Counterdiabatic Hamiltonian Monte Carlo

Reuben Cohn-Gordon
University of California, Berkeley
reubenharry@gmail.com

Uroš Seljak
University of California, Berkeley

Dries Sels
New York University

Abstract

Hamiltonian Monte Carlo (HMC) is a state of the art method for sampling from distributions with differentiable densities, but can converge slowly when applied to challenging multimodal problems. Running HMC with a time varying Hamiltonian, in order to interpolate from an initial tractable distribution to the target of interest, can address this problem. In conjunction with a weighting scheme to eliminate bias, this can be viewed as a special case of Sequential Monte Carlo (SMC) sampling [13]. However, this approach can be inefficient, since it requires slow change between the initial and final distribution. Inspired by [30], where a learned *counterdiabatic* term added to the Hamiltonian allows for efficient quantum state preparation, we propose *Counterdiabatic Hamiltonian Monte Carlo* (CHMC), which can be viewed as an SMC sampler with a more efficient kernel. We establish its relationship to recent proposals for accelerating gradient-based sampling with learned drift terms, and demonstrate on simple benchmark problems.

1 Introduction

Algorithms for sampling from probability distributions have many application in the physical sciences [33, 36] and Bayesian statistics [16]. For sampling from distributions with differentiable densities, *Hamiltonian Monte Carlo* (HMC) [5] is a state of the art scheme, and in conjunction with the No-U-Turn criterion for trajectory length [22], is a widely used *black box* method for inference [17].

Markov Chain Monte Carlo (MCMC), the family of algorithms to which HMC belongs, samples from a distribution with density π by constructing an ergodic Markov kernel which has this target as its stationary distribution. However, the number of steps in the resulting Markov chain required for the law of the chain to converge to the target can be prohibitively large. A motivating example is a mixture of two Gaussians with means separated by a distance much larger than the respective variances; if the chain is initialized near one of the two modes, attaining stationarity requires moving through areas of low probability density.

An intuitive solution to this problem is to vary the target distribution over time, so that π_0 is easy to sample from, while π_T is the desired target, and the kernel M_t at time t is an invariant kernel of π_t. Sequential Monte Carlo (SMC) samplers [11] do just this, running many chains (particles) in parallel, and using a weighting scheme to obtain unbiased samples from each π_i.

Hamiltonian Monte Carlo (HMC) is rooted in the connection between probabilistic inference and classical statistical physics, and constructs a Markov kernel M by simulating Hamiltonian dynamics. These dynamics have the canonical (Boltzmann) distribution as a stationary distribution in an extended space of position q and momentum p, and the Hamiltonian is chosen so that marginalizing over the momentum yields the target distribution π on q. In the context of HMC, moving through a series of kernels as above corresponds to evolving a physical system while changing its parameters, which is a central topic of statistical mechanics. From this perspective, varying the parameters too quickly results in *lag*, or more physically, *dissipation* [34], where the Hamiltonian flow cannot return the

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Frontiers in Probabilistic Inference: Sampling Meets Learning.

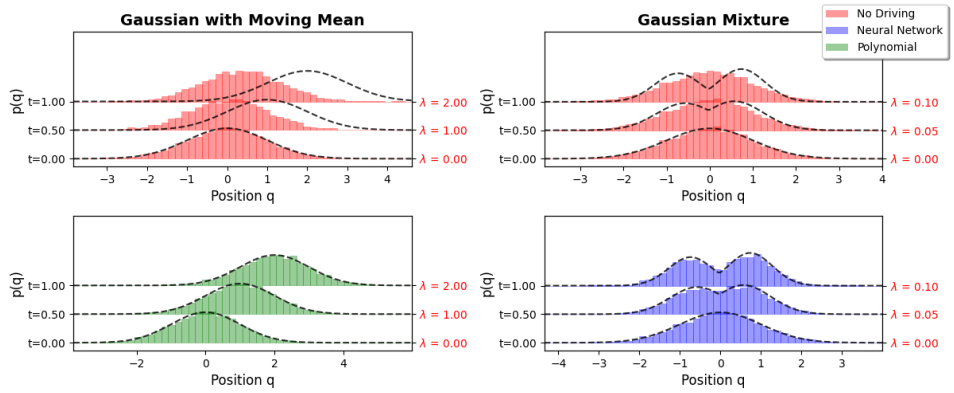

Figure 1: Illustration of lag induced by a time-varying Hamiltonian (top row), and the correction introduced by a learned *counterdiabatic term* (bottom row). The lefthand column shows a Gaussian with time-varying mean, and the righthand column an interpolation between a Gaussian and a mixture of two Gaussians. In the simple case, parametrizing A as a sum of polynomials with learned coefficients suffices, and in the more complex case, A is parameterized by a neural network.

particles to equilibrium quickly enough as the system's parameters change. Two examples of this lag are shown in Figure 1 in red.

Several papers [12, 4], both in the quantum and classical setting, have noted that one can compensate for this lag by adding a *counterdiabatic potential* to the Hamiltonian, which gives rise to a force that keeps the particles in the stationary distribution constantly. More recently, a scheme to approximate this term variationally has been proposed in a quantum setting [30].

Contributions We adapt this counterdiabatic approach to the setting of sampling, to obtain a method for transitioning quickly between a simple distribution and the distribution of interest. This is shown in the green and blue plots of Figure 1, where the counterdiabatic term accelerates the equilibration, so that a subsequent SMC reweighting scheme has less variance. We note that our approach is closely related to the proposals of [1, 35, 2], but in a Hamiltonian setting (see section 4).

2 Technical Background

Hamiltonian dynamics Suppose we have a physical system given by a d-dimensional vector $q \in \mathbb{R}^d$, which describes, for example, the position of a particle. Hamilton's equations of motion are a 1st order ODE defined on $z = (q, p) \in \mathbb{R}^{2d}$, where $p \in \mathbb{R}^d$ is a second variable called the momentum. The ODE is defined in terms of a real-valued *Hamiltonian* function $H(q, p)$.

$$\dot{q} = \nabla_p H(q, p); \qquad \dot{p} = -\nabla_q H(q, p) \tag{1}$$

We refer to the solution to the ODE as the *flow* ϕ_H, so that $\phi_H(t)(q_0, p_0)$ is the state of the system at time t given that the initial state is $z_0 = (q_0, p_0)$.

Poisson bracket Given two real-valued functions of (q, p), f and g, the Poisson bracket $\{f, g\} \equiv \nabla_q f \cdot \nabla_p g - \nabla_p f \cdot \nabla_q g$. The bracket allows a convenient description of Hamiltonian dynamics. In particular, the total derivative $\frac{df(q(t),p(t))}{dt}$ can be written as $\{f, H\} = \nabla_q f \cdot \nabla_p H - \nabla_p f \cdot \nabla_q H = \nabla_q f \cdot \dot{q} + \nabla_p f \cdot \dot{p} = \frac{df}{dt}$. We note that the Poisson bracket is antisymmetric, i.e. $\{A, B\} = -\{B, A\}$, and that if $f(q, p) = h(g(q, p))$ for some h, then $\{f, g\} = 0$.

Canonical distribution The distribution $\rho_H(q, p) \equiv (e^{-H(q,p)})/Z$, with normalization $Z = \int e^{-H(q,p)} dq dp$, is known as the *canonical distribution*. When H has no dependence on time except through q and p, then $\frac{d\rho_H}{dt} = \{\rho_H, H\} = 0$, with the last equality following by an above property of the Poisson bracket. We denote by ϕ_H^* the pushforward of the flow on a distribution, so that $\forall t, \phi_H^*(t)(\rho_H) = \rho_H$ is the statement that the canonical distribution is stationarity under the dynamics.

Hamiltonian Monte Carlo The principle of Hamiltonian Monte Carlo is to use a discretization of Hamiltonian dynamics to produce a kernel which has a stationary distribution ρ_H for which the marginal on q, i.e. $\pi(q) \propto \int \rho_H(q,p)dp$, is the distribution of interest. This can be achieved by choosing $H = -\log \pi(q) + \frac{1}{2}p^2 \equiv V(q) + T(p)$, which only requires knowing the density up to the normalization constant. The Metropolis-Hastings algorithm [8] is used to correct for bias coming from the discretization of the Hamiltonian dynamics, and momentum can be refreshed every n steps, or partially refreshed every step [7], in order to ensure ergodicity.

2.1 Counterdiabatic driving

Consider $H_{\lambda(t)}$, a Hamiltonian which depends on a parameter λ which varies with time. We can now consider the flow ϕ_{H_λ} that is the solution to Hamilton's equations. Crucially, we note that $\phi^*_{H_\lambda}(t')\rho_{H_{\lambda(t)}} \neq \rho_{H_{\lambda(t+t')}}$. In other words, the flow of the time-varying Hamiltonian does not preserve the time-varying distribution. To see why it fails, we inspect the total derivative:

$$\frac{d}{dt}\rho_{H_\lambda} = \{\rho_{H_\lambda}, H_\lambda\} + \dot{\lambda}\partial_\lambda \rho_{H_\lambda} = \dot{\lambda}\partial_\lambda H \rho_{H_\lambda} - \dot{\lambda}\frac{\partial \log Z_\lambda}{\partial \lambda}\rho_{H_\lambda} = (-\partial_\lambda H_\lambda + \mathbb{E}[\partial_\lambda H_\lambda])\dot{\lambda}\rho_{H_\lambda} \quad (2)$$

where the expectation is over ρ_{H_λ}. [12], [4] and [34, 18] note that one can express the lag in terms of a function A_λ of (q,p), known as the *counterdiabatic gauge potential*, with the property that at any λ,

$$\{A_\lambda, H_\lambda\} = \partial_\lambda H_\lambda - \mathbb{E}[\partial_\lambda H_\lambda] \quad (3)$$

We then have that $\frac{d}{dt}\rho_{H_\lambda} = \dot{\lambda}\partial_\lambda \rho_{H_\lambda} = \dot{\lambda}\{H_\lambda, A_\lambda\}\rho_{H_\lambda} = \{\rho_{H_\lambda}, -\dot{\lambda}A_\lambda\}$. This is the total derivative that would be generated by a Hamiltonian $-\dot{\lambda}A_\lambda$. This demonstrates a key result: if we choose as our Hamiltonian $\mathrm{HC}_{\lambda(t)} \equiv H_{\lambda(t)} + \dot{\lambda}(t)A_{\lambda(t)}$, we have that $\phi^*_{\mathrm{HC}_\lambda}(t')\rho_{H_{\lambda(t)}} = \rho_{H_{\lambda(t+t')}}$. Thus, by using HC_λ as our Hamiltonian, we obtain perfect transport to each $\rho_{H_{\lambda(t)}}$, including $\rho_{H_{\lambda(T)}}$.

A simple example of a Gaussian with a time-varying mean is given in Appendix B, where the counterdiabatic term can be derived analytically. In general however, it is necessary to learn A_λ.

2.2 Learning the counterdiabatic gauge potential

Lemma 2.1. *Let $\mathcal{L}_H(A) = \mathbb{E}[\|\{A,H\} - \partial_\lambda H\|^2]$, where the expectation is taken over ρ_H. Then equation 3 is satisfied by*

$$A^* = \arg\min_A \mathcal{L}_H(A)$$

This is shown in [30] in the quantum setting. We provide a similar proof in the present case in appendix D.

3 Counterdiabatic Hamiltonian Monte Carlo (CHMC)

The fact that we can learn an approximation of A_λ suggests an algorithm for sampling. We first choose a path of densities $\pi_{\lambda(t)}$, where $\pi_{\lambda(0)}$ is chosen to be straightforward to sample from, and where $\pi_{\lambda(T)}$ is the distribution of interest. This defines a time-varying Hamiltonian $H_{\lambda(t)}(p,q) = \frac{1}{2}|p|^2 - \log \pi_{\lambda(t)}(q) \equiv T(p) + V_{\lambda(t)}(q)$, and in turn a path of densities on the full space, $\rho_{H_{\lambda(t)}}$.

We define A^ϕ_λ as a neural network or sum of polynomials parametrized by a vector ϕ. We additionally discuss a scheme to use Hermite polynomials for the distribution over momentum in Appendix E. We initialize a population of particles distributed according to $\rho_{H_{\lambda(0)}}$. Choosing a step size ϵ, so that at each step j, $t = j\epsilon$, we learn a new ϕ by minimizing $\mathcal{L}$, taking the expectation with respect to the current population (Algorithm 1). We use $\mathrm{HC}_\lambda = H_{\lambda(t)} + \dot{\lambda}(t)A_{\lambda(t)}$ as our Hamiltonian at a given time t, and approximate Hamiltonian dynamics by a splitting integrator [24]. Since A is not generally separable, this will not be symplectic.

We can, and should, compute weights for each particle after every step, to obtain a consistent sampler. We explain this in more detail in Appendix A, as a special case of Sequential Monte Carlo sampling [11], and derive an expression for the weight from the non-equilibrium fluctuation theorem. The full algorithm is summarized in Algorithm 2.

4 Related work

From the perspective of stochastic optimal control [15] and optimal transport [29], the problem at hand is to design an ODE or SDE that induces a desired trajectory in the space of densities, from some π_0 to the target π_T. In this setting, Controlled Monte Carlo Diffusions [35], the Liouville Flow Importance Sampler [32], Path Guided Particle Sampling [14], the Path-Integral Sampler [37], the Adjoint sampler [20] and Non-equilibrium transport sampling [1] all learn a drift term of a differential equation, parametrized by a neural network.

Several of these approaches use weights to correct for remaining bias [1, 35]. [1] in particular presents a similar scheme to ours, but with overdamped Langevin dynamics, rather than Hamiltonian dynamics, with different loss functions, and with weights applied non-incrementally and not used in training. With weighting, these approaches can be understood in terms of Sequential Monte Carlo sampling [10], as described in Appendix A, and in that setting, [21] and [3] have both considered learning kernels M_t instead of setting M_t as the invariant kernel of each successive π_t.

Finally, adding a term to Hamiltonian dynamics to preserve stationarity through a series of intermediate distributions has been proposed several times in a quantum setting, as *engineered swift equilibration* [25] and counterdiabatic driving [12], where the corresponding problem with a quickly changing Hamiltonian is failure to preserve eigenstates. The core ideas carry over to a classical setting [28], often under the name *shortcuts to adiabaticity*, or *escorted free energy* (EFE) [34]. We recommend [19] for an overview. [30] proposes *learning* a driving term in a quantum context, but notes that the approach is transferable to a classical setting, which is the foundation for our approach.

Where CHMC differs from related work The work summarized above focuses either on overdamped dynamics, or Hamiltonian dynamics without learning. Since Hamiltonian Monte Carlo is the gold standard approach for sampling from differentiable densities, because of the geometric properties of Hamiltonian dynamics [6], it is natural that schemes for accelerating sampling in multimodal targets should focus on it. In this Hamiltonian setting, the learned term takes on an geometric significance, as a connection on a fiber bundle [23].

5 Evaluation

Here, we show the difference, on several simple problems, between the final particles that result from using CHMC and standard HMC, in order to highlight how counterdiabatic driving accelerates transport. We note that the goal is not to obtain unbiased samples from the target, but rather to improve the performance of subsequent SMC-style reweighting. We measure the standardized squared error $b^2(f) = \frac{(\hat{f}-\mathbb{E}[f])^2}{\mathrm{Var}[f]}$, where $\hat{f}$ is the Monte Carlo estimate $\sum_i^N f(q_i)$ of the final population of samples $\{q_i\}$ and $f(q) \equiv q^2$. We consider 3 problems: a 1D Gaussian with a moving mean, a 1D Gaussian with a changing variance, and an interpolation between a 1D Gaussian. See Appendix C for details.

System	True $E[x^2]$	Naive HMC	CHMC
Gaussian Moving Mean	2.0	1.12	2.1
Gaussian Annealing	0.1	86.5	0.65
Double Well	7.34	2.06	4.22

The takeaway is that when running at a fast schedule, standard HMC results in poor transport, because more than a single step of the kernel at each time is required to regain stationarity, but that using the counterdiabatic Hamiltonian results in better transport. A more complete evaluation, which includes the use of weights, a more powerful neural network, and higher dimensional benchmark problems in the spirit of [1], is beyond the present scope, but a topic for future work.

6 Conclusion

We have presented an application of learned counterdiabatic driving to the problem of sampling. This is closely related to the learned transport approach of [1] among other recent papers, but introduces a new physically inspired perspective. In particular, the extensive work in the physics literature on the choice of parametrization for A_λ and an optimal choice of $\lambda(t)$, and in applied statistics for the choice of ϵ and a mass matrix in HMC, will serve as the inspiration for future work.

References

[1] Michael S Albergo and Eric Vanden-Eijnden. "Nets: A non-equilibrium transport sampler". In: *arXiv preprint arXiv:2410.02711* (2024).

[2] Michael Arbel, Alex Matthews, and Arnaud Doucet. "Annealed flow transport monte carlo". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 318–330.

[3] Espen Bernton et al. "Schrodinger Bridge Samplers". In: *arXiv preprint arXiv:1912.13170* (2019).

[4] Michael Victor Berry. "Transitionless quantum driving". In: *Journal of Physics A: Mathematical and Theoretical* 42.36 (2009), p. 365303.

[5] Michael Betancourt. "A conceptual introduction to Hamiltonian Monte Carlo". In: *arXiv preprint arXiv:1701.02434* (2017).

[6] Michael Betancourt et al. "The geometric foundations of hamiltonian monte carlo". In: *Bernoulli* (2017), pp. 2257–2298.

[7] Xiang Cheng et al. "Underdamped Langevin MCMC: A non-asymptotic analysis". In: *Conference on learning theory*. PMLR. 2018, pp. 300–323.

[8] Siddhartha Chib and Edward Greenberg. "Understanding the metropolis-hastings algorithm". In: *The american statistician* 49.4 (1995), pp. 327–335.

[9] Gavin E Crooks. "Nonequilibrium measurements of free energy differences for microscopically reversible Markovian systems". In: *Journal of statistical physics* 90.5 (1998), pp. 1481–1487.

[10] Chenguang Dai et al. "An invitation to sequential Monte Carlo samplers". In: *Journal of the American Statistical Association* 117.539 (2022), pp. 1587–1600.

[11] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. "Sequential monte carlo samplers". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 68.3 (2006), pp. 411–436.

[12] Mustafa Demirplak and Stuart A Rice. "Adiabatic population transfer with control fields". In: *The Journal of Physical Chemistry A* 107.46 (2003), pp. 9937–9945.

[13] Arnaud Doucet, Nando De Freitas, and Neil Gordon. "An introduction to sequential Monte Carlo methods". In: *Sequential Monte Carlo methods in practice* (2001), pp. 3–14.

[14] Mingzhou Fan et al. "Path-guided particle-based sampling". In: *arXiv preprint arXiv:2412.03312* (2024).

[15] Wendell H Fleming and Raymond W Rishel. *Deterministic and stochastic optimal control*. Vol. 1. Springer Science & Business Media, 2012.

[16] Dani Gamerman and Hedibert F Lopes. *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*. Chapman and Hall/CRC, 2006.

[17] Andrew Gelman, Daniel Lee, and Jiqiang Guo. "Stan: A probabilistic programming language for Bayesian inference and optimization". In: *Journal of Educational and Behavioral Statistics* 40.5 (2015), pp. 530–543.

[18] David Guery-Odelin et al. "Driving rapidly while remaining in control: classical shortcuts from Hamiltonian to stochastic dynamics". In: *Reports on Progress in Physics* 86.3 (2023), p. 035902.

[19] David Guery-Odelin et al. "Shortcuts to adiabaticity: Concepts, methods, and applications". In: *Reviews of Modern Physics* 91.4 (2019), p. 045001.

[20] Aaron Havens et al. "Adjoint sampling: Highly scalable diffusion samplers via adjoint matching". In: *arXiv preprint arXiv:2504.11713* (2025).

[21] Jeremy Heng et al. "Controlled sequential monte carlo". In: *The Annals of Statistics* 48.5 (2020), pp. 2904–2929.

[22] Matthew D Hoffman, Andrew Gelman, et al. "The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo." In: *J. Mach. Learn. Res.* 15.1 (2014), pp. 1593–1623.

[23] Michael Kolodrubetz et al. "Geometry and non-adiabatic response in quantum and classical systems". In: *Physics Reports* 697 (2017), pp. 1–87.

[24] Benedict Leimkuhler and Sebastian Reich. *Simulating hamiltonian dynamics*. 14. Cambridge university press, 2004.

[25] Ignacio A Martinez et al. "Engineered swift equilibration of a Brownian particle". In: *Nature physics* 12.9 (2016), pp. 843–846.

[26] Pierre Moral. *Feynman-Kac formulae: genealogical and interacting particle systems with applications*. Springer, 2004.

[27] Takahiro Ohkuma and Takao Ohta. "Fluctuation theorems for non-linear generalized Langevin systems". In: *Journal of Statistical Mechanics: Theory and Experiment* 2007.10 (2007), P10010.

[28] Ayoti Patra and Christopher Jarzynski. "Shortcuts to adiabaticity using flow fields". In: *New Journal of Physics* 19.12 (2017), p. 125009.

[29] Gabriel Peyre, Marco Cuturi, et al. "Computational optimal transport: With applications to data science". In: *Foundations and Trends® in Machine Learning* 11.5-6 (2019), pp. 355–607.

[30] Dries Sels and Anatoli Polkovnikov. "Minimizing irreversible losses in quantum systems by local counterdiabatic driving". In: *Proceedings of the National Academy of Sciences* 114.20 (2017), E3909–E3916.

[31] David A Sivak, John D Chodera, and Gavin E Crooks. "Using Nonequilibrium Fluctuation Theorems to Understand and Correct Errors in Equilibrium and Nonequilibrium Simulations of Discrete Langevin Dynamics". In: *Physical Review X* 3.1 (2013), p. 011007.

[32] Yifeng Tian, Nishant Panda, and Yen Ting Lin. "Liouville flow importance sampler". In: *arXiv preprint arXiv:2405.06672* (2024).

[33] Mark E Tuckerman. *Statistical mechanics: theory and molecular simulation*. Oxford university press, 2023.

[34] Suriyanarayanan Vaikuntanathan and Christopher Jarzynski. "Escorted free energy simulations". In: *The Journal of chemical physics* 134.5 (2011).

[35] Francisco Vargas et al. "Transport meets variational inference: Controlled monte carlo diffusions". In: *arXiv preprint arXiv:2307.01050* (2023).

[36] Udo Von Toussaint. "Bayesian inference in physics". In: *Reviews of Modern Physics* 83.3 (2011), pp. 943–999.

[37] Qinsheng Zhang and Yongxin Chen. "Path integral sampler: a stochastic control approach for sampling". In: *arXiv preprint arXiv:2111.15141* (2021).

[38] Adrianne Zhong, Ben Kuznets-Speck, and Michael R DeWeese. "Time-asymmetric fluctuation theorem and efficient free-energy estimation". In: *Physical Review E* 110.3 (2024), p. 034121.

A Counterdiabatic HMC as a Sequential Monte Carlo (SMC) sampler

Sequential Monte Carlo sampling Sequential Monte Carlo is a general framework for probabilistic inference of path measures. The *SMC sampler* approach (SMCS) [11] applies SMC to the problem of sampling from a distribution π on $\mathbb{R}^d$. See [13] for SMC generally, [26] for its relationship to Feynman-Kac processes, and [11, 10] for SMCS in particular. [21] provides a compact overview of all three.

SMCS requires a series of densities π_j, with π_0 an initial distribution which is easy to sample from and π_J the distribution of interest. We refer to the unnormalized density as γ_j, so that $\pi_j(x) = \frac{\gamma_j(x)}{\int \gamma_j(x)dx}$.

Markov kernels M_j and L_j must also be defined; L_j only play a conceptual role in the algorithm, but the M_j must be implemented.

The algorithm initializes a population of particles $\{x_0^i\}_{i=1}^N$, and at each step j propagates them by a kernel M_j to obtain $\{x_{j+1}^i\}_{i=1}^N$, and weights each particle x_{j+1}^i by:

$$w_j^i = \frac{\gamma_j(x_j^i)L_{j-1}(x_{j-1}^i|x_j^i)}{\gamma_{j-1}(x_{j-1}^i)M_j(x_j^i|x_{j-1}^i)} \tag{4}$$

A.1 SMCS with physical dynamics: using the nonequilibrium fluctuation theorem

In the case of interest to us, the variable in question is $z = (q, p)$, and the series of distributions is $\rho_{H_{\lambda(j\epsilon)}}(z)$, for integer j and ϵ the step size of the discretization.

Further, we are concerned with kernels defined by Hamiltonian dynamics, so that $M_j^H(z'|z) = \delta(z' - \phi_{H(j\epsilon)}(\epsilon)(z))$ and $L_t^H(z|z') - \delta(\tilde{z}' - \phi_{H((j-1)\epsilon)}(\epsilon)(\tilde{z}))$. Here, for $z = (q, p)$, $\tilde{z} \equiv (q, -p)$.

We have written $H(t)$ to denote $H_{\lambda(t)}$, with the understanding that t varies over the course of the flow. We first consider the case of exact Hamiltonian dynamics, and then discuss the effect of discretization.

In this case, the ratio of M and L can be calculated by the non-equilibrium fluctuation theorem of [9], which we now review. The value of this perspective is that this theorem can be generalized to various settings [27], including to account for counterdiabatic driving.

First, we recall that

$$\frac{dH}{dt} = \frac{\partial H}{\partial t} + \nabla H \cdot \frac{d\mathbf{z}}{dt} \equiv dW + dQ$$

In physical terms, dW and dQ are the work and heat respectively, and mathematically, should be viewed as one-forms that are not exact, i.e. are not the differential of a function. By contrast, their sum dH is an exact one-form, namely the differential of the Hamiltonian. In particular, the integral $\Delta H = \int_\alpha dH$ over a trajectory α is independent of α, but is composed of two parts, $W[\alpha] = \int_\alpha dW$ and $Q[\alpha] = \int_\alpha dQ$ which are both dependent on α, so that $\Delta H = W[\alpha] + Q[\alpha]$.

The fluctuation theorem states that

$$\frac{\rho_H(z) P_{z\to z'}}{\rho_H(z') P_{z\leftarrow z'}} = \exp(W - \Delta F)$$

where $P_{z\to z'}$ is the density over a given trajectory under the SDE in question (here Hamilton's equations of motion) from z to z', and where W is the integral of $dW = \frac{\partial H}{\partial t}$ over the trajectory, and $\Delta F = -\log \frac{Z_{\lambda(t)}}{Z_{\lambda(t-1)}}$, for $Z_\lambda \equiv \int \rho_{H_\lambda}(z) dz$.

In the case of Hamiltonian dynamics, the density of paths is precisely the delta function $\delta(z' - \phi_H(z))$, so a rearrangement gives us $w_j^i = e^{-W}$, where W is calculated over the trajectories of w^i. Further, we have that

$$\frac{M_j(z_j|z_{j-1})}{L_{j-1}(z_{j-1}|z_j)} = e^W \frac{Z_j}{Z_{j-1}} \frac{Z_{j-1}}{Z_j} e^{-\Delta H} = e^{W-\Delta H} = e^{-Q}$$

where $Z_j = Z_{\lambda(j\epsilon)}$. Since Hamiltonian dynamics are measure-preserving, i.e. $Q = 0$, we see immediately that when using an unadjusted HMC kernel in the SMC sampling algorithm, the correct weight is given by $\frac{\gamma_{j+1}}{\gamma_j} = e^{-\Delta H}$, in agreement with Appendix B of [10].

The more interesting case concerns kernels $M_j^{\mathrm{HC}_\lambda}$ arising from a counterdiabatic Hamiltonian $\mathrm{HC}_\lambda = H + \dot{\lambda} A$. In this case, [38] derives an analogous fluctuation theorem, but in terms of a different work, which we refer to as W_c, obtained by integrating dW_c defined as follows:

$$dW_c = \left(\frac{\partial H}{\partial t} + \nabla H \cdot v_A - \nabla \cdot v_A \right) dt$$

where v_A is the drift induced by $\dot{\lambda} A$, i.e. $v_A = (\nabla_q \dot{\lambda} A, -\nabla_p \dot{\lambda} A)$. We then see that the third term vanishes, since Hamiltonian flow is dissipationless.

In this case, we also note that the physical intepretation of heat and work is preserved, in the sense that

$$\frac{dH}{dt} = dW_c + dQ$$

where dQ is as defined above, i.e. $dQ = \nabla H \cdot v_H$, with $v_H = (\nabla_q H, -\nabla_p H)$.

A.2 Discretization

The above discussion assumes exact Hamiltonian dynamics, but in practice, we must numerically integrate. For Hamiltonian dynamics without counterdiabatic driving, the discretization can be viewed

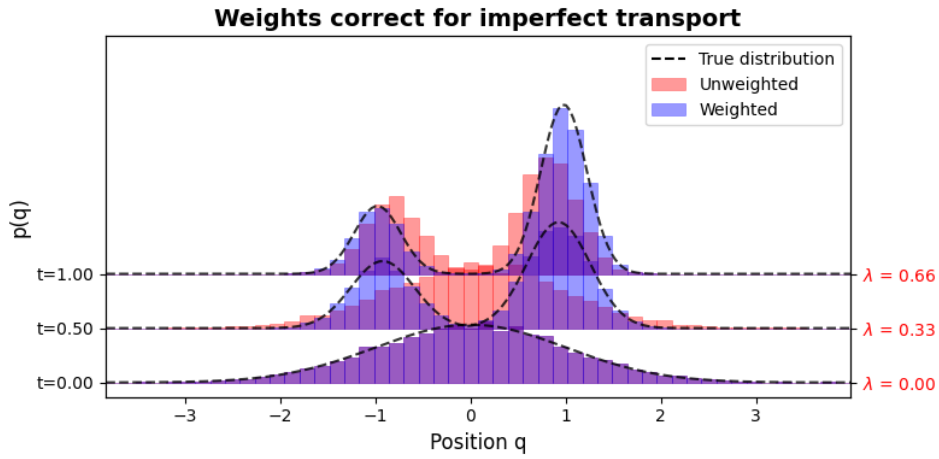

Figure 2: Here, the path of distributions is a geometric path between a standard normal and a mixture of two

as work [31]. This is because it is known that the discretized dynamics are the exact dynamics of a so-called shadow Hamiltonian H_S, so that the effect of discretization can be understood as an instantaneous change from H to H_S before the leapfrog step, and a corresponding change back afterwards. [31] refer to the work from the change of λ and from discretization as protocol work W_P and shadow work W_S respectively, with $\Delta H = W_S + W_P + Q$. Because the fluctuation theorem holds in the context of total work $W = W_S + W_P$, and because Hamiltonian dynamics have $Q = 0$, we have that $W = \Delta H$ and so $w \propto e^{-\Delta H}$.

The counterdiabatic case, without Langevin noise, can be approached similarly. In this case, the total work is $W = W_C + W_S = \Delta H$. The upshot is that the weight is the negative exponential of the change not in HC, but in the original H: $w \propto e^{-\Delta H}$.

This agrees with the weighting scheme in [1], in the special case of present interest where the flow is dissipationless.

As a demonstration of the weighting scheme working in practice, we show in Figure 2 a SMC scheme using the counterdiabatic kernel. Indeed, our implementation of CHMC in BlackJax uses the pre-existing SMC architecture.

B Example of simple counterdiabatic driving term

Suppose our time varying density $\pi_{\lambda(t)}$ is defined as a 1-dimensional Gaussian with unit variance and a time varying mean $\lambda(t)$, so that $\lambda(0) = 0$ and $\lambda(T) = 1$.

In this case, A is known, and is equal to p. To see this, calculate $\{p, H\} = -\frac{\partial H_\lambda}{\partial q} = -\frac{\partial V_\lambda}{\partial q}$, while $\partial_\lambda H_\lambda = -\frac{\partial V}{\partial q}$ also.

Then we have $\{A_\lambda, H_\lambda\} = \partial_\lambda H_\lambda$, which implies that $\mathbb{E}[\partial_\lambda H_\lambda] = \mathbb{E}[\{A, H\}] = \int \{A, H\}\rho_H = \int \{H, \rho_H\} A = 0$, by a standard property of the Poisson bracket.

A term p in the Hamiltonian generates motion towards the moving mean, so that probability mass is transported exactly along with the changing mean under the flow of the Hamiltonian.

It is also useful to consider the weight w in this setting, which as per Appendix A is defined as $e^{-\int dW_c}$, for

$$dW_c = \left(\frac{\partial H}{\partial t} + \nabla H \cdot v_A \right) dt$$

Algorithm 1 Fit adiabatic gauge potential A_ϕ at a given λ

Require: Real-valued function $A_\phi(q, p)$, optimizer (e.g. Adam) , schedule λ_i, population of particles $\{q_i, p_i\}_{i=1}^N$ and weights $w_{i=1}^N$, and T iterations.

for $t = 1, \ldots, T$ **do**
  Compute $l \leftarrow \frac{1}{N}\sum_{i=1}^N w_i \left(\{A_\phi, H_\lambda\}(q_i, p_i) - \partial_\lambda V_\lambda(q_i)\right)^2$
  Update ϕ using optimizer with loss l
end for
return ϕ

Since the transport is perfect, we know that the weights should be uniform, so the two terms in dW_c must cancel. Indeed, inspection shows that $\frac{\partial H}{\partial t} = -\dot{\lambda}(q - \lambda)$, while $\nabla H \cdot v_A = \dot{\lambda}(q - \lambda)$, giving the desired result.

C Implementation of Counterdiabatic Hamiltonian Monte Carlo

Here we review a number of details regarding the implementation of CHMC. Our implementation is in Jax, and for the learning of A we use the Adam optimizer. We use a 2 hidden layer feedforward neural network with ReLu activations, and hidden layers of 32 and 64. As an alternative, we implement a polynomial approximation, i.e. a sum of all polynomials in q and p up to a given order (our default is 5), where ϕ supplies the coefficients of each polynomial term. The latter is computationally cheaper to run and in low dimensions yields comparable results.

The code is available at `https://github.com/reubenharry/counterdiabatic-hmc/tree/working_branch`.

Choice of step size As in standard HMC, the choice of ϵ is a trade-off between the accuracy of the dynamics and the efficiency of the algorithm. We conjecture that adaptive tuning, as in tuning schemes for HMC step size [17] will prove useful. An advantage of the Hamiltonian setting is that large energy error can be used as a proxy for excessive step size.

Numerical Integration A key complication is that A_λ is not separable into a term depending on p and a term depending on q. This means that the Velocity Verlet integrator is not symplectic. In the current work, we overlook this problem and make sure to choose ϵ reasonably small.

Momentum refreshing In standard HMC, it is necessary to refresh momentum, either totally every n_l steps, or partially every step. The latter corresponds to a discretization of underdamped Langevin dynamics. In the case of CHMC, we find that momentum resampling is valuable. We use the strategy of refreshing every n_l steps, since the Langevin strategy results in dissipative dynamics which complicates the weight calculation. We fix $n_l = 2$, but note that performance is likely to be sensitive to this value, as in normal HMC, so we view automatically determining n_l as an important topic for future work.

For our experiments, we choose $\lambda(t) = 0.5t$. For the Gaussian Annealing and the Gaussian Moving Mean systems, we use $\epsilon = 2/3$ and 4 total timesteps, and for the Double Well, we use $\epsilon = 0.2$ and 10 timesteps. We use 1000 particles.

C.1 System Potentials

We evaluate our method on three one-dimensional systems with time-dependent potentials $V_\lambda(q)$.

C.1.1 Gaussian Moving Mean

The potential interpolates from a standard Gaussian to a Gaussian with shifted mean:

$$V_\lambda(q) = \frac{1}{2}(q - \lambda)^2 \tag{5}$$

This system evolves from $\mathcal{N}(0, 1)$ at $\lambda = 0$ to $\mathcal{N}(1, 1)$ at $\lambda = 1$.

Algorithm 2 Counterdiabatic HMC (CHMC) with population and weighting

Require: Population $\{q_i^{(0)}\}_{i=1}^N$ from initial distribution ; potential $V_\lambda(q) = -\nabla \log \pi_\lambda$ and gradients ; step size ε ; momentum refresh rate n ; schedule $\{\lambda_k\}_{k=1}^L$ with $\lambda_1 = 0$ (easy) and $\lambda_L = 1$ (target) .

Initialize weights $w_i^{(0)} = 1/N$ for all i
for $k = 1$ to $L - 1$ **do**
  for $i = 1$ to N **do**
    Set $(q, p) \leftarrow (q_i^{(k-1)}, p_i^{(k-1)})$
    If $k \mod n = 0$, sample $p \sim \mathcal{N}(0, I)$
    Update A_ϕ using the weighted population (Algorithm 1)
    Integrate equations of motion:

$$q_{\text{half}} \leftarrow q + \tfrac{\varepsilon}{2}\Big(p + \dot{\lambda}_k \, \nabla_p A_\phi(q, p)\Big)$$

$$p \leftarrow p - \varepsilon\Big(\nabla_q V_{\lambda_k}(q_{\text{half}}) + \dot{\lambda}_k \, \nabla_q A_\phi(q_{\text{half}}, p)\Big)$$

$$q \leftarrow q_{\text{half}} + \tfrac{\varepsilon}{2}\Big(p + \dot{\lambda}_k \, \nabla_p A_\phi(q_{\text{half}}, p)\Big)$$

    Increment work W:

$$W_i \leftarrow H_{\lambda_k}(q, p) - H_{\lambda_{k-1}}(q_i^{(k-1)}, p_i^{(k-1)})$$

    Update particle: $q_i^{(k)} \leftarrow q$
    Update weight: $w_i^{(k)} \leftarrow w_i^{(k-1)} \exp(-W_i)$
  end for
  Normalize weights: $w_i^{(k)} \leftarrow w_i^{(k)} / \sum_j w_j^{(k)}$
  Optionally resample if effective sample size is low
end for
return weighted population $\{(q_i^{(L)}, w_i^{(L)})\}_{i=1}^N$

C.1.2 Gaussian Annealing

The potential interpolates from a standard Gaussian to a Gaussian with reduced variance:

$$V_\lambda(q) = \frac{1}{2} k(\lambda) q^2 \tag{6}$$

where $k(\lambda) = 1 + 9\lambda$ interpolates from 1 to 10. This system evolves from $\mathcal{N}(0, 1)$ at $\lambda = 0$ to $\mathcal{N}(0, 0.1)$ at $\lambda = 1$.

C.1.3 Double Well

The potential interpolates from a harmonic oscillator to a double-well potential:

$$V_\lambda(q) = (1 - \lambda)\frac{1}{2} q^2 + \lambda \left((q^2 - 3)^2\right) \tag{7}$$

This system evolves from a single well centered at $q = 0$ to a double well with minima at $q = \pm\sqrt{3}$.

C.2 Kinetic Energy

All systems use the standard kinetic energy:

$$T(p) = \frac{p^2}{2m} \tag{8}$$

with $m = 1$.

D Learning A

To learn A satisfying property 3, it suffices to minimize $\mathbb{E}_{z\sim\rho_{H_{\lambda(t)}}}[|G(z)|^2]$, where

$$G(z) = \{A, H\}(z) - \partial_\lambda H(z) + \mathbb{E}[\partial_\lambda H(z)]$$

Multiplying out, we find $\mathbb{E}[|G(z)|^2] = \mathbb{E}[\{A,H\}^2 + (\partial_\lambda H)^2 + \mathbb{E}[\partial_\lambda H]^2 - 2\{A,H\}(\partial_\lambda H) + 2\{A,H\}\mathbb{E}[\partial_\lambda H] - 2(\partial_\lambda H)\mathbb{E}[\partial_\lambda H]] = \mathbb{E}[\{A,H\}^2 + (\partial_\lambda H)^2 - 2\{A,H\}(\partial_\lambda H) + 2\{A,H\}\mathbb{E}[\partial_\lambda H] - \mathbb{E}[\partial_\lambda H]^2]$.

The final term is irrelevant since it is constant with respect to A, while the penultimate term can be shown to vanish, since $\mathbb{E}[\{A,H\}] = \int \{A,H\}\rho_H = \int \{H, \rho_H\}A = 0$, by standard properties of the Poisson bracket.

This means that $\mathbb{E}[G(z)] = \mathbb{E}[\{A,H\}^2 + (\partial_\lambda H)^2 - 2\{A,H\}(\partial_\lambda H)] = \mathbb{E}[|\{A,H\} - \partial_\lambda H|^2] \equiv \mathcal{L}_H(A)$.

This means that we can minimize the loss $\mathcal{L}_H(A)$ instead, which is simple to calculate.

D.1 Choice of Gauge

The Hamiltonian HC_λ is presented in the main text as $H_\lambda + \dot{\lambda}A_\lambda$, but we are in fact free to choose instead of the first term H_λ, any function K of (q,p) which has $\{K, \rho_H\} = 0$. This is because, for a perfectly learned A, with $\epsilon \to 0$, such a term generates a flow under which ρ_H is stationary. In physical language this is known as a choice of *gauge*.

However, in the relevant regime, where ϵ is finite and A_λ is merely approximate, the question of the ideal gauge is open. Intuitively, using H_λ should allow the particles to (partially) re-equilibrate to the current stationary distribution at each step, and this is our justification for this choice. We also note that using H_λ is computationally cheap; although it requires the use of expensive gradients of the density π_λ, these are needed anyway in the fitting of A_λ, which happens subsequently.

E Hermite Polynomials

In our approach, we consider Hamiltonians of the form $H(q,p) = \frac{1}{2}p^2 + V(q,\lambda)$, so that the marginal stationary distribution over momentum is a Gaussian. This suggests an approach to parametrizing A more effectively, by making use of the orthogonal polynomials of the Gaussian distribution, which are the Hermite polynomials $\mathcal{H}_i$.

Concretely, we define $\phi_i := \mathcal{H}_i/\sqrt{i!}$, so that by a well-known property of the Hermite polynomials, $\mathbb{E}[\phi_m\phi_n] = \delta_{mn}$, where the expectation is taken with respect to the standard Normal distribution.

$$p\,\phi_i = \sqrt{i+1}\,\phi_{i+1} + \sqrt{i}\,\phi_{i-1}, \qquad \partial_p\phi_i = \sqrt{i}\,\phi_{i-1}, \qquad \phi_1(p) = p.$$

We proceed by defining A as follows:

$$A_{\theta,\tilde{\alpha}}(q,p) = f_\theta(q)\,g_{\tilde{\alpha}}(p), \qquad g_{\tilde{\alpha}}(p) = \sum_{\substack{i\geq 1\\ i \text{ odd}}} \tilde{\alpha}_i\,\phi_i(p).$$

Here, the parameters θ of f can be coefficients of polynomials or neural network weights, as before. Meanwhile, $\tilde{\alpha}$ are the parameters of g.

We now show that for fixed θ, $\arg\min_{\tilde{\alpha}} \mathcal{L}_H(A_{\theta,\tilde{\alpha}})$ is found by the solution of a tridiagonal linear system. That is:

Lemma E.1.

$$\mathcal{L}_H[\tilde{\alpha}] = b^\top\,\tilde{\alpha}^{(o)} \;+\; \tilde{\alpha}^{(o)\top} M\tilde{\alpha}^{(o)} + C.$$

where $\tilde{\alpha}^{(o)}$ is a vector of the odd coefficients, C is a constant, and b and M are a vector and matrix respectively to be specified below.

Proof. By definition,

$$\mathcal{L}_H[\tilde{\alpha}] \equiv \mathbb{E}\big[\|\partial_\lambda H - \{A,H\}\|^2\big] = \underbrace{\mathbb{E}[(\partial_\lambda H)^2]}_{\text{indep. of } \tilde{\alpha}} - 2\,\mathbb{E}[\partial_\lambda H\,\{A,H\}] \;+\; \mathbb{E}[\{A,H\}^2]$$

We now simplify, using the above properties of ϕ_i.

Linear term. By a standard property of the Poisson bracket:

$$-2\,\mathbb{E}[\partial_\lambda H\,\{A,H\}] = 2\,\mathbb{E}\big[\{\partial_\lambda H, H\}\,A\big] = 2\,\mathbb{E}\left[\underbrace{p\,\partial_q\partial_\lambda V(q)}_{\{\partial_\lambda H, H\}}\;\underbrace{f(q)}_{A\text{ (q-part)}}\;\underbrace{g(p)}_{A\text{ (p-part)}}\right].$$

Factoring and using $\phi_1 = p$, we have:

$$-2\,\mathbb{E}[\partial_\lambda H\,\{A,H\}] = 2\,\underbrace{\mathbb{E}_q\big[f(q)\,\partial_q\partial_\lambda V(q)\big]}_{=:L_q}\;\underbrace{\mathbb{E}_p\big[g(p)\,p\big]}_{=\sum_{i\text{ odd}}\tilde{\alpha}_i\,\mathbb{E}_p[\phi_i\phi_1]=\tilde{\alpha}_1}.$$

Hence

$$\boxed{\text{Linear term} \;=\; 2\,\tilde{\alpha}_1\,\mathbb{E}_q\big[f(q)\,\partial_q\partial_\lambda V(q)\big] \equiv 2\,\tilde{\alpha}_1\,L_q.}$$

Quadratic term. Expanding the Poisson bracket, we find:

$$\{A,H\} = (\partial_q A)\,(\partial_p H) - (\partial_p A)\,(\partial_q H) = \underbrace{(\partial_q f)}_{=:Q(q)}\;\underbrace{g(p)\,p}_{p\phi_i=\sqrt{i+1}\phi_{i+1}+\sqrt{i}\phi_{i-1}} - \underbrace{f(q)}\;\underbrace{g'(p)}_{\sum_i\tilde{\alpha}_i\sqrt{i}\,\phi_{i-1}}\;\underbrace{\partial_q V(q)}.$$

Using the identities (with sums over odd i):

$$\{A,H\} \;=\; \sum_{\substack{i\geq 1\\ i\text{ odd}}} \tilde{\alpha}_i\Big[\underbrace{Q(q)}_{=:Q}\,\sqrt{i+1}\,\phi_{i+1}(p) \;+\; \sqrt{i}\,\underbrace{\big(Q(q) - f(q)\,\partial_q V(q)\big)}_{=:R}\,\phi_{i-1}(p)\Big].$$

It then follows that

$$\mathbb{E}[\{A,H\}^2] = \sum_{i,j\text{ odd}} \tilde{\alpha}_i\tilde{\alpha}_j\Big\{\underbrace{\mathbb{E}_q[Q^2]}_{=:c_0}\,\mathbb{E}_p\big[\sqrt{i+1}\,\phi_{i+1}\,\sqrt{j+1}\,\phi_{j+1}\big] + \underbrace{\mathbb{E}_q[R^2]}_{=:c_1}\,\mathbb{E}_p\big[\sqrt{i}\,\phi_{i-1}\,\sqrt{j}\,\phi_{j-1}\big]$$
$$+\underbrace{\mathbb{E}_q[QR]}_{=:c_2}\,\mathbb{E}_p\big[\sqrt{i}\,\phi_{i-1}\,\sqrt{j+1}\,\phi_{j+1}\big] + c_2\,\mathbb{E}_p\big[\sqrt{i+1}\,\phi_{i+1}\,\sqrt{j}\,\phi_{j-1}\big]\Big\}.$$

Using orthonormality of ϕ_i, we have:

$$\boxed{\mathbb{E}[\{A,H\}^2] = \sum_{i\text{ odd}} \tilde{\alpha}_i^2\Big[c_0\,(i+1) + c_1\,i\Big] \;+\; 2\sum_{i\text{ odd}} \tilde{\alpha}_i\tilde{\alpha}_{i+2}\,c_2\,\sqrt{(i+1)(i+2)},}$$

with

$$\boxed{c_0 = \mathbb{E}_q[(\partial_q f)^2], \qquad c_1 = \mathbb{E}_q[(\partial_q f - f\,\partial_q V)^2], \qquad c_2 = \mathbb{E}_q[(\partial_q f)\,(\partial_q f - f\,\partial_q V)].}$$

Odd-sector tridiagonal form. Reindex $i = 2k+1$ and set $\tilde{\alpha}_k^{(o)} := \tilde{\alpha}_{2k+1}$ for $k = 0, 1, 2, \ldots$. Then

$$\boxed{\mathbb{E}[\{A,H\}^2] = \sum_{k\geq 0} \big(\tilde{\alpha}_k^{(o)}\big)^2\Big[c_0\,(2k+2) + c_1\,(2k+1)\Big] \;+\; 2\sum_{k\geq 0} \tilde{\alpha}_k^{(o)}\,\tilde{\alpha}_{k+1}^{(o)}\,c_2\,\sqrt{(2k+2)(2k+3)}\;,}$$

which is a *tridiagonal* quadratic form in the vector $(\tilde{\alpha}_0^{(o)}, \tilde{\alpha}_1^{(o)}, \tilde{\alpha}_2^{(o)}, \ldots)$ with entries

$$M_{k,k} = c_0\,(2k+2) + c_1\,(2k+1), \qquad M_{k,k+1} = M_{k+1,k} = c_2\,\sqrt{(2k+2)(2k+3)}.$$

Collecting terms. Let $b_k := 2\, L_q\, \mathbf{1}_{\{k=0\}}$ with $L_q = \mathbb{E}_q[f\, \partial_q \partial_\lambda V]$. Then

$$\boxed{\mathcal{L}_H[\tilde{\alpha}] = \underbrace{\mathbb{E}[(\partial_\lambda H)^2]}_{\text{constant}} \;+\; \sum_{k\geq 0} b_k\, \tilde{\alpha}_k^{(o)} \;+\; \tilde{\alpha}^{(o)\top} M \tilde{\alpha}^{(o)}.}$$

□