DIFFERENTIABLE MOLECULAR SIMULATIONS FOR CONTROL AND LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Molecular simulations use statistical mechanics at the atomistic scale to enable both the elucidation of fundamental mechanisms and the engineering of matter for desired tasks. Non-quantized molecular behavior is typically simulated with differential equations parameterized by a Hamiltonian, or energy function. The Hamiltonian describes the state of the system and its interactions with the environment. In order to derive predictive microscopic models, one wishes to infer a molecular Hamiltonian from macroscopic quantities. From the perspective of engineering, one wishes to control the Hamiltonian to achieve desired macroscopic quantities. In both cases, the goal is to modify the Hamiltonian such that bulk properties of the simulated system match a given target. We demonstrate how this can be achieved using differentiable simulations where bulk target observables and simulation outcomes can be analytically differentiated with respect to Hamiltonians. Our work opens up new routes for parameterizing Hamiltonians to infer macroscopic models and develops control protocols

1 INTRODUCTION

At the atomic level, physical processes are governed by differential equations containing many degrees of freedom. Macroscopic phenomena in matter emerge from microscopic interactions that can be simulated through numerically integrating the equations of motion. In classical simulations, these equations of motion are derived from a Hamiltonian function. In quantum simulations, they are derived from a Hamiltonian operator. Examples of microscopic quantities emerging from simulations are time series of positions, velocities, and forces on atoms and molecules. From these, a rich family of macroscopic observables can be calculated to describe the configurational and temporal correlation functions of atoms.

Classically, simulating the positions of points that preserves energy requires integrating the Hamiltonian equations of motions:

$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i} \qquad \frac{dq_i}{dt} = \frac{\partial H}{\partial p_i}, \tag{1}$$

where p_i and q_i are the respective momentum and position of the i^{th} particle. H is the Hamiltonian of the systems; for conservative systems, it is given by the sum of kinetic energy and the potential energy,

$$H(\mathbf{p}, \mathbf{q}) = U(\mathbf{q}) + \sum_i^N \frac{p_i^2}{2m_i}, \tag{2}$$

where boldface denotes the set of quantities for all particles, $U(\mathbf{q})$ is the potential energy and $p_i^2/(2m_i)$ is the kinetic energy of the i^{th} particle.

Simulating an entire system containing all degrees of freedom is computationally intractable. Typically one is interested in a small subset of a system, such as a molecule or protein, and concerned only with the influence of the environment on the system, but not the details of the environment itself. For this reason, one usually incorporates an environment Hamiltonian H_b with coarse-grained

macroscopic variables of interest into the original Hamiltonian: $H_{tot} = H + H_b$. The inclusion of H_b is important in simulating systems under certain thermodynamic conditions. For example, the crystallization and melting of water occur under constant pressure conditions. These conditions are imposed by the environment, which must therefore be incorporated into H_b. The environment, and therefore H_b, can also be explicitly controlled and optimized in an experiment. For example, H_b can represent an external laser that is varied to control chemical reaction dynamics.

Recent advances in differentiable solvers have shown that *differentiable simulations* may be performed, in which the result of a simulation may be analytically differentiated with respect to its inputs (1; 2; 3; 4; 5; 6). Our paper demonstrates the use of differentiable simulations in the context of molecular simulation. We show that a Hamiltonian can be learned such that the macroscopic observables computed through simulation trajectory to match a given target. This is done through automatic differentiation of the macroscopic observables computed from simulation trajectory with respect to the system Hamiltonian. Moreover, we show that the same principles can be used to control the system Hamiltonian to force the system towards a target state.

2 APPROACH

2.1 MOLECULAR SIMULATIONS

Molecular simulations with a control Hamiltonian In this work, we demonstrate the use of automatic differentiation in molecular dynamics simulations. To simulate systems under under fixed thermodynamic conditions with macroscopic thermodynamic controls, most applications require fixed temperature, pressure, or strain. A typical way to impose thermodynamic constraints is to introduce modified equations of motion that contain virtual external variables. For example, to control the temperature of a systems, a virtual heat bath variable is needed to couple to the system variables (7; 8). These modified equations of motion can be integrated using ODE solvers. In the Appendix we present the differentiable form of the Nose-Hover chain, a common method for imposing constant temperature conditions. We choose this integrator to represent realistic dynamics of particle systems at constant temperature for our experiments.

We also demonstrate the use of a graph neural network to represent a control Hamiltonian H_b in one of our experiments. Graph neural networks (GNN) learn the molecular potential energy function (9; 10; 11) while preserving physical symmetries (transnational, rotational and permutational). GNNs have shown flexible fitting power to represent molecular energies, and can be used to model control Hamiltonians (see Appendix).

Quantum dynamics Manipulation of the phase of light to control quantum behaviour, known as coherent control (12; 13), has been used to control the out-of-equilibrium behavior of various physical processes (14; 15; 16; 17; 18). A prototypical example of such control is of the retinal chromophore, a molecule whose light-induced isomerization controls human vision (19), light-sensing (20), and bacterial proton pumping (21). Computational and theoretical analysis of coherent control can explain and motivate experimental control protocols (22; 23; 24; 25). In Ref. (24), for example, an experimental control protocol was reproduced computationally and explained theoretically. A minimal model for retinal (26) was used to simulate the quantum yield (isomerization efficiency), and a genetic algorithm was used to shape the incident pulse. With the minimal retinal model used commonly in the literature (27; 28; 29; 24), we here analyze the control of the incident pulse through differentiable simulations. In particular, we show that back-propagation through a quantum simulation can be used to shape the incident pulse and control the isomerization quantum yield. This provides an example of out-of-equilibrium molecular control through backpropagation.

2.2 BACK-PROPAGATION WITH THE ADJOINT METHOD

To be able to differentiate molecular simulations to reach a control target, we adopt the reverse-mode automatic differentiation method from Chen *et al.*, which uses adjoint sensitivity methods (30; 1). Taking derivatives requires computation of the adjoint state $a(t) = dL/d(p(t), q(t))$. Evaluating the loss requires the reverse-time integration of the vector-Jacobian product:

$$\frac{dL}{d\theta} = \int_{t_{i+1}}^{t_i} a(t) \frac{df(q(t), p(t), \theta)}{d\theta} dt, \tag{3}$$

where $f(q, p, t)$ represents the Hamiltonian ODE defined in Eq. 1. The reverse-mode automatic differentiation computes the gradient through the adjoint states without backpropagating through the fowardpass coputations in the ODE solver. This has the advantage of constant memory cost. The ability to output positions and momenta at individual timesteps allow one to directly compute observables and correlation functions from a trajectory. Separate reverse adjoint integrations are performed for observations at different individual observation times.

3 Control Protocol for Molecular Quantum Dynamics

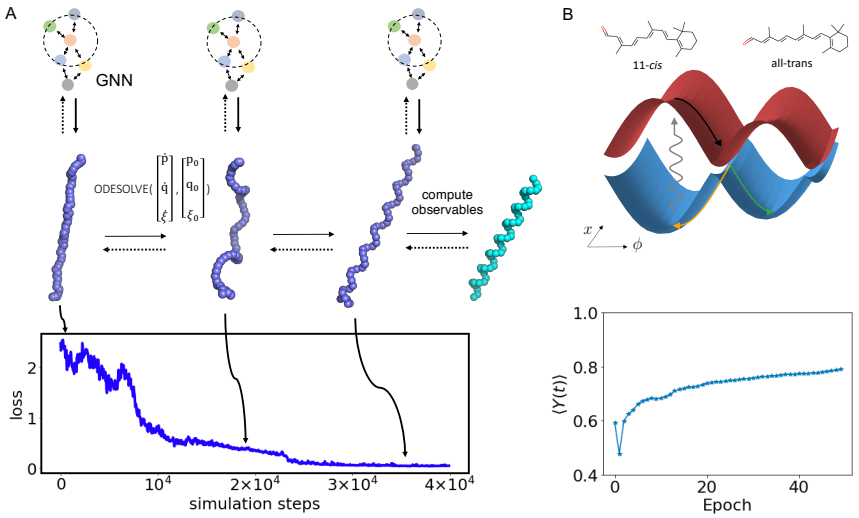

Figure 1: A We perform continuous model training during the simulations to bias a harmonic polymer chain toward a targeted helix shape. This is done by training a bias Hamiltonian parameterized by a GNN. We run the simulations for 4000 steps, and the loss is computed and differentiated to update GNN weights every 40 simulation steps. **B** Controlled isomerization of the model retinal Hamiltonian with a time-dependent electric field. The model consists of two electronic states, denoted with blue and red, a vibrational mode x, and a torsional mode ϕ. We show that the time-averaged quantum yield increases as a function of training epoch with differentiable control.

We use the model introduced in Ref. (26) for the retinal chromophore. The model Hamiltonian consists of two diabatic electronic states, a single torisonal mode ϕ for the isomerizing double bond, and a single stretching mode x (see Fig. 1B). Details of the model, the construction of the Hamiltonian and the operators of interest can be found in Refs. (26; 29; 27).

The total Hamiltonian of the system and the control field is given by

$$\hat{H}(t) = \hat{H}_{\mathrm{S}} + \hat{H}_b = \hat{H}_{\mathrm{S}} - \hat{\mu}E(t), \tag{4}$$

where $\hat{H}_{\mathrm{S}}$ is the system Hamiltonian, $\hat{H}_b$ is the control Hamiltonian, $\hat{\mu}$ is the dipole operator, and $E(t)$ is the electric field. The quantity to be optimized is the quantum yield, i.e. the efficiency of isomerization Y (the higher the better; further details are provided in the Appendix). During simulations, we backpropagate through the simulation trajectory to optimize both the magnitude and phase of the temporal electric field. The numerical results are shown in Fig. 1 B. The quantum yield begins at approximately 0.6, and after 50 epochs reaches 0.8 as the electric field is improved. These results show that differentiable simulations can be used to learn control protocols for electric fields driven isomerization.

4 Controlling Molecular Dynamics

Learning a bias control Hamiltonians is useful in performing pertubative free energy calculations and sampling rare events between two states in the state space (31; 32). Such bias control Hamiltonians are useful in both non-equilibrium and equilibrium cases as generalized by the Jarzynski equality (33; 34).

We show that differentiable molecular simulations allow us to control dynamics by pertubatively seeking transitions from a starting state to a target state. This learning scheme does not require a choice of reaction coordinate, as required by biasing potential methods like adaptive force biasing (35). We trained a GNN to represent H_b, to bias a linear chain with a harmonic Hamiltonian into a helix fold (see Fig. 1). The polymer simulations are performed at a constant temperature using the Nose-Hoover Chain integrator described above and in the Appendix. We back-propagate through the simulations during the course of the simulations to continuously update the GNN so that the loss function $L = \sum_i^{observables} (\phi_i(q(t_1)) - \phi_i(q_{helix}))^2$ is minimized where the set of functions ϕ_i include structural variables of the polymer chain: bond distances, angles and dihedrals angles.

5 Learning from observables

We demonstrate an example of fitting pair correlations in Lennard Jones (LJ) systems (Fig. 2). Pair correlation functions characterize structural and thermodynamic properties of condensed phase systems (36). We demonstrate that by differentiating through the simulation trajectories, one can actively modify parameters to match a target distributions function. To make the distribution function our differentiable target, we implement a differentiable histogram to approximate the typical non-differentiable histogram operation. This is done by summing over pair distances expanded in a basis of Gaussians ("Gaussian smearing"), followed by normalization to ensure that the histogram integration yields the total number of pair distances (see Appendix).

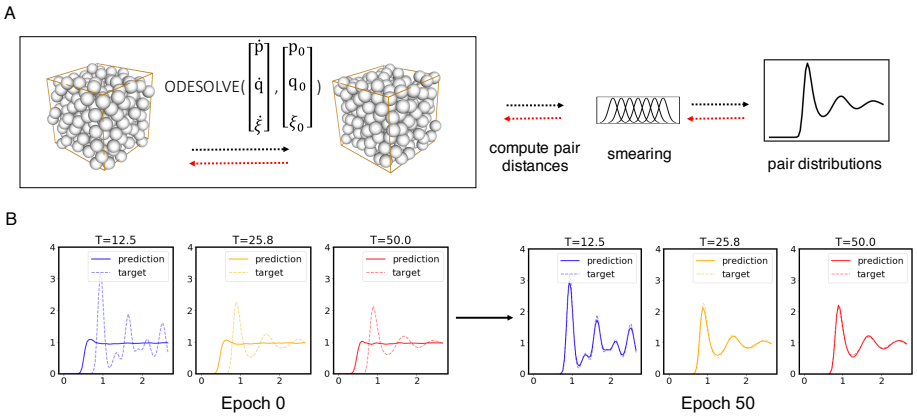

Figure 2: Computational workflow to fit pair distribution functions for an LJ system. The target distributions functions are obtained with σ = 1 and ϵ = 1. For each training epoch, we compute the pair distribution functions from simulated trajectories. We back-propagate the mean square loss between simulated and target pair distribution functions to update the LJ parameters.

We set up the experiment to learn the parameters of a Lennard Jones liquid (36) with pair interactions: $u_{ij} = 4\epsilon((\frac{\sigma}{r})^{12} - (\frac{\sigma}{r})^6)$, with the total potential energy of the system given by $U_{total} = \sum_{i \neq j} u_{ij}(r_{ij})$. We simulate the system at three different temperatures and obtain three different pair correlation functions $g(r)$. For each temperature the model learns to reproduce the same radial distribution function by minimizing the mean square loss between the simulated pair distribution function and the target. At each epoch, the parameters are updated with gradient descent.

6 Conclusions

In this work we proposed a framework for training molecular simulations based on macroscopic quantities to develop learning and control protocols. Our method is based on model learning through simulation time feedback from bulk observables. Our method also opens up new possibilities for designing control protocols for equilibrium and non-equilibrium simulations by incorporating bias Hamiltonians. This work can be extended to the simulation of other types of molecular systems with different thermodynamic boundary conditions and different control scenarios.

7 Related Work

Differentiable Simulations Several works have incorporated physics-based simulations to control and infer movements of mechanical objects. These are done by incorporating inductive biases that obey Hamiltonian dynamics (37; 38). Many works also focus on performing model control over dynamical systems (3; 39; 40). Differentiable simulations with automatic differentiation have also been utilized in constructing models from data in many differential equation settings like computational fluid dynamics (41), physics simulations (42; 43; 4), quantum chemistry (44), protein simulations (45) and normalizing flows (46; 47). Much progress has been made in developing differentiable frameworks for molecular dynamics (48), PDEs (49; 50; 51; 2) and ODEs (1).

Statistical Physics and Molecular Dynamics In machine learning for molecular dynamics, automatic differentiation has been applied in analyzing latent structure of molecular kinetics (52; 53; 54; 55; 56; 57), fitting models from quantum chemistry calculations (10; 58; 59; 60; 61), model reduction of atomistic simulations (2; 62; 63; 64; 65) and advanced sampling (66). For the computation of free energy in simulations, adaptive methods and variational methods have been proposed with bias Hamiltonians on a specified reaction coordinate (35), or invertible transformations (31) between initial and target configurations. For non-equilibrium simulations with finite-time driving forces, variational methods with gradients have also been applied in the computation of large deviation functions to better sample rare events (67; 68; 69).

References

[1] Chen, R. T. Q., Rubanova, Y., Bettencourt, J. & Duvenaud, D. Neural Ordinary Differential Equations. *Advances in neural information processing systems* (2018). URL https://arxiv.org/pdf/1806.07366.pdfhttp://arxiv.org/abs/1806.07366. 1806.07366.

[2] Lu, P. Y., Kim, S. & Soljačić, M. Extracting Interpretable Physical Parameters from Spatiotemporal Systems using Unsupervised Learning (2019). URL http://arxiv.org/abs/1907.06011. 1907.06011.

[3] Li, Y., He, H., Wu, J., Katabi, D. & Torralba, A. Learning Compositional Koopman Operators for Model-Based Control (2019). URL http://arxiv.org/abs/1910.08264. 1910.08264.

[4] Liang, J., Lin, M. & Koltun, V. Differentiable Cloth Simulation for Inverse Problems. *Advances in Neural Information Processing Systems 32* 771–780 (2019). URL http://papers.nips.cc/paper/8365-differentiable-cloth-simulation-for-inverse-problems.pdf.

[5] Holl, P., Koltun, V. & Thuerey, N. Learning to Control PDEs with Differentiable Physics (2020). URL http://arxiv.org/abs/2001.07457. 2001.07457.

[6] Lu, L., Meng, X., Mao, Z. & Karniadakis, G. E. DeepXDE: A deep learning library for solving differential equations (2019). URL http://arxiv.org/abs/1907.04502. 1907.04502.

[7] Nosé, S. A unified formulation of the constant temperature molecular dynamics methods. *The Journal of Chemical Physics* **81**, 511–519 (1984). URL http://aip.scitation.org/doi/10.1063/1.447334.

[8] Martyna, G. J., Klein, M. L. & Tuckerman, M. Nosé-Hoover chains: The canonical ensemble via continuous dynamics. *The Journal of Chemical Physics* **97**, 2635–2643 (1992). URL http://aip.scitation.org/doi/10.1063/1.463940.

[9] Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. Neural Message Passing for Quantum Chemistry. *arXiv:1704.01212* (2017). URL http://arxiv.org/abs/1704.01212http://www.arxiv.org/pdf/1704.01212.pdf. 1704.01212.

[10] Schütt, K. T. *et al.* SchNet: A continuous-filter convolutional neural network for modeling quantum interactions (2017). URL http://arxiv.org/abs/1706.08566. 1706.08566.

[11] Duvenaud, D. K. *et al.* Convolutional Networks on Graphs for Learning Molecular Fingerprints. In *Advances in Neural Information Processing Systems*, 2215–2223 (2015). URL http://arxiv.org/abs/1509.09292. 1509.09292.

[12] Shapiro, M. & Brumer, P. Principles of the quantum control of molecular processes. *Principles of the Quantum Control of Molecular Processes, by Moshe Shapiro, Paul Brumer, pp. 250. ISBN 0-471-24184-9. Wiley-VCH, February 2003.* 250 (2003).

[13] Judson, R. S. & Rabitz, H. Teaching lasers to control molecules. *Physical review letters* **68**, 1500 (1992).

[14] Zhu, L. *et al.* Coherent laser control of the product distribution obtained in the photoexcitation of hi. *Science* **270**, 77–80 (1995).

[15] Stievater, T. *et al.* Rabi oscillations of excitons in single quantum dots. *Physical Review Letters* **87**, 133603 (2001).

[16] Brinks, D. *et al.* Visualizing and controlling vibrational wave packets of single molecules. *Nature* **465**, 905–908 (2010).

[17] Haché, A. *et al.* Observation of coherently controlled photocurrent in unbiased, bulk gaas. *Physical Review Letters* **78**, 306 (1997).

[18] Warren, W. S., Rabitz, H. & Dahleh, M. Coherent control of quantum dynamics: the dream is alive. *Science* **259**, 1581–1589 (1993).

[19] Palczewski, K. Chemistry and biology of vision. *Journal of Biological Chemistry* **287**, 1612–1619 (2012).

[20] Briggs, W. R. & Spudich, J. L. *Handbook of photosensory receptors* (John Wiley & Sons, 2005).

[21] Schulten, K. & Tavan, P. A mechanism for the light-driven proton pump of halobacterium halobium. *Nature* **272**, 85–86 (1978).

[22] Lavigne, C. & Brumer, P. Ultrashort pulse two-photon coherent control of a macroscopic phenomena: light-induced current from channelrhodopsin-2 in live brain cells. *arXiv preprint arXiv:1907.07741* (2019).

[23] Lavigne, C. & Brumer, P. Considerations regarding one-photon phase control. *arXiv preprint arXiv:1910.13878* (2019).

[24] Lavigne, C. & Brumer, P. Interfering resonance as an underlying mechanism in the adaptive feedback control of radiationless transitions: Retinal isomerization. *The Journal of chemical physics* **147**, 114107 (2017).

[25] Pachón, L. A. & Brumer, P. Mechanisms in environmentally assisted one-photon phase control. *The Journal of chemical physics* **139**, 164123 (2013).

[26] Hahn, S. & Stock, G. Quantum-mechanical modeling of the femtosecond isomerization in rhodopsin. *The Journal of Physical Chemistry B* **104**, 1146–1149 (2000).

[27] Axelrod, S. & Brumer, P. Multiple time scale open systems: Reaction rates and quantum coherence in model retinal photoisomerization under incoherent excitation. *The Journal of chemical physics* **151**, 014104 (2019).

[28] Tscherbul, T. V. & Brumer, P. Excitation of biomolecules with incoherent light: Quantum yield for the photoisomerization of model retinal. *The Journal of Physical Chemistry A* **118**, 3100–3111 (2014).

[29] Tscherbul, T. V. & Brumer, P. Quantum coherence effects in natural light-induced processes: cis–trans photoisomerization of model retinal under incoherent excitation. *Physical Chemistry Chemical Physics* **17**, 30904–30913 (2015).

[30] Chambers, M. L. *et al.* The Mathematical Theory of Optimal Processes. *OR* **16**, 493 (1965).

[31] Jarzynski, C. Targeted free energy perturbation. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics* **65**, 5 (2001). URL `http://arxiv.org/abs/cond-mat/0109461http://dx.doi.org/10.1103/PhysRevE.65.046122. 0109461`.

[32] Rotskoff, G. M. & Vanden-Eijnden, E. Dynamical computation of the density of states using nonequilibrium importance sampling. *Physical Review Letters* **122** (2018). URL `http://arxiv.org/abs/1809.11132http://dx.doi.org/10.1103/PhysRevLett.122.150602. 1809.11132`.

[33] Jarzynski, C. Equilibrium free-energy differences from nonequilibrium measurements: A master-equation approach. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics* **56**, 5018–5035 (1997). `9707325`.

[34] Jarzynski, C. Nonequilibrium equality for free energy differences. *Physical Review Letters* **78**, 2690–2693 (1997). `9610209`.

[35] Darve, E., Rodríguez-Gómez, D. & Pohorille, A. Adaptive biasing force method for scalar and vector free energy calculations. *Journal of Chemical Physics* **128**, 144120 (2008). URL http://aip.scitation.org/doi/10.1063/1.2829861.

[36] Jones, J. E. On the Determination of Molecular Fields. II. From the Equation of State of a Gas. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **106**, 463–477 (1924).

[37] Greydanus, S., Dzamba, M. & Yosinski, J. Hamiltonian Neural Networks (2019). URL http://arxiv.org/abs/1906.01563. 1906.01563.

[38] Sanchez-Gonzalez, A., Bapst, V., Cranmer, K. & Battaglia, P. Hamiltonian Graph Networks with ODE Integrators (2019). URL http://arxiv.org/abs/1909.12790. 1909.12790.

[39] Battaglia, P. W., Pascanu, R., Lai, M., Rezende, D. & Kavukcuoglu, K. Interaction Networks for Learning about Objects, Relations and Physics. *Advances in Neural Information Processing Systems* 4509–4517 (2016). URL http://arxiv.org/abs/1612.00222. 1612.00222.

[40] Zhong, Y. D., Dey, B. & Chakraborty, A. Symplectic ODE-Net: Learning Hamiltonian Dynamics with Control (2019). URL http://arxiv.org/abs/1909.12077. 1909.12077.

[41] Schenck, C. & Fox, D. SPNets: Differentiable Fluid Dynamics for Deep Neural Networks (2018). URL http://arxiv.org/abs/1806.06094. 1806.06094.

[42] Hu, Y. *et al.* ChainQueen: A real-time differentiable physical simulator for soft robotics. In *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, 6265–6271 (Institute of Electrical and Electronics Engineers Inc., 2019). 1810.01054.

[43] Hu, Y. *et al.* DiffTaichi: Differentiable Programming for Physical Simulation (2019). URL http://arxiv.org/abs/1910.00935. 1910.00935.

[44] Tamayo-Mendoza, T., Kreisbeck, C., Lindh, R. & Aspuru-Guzik, A. Automatic Differentiation in Quantum Chemistry with Applications to Fully Variational Hartree-Fock. *ACS Central Science* **4**, 559–566 (2018). URL https://pubs.acs.org/doi/10.1021/acscentsci.7b00586. 1711.08127.

[45] Ingraham, J., Riesselman, A., Sander, C. & Marks, D. Learning Protein Structure with a Differentiable Simulator. In *International Conference on Learning Representations* (2019).

[46] Zhang, L., E, W. & Wang, L. Monge-Amp\‘ere Flow for Generative Modeling (2018). URL http://arxiv.org/abs/1809.10188. 1809.10188.

[47] Grathwohl, W., Chen, R. T., Bettencourt, J., Sutskever, I. & Duvenaud, D. Ffjord: Free-form continuous dynamics for scalable reversible generative models. In *7th International Conference on Learning Representations, ICLR 2019* (International Conference on Learning Representations, ICLR, 2019). 1810.01367.

[48] Schoenholz, S. S. & Cubuk, E. D. JAX, M.D.: End-to-End Differentiable, Hardware Accelerated, Molecular Dynamics in Pure Python (2019). URL http://arxiv.org/abs/1912.04232. 1912.04232.

[49] Han, J., Jentzen, A. & Weinan, E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences of the United States of America* **115**, 8505–8510 (2018). URL www.pnas.org/cgi/doi/10.1073/pnas.1718942115. 1707.02568.

[50] Long, Z., Lu, Y., Ma, X. & Dong, B. PDE-Net: Learning PDEs from Data. *35th International Conference on Machine Learning, ICML 2018* **7**, 5067–5078 (2017). URL http://arxiv.org/abs/1710.09668. 1710.09668.

[51] Long, Z., Lu, Y. & Dong, B. PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network. *Journal of Computational Physics* **399** (2019). URL http://arxiv.org/abs/1812.04426http://dx.doi.org/10.1016/j.jcp.2019.108925. 1812.04426.

[52] Mardt, A., Pasquali, L., Wu, H. & Noé, F. VAMPnets for deep learning of molecular kinetics. *Nature Communications* **9**, 5 (2018). URL http://www.nature.com/articles/s41467-017-02388-1.

[53] Wu, H., Mardt, A., Pasquali, L. & Noe, F. Deep Generative Markov State Models. *arXiv:1805.07601* (2018). URL http://arxiv.org/abs/1805.07601.

[54] Wehmeyer, C. & Noé, F. Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics. *The Journal of Chemical Physics* **148**, 241703 (2018). URL https://arxiv.org/pdf/1710.11239.pdfhttp://arxiv.org/abs/1710.11239http://aip.scitation.org/doi/10.1063/1.5011399. 1710.11239.

[55] Ceriotti, M. Unsupervised machine learning in atomistic simulations, between predictions and understanding (2019). 1902.05158.

[56] Post, M., Wolf, S. & Stock, G. Principal component analysis of nonequilibrium molecular dynamics simulations. *Journal of Chemical Physics* **150** (2019). 1903.08105.

[57] Wang, Y., Ribeiro, J. M. L. & Tiwary, P. Past–future information bottleneck for sampling molecular reaction coordinate simultaneously with thermodynamics and kinetics. *Nature Communications* **10**, 1–8 (2019).

[58] Yao, K., Herr, J. E., Toth, D., Mckintyre, R. & Parkhill, J. The TensorMol-0.1 model chemistry: a neural network augmented with long-range physics. *Chemical Science* **9**, 2261–2269 (2018). URL http://xlink.rsc.org/?DOI=C7SC04934J.

[59] Behler, J. & Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical Review Letters* **98**, 146401 (2007).

[60] Mailoa, J. P. *et al.* A fast neural network approach for direct covariant forces prediction in complex multi-element extended systems. *Nature Machine Intelligence* **1**, 471–479 (2019). 1905.02791.

[61] Zhang, L., Han, J., Wang, H., Car, R. & Weinan, E. Deep Potential Molecular Dynamics: A Scalable Model with the Accuracy of Quantum Mechanics. *Physical Review Letters* **120**, 143001 (2018). 1707.09571.

[62] Ma, C., Wang, J. & E, W. Model Reduction with Memory and the Machine Learning of Dynamical Systems. *arxiv:1808.04258* (2018). URL http://arxiv.org/abs/1808.04258.

[63] Wang, W. & Gómez-Bombarelli, R. Coarse-graining auto-encoders for molecular dynamics. *npj Computational Materials* **5** (2019). URL http://dx.doi.org/10.1038/s41524-019-0261-5.

[64] Wang, J. *et al.* Machine Learning of coarse-grained Molecular Dynamics Force Fields. *arXiv:1812.01736* acscentsci.8b00913 (2018). URL http://pubs.acs.org/doi/10.1021/acscentsci.8b00913http://arxiv.org/abs/1812.01736.

[65] Zhang, L., Han, J., Wang, H., Car, R. & E, W. DeePCG: Constructing coarse-grained models via deep neural networks. *The Journal of Chemical Physics* **149**, 34101 (2018).

[66] Noé, F. & Wu, H. Boltzmann Generators-Sampling Equilibrium States of Many-Body Systems with Deep Learning. *arXiv:1812.01729* (2018). URL https://arxiv.org/pdf/1812.01729.pdf.

[67] Nguyen, M. & Vaikuntanathan, S. Design principles for nonequilibrium self-assembly. *Proceedings of the National Academy of Sciences of the United States of America* **113**, 14231–14236 (2016). 1507.08971.

[68] Das, A. & Limmer, D. T. Variational control forces for enhanced sampling of nonequilibrium molecular dynamics simulations. *Journal of Chemical Physics* **151**, 244123 (2019). 1909.03589.

[69] Dolezal, J. & Jack, R. L. Large deviations and optimal control forces for hard particles in one dimension. *Journal of Statistical Mechanics: Theory and Experiment* **2019**, 123208 (2019). URL http://arxiv.org/abs/1906.07043http://dx.doi.org/10.1088/1742-5468/ab4801. 1906.07043.

[70] Parrinello, M. & Rahman, A. Strain fluctuations and elastic constants. *The Journal of Chemical Physics* **76**, 2662–2666 (1982). URL http://aip.scitation.org/doi/10.1063/1.443248.

[71] Paszke, A. *et al.* Automatic differentiation in PyTorch. In Wallach, H. *et al.* (eds.) *Advances in Neural Information Processing Systems 32*, 8024–8035 (2019). URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[72] Abadi, M. *et al.* TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems (2016). URL http://arxiv.org/abs/1603.04467. 1603.04467.

8 APPENDIX

8.1 NOSE-HOVER CHAIN INTEGRATOR

Here we describe the Nose-Hover Chain (7; 8), the constant temperature integrator algorithm mentioned in the paper. We applied this integrator to the Lennard Jones system and polymer examples to simulate systems with constant temperature control. Here we define the variables used in the integrator:

- N: number of particles
- K: number of virtual variables used in the chain
- i: index for individual degrees of freedom, $i : 1, ..., 3N$
- j: index for virtual variables in the chain $j : 1, ..., K$
- p_i: momentum for each degree of freedom i
- q_i: position for each degree of freedom i
- m_i: mass for each particle in the simulation
- Q_j: coupling strengths to the heat baths variable in the chain
- η_j: virtual momenta

The coupled equations of motion are:

$$\begin{aligned}
\frac{dp_i}{dt} &= -\frac{\partial H}{\partial q_i}, \\
\frac{dq_i}{dt} &= \frac{\partial H}{\partial p_i} - p_i \frac{\eta_1}{Q_1} \\
\frac{d\eta_1}{dt} &= \big(\sum_i^{3N} \frac{p_i^2}{2m_i} - N k_B T\big) - \eta_1 \frac{\eta_2}{Q_2} \\
&... \\
\frac{d\eta_j}{dt} &= \big(\frac{\eta_{j-1}}{Q_{j-1}} - k_B T\big) - \eta_j \frac{\eta_{j+1}}{Q_{j+1}} \\
&... \\
\frac{d\eta_K}{dt} &= \big(\frac{\eta_{j-1}}{Q_{j-1}} - k_B T\big)
\end{aligned} \tag{5}$$

The Nose Hover Chain integrator performs effective temperature control, and the integrated dynamics sample the Boltzmann distribution. The integrator is deterministic and time-reversible. The control of other thermodynamic variables can be realized with other integrator protocols, such as the Rahman-Parrinelo method to maintain constant pressure (70).

8.2 MORE ON TWO-STATE ISOMERIZATION

Here we provide details for the two-state isomerization example in this section. The control incident electric field is initialized with a Gaussian form.

$$E(t) = E_0 \cos(\omega_0 (t - t_p)) \exp(-(t - t_p)^2/\tau^2), \tag{6}$$

where E_0 is the amplitude of the field, ω_0 is the center frequency, t_p is the pulse arrival time, and τ is the pulse duration. The pulse duration is set to $\tau = 10$ fs, the center frequency to $\omega_0 = 2.4$ eV, and the arrival time to $t_p = 3\tau = 30$ fs. The short pulse duration is chosen to approximate a delta-function, the arrival time to ensure that the pulse is completely contained within the simulation, and the center frequency to approximately match the electronic excitation energy. The field amplitude is chosen as $E_0 = 1$ in atomic units, since, as explained below, the quantum yield is normalized with respect to excited state population, and hence to the field intensity.

The quantity to be optimized is the quantum yield, i.e. the efficiency of isomerization. Denoting the *cis* projection operator as $\hat{P}_c$ and the *trans* projection operator as $\hat{P}_T$, the quantum yield is then

$$Y(t) = \frac{\langle \hat{P}_T^{11}(t) \rangle}{\langle \hat{P}_T^{11}(t) + \hat{P}_C^{00}(t) \rangle - p_g(t)}, \tag{7}$$

where $\langle ... \rangle = \langle \psi | ... | \psi \rangle$ denotes a quantum expectation value, $\hat{P}_T^{11} = \hat{P}_T \left| \psi_{e_1} \right\rangle \left\langle \psi_{e_1} \right|$ is the projection of $\hat{P}_T$ onto the diabatic excited electronic state, $\hat{P}_C^{00} = \hat{P}_C \left| \psi_{e_0} \right\rangle \left\langle \psi_{e_0} \right|$ projects onto the ground diabatic state, and p_g is the ground state population. Subtraction of p_g ensures that any population remaining in the ground state does not contribute to the quantum yield. Since the quantum yield depends on time, we optimize its average over a time period after the pulse is over:

$$\langle Y(t) \rangle_t = \frac{1}{T} \int_{t_0}^{t_0+T} dt\, Y(t), \tag{8}$$

where $\langle ... \rangle_t$ denotes a time average. Here, t_0 is the time at which the yield is first recorded, and T is the averaging time. We set $t_0 = 0.5$ ps and $T = 1.5$ ps.

The dynamics are discretized with a timestep of $\Delta t = 0.05$ fs. The electric field is discretized with a time step of $5\Delta t = 0.25$ fs. This is done to limit the timescale over which the electric field can vary.

8.3 Graph Neural Networks

The model is based on graph convolution, which has achieved state-of-the-art predictive performance for chemical properties and molecular energies/forces (11; 10; 61; 58; 60). In our work, we utilized the SchNet (10) architecture to learn the control Hamiltonian. The model consists of a message step and update step to systematically gather information from neighboring atoms. A 3D molecular graph is used, and the existence of a connection between atoms is decided by a fixed distance cutoff. Defining v as the index for each atom and its neighbors as $N(v)$, the graph convolutions process iteratively updates the atomic embedding h_v by aggregating "messages" from their connected atoms v and their edge features e_{uv}. This update process is summarized by

$$h_v^t = h_v^{t-1} + \sum_{u \in N(v)} Message^t(h_u, e_{uv}). \tag{9}$$

By performing this operation several times, a many-body correlation function can be constructed to represent the potential energy surface of a molecular system. In the case of SchNet, the update function is simply a summation over the atomic embeddings. The message function is parameterized by the following equations:

$$Message^t(e_{uv}, h_v) = MLP_3(MLP_1(e_{uv}) \circ MLP_1(h_v) \tag{10}$$

where the MLP_i are independent multi-layer perceptrons (MLP). For each convolution t, a separate message function is applied to characterize atom correlations at different scales. After taking element-wise products of atomic fingerprints h_v and pair-interaction fingerprints e_{uv}, the joint fingerprint is further parameterized by another MLP to incorporate more non-linearity into the model. The final updated fingerprints are used as inputs to two fully-connected layers that yield atom-wise energies. The sum of these energies gives the total energy of the system. The atomic forces are the negative gradients of the energy with respect to atomic positions. They are easily computed through automatic differentiation implemented in PyTorch (71) and Tensorflow (72). We used PyTorch in our demonstrations.

8.4 Differential Histograms

In this section, we present the calculation of an approximate histogram with Gaussian smearing functions. Given an observation r that we wish to approximately map onto a histogram with k bins, the Gaussian-smeared function $\rho_k(r)$ is given by

$$\rho_k(r) = e^{-\frac{(r-r_k)^2}{\delta}} / \sum_k e^{-\frac{(r-r_k)^2}{\delta}}, \tag{11}$$

where δ approximates the bin width. A Gaussian basis is used to replace the non-differentiable Dirac Delta functions to compute an approximate histogram. The total normalized histogram is the expected value over individual samples of r over the all the observations of pair distances (between atoms i and j) in the trajectory:

$$p(r) = \mathbb{E}_{i,j}\rho(r_{ij}). \tag{12}$$

The pair correlation function $g(r)$ is obtained by normalizing by differential volume increases:

$$g(r) = \frac{V}{N^2 4\pi r^2} p(r). \tag{13}$$