Quantum dynamical Hamiltonian Monte Carlo

Owen Lockwood [1,2,*] Peter Weiss,[3] Filip Aronshtein,[3] and Guillaume Verdon[4,5,2,†]

[1]*Department of Computer Science, Rensselaer Polytechnic Institute, Troy, New York 12180, USA*
[2]*Extropic Corp., San Francisco, California 94111, USA*
[3]*Dirac Inc., New York, New York 10001, USA*
[4]*Institute for Quantum Computing, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada*
[5]*Department of Applied Mathematics, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada*

One of the open challenges in quantum computing is to find meaningful and practical methods to leverage quantum computation to accelerate classical machine-learning workflows. A ubiquitous problem in machine-learning workflows is sampling from probability distributions that we only have access to via their log probability. To this end, we extend the well-known Hamiltonian Monte Carlo (HMC) method for Markov chain Monte Carlo (MCMC) sampling to leverage quantum computation in a hybrid manner as a proposal function. Our new algorithm, Quantum Dynamical Hamiltonian Monte Carlo (QD-HMC), replaces the classical symplectic integration proposal step with simulations of quantum-coherent continuous-space dynamics on digital or analog quantum computers. We show that QD-HMC maintains key characteristics of HMC, such as maintaining the detailed balanced condition with momentum inversion, while also having the potential for polynomial speedups over its classical counterpart in certain scenarios. As sampling is a core subroutine in many forms of probabilistic inference, and MCMC in continuously parametrized spaces covers a large class of potential applications, this work widens the areas of applicability of quantum devices.

I. INTRODUCTION

Fueled by the success of machine learning (ML) [1] and recent developments in quantum computing hardware [2–7], substantial interest has developed at the intersection of these fields [8–10]. Sampling from difficult distributions is key to many classical machine-learning workflows. Sampling routines are often leveraged as a critical component of workloads such as Bayesian inference [11], optimization [12], machine learning [13], statistical inference/modeling [14], and energy-based models (EBMs) [15,16], to name a few. Research at the intersection of classical and near-term quantum machine learning has been heavily focused on parametrized quantum circuits [17] for problems such as quantum simulation [18–20], reinforcement learning [21–23], and mathematical applications [24–26]; however, we explore a different direction to accelerate classical machine-learning workflows. Specifically, we investigate the potential for quantum computers to accelerate Hamiltonian Monte Carlo (HMC) [27] proposals. HMC is a continuous parameter-space Markov chain Monte Carlo (MCMC) method that approximates Hamiltonian dynamics to provide improved proposals [28,29].

MCMC methods [30], such as HMC, are some of the most established methods in classical machine learning and provide a general-purpose toolkit for sampling from target probability distributions. Accelerating MCMC with quantum proposals is a research direction that has only just begun [31–37].

As novel physics-based accelerators for probabilistic sampling are on the horizon, it is important to benchmark the performance of quantum computers for this task. In this paper we take a first step towards porting a core sampling algorithm to quantum devices. Here, we propose a method for extending the HMC approach to leverage quantum computation via an algorithm we call Quantum Dynamical Hamiltonian Monte Carlo (QD-HMC). Our algorithm directly builds upon the work of Layden *et al.* [32], in which they proposed the use of quantum simulations to generate MCMC proposals in discrete state spaces, and Verdon *et al.* [38], which proposed a continuous variable (CV) quantum approximate optimization algorithm (QAOA) [39]. CV-QAOA is effectively a variational Trotterization of continuous space dynamics, similar to QD-HMC's randomized Trotterization of dynamics. Our method utilizes quantum computers to more efficiently simulate the Hamiltonian dynamics used for the proposal for HMC. We outline the theory behind this algorithm and present initial small simulations of QD-HMC. Given the classical complexity of simulating quantum dynamics, there is clear potential in using quantum hardware to generate proposals. QD-HMC leverages hybrid quantum computing to effectively create a method for quantum devices to be relevant to probabilistic and Bayesian inference at large scale. We hope that this link will fuel further explorations of scalability and performance for QD-HMC.

*Contact author: owen@extropic.ai
†Contact author: gv@extropic.ai

II. BACKGROUND

There are many cases in physics, machine learning, and optimization in which we do not have access to a desired target distribution, but we do have access to an energy function (or an oracle for the un-normalized negative log likelihood) [40]. To sample from this target distribution we can use MCMC methods, which often requires only the ability to evaluate the log probability (i.e., the energy). Mathematically, we can represent these distributions as a Boltzmann distribution where $P(x) = \frac{1}{\mathcal{Z}} e^{-E(x)/kT}$ with $\mathcal{Z}$ being the partition function $\int e^{-E(x)/kT} dx$. In many cases, such as that of modern deep neural network-based EBMs, kT is just set to 1. One of the earliest and most popular MCMC sampling methods is the Metropolis-Hastings algorithm [41], which achieves sampling from the target distribution by repeatedly alternating between proposal and acceptance steps. In Metropolis-Hasting, a proposal $q(y|x)$ is generated using any one of many possible methods, e.g., random walk, Langevin dynamics [42,43], splines [44], or normalizing flows [45], which is then fed into the acceptance function $A(y|x) = \min[1, \frac{\pi(y)q(y|x)}{\pi(x)q(x|y)}]$. In our notation, $\pi(x) \equiv P(x)$, which represents the target distribution. In many cases, the proposal function is symmetric [i.e., $q(x|y) = q(y|x)$], which eliminates the need for the so called "Hasting's correction" and reduces $A(y|x)$ to $\min[1, \frac{\pi(y)}{\pi(x)}]$. In the case of energy-based models, this becomes $\min[1, e^{E(x)-E(y)}]$.

In continuous high-dimensional state spaces, HMC provides a way to leverage Hamiltonian dynamics and the gradient information of the log probability to simulate the trajectories of particles traversing the energy landscape [46]. Specifically, HMC adds an auxiliary variable momentum p to create a kinetic energy term (in addition to the log probability term defined to be the potential energy). Given the function $\pi(x)$, HMC draws from a joint density $\pi(x, p) = \pi(p|x)\pi(x)$. The Hamiltonian of this joint density is $H = -\ln \pi(x, p) = -\ln \pi(p|x) - \ln \pi(x) = K(x, p) + U(x)$ (i.e., the sum of kinetic and potential energies). By picking a value for the momentum, one can simulate the Hamiltonian dynamics $\frac{dx}{dt} = \frac{\partial H}{\partial p} = -\frac{\partial}{\partial p} \ln \pi(p|x)$ and $\frac{dp}{dt} = -\frac{\partial H}{\partial x} = \frac{\partial}{\partial x} \ln \pi(x)$. Although there are a variety of methods for integrating approximations for these dynamics [47], one of the most common methods is a leapfrog integrator that approximates these dynamics via the following finite difference estimation:

$$
\begin{aligned}
p_{i+1} &= p_i - \frac{\epsilon}{2} \frac{\partial}{\partial x} \ln \pi(x_i), \\
x_{i+1} &= x_i + \epsilon M^{-1} p_{i+1}, \\
p_{i+1} &= p_{i+1} - \frac{\epsilon}{2} \frac{\partial}{\partial x} \ln \pi(x_{i+1}),
\end{aligned}
\tag{1}
$$

where ϵ is the step size and M is the mass matrix (often a multiple of the identity matrix) [48]. This proposal is then fed into the acceptance function $\min[1, \exp(-H(x_{i+1}, p_{i+1}) + H(x_i, p_i))]$.

Since HMC proposals (ideally) conserve energy [47], their acceptance probability can be much higher than other MCMC methods. Naturally, if we could perfectly simulate Hamiltonian dynamics the acceptance probability would be 1 [27]. Since the Hamiltonian dynamics are energy conserving, the acceptance probability would always be $e^0 = 1$. Our aim is to show how these classical dynamics (classically simulated via symplectic leapfrog integration) can be replaced using a quantum computational simulation of dynamics, and how this confers a potential quantum advantage for the task of sampling from distributions in continuous spaces.

III. QUANTUM DYNAMICAL HAMILTONIAN MONTE CARLO

Quantum Dynamical Hamiltonian Monte Carlo (QD-HMC) consists of leveraging digitally simulated or analog continuous quantum dynamics in order to suggest proposals for a Metropolis-Hastings acceptance step in continuous spaces. Let us first outline the steps of the QD-HMC algorithm before demonstrating its theoretical and empirical justifications. This can also be seen in Algorithm 1. We present the algorithm for digitally simulated quantum dynamics here.

QD-HMC Algorithm. First, prepare bitstring $|x\rangle$. With $f(\hat{x})$ as the target distribution, define $\hat{H}_{\eta\lambda} = \eta \frac{\hat{p}^2}{2} + \lambda f(\hat{x})$ with $\hat{p}$ being the momentum and chosen hyperparameters η, λ. Via random Trotterization, apply $\hat{U}_{\eta\lambda} \equiv e^{-i\hat{H}_{\eta\lambda}t}$. Sample bitstring measurement $y \sim |\langle y| \hat{U}_{\eta\lambda} |x\rangle|^2$. Accept with probability $A(y|x) = \min\{1, e^{[-f(x)+f(y)]}\}$.

To understand how QD-HMC relates to traditional HMC, we need to look at the ansatz and how each layer in the simulated evolution maps the position and momenta in the Heisenberg picture. Our algorithm can be seen as a merging of quantum-enhanced MCMC [32] with HMC using the continuous evolution techniques of Verdon *et al.* [38]. The evolution of these Trotter updates are very similar to those of a CV-QAOA, a connection readily seen in the following section.

Suppose the log probability is a function of N variables; $f(\mathbf{x}) : \mathbb{R}^N \mapsto \mathbb{R}$. As a quantum operator, this function becomes part of the Hamiltonian $f(\hat{\mathbf{x}})$, where $\hat{\mathbf{x}}$ is the position operator corresponding to a quadrature of a simulated N-dimensional quantum system. In photonic or analog systems, such quadrature operators could be represented directly, while in digital quantum systems, this quadrature can be represented in the one-hot or binary representation. In this work we follow the convention of Refs. [49,50]. That is to say, we represent a discretized position operator $\hat{x}_d$ over d qubits via

$$
\hat{x}_d = \sqrt{\frac{2\pi}{N}} \left(-\hat{J} + \left(\frac{N}{2} - 1 \right) I \right),
\tag{2}
$$

where $N = 2^d$ and $\hat{J}_d \equiv \sum_{j=0}^{d-1} 2^{(d-1)-j} |1\rangle \langle 1|^{(j)}$ with $|1\rangle \langle 1|^{(j)}$ acting on the jth qubit only. A computational basis state $|k\rangle$ on d qubits is an eigenvector of $\hat{x}_d$ with the eigenvalue $x_k = \sqrt{2\pi/N}(k - N/2)$. The states $|k\rangle$ therefore represent positions on a one-dimensional grid with half-open interval $\sqrt{2\pi}[-\sqrt{N}/2, \sqrt{N}/2)$, and the binary representation of k provides a little-endian description of position with respect to the negative bound of the domain. Using a centered Fourier transform $\hat{F}_c$ [49], defined as $\hat{F}_c = \hat{X}_0 \hat{F} \hat{X}_0$ with $\hat{X}_0$ being the Pauli X gate on the 0th qubit and $\hat{F}$ being the quantum Fourier transform [51], the momentum operator in this space is defined as $\hat{p}_d = \hat{F}_c \hat{x}_d \hat{F}_c^\dagger$ and has corresponding momentum
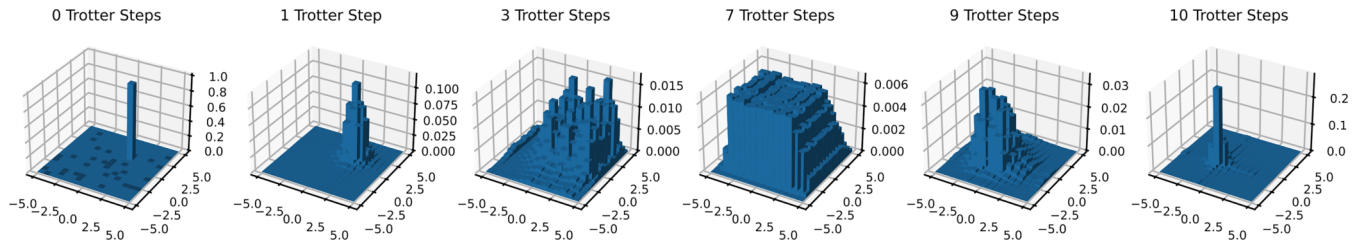
FIG. 1. A QD-HMC update: 10 Trotter Steps showing the evolution of the wave-function probability for a 2D Gaussian $\mu = (0, 0)$.

eigenstates $|\rho_k\rangle = \hat{F}|k\rangle$ with eigenvalues identical to the position eigenvalues.

Now that we have outlined how to represent the function of interest, we can focus on understanding the Hamiltonian $\hat{H}_{\eta\lambda}$. First, let us focus on what is called the kinetic term $\hat{K} = \frac{1}{2}\sum_{j=1}^{N}\hat{p}_j^2 := \frac{1}{2}\hat{p}^2$ [38]. For analog devices, this represents the momentum, or the canonically conjugate variable to the position where we have the commutation relation $[\hat{x}_j, \hat{p}_k] = i\delta_{jk}$. For the binary representation of these operators, we can simply use the fact that momentum is analogous to position in Fourier space. Using a centered Fourier transform $\hat{F}_c$ [49], the momentum operator in this space is thus defined as $\hat{p}_d = \hat{F}_c^\dagger \hat{x}_d \hat{F}_c$.

Next, let us examine the action of each layer in the random Trotterization of the Hamiltonian. In the Heisenberg picture, the kinetic term of the Hamiltonian generates a change in position of the form

$$e^{i\eta\hat{p}^2/2}\,\hat{x}\,e^{-i\eta\hat{p}^2/2} = \hat{x} + \eta\hat{p}, \tag{3}$$

where η is a choice of parameter analogous to inverse mass, $\eta \cong m^{-1}$. With this, we can see that the position gets updated by the momentum divided by mass (i.e., the velocity). Now, for the next step, the momentum operator is translated through evolution under the target Hamiltonian as

$$e^{i\lambda f(\hat{x})}\hat{p}e^{-i\lambda f(\hat{x})} = \hat{p} - \lambda\,\nabla f(\hat{x}), \tag{4}$$

thus the momentum is shifted in a manner proportional to the negative gradient of the target function. The details of these derivations are presented in Appendix A. Evolving under both the target and kinetic Hamiltonians yields

$$\hat{x} \to \hat{x} + \eta\hat{p} - \eta\lambda\,\nabla f(\hat{x}). \tag{5}$$

This is analogous to gradient descent with momentum, or classical kinematics, with $\eta = \Delta t/m$ and $\lambda = \Delta t$ for time-step size Δt. For infinitesimal time steps, alternating between Eq. (3) and Eq. (4), two steps is equivalent to the quantum dynamics of a particle undergoing motion in a high-dimensional potential, i.e., kinematic evolution.

As an illustrative example of the dynamics of the Hamiltonian simulation done by the QD-HMC algorithm, see Fig. 1. For a log probability, or target function, that is a two-dimensional (2D) Gaussian with mean at (0,0) we show the probability of the wave function over 10 trotter steps. The wave function is initialized to a location that is away from the mean and we can see the initial wave function is 100% in the location because every step of QD-HMC is initialized to the single state $|x\rangle$. As the evolution progresses, we see the final result is a wave function localized at the desired mean

(0,0). This is not steps of HMC, but Trotter steps within a single run of our quantum proposal function. Although this evolution is heavily dependent on the number of Trotter steps and the time simulated by these steps ($t = 2$ in this example), this provides intuition and pedagogical insight into how the algorithm proposal step works.

Let us highlight the connection between the above dynamics and the symplectic integration more explicitly. Taking Eq. (1) and looking only at the position update, we can see the leapfrog integrator's x update is

$$x \to x + \epsilon M^{-1}\left(p - \frac{\epsilon}{2}\nabla \ln\pi(x)\right). \tag{6}$$

With $\pi(x) = e^{f(x)}$ we can see this reduces to

$$x \to x + \epsilon M^{-1}p - \frac{\epsilon^2}{2}M^{-1}\nabla f(x). \tag{7}$$

Since the inverse of the diagonal mass matrix is the diagonal matrix of inverses $M_{i,i}^{-1} = (M_{i,i})^{-1}$, and we have established $\eta \approx \frac{\Delta t}{m}$, then we can see these terms are equivalent with $\epsilon = \Delta t$. With $\eta = \frac{\epsilon}{m}$, we can set $\lambda = \frac{\Delta t}{2} = \frac{\epsilon}{2}$. We can now see clearly the similarities of the Hamiltonian dynamics between symplectic leapfrog integration in Eq. (7) and the quantum Hamiltonian dynamics in Eq. (5). We see that we recover the symplectic integrator behavior, except this is done quantum coherently, allowing for exponentially complex superpositions over position values to be simultaneously symplectically integrated.

Now that we have illustrated the mechanisms of Hamiltonian simulation, we will outline how QD-HMC functions as a Markov chain Monte Carlo (MCMC) algorithm. First, we show that our QD-HMC algorithm meets the requirement of conserved energy. In the classical case this can be easily shown via $\frac{dH}{dt} = \sum_i[\frac{\partial H}{\partial x_i}\frac{dx_i}{dt} + \frac{\partial H}{\partial p_i}\frac{dp_i}{dt}] = \sum_i[\frac{\partial H}{\partial x_i}\frac{\partial H}{\partial p_i} - \frac{\partial H}{\partial p_i}\frac{\partial H}{\partial x_i}] = 0$ [48]. In the quantum case, we can use inspiration from the Ehrenfest theorem to show our quantum Hamiltonian conserves energy. As was shown in Appendix A, $\frac{d\hat{x}}{dt} = \eta\hat{p} = \frac{\partial\hat{H}}{\partial\hat{p}}$ and $\frac{d\hat{p}}{dt} = -\lambda\nabla f(\hat{x}) = -\frac{\partial\hat{H}}{\partial\hat{x}}$. Thus, following these derivations, $\frac{d\hat{H}}{dt} = \sum_i[\frac{\partial\hat{H}}{\partial x_i}\frac{dx_i}{dt} + \frac{\partial\hat{H}}{\partial p_i}\frac{dp_i}{dt}] = 0$, since this becomes identical to the classical case. Using these same derivatives, it is straightforward to show the quantum Hamiltonian dynamics preserve volume since the divergence of the vector field $\nabla \cdot \mathbf{F}$, $\mathbf{F} = (\hat{x}, \hat{p}) \to (\frac{\partial\hat{H}}{\partial\hat{p}}, -\frac{\partial\hat{H}}{\partial\hat{x}})$ is 0 (this is the same approach as in Ref. [48]).

Next, a key requirement of MCMC algorithms is reversibility, i.e., it must satisfy the detailed balance condition. This condition states that for a stationary distribution π and
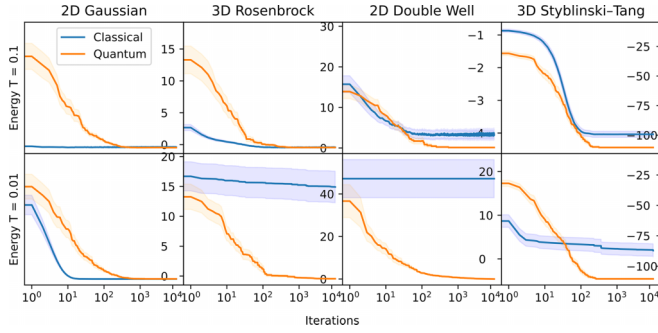
FIG. 2. Comparison of QD-HMC and HMC samples energies over 10 000 iterations on a variety of optimization problems for temperature 0.1 and 0.01.

FIG. 3. Estimation of autocorrelation time τ as a function of number of samples for a double well.

transition probability $p_{i,j} = P(x_{t+1} = j | x_t = i)$,

$$\pi_i p_{i,j} = \pi_j p_{j,i} \quad \forall i, j. \tag{8}$$

We show that QD-HMC obeys this condition by proving the symmetry of the proposal function $|\langle y | \hat{U}_{\eta\lambda} | x \rangle|^2 = |\langle x | \hat{U}_{\eta\lambda} | y \rangle|^2$ for our algorithm. The proof is presented in Appendix B. Since our proposal is symmetric, we can augment it with a classical acceptance/rejection step which meets the detailed balance condition. In HMC, one needs to add a sign flip to the momentum term to ensure reversibility. Intuitively, the Hamiltonian dynamics only go "forward," so $q(x|y) \to 0$, which means proposals would never be accepted. Adding a sign flip on the momentum, $(x, p) \to (x, -p)$ enables this reversibility (but is of no practical importance since the momentum is resampled), see Ref. [28, sec 5.4] for more information. Analogous to this, there is a momentum flip necessary at the end of our quantum proposal to ensure reversibility, but as with HMC is of no practical relevance due to the random Trotterization. Having established the theory of QD-HMC, we now present some example experiments to outline how the algorithm functions.

IV. SIMULATIONS & EXPERIMENTS

We evaluate our algorithm on a number test functions [52] as is common in optimization literature [53,54]. The QD-HMC and classical comparisons are generated from implementations built on TensorFlow Probability [55], TensorFlow Quantum [56] and Continuous Variable TensorFlow Quantum [57]. All code is available in Ref. [58]. Note that all results come from noiseless and exact state-vector simulations. The results of these experiments can be seen in Fig. 2, which compares the energy over 10 000 proposals on a set of functions for low temperatures $T = 0.1$ and 0.01. We add this temperature so the adjusted log probability becomes $\ln p(x)/T$ (i.e., the same way temperature is included in the Boltzmann distribution). The x axis is scaled logarithmically. For each function, we used Optuna [59] to optimize the classical and quantum hyperparameters independently. It is important to emphasize that these empirical results are not meant to suggest that QD-HMC is universally better than HMC, or that QD-HMC should be used (with simulations) as a replacement. These results are purely a proof of concept and of intuitive, explanatory, and pedagogical interest.
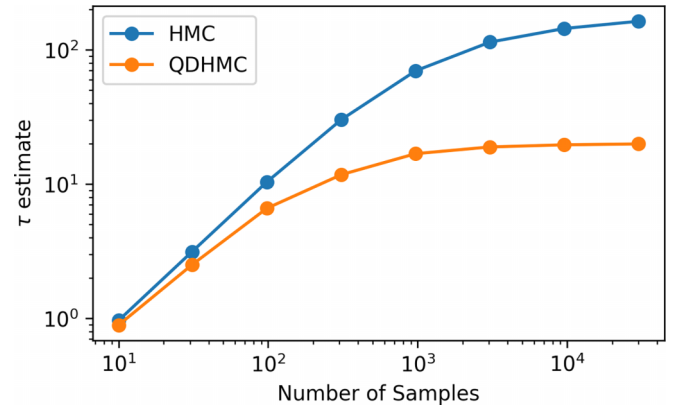
There are many HMC improvements that could likely do even better on these problems [60,61]. The log probability for each function are defined as followed; Gaussian: $\sum_i -x_i - x_i^2$, Rosenbrock: $-\sum_i 10(x_{i+1} - x_i)^2 + (1 - x_i)^2$, double well: $-(x_0^4 - 4x_0^2 + x_1^2) - 0.5x_0$, and Styblinski-Tang: $-\frac{1}{2}\sum_i x_i^4 - 16x_i^2 + 5x_i$. These show the minimization of the negative log probability (energy) function (labeled at the top) for low temperatures, making this energy a good proxy for free energy. Although the results are comparable at higher temperatures with HMC performing very well, the low-temperature results highlight the potential of quantum dynamical simulation. The initial points for optimization are selected randomly across the range of possible digital quantum representations (which is often far from the minimum, something as a qualitative point HMC sometimes struggles with overcoming). These results are averaged over 50 repetitions with different initial points.

To further understand the advantages QD-HMC might yield, we also estimate the autocorrelation time across these problems for QD-HMC and HMC. The integrated autocorrelation time τ is estimated for a finite chain observable $\{f_i\}_{i=1}^N$ via [62,63]

$$\tau = \frac{1}{2} + \sum_{t=1}^{\infty} \rho_f(t),$$
$$\rho_f(t) = \frac{N \sum_{i=1}^{N-t}(f_i - \mu_f)(f_{i+t} - \mu_f)}{(N - t)\sum_{i=1}^N (f_i - \mu_f)^2}. \tag{9}$$

We plot an example of the estimation of autocorrelation time as a function of samples in Fig. 3. The plots for the other problems are available in Appendix C. These experiments were all performed (and averaged) over the 2D version of functions and were conducted at $T = 5.0$. As these experiments reveal, QD-HMC is able to achieve lower autocorrelation times, which means samples become independent faster and can result in improved sampling convergence rates. Although there are a multitude of methods when it comes to evaluating MCMC samplers, autocorrelation time is a common and important foundation. For example, it forms the backbone of effective sample size (ESS) estimation (ESS is proportional to $\frac{1}{\tau}$). As before, these results are not meant to offer strong and universal claims of quantum advantage but
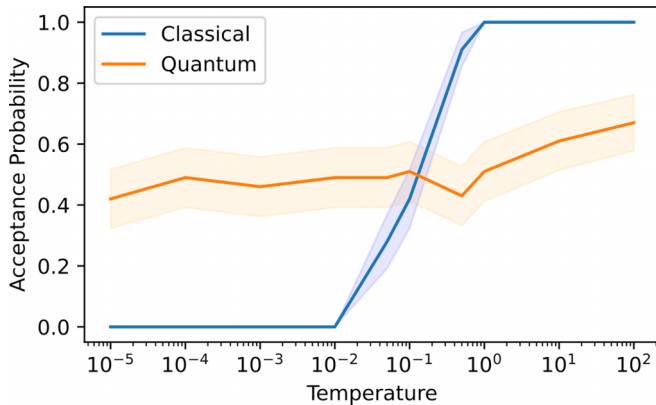
FIG. 4. Acceptance probability comparison as a function of temperature for a 2D double well.

FIG. 5. High-temperature trajectory comparison between HMC (left) and QD-HMC (right).

to investigate regimes of interest and mechanisms of potential advantage (such as having a lower autocorrelation time).

To explore QD-HMC further we focus on low-temperature regimes. We can see the resilience of the proposals empirically when varying the temperature and observing the acceptance rates. Figure 4 shows an example of this, in which we plot the acceptance probability of the quantum and classical proposals for 100 different random points for a 2D double-well function. The hyperparameters used are the same as the $T = 0.1$ double well above. As can be seen, the quantum proposals average around 50% independent of temperature, whereas the classical proposals are strongly positively correlated with temperature. The classical acceptance rates could likely be increased with hyperparameter optimization for each temperature, however, this often results in infeasible step sizes at very low temperatures. This result demonstrates that the areas in which it might be best to probe for the advantage of QD-HMC are likely at low temperatures and this highlights the potential benefit of lessened dependence on properly tuned hyperparameters.

V. DISCUSSION AND CONCLUSION

First, let us remark on the important distinctions between QD-HMC and the quantum enhanced MCMC (QEMCMC) of Ref. [32]. While we take inspiration and build upon their work, QD-HMC operates on a different class of problems (general continuous optimization vs discrete Ising models) and which we show is analogous to performing HMC on a quantum computer. This allows for implementation on both continuous variable and discrete quantum computers. Additionally, continuous dynamics adds difficulty in analytically computing advantage bounds. Unlike in the discrete case, we cannot materialize the transition matrix for exact analysis. Orfi and Sels [64] demonstrated by analyzing the spectral gap of the transition matrix that the QEMCMC of Layden *et al.* [32] offers no speedup in the general worst case. Although this proof of no speedup does not directly apply to QD-HMC, it does motivate our experiments to focus on areas in which QD-HMC could provide practical advantages and to focus on the mechanisms of potential advantages. The potential for a polynomial speedup in low-temperature regimes occurs as the quantum dynamics become a Grover
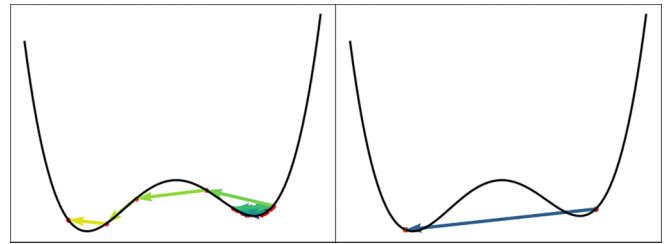
search [65] with a narrow Gaussian as the target state [38]. In contrast to the low-temperature evaluations above, as the wave function becomes delocalized during the evolution at high temperatures, the proposals find themselves in a highly nonlocal part of the landscape (i.e., traversing larger distances). Under high-temperature conditions, we see high-weight updates, compared to simulated annealing or HMC. Consider, as an illustrative example, Fig. 5. These show the trajectory HMC and QD-HMC algorithms take, respectively, for a high-temperature ($T = 100$) optimization in a simple one-dimensional (1D) double well. The QD-HMC updates are substantially larger than the classical due to the aforementioned delocalization. Note that this cannot be trivially solved by increasing the HMC step size or integration steps, as this results in convergence failures on this example. The coherent evolution and delocalization in position space enables tunneling, since the potential and kinetic strengths are randomly sampled.

This work begins to unlock a number of interesting future directions. There are improvements that could be made to this algorithm, e.g., the optimization of hyperparameters that defined the kicking magnitudes, how to tune all hyperparameters in certain optimization or probabilistic inference scenarios, etc. For example, one could tune the temperature synchronously with the temperature hyperparameters [66]. Another potential area of exploration would be the potential of population transfer for very low-temperature updates [67]. Experimenting with real quantum hardware is another important future direction. Adapting any algorithm to hardware is a challenge as the simulation of the Hamiltonian induces many two-qubit gates (which have substantially higher error rates) presenting an added challenge for this algorithm. However, the CV-QAOA subroutine, which is highly similar in structure to the transition kernel used in this work, was the subject of a recent hardware implementation [68]. For digital simulation of continuous dynamics, accessing the higher energy states of qutrits (e.g., through superconducting transmon hardware [69]) may be helpful for hardware-efficient emulation of qudit-based qumodes.

In this paper we demonstrated how to generalize HMC to quantum computing by leveraging coherent quantum dynamics. Building upon work done for quantum enhanced MCMC and CV-QAOA, we presented a new algorithm with promising machine learning use cases. This algorithm, QD-HMC, is theoretically outlined within the quantum mechanical and MCMC frameworks. We provide an empirical and theoretical analysis of our algorithm to demonstrate the potential

advantages QD-HMC may offer. Given the prevalence and importance of classical HMC in a variety of applications, improvements on it are meaningful to substantial portions of the Bayesian, machine learning, and optimization communities. QD-HMC expands the potential applications of quantum hardware for all of these areas.

APPENDIX A: HEISENBERG PICTURE UPDATE RULES

In the Heisenberg picture the wave function defines a time-independent basis with time-dependent operators, as opposed to the Schrödinger picture in which the wave function is a function of time and the operators are time independent. The resulting Heisenberg observables $O^{(H)}$ can be described in relation to the Schrödinger observables $O^{(S)}$ via

$$O^{(H)}(t) = e^{i\hat{H}t/\hbar} O^{(S)} e^{-i\hat{H}t/\hbar}. \tag{A1}$$

We can see the evolution of our operators at each layer of in the Trotterization. The evolution of these operators in the Heisenberg picture can be represented via

$$\frac{d}{dt} O^{(H)}(t) = \frac{i}{\hbar} [H, O^{(H)}(t)]. \tag{A2}$$

With this background, we can now derive the updates in the right-hand side of Eqs. (3) and (4) to see the evolution of the position and momentum operators (for a single time step). Starting with the momentum operator we have the single (discrete Trotter) update, i.e., when $t = 1$.

Since

$$\hat{H} = \eta \frac{\hat{\mathbf{p}}^2}{2} + \lambda f(\hat{\mathbf{x}}), \tag{A3}$$

we have

$$\begin{aligned}
\frac{d}{dt}\hat{\mathbf{x}} &= \frac{i}{\hbar}\left[\eta \frac{\hat{\mathbf{p}}^2}{2} + \lambda f(\hat{\mathbf{x}}), \hat{\mathbf{x}} \right] \\
&= \frac{i}{\hbar}\left(\left(\eta \frac{\hat{\mathbf{p}}^2}{2} + \lambda f(\hat{\mathbf{x}})\right)\hat{\mathbf{x}} - \hat{\mathbf{x}}\left(\eta \frac{\hat{\mathbf{p}}^2}{2} + \lambda f(\hat{\mathbf{x}})\right) \right) \\
&= \frac{i}{\hbar}\left(\eta \frac{\hat{\mathbf{p}}^2}{2}\hat{\mathbf{x}} + \lambda f(\hat{\mathbf{x}})\hat{\mathbf{x}} - \hat{\mathbf{x}}\eta \frac{\hat{\mathbf{p}}^2}{2} - \hat{\mathbf{x}}\lambda f(\hat{\mathbf{x}}) \right).
\end{aligned} \tag{A4}$$

Using a temporary wave function for clarity, and knowing that $f(\hat{\mathbf{x}})$ and $\hat{\mathbf{x}}$ commute, we can see

$$\begin{aligned}
\frac{i}{\hbar}&\left(\eta \frac{\hat{\mathbf{p}}^2}{2}\hat{\mathbf{x}} + \lambda f(\hat{\mathbf{x}})\hat{\mathbf{x}} - \hat{\mathbf{x}}\eta \frac{\hat{\mathbf{p}}^2}{2} - \hat{\mathbf{x}}\lambda f(\hat{\mathbf{x}}) \right)\psi \\
&= \frac{i}{\hbar}\left(\eta \frac{\hat{\mathbf{p}}^2}{2}\hat{\mathbf{x}}\psi - \hat{\mathbf{x}}\eta \frac{\hat{\mathbf{p}}^2}{2}\psi \right) \\
&= \frac{i}{2\hbar}\eta[\hat{\mathbf{p}}^2, \hat{\mathbf{x}}] \\
&= \frac{i}{2\hbar}\eta([\hat{\mathbf{p}}, \hat{\mathbf{x}}]\hat{\mathbf{p}} + \hat{\mathbf{p}}[\hat{\mathbf{p}}, \hat{\mathbf{x}}])
\end{aligned}$$

$$= \frac{i}{2\hbar}\eta(-i\hbar\hat{\mathbf{p}} - i\hbar\hat{\mathbf{p}})$$
$$= \eta\hat{\mathbf{p}}. \tag{A5}$$

Thus, we can see that the discrete Trotter step enacts a change on $\hat{\mathbf{x}}$ via $\eta\hat{\mathbf{p}}$, thus recovering the update in Eq. (3).

Now let us walk through the same steps, but for momentum.

We have

$$\begin{aligned}
\frac{d}{dt}\hat{\mathbf{p}} &= \frac{i}{\hbar}\left[\eta \frac{\hat{\mathbf{p}}^2}{2} + \lambda f(\hat{\mathbf{x}}), \hat{\mathbf{p}} \right] \\
&= \frac{i}{\hbar}\left(\left(\eta \frac{\hat{\mathbf{p}}^2}{2} + \lambda f(\hat{\mathbf{x}})\right)\hat{\mathbf{p}} - \hat{\mathbf{p}}\left(\eta \frac{\hat{\mathbf{p}}^2}{2} + \lambda f(\hat{\mathbf{x}})\right) \right) \\
&= \frac{i}{\hbar}\left(\eta \frac{\hat{\mathbf{p}}^3}{2} + \lambda f(\hat{\mathbf{x}})\hat{\mathbf{p}} - \eta \frac{\hat{\mathbf{p}}^3}{2} - \hat{\mathbf{p}}\lambda f(\hat{\mathbf{x}}) \right).
\end{aligned} \tag{A6}$$

Adding in a wave function once again, we can see

$$\begin{aligned}
\frac{i}{\hbar}&\left(\eta \frac{\hat{\mathbf{p}}^3}{2} + \lambda f(\hat{\mathbf{x}})\hat{\mathbf{p}} - \eta \frac{\hat{\mathbf{p}}^3}{2} - \hat{\mathbf{p}}\lambda f(\hat{\mathbf{x}}) \right)\psi \\
&= \frac{i}{\hbar}\lambda(f(\hat{\mathbf{x}})\hat{\mathbf{p}}\psi - \hat{\mathbf{p}}f(\hat{\mathbf{x}})\psi) \\
&= \lambda(f(\hat{\mathbf{x}})\nabla\psi - \nabla(f(\hat{\mathbf{x}})\psi)) \\
&= \lambda(f(\hat{\mathbf{x}})\nabla\psi - f(\hat{\mathbf{x}})\nabla\psi - \psi\nabla f(\hat{\mathbf{x}})) \\
&= -\lambda\nabla f(\hat{\mathbf{x}}).
\end{aligned} \tag{A7}$$

Thus, we can see that the discrete Trotter step enacts a change on $\hat{\mathbf{p}}$ via $-\lambda\nabla f(\hat{\mathbf{x}})$, thus recovering Eq. (4).

APPENDIX B: QD-HMC AND DETAILED BALANCE

In this section we will prove that QD-HMC, as outlined in Algorithm 1, obeys the detailed balance condition, as it is essential to showing that the sampled distribution converges to the true target Boltzmann distribution in the asymptotic limit.

ALGORITHM 1: QD-HMC algorithm.

---

1. Prepare bitstring $|x\rangle$.
2. Where $f(\hat{x})$ is the problem function, define

$$\hat{H}_{\eta\lambda} = \eta \frac{\hat{p}^2}{2} + \lambda f(\hat{x})$$

with chosen hyperparameters η, λ.
3. Via random Trotterization, apply

$$\hat{U}_{\eta\lambda} \equiv e^{-i\hat{H}_{\eta\lambda}t}.$$

4. Optionally, flip the momentum via Algorithm 2.
4. Sample bitstring measurement

$$y \sim |\langle y| \hat{U}_{\eta\lambda} |x\rangle|^2.$$

5. Accept with probability

$$A(y|x) = \min\{1, e^{-[f(y)-f(x)]}\}.$$

Claim.

$$|\langle y|\hat{U}_{\eta\lambda}|x\rangle|^2 = |\langle x|\hat{U}_{\eta\lambda}|y\rangle|^2 \quad \forall x, y, \eta, \lambda. \tag{B1}$$

Proof.

Note. $(e^{\hat{A}})^T = e^{\hat{A}^T}$

Eq. (B1) holds if $\hat{U}_{\eta\lambda} = \hat{U}_{\eta\lambda}^T$.

This equality is equivalent to

$$\iff [\eta\hat{p}^2 + \lambda f(\hat{x})]^T = \eta\hat{p}^2 + \lambda f(\hat{x}),$$
$$\iff (\hat{p}^2)^T = \hat{p}^2. \tag{B2}$$

Note. $\hat{p} = \hat{F}\hat{x}\hat{F}^\dagger$ and $\hat{F}^T = \hat{F}$.

Therefore,

$$\hat{p}^T = (\hat{F}^\dagger)^T \hat{x}\hat{F}^T = \hat{F}^\dagger \hat{x}\hat{F}$$
$$= (\hat{F}^\dagger)^2 \hat{p}\hat{F}^2 = -\hat{p}. \tag{B3}$$

We can make this transition invertible by adding a momentum flip as seen in Algorithm 2.

$$\hat{U}_{\eta\lambda} \mapsto \hat{M}\hat{U}_{\eta\lambda} \mid \hat{M}\hat{U}_{\eta\lambda}\hat{M} = \hat{U}_{\eta\lambda}^T. \tag{B4}$$

ALGORITHM 2: Momentum flip.

---

1. Momentum flip:
$$\hat{M} \equiv \hat{F}_c^\dagger \hat{X}_{j=n} \hat{F}_c, \quad \hat{M}^\dagger = \hat{M}.$$
2. Convert to momentum space.
3. Flip most significant bit.
4. Convert back.

Since

$$\hat{M}\hat{M}^T = \hat{F}_c^\dagger \hat{X}_n \hat{F}_c \hat{F}_c \hat{X}_n \hat{F}_c^\dagger$$
$$= -\hat{F}_c^\dagger \hat{F}_c \hat{F}_c \hat{F}_c^\dagger$$
$$= -I, \tag{B5}$$

it follows that

$$(\hat{M}\hat{U})^T = \hat{U}^T \hat{M}^T$$
$$= \hat{M}\hat{U}\hat{M}\hat{M}^T$$
$$= \hat{M}\hat{U}. \tag{B6}$$

Thus,

$$|\langle y|\hat{U}_{\eta\lambda}|x\rangle|^2 = |\langle x|\hat{U}_{\eta\lambda}|y\rangle|^2 \quad \forall x, y, \eta, \lambda. \tag{B7}$$
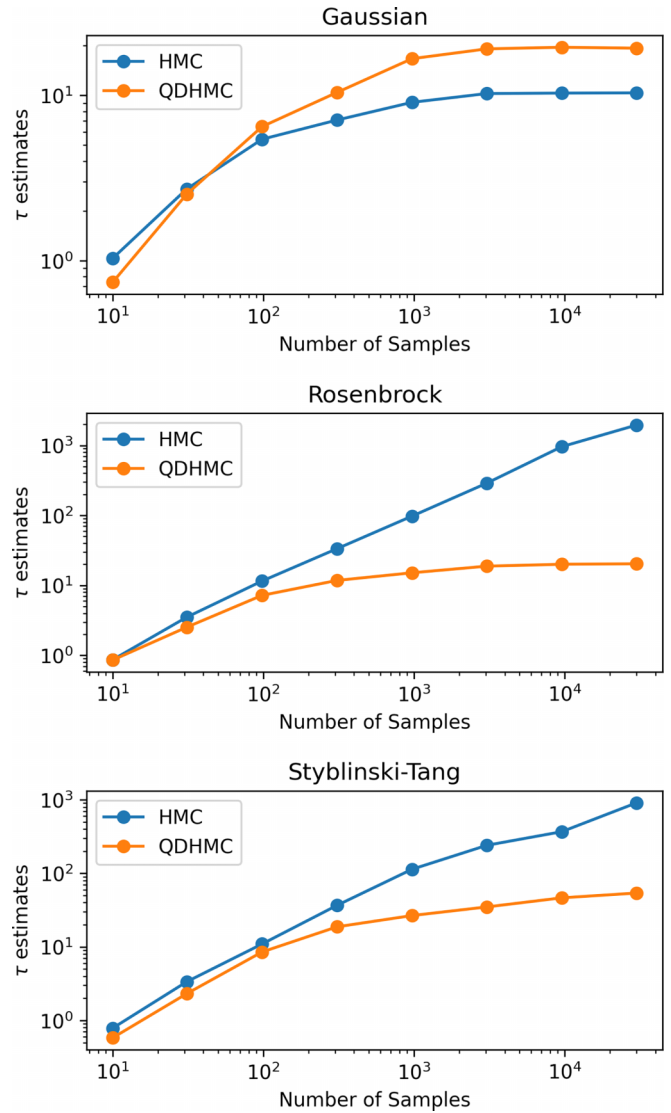
∎

FIG. 6. Estimation of autocorrelation time τ as a function of number of samples.

APPENDIX C: FURTHER AUTOCORRELATION ESTIMATES

Figure 6 shows the estimates of τ for different 2D problems.

[1] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature (London) **521**, 436 (2015).

[2] Y. Kim, A. Eddins, S. Anand, K. X. Wei, E. Van Den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme *et al.*, Evidence for the utility of quantum computing before fault tolerance, Nature (London) **618**, 500 (2023).

[3] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell *et al.*, Quantum supremacy using a programmable superconducting processor, Nature (London) **574**, 505 (2019).

[4] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu *et al.*, Quantum computational advantage using photons, Science **370**, 1460 (2020).

[5] J. Chow, O. Dial, and J. Gambetta, IBM quantum breaks the 100-qubit processor barrier, IBM Research Blog (2021),

https://www.ibm.com/quantum/blog/127-qubit-quantum-processor-eagle.

[6] Y. Wu, W.-S. Bao, S. Cao, F. Chen, M.-C. Chen, X. Chen, T.-H. Chung, H. Deng, Y. Du, D. Fan *et al.*, Strong quantum computational advantage using a superconducting quantum processor, Phys. Rev. Lett. **127**, 180501 (2021).

[7] L. S. Madsen, F. Laudenbach, M. F. Askarani, F. Rortais, T. Vincent, J. F. Bulmer, F. M. Miatto, L. Neuhaus, L. G. Helt, M. J. Collins *et al.*, Quantum computational advantage with a programmable photonic processor, Nature (London) **606**, 75 (2022).

[8] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature (London) **549**, 195 (2017).

[9] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[10] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke *et al.*, Noisy intermediate-scale quantum algorithms, Rev. Mod. Phys. **94**, 015004 (2022).

[11] U. von Toussaint, Bayesian inference in physics, Rev. Mod. Phys. **83**, 943 (2011).

[12] D. Bertsimas and J. Tsitsiklis, Simulated annealing, Stat. Sci. **8**, 10 (1993).

[13] I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling, Fast collapsed gibbs sampling for latent dirichlet allocation, in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, New York, 2008), pp. 569–577.

[14] R. van de Schoot, S. Depaoli, R. King, B. Kramer, K. Märtens, M. G. Tadesse, M. Vannucci, A. Gelman, D. Veen, J. Willemsen *et al.*, Bayesian statistics and modelling, Nat. Rev. Methods Primers **1**, 1 (2021).

[15] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang, A tutorial on energy-based learning, Predicting Structured Data **1** (2006).

[16] Y. Du and I. Mordatch, Implicit generation and modeling with energy based models, in *Advances in Neural Information Processing Systems*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., 2019), Vol. 32.

[17] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, Parameterized quantum circuits as machine learning models, Quantum Sci. Technol. **4**, 043001 (2019).

[18] X. Yuan, S. Endo, Q. Zhao, Y. Li, and S. C. Benjamin, Theory of variational quantum simulation, Quantum **3**, 191 (2019).

[19] S. Endo, J. Sun, Y. Li, S. C. Benjamin, and X. Yuan, Variational quantum simulation of general processes, Phys. Rev. Lett. **125**, 010501 (2020).

[20] F. M. Sbahi, A. J. Martinez, S. Patel, D. Saberi, J. H. Yoo, G. Roeder, and G. Verdon, Provably efficient variational generative modeling of quantum many-body systems via quantum-probabilistic information geometry, arXiv:2206.04663.

[21] S. Y.-C. Chen, C.-H. H. Yang, J. Qi, P.-Y. Chen, X. Ma, and H.-S. Goan, Variational quantum circuits for deep reinforcement learning, IEEE Access **8**, 141007 (2020).

[22] O. Lockwood and M. Si, Reinforcement learning with quantum variational circuit, in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Vol. 16 (AAAI, Washington, 2020), pp. 245–251.

[23] O. Lockwood and M. Si, Playing atari with hybrid quantum-classical reinforcement learning, in *NeurIPS 2020 Workshop on Pre-Registration in Machine Learning* (PMLR, 2021), pp. 285–301.

[24] E. Anschuetz, J. Olson, A. Aspuru-Guzik, and Y. Cao, Variational quantum factoring, in *Quantum Technology and Optimization Problems First International Workshop, QTOP 2019, Munich, Germany, March 18, 2019, Proceedings 1* (Springer, Berlin, 2019), pp. 74–85.

[25] K. Kubo, Y. O. Nakagawa, S. Endo, and S. Nagayama, Variational quantum simulations of stochastic differential equations, Phys. Rev. A **103**, 052425 (2021).

[26] M. Lubasch, J. Joo, P. Moinier, M. Kiffner, and D. Jaksch, Variational quantum algorithms for nonlinear problems, Phys. Rev. A **101**, 010301(R) (2020).

[27] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, Hybrid Monte Carlo, Phys. Lett. B **195**, 216 (1987).

[28] M. Betancourt, A conceptual introduction to Hamiltonian Monte Carlo, arXiv:1701.02434.

[29] A. Beskos, N. Pillai, G. Roberts, J.-M. Sanz-Serna, and A. Stuart, Optimal tuning of the hybrid Monte Carlo algorithm, Bernoulli **19**, 1501 (2013).

[30] C. Robert and G. Casella, A short history of Markov chain Monte Carlo: Subjective recollections from incomplete data, Stat. Sci. **26**, 102 (2011).

[31] G. Mazzola, Sampling, rates, and reaction currents through reverse stochastic quantization on quantum computers, Phys. Rev. A **104**, 022431 (2021).

[32] D. Layden, G. Mazzola, R. V. Mishmash, M. Motta, P. Wocjan, J.-S. Kim, and S. Sheldon, Quantum-enhanced Markov chain Monte Carlo, Nature (London) **619**, 282 (2023).

[33] Y. Nakano, H. Hakoshima, K. Mitarai, and K. Fujii, Markov-chain Monte Carlo method enhanced by a quantum alternating operator ansatz, Phys. Rev. Res. **6**, 033105 (2024).

[34] A. Orfi, Near-term quantum Algorithms for classical sampling, Master's thesis, University of Waterloo, 2023.

[35] M. B. Mansky, J. Nüßlein, D. Bucher, D. Schuman, S. Zielinski, and C. Linnhoff-Popien, Sampling problems on a quantum computer, in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, Vol. 1 (IEEE, 2023), pp. 485–495.

[36] G. Mazzola, Quantum computing for chemistry and physics applications from a Monte Carlo perspective, J. Chem. Phys. **160**, 010901 (2024).

[37] S. Ferguson and P. Wallden, Quantum-enhanced Markov chain Monte Carlo for systems larger than your quantum computer, arXiv:2405.04247.

[38] G. Verdon, J. M. Arrazola, K. Brádler, and N. Killoran, A quantum approximate optimization algorithm for continuous problems, arXiv:1902.00409.

[39] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv:1411.4028.

[40] P. Huembeli, J. M. Arrazola, N. Killoran, M. Mohseni, and P. Wittek, The physics of energy-based models, Quantum Machine Intelligence **4**, 1 (2022).

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

[42] G. O. Roberts and R. L. Tweedie, Exponential convergence of langevin distributions and their discrete approximations, Bernoulli **2**, 341 (1996).

[43] X. Cheng and P. Bartlett, Convergence of langevin mcmc in kl-divergence, in *Algorithmic Learning Theory* (PMLR, 2018), pp. 186–211.

[44] W. Shao, G. Guo, F. Meng, and S. Jia, An efficient proposal distribution for Metropolis–Hastings using a β-splines technique, Comput. Stat. Data Anal. **57**, 465 (2013).

[45] J. Brofos, M. Gabrié, M. A. Brubaker, and R. R. Lederman, Adaptation of the independent metropolis-hastings sampler with normalizing flow proposals, in *International Conference on Artificial Intelligence and Statistics* (PMLR, 2022), pp. 5949–5986.

[46] M. Betancourt, S. Byrne, S. Livingstone, and M. Girolami, The geometric foundations of Hamiltonian Monte Carlo, Bernoulli **23**, 2257 (2017).

[47] N. Bou-Rabee and J. M. Sanz-Serna, Geometric integrators and the Hamiltonian Monte Carlo method, Acta Numerica **27**, 113 (2018).

[48] R. M. Neal *et al.*, MCMC using Hamiltonian dynamics, in *Handbook of Markov Chain Monte Carlo*, edited by S. Brooks, A. Gelman, G. Jones, and X.-L. Meng (Chapman and Hall/CRC, 2011), Chap. 5, 1st ed.

[49] R. D. Somma, Quantum simulations of one dimensional quantum systems, arXiv:1503.06319.

[50] G. Verdon, J. Pye, and M. Broughton, A universal training algorithm for quantum deep learning, arXiv:1806.09729.

[51] D. Coppersmith, An approximate fourier transform useful in quantum factoring, arXiv:quant-ph/0201067.

[52] X.-S. Yang, Test problems in optimization, arXiv:1008.0549.

[53] V. Beiranvand, W. Hare, and Y. Lucet, Best practices for comparing optimization algorithms, Optim. Eng. **18**, 815 (2017).

[54] J. J. Moré and S. M. Wild, Benchmarking derivative-free optimization algorithms, SIAM J. Optim. **20**, 172 (2009).

[55] J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, and R. A. Saurous, Tensorflow distributions, arXiv:1711.10604.

[56] M. Broughton, G. Verdon, T. McCourt, A. J. Martinez, J. H. Yoo, S. V. Isakov, P. Massey, R. Halavati, M. Y. Niu, A. Zlokapa, E. Peters, O. Lockwood, A. Skolik, S. Jerbi, V. Dunjko, M. Leib, M. Streif, D. Von Dollen, H. Chen, S. Cao *et al.*, Tensorflow quantum: A software framework for quantum machine learning, arXiv:2003.02989.

[57] https://github.com/QuantumVerd/cv-tfq.

[58] https://github.com/diracq/qdhmc.

[59] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (ACM, New York, 2019), pp. 2623–2631.

[60] M. D. Hoffman and A. Gelman, The no-u-turn sampler: adaptively setting path lengths in hamiltonian Monte Carlo, J. Mach. Learn. Res. **15**, 1593 (2014).

[61] T. Chen, E. Fox, and C. Guestrin, Stochastic gradient Hamiltonian Monte Carlo, in *International Conference on Machine Learning* (PMLR, 2014), pp. 1683–1691.

[62] A. Sokal, Monte Carlo methods in statistical mechanics: Foundations and new algorithms, in *Functional integration: Basics and Applications* (Springer, 1997), pp. 131–192.

[63] D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman, emcee: The MCMC hammer, Publ. Astron. Soc. Pac. **125**, 306 (2013).

[64] A. Orfi and D. Sels, Bounding speedup of quantum-enhanced markov chain Monte Carlo, arXiv:2403.03087.

[65] L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* (ACM, New York, 1996), pp. 212–219.

[66] A. Bapat and S. Jordan, Bang-bang control as a design principle for classical and quantum optimization algorithms, Quantum Info. Comput. **19**, 424 (2019).

[67] V. N. Smelyanskiy, K. Kechedzhi, S. Boixo, S. V. Isakov, H. Neven, and B. Altshuler, Nonergodic delocalized states for efficient population transfer within a narrow band of the energy landscape, Phys. Rev. X **10**, 011017 (2020).

[68] Y. Enomoto, K. Anai, K. Udagawa, and S. Takeda, Continuous-variable quantum approximate optimization on a programmable photonic quantum processor, Phys. Rev. Res. **5**, 043005 (2023).

[69] A. Cervera-Lierta, M. Krenn, A. Aspuru-Guzik, and A. Galda, Experimental high-dimensional Greenberger-Horne-Zeilinger entanglement with superconducting transmon qutrits, Phys. Rev. Appl. **17**, 024062 (2022).