variables from $(x, q)$ to $(x', q')$ with the transformation:
\[
\begin{align*}
    x' &= x \\
    q' &= M(t)^{-1/2} q
\end{align*}
\]
where $M^{-1/2}$ is defined as $VD^{-1/2}V^T$ with the diagonalization of $M$ as $M = VDVT$. After changing variables, Eq. (4) is transformed to
\[
    d \left( \frac{x'}{q'} \right) = dt \left( M(t)^{-1/2} 0 \right) \left( M(t)^{-1/2} \right)^{-1} \left( -\nabla U(x') \right)
\]
Because $M(t)^{-1/2}$ is a symmetric matrix, according to Lemma 1, we have a unique steady distribution for $(x', q')$ such that $p_s'(x', q') \propto \exp(-U(x') + q'^T q'/2)$. After transforming $(x', q')$ back to the $(x, q)$ space, the distribution $p_s(x, q, t) \propto \sqrt{\det(M(t))} \exp(-U(x) + q^T M(t)^{-1} q/2)$ is dependent on $t$, where $\det(M(t))$ arises from the Jacobian of variable transformation:
\[
    p_s(x, q, t) = p'_s(x', q', t) \det \left( \frac{\partial x}{\partial x'} \frac{\partial q}{\partial q'} \right) \propto \sqrt{\det(M(t))} \exp(-U(x) + q^T M(t)^{-1} q/2).
\]
After integrating over the momentum $q$ we obtain the marginal probability density $p_s(x) \propto \exp(-U(x))^\frac{1}{2}$ which is again independent of $t$ hence a steady distribution.

The intuition of Theorem 1 is as follows. The time-varying matrix $M(t)$ has an effect of increasing or shrinking the time scale, but it does not change the steady distribution in the $x$ space. As a corollary of Theorem 1, we show the steady distribution of our discretized QHMC algorithm indeed is the desired posterior distribution.

**Corollary 1.1.** Consider a piecewise $M(t)$. The continuous time is divided into pieces as $t_{n-1} < t_n < t_{n+1} < \cdots$ and on each time interval $M$ is a constant matrix i.e. $M(t) = M_n$ if $t_n \leq t < t_{n+1}$. By defining $M_n = m_n I$, $m_n = 10^{\omega_n}$, $\omega_n \sim N(\mu_m, \sigma_m^2)$, we come to the case in the Alg. 2. Each interval has the same and correct steady distribution $p_s(x) \propto \exp(-U(x))$.

**Equivalence between changing time step size and changing mass:** The step size $\epsilon$ in Eq. (5) and Alg. 2 is used to discretize the continuous time variable, and it has no analogy from the physical world. Actually Theorem 1 shows that there exists an close relation between mass $M(t)$ and the time-modifier $A(t)$ [step size $\epsilon$ is the discretized version of $A(t)$]. The key idea is summarized as follows: we have $M(t) \sim A(t)^{-2}$ (in Theorem 1) or $m \sim \epsilon^{-2}$ (see the discussion in the next paragraph). Therefore, a larger step size corresponds to a smaller mass (light particle), and a smaller step size corresponds to a larger mass (heavy particle). As a result, having a time-varying step size is equivalent to having a time-varying mass.

To gain more insights why $m$ is equivalent to $\epsilon^{-2}$ in the scalar case, we need to show $m \sim \epsilon^{-2}$ for both Hamiltonian dynamics and the momentum resampling step. For the Hamiltonian dynamics described by Eq. (1), we point out a ‘scaling property’ such that: given a trajectory $t(x(t), q(t))$ for a particle with mass $m$, the rescaled trajectory $t \to \alpha t$, $q \to \alpha q$, $x \to x$ for a particle with mass $m \to m^2$ is just the original trajectory but with different time scale, because the rescaled trajectory also satisfies Eq. (1). As for the momentum resampling step, the momentum distribution is proportional to $\exp(-q^T q/2m)$, so the scaling $q \to \alpha q$, $m \to \alpha^2 m$ will leave the probability density invariant. As a result, original dynamics is equivalent to the rescaled dynamics, only with the rescaled $m$ and rescaled time $\epsilon$.

\[^{1}\]Here $\det(M(t))$ is a scaling factor independent of $x$ and $q$, and it vanishes after normalization.
3.3 Time-Averaged Distribution (Time Domain)

Rather than simulating a set of particles simultaneously, HMC-type methods simulate one particle and collect a set of samples in a sequence in time points. Therefore, the time-averaged distribution $p_t(x)$, rather than the space-averaged distribution (steady distribution, denoted as $p_\infty(x)$), is more immediate and essential to the sampling procedure. Here the time-averaged distribution is defined as:

$$p_t(x) = \begin{cases} 
\lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T} \mathbb{1}(x(t) = x) & \text{(discrete)} \\
\lim_{T \to \infty} \frac{1}{T} \int_{0}^{T} \delta(x(t) - x) \, dt & \text{(continuous)}
\end{cases} \quad (14)$$

Where $\mathbb{1}(x = x)$ is the indicator function (1 if the argument in the bracket is true and 0 otherwise), and $\delta(x(t) - x)$ is the Dirac delta function ($+\infty$ if the number in the bracket is 0 and 0 otherwise, plus a normalization criteria as $\int_0^T \delta(x(t) - x) \, dx = 1$)\footnote{One can show that $p_t(x)$ is a probabilistic density function because $\sum_x p_t(x) = 1$ or $\int p_t(x) \, dx = 1$.}. The physical meaning of $p_t(x)$ corresponds to drawing a histogram of samples obtained in the time sequence in HMC and $T$ is the number of simulation paths. In most HMC methods where the step size or mass is fixed, the equivalence between $p_t(x)$ and $p_\infty(x)$ can be justified with ergodicity theory from the mathematical perspective \cite{27,29} or ensemble theory from the physical perspective \cite{30,31}. Also Eq. (1) has the steady distribution $p_\infty(x) = p(x|\mathcal{D})$. As a result, both $p_t(x)$ and $p_\infty(x)$ can produce the correct posterior distribution $p(x|\mathcal{D})$. However for QHMC, the step size or mass is effectively modified by $A(t)$ (c.f. Theorem 1 and Lemma 1), or more generally $A(x, \mathbf{q}, t)$. As a result, it is non-trivial to obtain the equivalence between $p_t(x)$ and $p(x|\mathcal{D})$ in QHMC.

Observations. We first illustrate the above point by considering the piecewise energy function

$$U(x) = \begin{cases} 
-x & (x < 0) \\
3x & (x \geq 0).
\end{cases} \quad (15)$$

Because $U(x)$ has larger gradients in $x \geq 0$ than $x < 0$, one may want to change the mass (and thus the step size) in the QHMC simulation. We consider two different schemes:

- **Explicit mass adaption.** One may use a small mass $m = 0.1$ (or equivalently a large step size) for $x < 0$ and $m = 1$ for $x > 0$. However, this explicit adaptation can lead to a wrong time-averaged distribution. As shown in Fig. 2(a), the region $x > 0$ has more samples than required by the true posterior distribution. This is because a small step size results in more accepted samples than the ground truth \footnote{In this example, one is able to produce the true posterior distribution with an explicit step size adaption as $\epsilon(x) \sim |\nabla U(x)|^{-1}$ by fixing the time interval. However, this method is generally less efficient than QHMC: QHMC makes a proposal every $L$ steps, whereas this explicit mass-adaption method does not have a proper upper bound for the number of steps. For instance, in very spiky regions ($|\nabla U(x)|$ very large), it takes $O(|\nabla U(x)|)$ steps to make a new proposal.}.

- **Implicit mass adaption.** The mass is a random variable with a Bernoulli distribution: we have an equal probability to choose either $m = 0.1$ or $m = 1$. This choice is independent of $x$, and it produces the correct time-averaged distribution as shown in Fig. 2(b).

Our QHMC method employs the implicit mass adaption. The benefits of this implicit method will be theoretically justified in Section 4. Here we first make the aforementioned intuitions more rigorous by proposing Theorem 2.

**Theorem 2.** Denote $\mathbf{z} = (x, \mathbf{q})$, and consider the trajectory $\mathbf{z}(t)$ described by the dynamics \footnote{One can show that $p_t(x)$ is a probabilistic density function because $\sum_x p_t(x) = 1$ or $\int p_t(x) \, dx = 1$.}:

$$d\begin{pmatrix} x \\ \mathbf{q} \end{pmatrix} = dt \begin{pmatrix} A(\mathbf{z}, t) & 0 \\ 0 & A(\mathbf{z}, t) \end{pmatrix} \begin{pmatrix} \mathbf{q} \\ -\nabla U(x) \end{pmatrix} \quad (16)$$

The only difference between Eq. (16) and Eq. (8) is that $A(\mathbf{z}, t)$ replaces $A(t)$. The time-averaged distribution of the particle can be defined as

$$p_t(\mathbf{z}) = \lim_{T \to \infty} \frac{1}{T} \int_{0}^{T} \delta(\mathbf{z}(t) - \mathbf{z}) \, dt \quad (17)$$
We restrict our discussions to the case $\mathbf{A}(\mathbf{z}, t) = a(\mathbf{z}, t)\mathbf{I}$. We denote $p(\mathbf{z}|\mathcal{D}) = p(\mathbf{x}|\mathcal{D})p(\mathbf{q}) = p(\mathbf{x}|\mathcal{D})\exp(-\frac{1}{2}q^T \mathbf{q}/2)$. For the special case (implicit mass adaptation) $a(\mathbf{z}, t) = a(t)$, we have $p_t(\mathbf{z}) = p(\mathbf{z}|\mathcal{D})$. However, $p_t(\mathbf{z}) \neq p(\mathbf{z}|\mathcal{D})$ for the general case (in an explicit mass adaptation) $a(\mathbf{z}, t) \neq a(t)$.

**Proof.** Our goal is to calculate $p_t(\mathbf{z})$ and compare it with $p(\mathbf{z}|\mathcal{D})$. The key idea of proof includes two steps:

- **Step 1:** Change the time variable from $t$ to $t'$ and define $p_{t'}(\mathbf{z})$. We find $p_{t'}(\mathbf{z}) = p(\mathbf{z}|\mathcal{D})$.
- **Step 2:** Establish the relation between $p_t(\mathbf{z})$ and $p_{t'}(\mathbf{z})$. We find $p_t(\mathbf{z}) \propto E_t(\frac{1}{a(\mathbf{z}, t)})p_{t'}(\mathbf{z})$.

**Step 1:** In Eq. (16) we can define a new time variable $t'$ such that $dt' = a(\mathbf{z}, t)dt$, and rewrite the trajectory $\mathbf{z}(t)$ as $\mathbf{z}'(t')$. With the new time variable $t'$, the original time-varying Eq. (16) becomes the standard HMC dynamics in (1). Therefore, we have $p_{t'}(\mathbf{z}) = p(\mathbf{z}|\mathcal{D})$ where $p_{t'}(\mathbf{z})$ is defined as

$$p_{t'}(\mathbf{z}) = \lim_{T' \to \infty} \frac{1}{T'} \int_0^{T'} \delta(\mathbf{z}'(t') - \mathbf{z}) dt'.$$

**Step 2:** First we note

$$\int_0^T \delta(\mathbf{z}(t) - \mathbf{z}) dt = \sum_{i=1}^{N_T(\mathbf{z})} \frac{1}{|\mathbf{z}(t_i) = \mathbf{z}|} \left| \frac{d\mathbf{z}}{dt}(t_i) \right|,$$

where $N_T(\mathbf{z})$ counts the number of $t_i \leq T$ where $\mathbf{z}(t_i) = \mathbf{z}$. Consequently, both $p_t(\mathbf{z})$ and $p_{t'}(\mathbf{z})$ can be rewritten as

$$p_t(\mathbf{z}) = \lim_{T \to \infty} \frac{1}{T} \sum_{i=1}^{N_T(\mathbf{z})} \frac{1}{|\mathbf{z}(t_i) = \mathbf{z}|} \left| \frac{d\mathbf{z}}{dt}(t_i) \right|,$$

$$p_{t'}(\mathbf{z}) = \lim_{T' \to \infty} \frac{1}{T'} \sum_{i=1}^{N_{T'}(\mathbf{z})} \frac{1}{|\mathbf{z}'(t'_i) = \mathbf{z}|} \left| \frac{d\mathbf{z}'}{dt'}(t'_i) \right|.$$

We further note

$$\left| \frac{d\mathbf{z}}{dt}(t_i) \right| = \frac{dt'}{dt} \left| \frac{d\mathbf{z}'}{dt'}(t'_i) \right| = a(\mathbf{z}(t_i), t_i) \left| \frac{d\mathbf{z}'}{dt'}(t'_i) \right|.$$

Combining Eq. (20), (21) and (22), we have

$$p_t(\mathbf{z}) = \lim_{T' \to \infty} \frac{1}{T'} \sum_{i=1}^{N_{T'}(\mathbf{z})} \frac{1}{\left| \frac{d\mathbf{z}'}{dt'}(t'_i) \right|} \left| \frac{d\mathbf{z}'}{dt'}(t'_i) \right| \propto E_t(\frac{1}{a(\mathbf{z}, t)})p_{t'}(\mathbf{z}) = E_t(\frac{1}{a(\mathbf{z}, t)})p(\mathbf{z}|\mathcal{D}).$$

Figure 2: (a) Explicit mass adaptation can lead to wrong time averaged distribution; (b) Implicit mass adaptation produces the correct time averaged distribution.
Where the last equality holds due to \( p_t(z) = p(z|D) \). When \( a(z, t) = a(t) \), \( E_t(1/a(z, t)) \) is independent of \( z \), and \( p_t(z) = p(z|D) \). Otherwise, \( p_t(z) \neq p(z|D) \) in general when \( a(z, t) \) explicitly depends on \( z \).

In the QHMC algorithm (Alg. 3), we set \( a(x, q, t) = a(t) \), so the equivalence between \( p_t(x) \) and \( p(x|D) \) is proved by Theorem 2 with marginalizing over \( q \). By contrast, explicit mass adaptation has \( p_t(x) \neq p(x|D) \) in general. In practical simulations, only a finite number of samples are obtained (i.e. \( T \) is finite). The asymptotic distribution error for a general ergodic Markov process has an exponential decay rate (i.e. linear convergence rate) \([29]\), such that \( |p_t(x) - p(x|D)|_\infty \leq C|\rho|^t \) where \( p_t(x) \) is the finite-time sample distribution, \( \rho < 1 \) and \( C \) are constant. Therefore, the obtained sample distribution \( p_t(x) \) can be arbitrarily close to the true posterior distribution \( p(x|D) \) as long as the stochastic process is simulated for long enough time.

### 3.4 Discussions: Combine QHMC with Other Tricks

The proposed QHMC method has barely extra costs compared with the standard HMC: for each path of Hamiltonian simulation, we only need to resample the scalar mass \( m \) according to \( \log_{10} m \sim \mathcal{N}(\mu_m, \sigma_m^2) \) and set the mass matrix as \( M = m I \) before resampling the momentum variable \( q \). Due to the ease of implementation, QHMC can be easily combined with other state-of-the-art techniques, such as the Riemannian manifold method \([22]\), the continuous tempering technique \([23]\), and the “no U-Turn” sampler \([32]\). QHMC alone can handle ill-conditioned high-dimensional problems, but we can also combine it with the preconditioning mass method \([22]\). The combination is straightforward: firstly we determine the mass matrix calculated from the curvature of the potential energy, and then perturb each diagonal element by some uncertainties that obey a distribution. Furthermore, although QHMC can handle multimodal distributions, we may further improve its performance by combining QHMC with other tricks such as the continuous tempering HMC \([23]\), magnetic HMC \([18]\) and wormhole HMC \([21]\).

### 4 Implicit Mass Adaption in QHMC

The mass is set as a random parameter in our QHMC method. In this section, we explain how this treatment can benefit the Bayesian sampling in various scenarios.

#### 4.1 Light Particles for Handling Smooth Energy Functions

While using heavy particles (and small step size) can produce more accurate simulation results and higher acceptance rates of a Hamiltonian dynamics, the simulation suffers from random walking behavior and low mixing rates. Here we show that the mass should be appropriately small in order to sample a distribution in a broad region when the associated energy function is very smooth.

We consider a continuous energy function \( U(x) \) in \( x \in \mathbb{R}^d \) and with \( \beta \)-smoothness:

\[
||\nabla U(x) - \nabla U(y)||_2 \leq \beta||x - y||_2
\]  
(24)

We further consider an HMC simulation with the leapfrog scheme\(^8\):

\[
x_{n+1} = x_n + \epsilon M^{-1} q_n
\]

\[
q_{n+1} = q_n - \epsilon \nabla U(x_{n+1})
\]  
(25)

For simplicity we assume a scalar mass parameter \( m \) such that \( M = m I \).

In the following, we qualitatively measure the random walking effects with different choices of mass parameters. We first investigate a quadratic energy function in Lemma 2.

**Lemma 2.** For the quadratic energy function \( U(x) = \frac{1}{2} x^T A x \) (with \( A \) being positive definite), the discrete dynamical system in Eq. 25 has bounded trajectories if and only if \( m \geq \frac{\epsilon^2}{\beta} \lambda_{\text{max}} \), where \( \lambda_{\text{max}} \) is the largest eigenvalue of \( A \).

---

\(^8\)Here we have ignored the half-step momentum updates at the beginning and end of each simulation path.
Proof. One can easily find an orthogonal transformation \( \mathbf{V} \) to diagonalize \( \mathbf{A} \) such that \( \mathbf{D} = \mathbf{V}^T \mathbf{A} \mathbf{V} \). Therefore, without loss of generality we assume that \( \mathbf{A} \) is a diagonal matrix with diagonal elements being \( \lambda_i \) \((i = 1, 2, \cdots, d)\). By eliminating \( q_m \) in (25) we have the second-order recurrence relation:

\[
x_{n+2} - 2x_{n+1} + x_n + \frac{c^2}{m} \nabla U(x_{n+1}) = 0.
\] (26)

Because \( U(x) = \frac{1}{2}x^T \mathbf{A} x \), the Eq. (26) becomes \( x_{n+2} + (\frac{c^2}{m} \mathbf{A} - 2\mathbf{I})x_{n+1} + x_n = 0 \). Because \( \mathbf{A} \) is diagonal, all the components are decoupled such that the equation of \( i \)-th component can be written as \( x_{n+2}(i) + (\frac{c^2}{m} \lambda_i - 2)x_{n+1}(i) + x_n(i) = 0 \). This discrete system is stable if and only if \( 2 \geq \frac{c^2 \lambda_i}{m} - 2 \geq -2 \), implying \( m \geq \frac{c^2 \lambda_i}{4} \) for \( i = 1, 2, \cdots, d \). As a result, \( m \geq \frac{c^2 \lambda_{\text{max}}}{4} \).  

Next we consider a more general smooth and continuous energy function.

Theorem 3. Assume \( U(x) \) is a differentiable and continuous energy function with \( \beta \)-smoothness, and assume the mass parameter \( m > \frac{\beta \epsilon^2}{4} \). Without loss of generality we assume \( \nabla U(x^*) = 0 \) for \( x^* = 0 \), and we denote \( c = (\frac{\beta \epsilon^2}{2m} - 1) \in (-1, 2) \). Lemma 2 implies that the sequence \( \{x^*_n\} \) generated by \( x^*_{n+2} + c x^*_{n+1} + x^*_n = 0 \) is bounded, i.e. \( ||x^*_n|| < \Delta \). There exist constants \( \Delta \) and \( \lambda \), such that \( ||x_n|| < \Delta \) for all \( x_n \) \((n < N_0)\) generated by Eq. (26). Here \( \Delta \) only depends on the initial condition \( x_0 \), and \( N_0 \) is given by

\[
N_0 = \log_\lambda(\frac{\Delta}{\Delta_0}), \quad \lambda = 4 + \sqrt{17}
\] (27)

Proof. Given the \( \beta \)-smoothness condition, we have \( ||\nabla U(x) - \nabla U(x^*)|| \leq \beta ||x - x^*|| \) thus \( ||\nabla U(x)|| \leq \beta ||x|| \). The sequences \( \{x^*_n\} \) and \( \{x_n\} \) are obtained as

\[
x^*_{n+2} = -c x^*_{n+1} - x^*_n, \\
x_{n+2} = -(\frac{c^2}{m} \nabla U(x_{n+1}) - 2x_{n+1}) - x_n.
\] (28)

Taking the difference between the above two equations we get

\[
x_{n+2} = (c x^*_{n+1} + x^*_n + x^*_{n+2}) - (\frac{c^2}{m} \nabla U(x_{n+1}) - 2x_{n+1}) - x_n.
\] (29)

Denote \( \Delta_n = ||x_n|| \) we have

\[
\Delta_{n+2} = ||x_{n+2}|| \leq ||c x^*_{n+1} + x^*_n + x^*_{n+2}|| + \frac{c^2}{m} ||\nabla U(x_{n+1})|| + 2 ||x_{n+1}|| + ||x_n|| + \Delta_n + \Delta_{n+1} < 4\Delta_0 + 8\Delta_{n+1} + \Delta_n.
\] (30)

In the extreme case, \( \Delta_n \) grows exponentially as \( \lambda^n \) where \( \lambda = 4 + \sqrt{17} \). Therefore, there should exist a constant \( A \) which only depends on the initial conditions, satisfying that

\[
\Lambda_n < A \Delta_0 \lambda^n
\] (31)

By enforcing \( A \Delta_0 \lambda^n < \Delta \) for all \( n < N_0 \), we immediately have Eq. (27).

Following Theorem 3, now we provide a necessary condition for efficiently sampling the posterior density associated with a smooth energy function.

Corollary 3.1. A sampler can explore the smooth energy function efficiently, if it can find a path from \( ||x_0|| \leq \Delta_0 \) to \( ||x_i|| \geq \Delta \) with no more than \( N_0 \) steps. A necessary condition is

\[
m \leq \frac{\beta \epsilon^2}{6}
\] (32)

This implies that the sampler cannot explore the energy function efficiently if the mass is above a certain bound, which is linear with respect to \( \beta \).
Here “spiky” means a distribution whose energy function \( U(x) \) has a very large (or even infinite) gradient around some points. Fig. 3 left shows an energy function \( U(x) \) with a large gradient at \( x = 0 \). A representative family of spiky distributions is \( \exp(-\lambda \| x \|_p^p) \) with \( p \in (0,1) \), which is widely used as a prior density for sparse modeling and model selection \([33, 34, 14, 12, 11, 34]\). Here the \( \ell_p \) norm is defined in a loose sense as \( \| x \|_p = \left( \sum_{i=1}^{d} |x_i|^p \right)^{1/p} \). As shown in Fig. 3 middle, \( \| x \|_p \) is non-convex and has divergent gradients around \( x = 0 \) when \( 0 < p < 1 \), causing troubles in traditional HMC samplers. Sampling from such a spiky distribution is a challenging task in Bayesian learning \([9, 10]\).

The Hamiltonian dynamical system \([1]\) becomes stiff if the energy function \( U(x) \) has a large gradient at some points. This issue can cause unstable numerical simulations. A naive idea is to adapt the step size based on the local gradient \( \nabla U(x) \). Unfortunately, this explicit step-size tuning is equivalent to using a state-dependent time modifier \( a(x, t) \), and will produce a wrong time-averaged distribution \( p_t(x) \neq p(x|D) \), as proved in Theorem 2. In contrast, the implicit and stochastic mass adaptation in our QHMC is equivalent to using a state-independent time modifier \( a(t) \), and it ensures the produced time-averaged distribution converging to the desired posterior density \( p(x|D) \).

The advantage of the implicit adaptation strategy in QHMC can be illustrated with a toy spiky distribution with the following potential energy function \( U(x) \):

\[
U(x) = \begin{cases} 
U_1(x) = |x| + 999x_0 & \text{if } |x| > x_0, \\
U_2(x) = 1000|x| & \text{if } |x| \leq x_0.
\end{cases}
\]
When a particle has momentum uncertainty \( \Delta \beta \) Alg. 2. In order to adapt to widespread 
although \( \ell \) matters with \( \mu \) and \( \sigma^2 \) choices in HMC:

4.3 Explore Multimodal Energy Functions: Quantum Tunneling Effects

Finally, we show that the random mass distribution can help sample a multimodal distribution because it can manifest the “quantum tunneling” effect.

The “quantum tunneling” effect describes that a microscopic particle can climb over a potential peak even if its total energy is below that peak, which is very different from the case in classical mechanics. In quantum mechanics, a particle should be treated as wave permeable in the whole space rather than a localized object. As a result, the particle (the wave) always has a non-zero probability to climb over the peak. The larger step the particle takes, the more likely a quantum tunneling will happen.

Now we analyze the quantum tunneling effect of our QHMC method in a semi-quantitative way. When a particle has momentum uncertainty \( \Delta q \), it should also have a real space uncertainty \( \Delta x \) in quantum mechanics such at \( \Delta x \Delta q \sim \hbar \), where \( \hbar \approx 1.05 \times 10^{-34} \) Js is the Planck’s constant and represents the unit of time in the quantum world. In QHMC, the step size \( \epsilon \) plays the role of time unit, so we instead have \( \Delta x \Delta q \sim \epsilon \). The momentum variable \( q \) has a distribution \( \propto \exp(-q^2/2m) \) at the thermal equilibrium point, therefore we have momentum uncertainty \( \Delta q \sim \sqrt{m} \) and position

Figure 5: The “quantum tunneling” effects for QHMC. (a) In HMC, a relatively large mass (equivalently, small step size) is chosen to guarantee accurate Hamiltonian simulations. However, the particle can be trapped within one well. (b) In QHMC, the mass is random, and there is chance that one uses a small mass in one path. The light particle has a higher chance to jump to another well through “tunneling”.

Here \( U(x) \) can be a simple yet good approximation to an \( \ell_p \) function, as shown on the right of Fig.3. The spiky and smooth region of \( U(x) \) can be effectively separated by \(-x_0\) and \( x_0\). We care if both regions can be sampled accurately. When \( x_0 = \log(1001)/1000 \), it can be shown that the probability of \( x \) being located in either region is 0.5, therefore \( U_1(x) \) and \( U_2(x) \) are equally important for sampling and we study them independently. Because \( U_1(x) \) and \( U_2(x) \) are both symmetric, their associated mean values should be zero, and we measure the numerical performance of a sampler by estimating the error bars obtained from 20 independent experiments. Suppose that we have two mass choices in HMC: \( m = 1 \) or \( m = 0.01 \). In QHMC, we adapt the mass implicitly by allowing \( m = 1 \) and \( m = 0.01 \) with an equal probability. Because \( U_1(x) = |x| + 999|x_0| (|x| > x_0) \) is very smooth, a small mass \( m = 0.01 \) is preferred, as shown in Fig.4(a). Although QHMC slightly underperforms the HMC implemented with \( m = 0.01 \), it significantly outperforms the HMC implemented with \( m = 1 \). On the other hand, \( U_2(x) = 1000|x| \) has a very large gradient, therefore a relatively large mass \( m = 1 \) is preferred, as shown in Fig.4(b).

Similarly, QHMC has slightly worse performance than the HMC implemented with \( m = 1 \), but it performs much better than the HMC implemented with \( m = 0.01 \). In summary, the HMC cannot explore efficiently \( U_1 \) and \( U_2 \) simultaneously no matter \( m = 1 \) or \( m = 0.01 \), however our QHMC with an implicit mass adaptation can have excellent performance in both regions.

Remark: Adapting Mass in an Exponential Scale. Corollary 3.1 implies that one needs to choose a small mass \( m \leq \frac{\hbar^2}{m} \) to efficiently explore a smooth distribution characterized by \( \beta \)-smoothness. However, \( \beta \)-smoothness remains a valid description for spiky distributions, only with very large \( \beta \). Although \( \ell_p (0 < p < 1) \) function has no global \( \beta \)-smoothness, one can define \( \beta \) locally. In the example above, for \( x > x_0 \) or \( x < -x_0, \beta = 1 \); for \(-x_0 < x < x_0, \beta = 1000 \). The wide spread of \( \beta \) from 1 to 1000 justifies our choice of a widespread mass distribution as \( \log m \sim N(\mu_m, \sigma^2_m) \) in Alg. 2. In order to adapt to widespread \( \beta \), we should adapt \( m \) in an exponential scale rather than in a linear scale.
uncertainty $\Delta x \sim \epsilon/\Delta q = \epsilon/\sqrt{m}$. When $m$ is small the particle is less likely to be trapped in a single well. Fig. 5(b) shows the intuition of the quantum tunnelling effect.

5 Stochastic-Gradient Implementation

5.1 Quantum Stochastic Gradient Nosé-Hoover Thermostat (QSGNHT)

Similar to HMC, the proposed QHMC method suffers from a high computational cost when the training data size is huge. Consider a machine learning problem with $N$ training samples, the loss (energy) function $U(x)$ is commonly defined as the average loss over all training samples: $U(x) = \frac{1}{N} \sum_{i=1}^{N} U_i(x)$, where $U_i(x)$ depends only on the $i$-th training sample. Calculating the full gradient $\nabla U(x)$ needs computation over every training sample. Instead, one may replace the true loss with a stochastic estimation $\tilde{U}(x)$, and the stochastic gradient is computed efficiently with only a small batch of samples $\nabla \tilde{U}(x) = \frac{1}{b} \sum_{i=1}^{b} \nabla U_i(x)$. Here $b$ is called the batch size. However, the mini-batch estimation of gradients will introduce extra noise. According to the central limit theorem, we can approximate the stochastic gradient as the true gradient plus a Gaussian noise with covariance $\mathcal{N}(0, \sigma_i^2)$.

However, such a naive stochastic gradient implementation can result in incorrect steady distribution, and one can add a friction term to compensate for the extra noise in a stochastic-gradient HMC [19]. Different from the friction formulation in [19], we utilize the thermostat technique [24, 35] to correct the steady distribution. Specifically, we treat the gradient uncertainty term $\xi$ as a noise with an unknown magnitude, and use the Nosé-Hoover thermostat to avoid the explicit estimation of this gradient noise term. The resulting update rule for $(x, q, \xi)$ is shown in Eq. (34) with step size $\epsilon$:

$$
\begin{align*}
\omega_i &\sim \mathcal{N}(\mu_i, \sigma_{m_i}^2), m_i = 10^{\omega_i}, M_i = m_i I, \\
x_{i+1} &\leftarrow x_i + \epsilon M_i^{-1} q_i, \\
qu_{i+1} &\leftarrow q_i - \epsilon \nabla \tilde{U}(x_{i+1}) - \epsilon \xi q_i + \sqrt{2A} m_i^{\frac{1}{2}} \mathcal{N}(0, \epsilon I) \\
\xi_{i+1} &\leftarrow \xi_i + \frac{\epsilon}{m_\mu} (q_{i+1}^T M_i^{-2} q_{i+1} - d_0 T m_i^{-\frac{1}{2}})
\end{align*}
$$

where $A$ indicates the magnitude of injected noise, $m_\mu$ is the thermal mass term, $d_0$ is the dimension of $x$, and $T$ is in the definition of the energy function $U(x) = -T\log(p(x|D))$. We set $m_\mu = 1$ (like in [35]), $A = 1$ and $T = 1$. We refer to the proposed method with thermostat as quantum stochastic gradient Nosé-Hoover thermostat (QSGNHT). The algorithm flow of QSGNHT is shown in Alg. 3.

5.2 Theoretical Analysis based on Stochastic Differential Equation (SDE)

In this subsection, we prove that the above QSGNHT implementation in Alg. 3 indeed produces the desired posterior density $p(x|D)$. Setting $M(t) = m(t) I$, we first provide the continuous-time stochastic differential equation (SDE) for QSGNHT:

$$
\begin{align*}
\begin{pmatrix} d\mathbf{x} \\ d\mathbf{q} \\ d\xi
\end{pmatrix} =
\begin{pmatrix}
\mathbf{0} \\ -\frac{1}{m}(\mathbf{q}^T m^{-2} \mathbf{q} - m(t)^{-\frac{1}{2}} d_0) \\ \sqrt{2A} m(t)^{-\frac{1}{2}} dW(t)
\end{pmatrix}.
\end{align*}
$$

(35)

The gradient noise term is dominated by the injected wiener process $W(t)$, therefore it can be ignored in this continuous-time formulation, as in [35, 24].

The major challenge in the theoretical analysis is that the mass matrix $M(t)$ is time-varying. Therefore, before presenting our result in Theorem 3, we review the result for a general continuous-time Markov process in Lemma 4 and study a general time-dependent system in Lemma 4.

Lemma 3. (Theorem 1 in [35]) A general continuous-time Markov process can be written as a stochastic differential equation (SDE) in this form:

$$
d\mathbf{z} = \mathbf{f}(\mathbf{z}) dt + \sqrt{2\mathbf{D}(\mathbf{z})} dW(t)
$$

(36)
Algorithm 3: Quantum Stochastic Gradient Nosé-Hoover Thermostat (QSGNHT)

**Input**: starting point $x_0$, step size $c$, simulation steps $L$, and mass distribution parameters $\mu_m$ and $\sigma_m$, thermal mass $m_p$, batch size $b$

dimension of $x$ is $d$, temperature $T$, diffusion strength $A$
initialization;

for $i = 1, 2, \ldots$ do

Randomly select $b$ samples out of the $N$ training samples and compute $\tilde{U}(x) = \frac{1}{b} \sum_{i=1}^{b} U_i(x)$;

Resample $\omega_i \sim \mathcal{N}(\mu_m, \sigma_m^2)$, $m_i = 10\omega_i$, $M_i = m_i I$;

Resample $q \sim \mathcal{N}(0, m_i T)$;

$(x_0, q_0) = (x^{(i)}, q^{(i)})$;

Simulate dynamics based on Eq. (34):

$\tilde{q} \leftarrow q_0 - \frac{c}{2} \nabla \tilde{U}(x_0)$;
for $j = 1, \ldots, L - 1$ do

$x_{j+1} \leftarrow x_j + c M_i^{-1} q_j$;
$q_{j+1} \leftarrow q_j - c \tilde{U}(x_{j+1}) + \sqrt{2 AM_i^2} \mathcal{N}(0, c I)$;
$\xi_{j+1} \leftarrow \xi_j + \frac{c}{m_p} (q_{j+1}^T M_i^{-2} q_{j+1} - dT m_i^{-2})$;
end

$q_L \leftarrow q_L - \frac{c}{2} \nabla \tilde{U}(x_L)$;
$(x, q) = (x_L, q_L)$;

M-H step: $u \sim \text{Uniform}[0, 1]$;

Define $H(x, q) = \tilde{U}(x) + \frac{1}{2} q^T M_i^{-1} q$;

$\rho = e^{-H(x, q) - H(x^{(i)}, q^{(i)})}$;

if $u < \min(1, \rho)$ then

$(x^{(t+1)}, q^{(t+1)}) = (x, q)$
else

$(x^{(t+1)}, q^{(t+1)}) = (x^{(i)}, q^{(i)})$
end
end

Output: $\{x^{(1)}, x^{(2)}, \ldots\}$

where $z$ can be a general vector and $D(z)$ represents the magnitude of the Wiener diffusion process. Then, $p_s(z) \propto \exp(-H(z))$ is a steady distribution of the above SDE if $f(z)$ can be written as:

$$f(z) = - (D(z) + Q(z)) \nabla H(z) + \Gamma(z), \quad \Gamma(z) = \sum_{i=1}^{d} \frac{\partial}{\partial z_i} (D_{ij}(z) + Q_{ij}(z))$$ (37)

where $H(z) = U(x) + g(x, q)$ is the Hamiltonian of the system, $U(x) = - \log p(x|D)$ is the potential energy, $g(x, q) = \frac{1}{2} q^T M(t)^{-1} q/2$ is the kinetic energy, $Q(z)$ determines the deterministic transverse dynamics, $D(z)$ is positive semidefinite, and $Q(z)$ skew-symmetric. The steady distribution $p_s(z)$ is unique if $D(z)$ is positive definite, or if ergodicity\footnote{The ergodicity of a Markov process requires the coexistence of irreducibility, aperiodicity and positive recurrence. Intuitively, ergodicity means that every point in the state space can be hit within finite time with probability one.} can be shown.

**Proof**: The proof is based on [36]. According to Eq. (36) we have a corresponding Fokker-Planck equation to describe the evolution of the probability density:

$$\partial_t p(z, t) = - \sum_i \frac{\partial}{\partial z_i} (f_i(z)p(z, t)) + \sum_{i,j} \frac{\partial^2}{\partial z_i \partial z_j} (D_{ij}(z)p(z, t))$$ (38)

Eq. (38) can be further written in a more compact form:

$$\partial_t p(z, t) = \nabla^T \cdot ([D(z) + Q(z)][p(z, t) \nabla H(z) + \nabla p(z, t)])$$ (39)
We are able to verify $p_s(x, q, \xi) \propto \exp(-H(x, q, \xi))$ is invariant under Eq. \[(38)\] by calculating $e^{-H(x)} \nabla H(z) + \nabla e^{-H(x)} = 0$. If the process is ergodic, then the stationary distribution is unique. The ergodicity of the Markov process requires three conditions: (a) irreducibility (b) aperiodicity (c) positive recurrence. Irreducibility and aperiodicity can be guaranteed by non-zero diffusion noises, while positive recurrence is satisfied if $U(x) \to \infty$ when $|x| \to \infty$. □

**Lemma 4.** Consider the following time-dependent thermostat with a positive definite matrix $B(t)$:

$$
\frac{d}{dt} \begin{pmatrix} x \\ q \\ \xi \end{pmatrix} = B(t) \begin{pmatrix} x \\ q \\ \xi \end{pmatrix} + \begin{pmatrix} -\nabla U(x) - \xi q \\ \frac{1}{m_\mu}(q^T B(t)q - d) \end{pmatrix} + \begin{pmatrix} 0 \\ \sqrt{2AB(t)}W(t) \end{pmatrix}
$$

(40)

where $d$ is the dimension of $x$. The thermostat SDE have the unique steady distribution $p_s(x, q, \xi) \propto \exp(-U(x) - q^T q/2 - m_\mu(\xi - A)^2/2)$.

**Proof.** Compare Eq. (40) with Eq. (36) and (37) we can determine the corresponding Hamiltonian and coefficient matrices $D$ and $Q$:

$$
H(x, q, \xi) = U(x) + \frac{1}{2}q^T q + \frac{m_\mu}{2}(\xi - A)^2
$$

$$
D(x, q, \xi) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & AB(t) & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad Q(x, q, \xi) = \begin{pmatrix} 0 & -B(t) & B(t)q/m_\mu \\ 0 & -q^T B(t)/m_\mu & 0 \\ 0 & 0 & 0 \end{pmatrix}
$$

(41)

It is clear that $D(x, q, \xi)$ is positive semi-definite and $Q(x, q, \xi)$ is skew-symmetric. By denoting $z = (x, q, \xi)$, we know that there exists a steady distribution $p_s(z) \propto \exp(-H(z)) = \exp(-U(x) - q^T q/2 - m_\mu(\xi - A)^2/2)$. Due to non-zero diffusion error $A > 0$ in the system, $p_s(x, q, \xi)$ is the unique steady distribution. □

**Theorem 4.** For the continuous-time QSGNHT as described in Eq. (35), the unique steady distribution is $p_s(x) \propto \exp(-U(x))$.

**Proof.** We change variables from $(x, q, \xi)$ to $(x', q', \xi')$ with the relation:

$$
\begin{align*}
x' &= x \\
qu' &= m(t)^{-\frac{1}{2}}q \\
\xi' &= m(t)^{\frac{1}{2}}\xi.
\end{align*}
$$

(42)

As a result, Eq. (35) is transformed to

$$
\frac{d}{dt} \begin{pmatrix} x' \\ q' \\ \xi' \end{pmatrix} = \begin{pmatrix} m(t)^{-\frac{1}{2}}I & 0 & 0 \\ 0 & m(t)^{-\frac{1}{2}}I & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} q' \\ -\nabla U(x') - \xi' q' \\ \frac{1}{m_\mu}(q'^T m(t)^{-\frac{1}{2}} q' - d) \end{pmatrix} + \begin{pmatrix} 0 \\ \sqrt{2Am(t)^{-\frac{1}{2}}}dW(t) \end{pmatrix}
$$

(43)

According to Lemma4 by letting $B(t) = m(t)^{-1/2}I$, we know that there exists a steady distribution $p_s(x', q', \xi') \propto \exp(-U(x') - q'^T q'/2 - m_\mu(\xi' - A)^2/2)$. Furthermore, due to non-zero diffusion term in the SDE, $p_s(x', q', \xi')$ is the unique steady distribution. By marginalization over $q'$ and $\xi'$ and noticing $x' = x$, we have $p_s(x) \propto \exp(-U(x))$ which is also unique. □

### 6 Numerical Experiments and Applications

This section verifies our proposed QHMC method by several synthetic examples, and shows its applications in some machine learning tasks such as sparse bridge regression, image denoising, and neural network pruning. The number of simulation steps and the step size are set as $L = 5$ and $\epsilon = 0.03$ in QHMC, if not stated explicitly otherwise\[10\]. Our codes are implemented in MATLAB and Python, and all experiments are run on a computer with 4-core 2.40 GHz CPU and 8.0G memory. Codes are available at [https://github.com/KindXiaoming/QHMC](https://github.com/KindXiaoming/QHMC)

\[10\] We use a small $L$ and a large $\epsilon$ due to the efficiency consideration. Changing mass in QHMC is equivalent to implicitly changing $\epsilon$ and $L$ in HMC, therefore $\epsilon$ and $L$ can be chosen with lots of freedom.
6.1 Synthetic Examples

6.1.1 One-Dimensional $\ell_p$ Norm

The $\ell_p$ \((0 < p < 1)\) norm can be used as a regularizer in an optimization problem for model selection and to enforce model sparsity \([11, 12]\). In a Bayesian setting, employing an $\ell_p$ regularization is equivalent to placing a prior density $\exp(-\|x\|_p^p)$. 
We consider the flat energy function $U$ when the median mass value, which is already in the typical set with different mass values; (d)-(f) QHMC results obtained with different distributions for the mass parameter.

In this experiment, we use QHMC to sample from a 1-D distribution $p(x) \propto \exp(-|x|^p)$ whose corresponding potential energy function is $U(x) = |x|^p$ for $p = \{1, 0.5, 0.1\}$. We start from $x_0 = 0.1$, which is already in the typical set, therefore burn-in steps are not needed. Results for $p = 1, 0.5, 0.1$ are presented in Fig. 6 to Fig. 8 respectively. One can see that as $p$ approaches zero, the advantages of QHMC over HMC becomes increasingly significant.

We compare HMC and QHMC implemented with different mass parameters. Take the results of $p = 0.5$ as an example (see Fig. 7). In each row, the mass value in HMC is set as the the median mass value of QHMC. Fig. 7(a)-(c) show that the distribution can be well sampled by HMC only if the mass value is properly chosen. The region around $x = 0$ can be hardly explored when the mass value is too small, as shown in Fig. 7(a). In Fig. 7(c), the whole distribution cannot be well explored due to the random walk caused by a large mass. However, the proposed QHMC method does not suffer from this issue. As shown in Fig. 7(d)-(f), the QHMC can produce accurate sample distributions when the median mass value $10^{\alpha}$ changes from $10^{-3}$ to $10^{3}$.

6.1.2 One-Dimensional Distribution with Both Smooth and Spiky Regions

Section 4 states that different mass magnitudes are preferred for smooth and spiky regions. In this example, we consider the following piecewise (non-normalized) potential energy function:

$$U(x) = \begin{cases} 
-x - 3 & \text{if } x \leq -3 \\
0 & \text{if } -3 < x \leq 0 \\
8x(x - 1) & \text{if } 0 < x \leq 1 \\
x - 1 & \text{if } x > 1.
\end{cases}$$ (44)

This function is very flat and smooth in the range $[-3, 0]$ and very spiky in the interval $[0, 1]$.

In order to understand why QHMC outperforms HMC in this synthetic example, we provide a semi-quantitative analysis by introducing the diffusion coefficient $D$ and characteristic length $L \epsilon$. We consider the flat energy function $U(x) = 0$ for $-3 < x < 0$. Suppose that we use a fixed step size $\epsilon$ and $L$ leap-frog steps in each simulation path, and that we run $N$ simulation paths. Because $q_i(i = 1, 2, \cdots, N)$ are drawn from a Gaussian distribution, their sum also has a Gaussian distribution.

The “typical set” is a set that contains almost all elements. For instance, $[-3\sigma, 3\sigma]$ can be considered as a typical for a 1-D Gaussian distribution with standard deviation $\sigma$, because this set contains 99.7% of its elements. In the $\ell_{1/2}$ case, we consider $[-20, 20]$ as the typical set because 98.7% of its elements are in this range.

\[\text{Figure 8: The results of HMC and QHMC for the energy function } U(x) = 20|x|^{1/10}. \text{ The red dashed line is the true distribution, and the blue histogram is the distribution with 50000 simulation samples.} (a)-(c): HMC with different mass values; (d)-(f) QHMC results obtained with different distributions for the mass parameter.\]
We consider a double-well posterior distribution as

\[ p(x) \propto \exp(-U(x)) \]

where the potential energy function \( U(x) = x^4 - 4x^2 \), as shown in Fig. 10(a). We simulate 200 particles starting from the right minimum point \( x_0 = \sqrt{2} \), and we check if each particle has crossed the peak at \( x = 0 \) after every 50 iterations. If yes, then the particle has successfully escaped from the right well. We report the ratio of particles that have escaped from the right wells after certain iterations in Fig. 10(b). We observe that a moderate variance \( \sigma_m \sim 2 \) of the log-mass distribution can lead to significantly better performance.

---

12The characteristic length \( r_c \) can be seen as the size of typical set, e.g. the standard deviation of a 1-D distribution. Here we do not pursue a rigorous definition and just use it for semi-quantitative analysis.
than simply fixing the mass in standard HMC. We also observe that the particles can escape from the right well more quickly once the log-mass distribution has a larger variance.

We further compare the proposed QHMC algorithm with standard HMC and tempered HMC. The HMC with tempering is implemented by setting a low temperature \( T_l = 1 \) and high temperature \( T_h = 25 \) and run a high-temperature step every 30 paths, including 10 paths of burn-in and 20 paths to collect samples. For each method, we run 100 trials of sampling results and sort them by the Wasserstein distance. We show the sample distributions with the 20th smallest Wasserstein distance.

6.2 Application: Sparse Modeling via Bridge Regression

In data mining and machine learning, the following loss function is often minimized

\[
L(\beta) = \frac{\mu}{2\tau} |y - X\beta|^2_F + \lambda |\beta|^p_p = L_{ls}(\beta) + L_{re}(\beta) \tag{45}
\]

in order to learn a model. In Bayesian learning, this is equivalent to a linear model with Gaussian noise and \( \ell_p \) prior. The likelihood function and prior distribution are \( p(D|\beta) \propto \exp(-L_{ls}(\beta)) \) and \( p(\beta) \propto \exp(-L_{re}(\beta)) \) respectively, so the posterior distribution is \( p(\beta|D) \propto p(\beta)p(D|\beta) \propto \exp(-L(\beta)) \). In “bridge regression” [38, 33], the parameter \( p \) is chosen in the range \((0, 1)\) in order to select proper features and to enforce model sparsity. In a Bayesian setting, this is equivalent to placing a prior \( \exp(-\|\beta\|_p^p/\tau) \) over the unknown model parameters \( \beta \).

In this experiment, we consider the case \( p = 1/2 \) and perform bridge regression using the Stanford diabetes dataset [13]. This dataset includes \( n = 442 \) people, 10 attributes (AGE, SEX, BMI, BP, S1–S6) and 1 health indicator (Y). The goal is to select as few attributes as possible but they still

\[\text{https://web.stanford.edu/~hastie/Papers/LARS/diabetes.data}\]
accurately predict the target Y. We split the dataset into a training set with 300 people and a testing set with 142 people. The hyper-parameters µ and λ can be automatically determined if by further introducing some hyper-priors over λ and µ [38, 33]. However, the non-existence of analytical conjugate priors [33] makes updates (Gibbs sampling) of µ and λ only approximate, but not exact. For the sake of simplicity, we utilize a grid search method in the (λ, µ) plane with the testing MSE (mean-squared error) as a criteria to choose the hyper-parameters. We start from β = 0 and use 1000 paths of gradient descent as a burn-in process. After that, we run another 1000 paths of HMC/QHMC and collect the samples. The results are shown in Table 2 for both HMC and QHMC. When the regularization λ is small (λ = 0.1, 1, 10), the difference between HMC and QHMC is insignificant; however, when large regularization is required for very sparse models (e.g. λ = 100 or 1000), QHMC can produce models with higher accuracy.

For the sake of simplicity, we utilize a grid search method in the (λ, µ) plane with the testing MSE (mean-squared error) as a criteria to choose the hyper-parameters. We start from β = 0 and use 1000 paths of gradient descent as a burn-in process. After that, we run another 1000 paths of HMC/QHMC and collect the samples. The results are shown in Table 2 for both HMC and QHMC. When the regularization λ is small (λ = 0.1, 1, 10), the difference between HMC and QHMC is insignificant; however, when large regularization is required for very sparse models (e.g. λ = 100 or 1000), QHMC can produce models with higher accuracy.

We consider the case with λ = 10 and µ = 100, and compare QHMC with standard HMC, NUTS [32] and RHMC [22]. The resulting test MSE and CPU time are reported in Table 3. RHMC is implemented with (X^T X) (with X being the data matrix) being the metric of the Riemannian manifold in [22], and NUTS is implemented based on Alg. 2 in [32]. Although RHMC is an adaptive HMC, it tends to degrade for spiky energy functions. Because the ℓ_p penalty is isotropic (i.e., the regularization coefficients for all attributes are the same), imposing an anisotropic metric renders the sampling performance even worse than HMC. NUTS can achieve comparable test MSE with QHMC, nonetheless it consumes 25× more CPU time than QHMC due to the expensive recursions of building balanced binary trees. Compared with HMC, QHMC has three advantages. Firstly, QHMC is more accurate as shown by the lower test MSE. Secondly, QHMC has a much shorter burn-in phase than HMC, as shown in Fig. 11 (a). This is expected because the possible small masses in QHMC can speed up the burn-in phase. Thirdly, QHMC can produce sparser models than HMC as expected. As shown in Fig. 11 (b), seven attributes have nearly zero mean (except BMI, S3 and S5). The samples from QHMC have smaller variance for these seven attributes, implying a more spiky posterior sample distribution of β.

### 6.3 Application: Image Denoising

In this experiment, we apply QHMC to solve an image denoising problem. Specifically, we consider a two-dimensional gray-level image, but the extension to high-order tensors is available [39]. We employ the robust low-rank matrix factorization model [40] which models a corrupted image Y as the sum of a low-rank matrix L = AB, a sparse matrix S describing outliers and some i.i.d. Gaussian noise. Standard robust PCA uses ℓ_1-norm to enforce the sparsity of S. Instead, we employ ℓ_p norm with p = 1/2 because it is closer to the exact sparsity measurement ℓ_0-norm. As a result, we can define the potential energy function as follows:

\[
U(A, B, S) = L_g + L_i + L_n = \frac{1}{2} \mu ||Y - AB - S||_F^2 + \frac{1}{2} \lambda_1(||A||_F^2 + ||B||_F^2) + \lambda_2 ||S||_p^p \quad (46)
\]
Figure 11: (a) The log testing error as a function of the iteration step. The number of steps required by burn-in process in QHMC (∼ 500 steps) is much smaller than that in HMC (∼ 2000 steps). (b) Distribution of all attributes in the diabetes dataset using QHMC, HMC, NUTS and RHMC with μ = 100 and λ = 10. QHMC is better at sampling sparser models than standard HMC (QHMC produces smaller variance in coefficients) and has smaller MSE on the testing data.

In the Bayesian setting, exp(−Lg) refers to the likelihood function; exp(−L1) refers to the prior density of the low rank part; exp(−Ln) refers to the prior of the salt-and-pepper noise (which is sparse in nature). The gradient of the above loss function with respect to A, B and S are:

\[
\begin{align*}
\frac{\partial U}{\partial A} &= \mu (AB + S - Y)B^T + \lambda_1 A \\
\frac{\partial U}{\partial B} &= \mu A^T (AB + S - Y) + \lambda_1 B \\
\frac{\partial U}{\partial S} &= \mu (AB + S - Y) + \lambda_2 \frac{p_0}{S^{1-p_0} + \epsilon_0} \text{sign}(S).
\end{align*}
\] (47)

In the second term of the gradient with respect to S, all operations are element-wise, and sign(x) is defined as

\[
\text{sign}(x) = \begin{cases} 
1, & x \geq 0 \\
-1, & x < 0.
\end{cases}
\] (48)

The parameters are fine tuned and set as μ = 100, λ1 = 1, λ2 = 10, p = 1/2 and the rank r = 20. It is possible to automatically determine the rank by introducing some hyper-priors (e.g. [41]), but here we focus on the sampling of a given model with a spiky loss function.

We use the proposed QHMC method to sample images from the model, and we compare it with HMC, RHMC [22] and NUTS [32]. In all experiments we initialize the model with Singular Value Decomposition (SVD) of the corrupted image and run 300 iterations as burn-in and another 200 iterations to collect the sample images. In HMC and QHMC, we use three groups of parameter choices for the mass matrix. The metric of the Riemannian manifold [22] in RHMC is chosen as a diagonal matrix whose diagonal elements are the singular values of the corrupted image. The NUTS is implemented based on Alg. 2 in [32].

We show the image denoising results and convergence behaviors of QHMC, HMC, NUTS and RHMC in Fig. 12(a)–(e). We use the peak signal to noise ratio (PSNR) as the measure of denoising
Figure 12: Image denoising example. (a)(b) HMC can achieve good reconstruction only for the second parameters, while QHMC has good performance for all settings and its performance is much less sensitive to the choice of mass parameters. Also, Neither NUTS or RHMC can achieve comparable results with QHMC. (c) For small step size (large mass), QHMC can speed up convergence in the large gradients phase; (d) For proper step size, HMC and QHMC show no significant difference. (e) For large step size (small mass), QHMC can still make progress in terms of decreasing the loss function, while HMC gets stuck at high loss.

Table 4: Image denoising example: PSNR for different algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>HMC with $m = 10^{\mu_m}$</th>
<th>HMC with $m = 10^{\mu_m}$</th>
<th>NUTS</th>
<th>RHMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>23.81</td>
<td>23.92</td>
<td>24.04</td>
<td>22</td>
</tr>
</tbody>
</table>

effects and report the results in Table 4. The standard HMC method is very sensitive to parameter choices, because it can only provide good results with the 2nd group of parameter setting. In contrast, our QHMC method shows good performance for all parameters choices. RHMC achieves similar performance with HMC ($\mu_m = 1$), only decreasing the training loss faster for larger iteration steps. However, RHMC still fails to produce comparable denoising effects with QHMC. NUTS converges very slowly for this example because the “U-Turn” criteria is easily satisfied in the spiky regions. Consequently, the recursions of balanced binary tree terminate quickly in NUTS, leading to low mixing rates.

6.4 Application: Bayesian Neural Network Pruning

Finally we apply the QHMC method to the Bayesian pruning of neural networks. Neural networks have achieved great success in wide engineering applications, but they are often over-parameterized. In order to reduce the memory and computational cost of deploying neural networks, pruning techniques have been developed to remove redundant model parameters (e.g. weight parameters with tiny absolute values).

In this experiment we consider training the following two-layer neural network

$$\hat{y} = \text{softmax}(\text{relu}(W_2\text{relu}(W_1x + b_1) + b_2))$$

(49)

to classify the MNIST dataset. Here “relu” means ReLU activation functions; $W_1 \in \mathbb{R}^{784 \times 200}$ and $W_2 \in \mathbb{R}^{200 \times 10}$ are the weight matrices for the 1st and 2nd fully connected layers; a softmax layer is used before the output layer. The log-likelihood of this neural network is a cross-entropy function. We introduce the $\ell_p$ ($p = 1/2$) priors, $\exp(-\lambda ||\text{Vec}(W_1)||_p)$ and $\exp(-\lambda ||\text{Vec}(W_2)||_p)$
Table 5: Bayesian pruning results of a 2-layer neural network classifier for the MNIST dataset.

<table>
<thead>
<tr>
<th></th>
<th>SGNHT([24])</th>
<th>SGHMC([19])</th>
<th>SGLD([37])</th>
<th>QSGNHT(proposed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>compression rate</td>
<td>47.6 ± 1.3</td>
<td>32.2 ± 1.5</td>
<td>30.8 ± 1.4</td>
<td>54.2 ± 2.1</td>
</tr>
<tr>
<td>testing accuracy (%)</td>
<td>94.4 ± 0.2</td>
<td>95.2 ± 0.2</td>
<td>93.2 ± 0.2</td>
<td>96.6 ± 0.2</td>
</tr>
</tbody>
</table>

Figure 13: Bayesian pruning of a neural network with QSGNHT (proposed), SGNHT, SGHMC and SGLD. Left: testing accuracy; Right: compression rate.

with $\lambda = 0.0001$, for the weight matrices $W_1$ and $W_2$ to enforce their sparsity. The resulting potential energy function is thus

$$U(W_1, W_2, b_1, b_2) = \sum_{i=1}^{N} \sum_{j=1}^{c} -y_{i,c}\log(\hat{y}_{i,c}) + \lambda||\text{Vec}(W_1)||_p^p + \lambda||\text{Vec}(W_2)||_p^p$$

(50)

where $c$ is the number of classes (10 for MNIST), $y$ is the ground truth label vector, and Vec is an operator that vectorizes the matrix. For simplicity we employ uniform prior distributions for $b_1$ or $b_2$. Due to the large training data set (with 60000 samples), we use stochastic gradient implementations with a batch size of 64 in the stochastic gradient Nosé-Hoover thermostat (SGNHT) [24] and in our proposed quantum stochastic gradient Nosé-Hoover thermostat (QSGNHT). In [19] it is shown that a naive implementation of stochastic gradient HMC can produce wrong steady distributions, therefore we utilize the thermostat method [24] to correct the steady distributions, and adopt the QSGNHT algorithm proposed in Section 5. Our proposed QSGNHT is slightly different from SGHMC [19] in the practical implementation. Firstly, we implement the MH step where a Hamiltonian is estimated based on a batch of samples, although no MH steps are used in [19] due to efficiency considerations. Secondly, we set a lower bound $m_0$ for $m$: if a sampled mass $m$ is smaller than $m_0$, then we set $m = m_0$. Both tricks help avoid very large model updates because $m$ can be arbitrarily close to 0 if there is no lower bound for $m$.

We choose $m = 10^1$ for SGNHT and $m \sim 10^{\lambda(1.0,0.5)}$, $m_0 = 1$ for QSGNHT, and run 1000 steps to collect random samples based on Hamiltonian simulations. We set a weight parameter to 0 if its absolute value is below 0.01. The resulting compression rate is measured as:

$$\text{Compression rate} = \frac{\# \text{of all weight parameters}}{\# \text{of parameters after pruning}}$$

(51)

We plot the test accuracy estimated on 10000 test samples and the compression rate as a function of the total number of training batches accessed in gradient evaluations, shown in Fig [13]. Our method is compared with SGNHT [24], SGHMC [19] and SGLD [37]. SGHMC is implemented in the framework of SGNHT with the parameter setting $\xi = 1$ and $m_\mu = \infty$. SGLD is implemented by excluding the momentum in SGHMC and directly updating the network with gradients. We consider the first 800 batches as the burn-in phase and use the simulation samples in last 200 batches. The test accuracy and compression rate for the neural network samples are reported in Table [5]. For test accuracy, our proposed QSGNHT achieves 2% accuracy improvement compared with
SGNHT [24]. SGHMC and SGLD also underperform QSGNHT in terms of test accuracy. Regarding the compression rate, the third-order Langevin dynamics (QSGNHT and SGNHT) outperform the second-order Langevin dynamics (SGHMC) and first-order (SGLD) Langevin dynamics. Besides, the proposed QSGHNT achieves 14% more compression rate than SGNHT.

7 Conclusions

Leveraging the energy-time uncertainty relation in quantum mechanics, we have proposed a quantum-inspired Hamiltonian Monte Carlo method (QHMC) for Bayesian sampling. Different from the standard HMC, our method sets the particle mass as a random variable associated with a probability distribution. We have proved the convergence of its steady state (both in space and in time sequence) to the true posterior density, and have theoretically justified its advantage over standard HMC in sampling from spiky distributions and multimodal distributions. In order to improve the efficiency of QHMC in massive-data scenarios, we have proposed a stochastic-gradient implementation with Nosé-Hoover thermostat terms and proved its theoretical properties. We have verified the effectiveness of our method and demonstrated its advantage over HMC by several synthetic examples. Finally, we have applied our methods to solve several popular Bayesian learning problems including sparse modeling, image denoising and neural network pruning. Our method outperforms HMC, NUTS, RHMC and several stochastic-gradient implementations on these realistic examples. In the future, we plan to develop a deeper theoretical understanding of QHMC and more robust and efficient implementation for large-scale learning problems.

Acknowledgement

This work was partly supported by NSF CCF-1817037 and NSF CAREER Award No. 1846476. The authors would like to thank Kaiqi Zhang, Cole Hawkins and Dr. Chunfeng Cui for their valuable technical suggestions.

References


