Kohn-Sham equations as regularizer: building prior knowledge into machine-learned physics

Li Li
Google Research
Mountain View, CA 94043, USA
leeley@google.com

Stephan Hoyer
Google Research
Mountain View, CA 94043, USA
shoyer@google.com

Ryan Pederson
University of California
Irvine, CA 92697, USA
pedersor@uci.edu

Ruoxi Sun
Google Research
Mountain View, CA 94043, USA
ruoxis@google.com

Ekin D. Cubuk
Google Research
Mountain View, CA 94043, USA
cubuk@google.com

Patrick Riley
Google Research
Mountain View, CA 94043, USA
pfr@google.com

Kieron Burke
University of California
Irvine, CA 92697, USA
kieron@uci.edu

Abstract

Including prior knowledge is important for effective machine learning models in physics, and is usually achieved by explicitly adding loss terms or constraints on model architectures. Prior knowledge embedded in the physics computation itself rarely draws attention. We show that solving the Kohn-Sham equations when training neural networks for the exchange-correlation functional provides an implicit regularization that greatly improves generalization. Two separations suffice for learning the entire one-dimensional H_2 dissociation curve within chemical accuracy, including the strongly correlated region. Our models also generalize to unseen types of molecules and overcome self-interaction error.

1 Introduction

Differentiable programming [1] is a general paradigm of deep learning, where parameters in the computation flow are trained by gradient-based optimization. Based on the enormous development in automatic differentiation libraries [2–5], hardware accelerators [6] and deep learning [7], this emerging paradigm is relevant for scientific computing. It keeps rigorous components where we have extremely strong physics prior knowledge and well-established numerical methods [8] and parameterizes the approximation by a neural network, which can approximate any continuous function [9]. Recent highlights include discretizing partial differential equations [10], structural optimization [11], sampling equilibrium configurations [12], differentiable molecular dynamics [13], differentiable programming tensor networks [14], optimizing basis sets in Hartree-Fock [15] and variational quantum Monte Carlo [16–18].

Third Workshop on Machine Learning and the Physical Sciences (NeurIPS 2020), Vancouver, Canada.

Density functional theory (DFT), an approach to electronic structure problems, took an enormous step forward with the creation of the Kohn-Sham (KS) equations [19], which greatly improves accuracy [20–22]. The results of solving the KS equations are reported in tens of thousands of papers each year [23]. Given an approximation to the exchange-correlation (XC) energy, the KS equations are solved self-consistently. Results are limited by the quality of such approximations, and a standard problem of KS-DFT is to calculate accurate bond dissociation curves [24]. The difficulties are an example of strong correlation physics as electrons localize on separate nuclei [25].

Naturally, there has been considerable interest in using machine learning (ML) methods to improve DFT approximations. Initial work [26, 27] focused on the KS kinetic energy, as a sufficiently accurate approximation would allow by-passing the solving of the KS equations [28, 29]. For XC, recent works focus on learning the XC potential (not functional) from inverse KS [30], and use it in the KS-DFT scheme [31–34]. An important step forward was made last year, when it was shown that a neural network could find functionals using only three molecules, by training on both energies and densities [35], obtaining accuracy comparable to human-designed functionals, and generalizing to yield accurate atomization energies of 148 small molecules [36]. But this pioneering work does not yield chemical accuracy, nor approximations that work in the dissociation limit. Moreover, it uses gradient-free optimization which usually suffers from poor convergence behavior on the large number of parameters used in modern neural networks [37–39].

Here, we show that all these limitations are overcome by incorporating the KS equations themselves into the neural network training by backpropagating through their iterations – a *KS regularizer* (KSR) to the ML model. In a traditional KS calculation, the XC is given, the equations are cycled to self-consistency, and all previous iterations are ignored in the final answer. In other ML work, functionals are trained on either energies alone [40–43], or even densities [32, 33, 44], but only after convergence. By incorporating the KS equations into the training, thereby learning the relation between density and energy at every iteration, we find accurate models with very little data and much greater generalizability. More details on experiments and discussions are available in the full paper.

2 Kohn-Sham self-consistent calculations as a differentiable program

Forward — Modern DFT finds the ground-state electronic density by solving the Kohn-Sham equations:

$$\left\{-\frac{\nabla^2}{2}+v_{\rm S}[n](\mathbf{r})\right\}\phi_i(\mathbf{r})=\epsilon_i\phi_i(\mathbf{r}). \tag{1}$$

The electronic density is obtained from occupied orbitals $n(\mathbf{r}) = \sum_i |\phi_i(\mathbf{r})|^2$. Here $v_{\rm S}[n](\mathbf{r}) = v(\mathbf{r}) + v_{\rm H}[n](\mathbf{r}) + v_{\rm XC}[n](\mathbf{r})$ is the KS potential consisting of the external one-body potential and the density-dependent Hartree (H) and XC potentials. The XC potential $v_{\rm XC}[n](\mathbf{r}) = \delta E_{\rm XC}/\delta n(\mathbf{r})$ is the functional derivative of the XC energy functional $E_{\rm XC}[n] = \int \epsilon_{\rm XC}[n](\mathbf{r})n(\mathbf{r})d\mathbf{r}$, where $\epsilon_{\rm XC}[n](\mathbf{r})$ is the XC energy per electron. The total electronic energy E is then given by the sum of the non-interacting kinetic energy $T_s[n]$, the external one-body potential energy $V[n]$, the Hartree energy $U[n]$, and XC energy $E_{\rm XC}[n]$.

The KS equations are in principle exact given the exact XC functional [19, 45], which in practice is the only term approximated in DFT. From a computational perspective, the eigenvalue problem of Eq. (1) is solved repeatedly until the density converges to a fixed point, starting from an initial guess. We use linear density mixing [46] to improve convergence, $n^{\rm (in)}_{k+1} = n^{\rm (in)}_k + \alpha(n^{\rm (out)}_k - n^{\rm (in)}_k)$. Figure 1(a) shows the unrolled computation flow. We approximate the XC energy per electron using a neural network $\epsilon_{{\rm XC},\theta}[n]$, where θ represents the trainable parameters. Together with the self-consistent KS iterations in Figure 1(b), the combined computational graph resembles a recurrent neural network [47] or deep equilibrium model [48] with additional fixed computational components. Density mixing has the same form as residual connections in deep neural networks [49]. In addition to improving convergence for the forward problem of KS self-consistent calculations, density mixing helps backpropagate gradients efficiently through long computational procedures.

Backward — If the neural XC functional were exact, KS self-consistent calculations would output the exact density and the intermediate energies over iterations would converge to the exact energy. This intention can be translated into a loss function and the neural XC functional can be updated end-to-end by backpropagating through the KS self-consistent calculations. This procedure differentiates through KS calculations and is general regardless of the dimensionality of the system. Throughout, experiments are performed in one dimension where accurate quantum solutions could be relatively easily generated via density matrix renormalization group (DMRG) [50]. We design the loss function as an expectation $\mathbb{E}$ over training molecules,

$$L(\theta) = \underbrace{\mathbb{E}_{\text{train}} \left[\int dx (n_{\text{KS}} - n_{\text{DMRG}})^2 / N_e \right]}_{\text{density loss } L_n} + \underbrace{\mathbb{E}_{\text{train}} \left[\sum_{k=1}^{K} w_k (E_k - E_{\text{DMRG}})^2 / N_e \right]}_{\text{energy loss } L_E},$$

where N_e is the number of electrons. L_n minimizes the difference between the final density with the exact density. L_E optimizes the trajectory of energies in total K iterations. The neural XC functional needs to not only output accurate ϵ_{XC} in each iteration, but also drive the iterations to quickly converge to the exact energy. The trajectory loss also makes backpropagation more efficient by directly flowing gradients to early iterations [51]. w_k are arbitrary non-negative weights associated with each iteration. The optimal neural network parameters are selected with minimal mean absolute energy per electron on the validation set.

Neural networks with physics intuition tailored for XC — Hundreds of useful XC functional approximations have been proposed by humans [52]. Here we build a neural XC functional with several differentiable components with physics intuition tailored for XC in Figure 1(c). A global convolution layer captures the long range interaction, $G(n(x), \xi_p) = \frac{1}{2\xi_p} \int dx' n(x') \exp(-|x - x'|/\xi_p)$. Note two special cases retrieve known physics quantities, Hartree energy density $G(n(x), \kappa^{-1}) \propto \epsilon_{\text{H}}$ and electronic density $G(n(x), 0) = n(x)$. Global convolution contains multiple channels and ξ_p of each channel is trainable to capture interaction in different scales. Although the rectified linear unit [53] is popular, we use the sigmoid linear unit (SiLU) [54] (or swish [55]) $f(x) = x/(1+\exp(-x))$ because the infinite differentiability of SiLU guarantees the smoothness of v_{XC}, the first derivative, and the second and higher order derivatives of the neural network used in the L-BFGS training [56]. We do not enforce a specific choice of ϵ_{XC} (sometimes called a gauge [57]), but we do enforce some conditions, primarily to aid convergence of the algorithm. We require ϵ_{XC} to vanish whenever the density does, and that it be negative if at all possible. We achieved the former using the linearity of SiLU near the origin and turning off the bias terms in convolution layers. We softly impose the latter by a negative transform layer at the end, where a negative SiLU makes most output values negative. Finally, we design a self-interaction gate (SIG) that mixes in a portion of $-\epsilon_{\text{H}}$ to cancel the self-interaction error, $\epsilon_{\text{XC}}^{(\text{out})} = \epsilon_{\text{XC}}^{(\text{in})}(1 - \beta) - \epsilon_{\text{H}}\beta$. The portion is a gate function $\beta(N_e) = \exp(-(N_e - 1)^2/\sigma^2)$. When $N_e = 1$, then $\epsilon_{\text{XC}}^{(\text{out})} = -\epsilon_{\text{H}}$. For more electrons, σ can be fixed or adjusted by the training algorithm to decide the sensitivity to N_e. For H_2 as $R \to \infty$, ϵ_{XC} tends to a superposition of the negative of the Hartree energy density at each nucleus and approaches half that for H_2^+.

3 Experiments

Our results are illustrated in Figure 2, which is for a one-dimensional mimic of H_2 designed for testing electronic structure methods [58]. The distribution of curves of the ML model directly

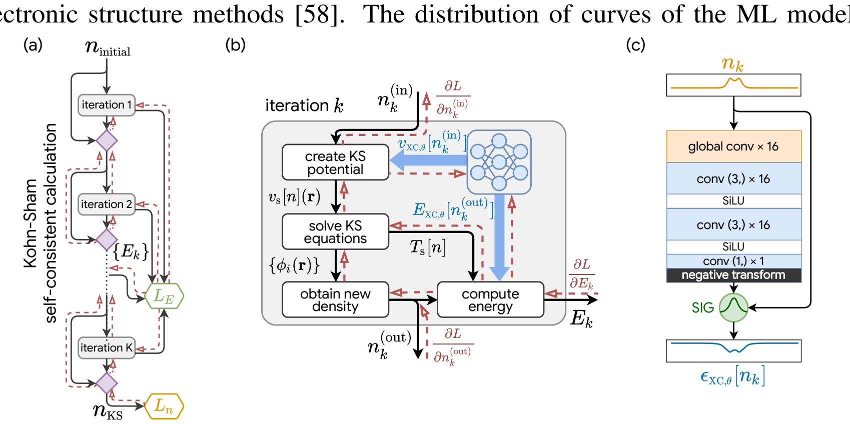

Figure 1: (a) KS-DFT as a differentiable program. Black arrows are the conventional computation flow of KS self-consistent calculations with linear density mixing (purple diamonds). The gradients flow along red dashed arrows to minimize the energy loss L_E (green hexagon) and density loss L_n (orange hexagon). (b) In each single KS iteration, neural XC functional produces $v_{\text{XC},\theta}[n]$ and $E_{\text{XC},\theta}[n]$. (c) Architecture of global XC functional $\epsilon_{\text{XC},\theta}[n]$.

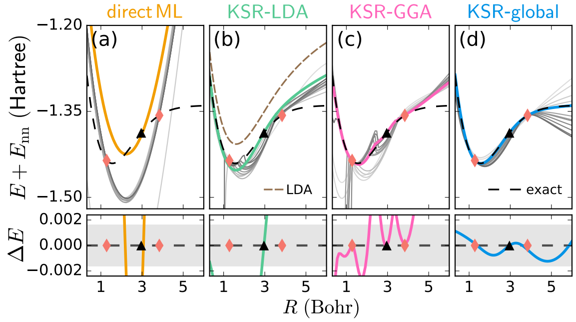

Figure 2: One-dimensional H_2 dissociation curves trained from two molecules (red diamonds). (a) A ML model that directly predicts E from geometries, clearly fails to capture the physics from very limited data. (b) Comparison of LDA found with KSR and that from uniform gas (brown), and (c) same as (b) but for GGA, (d) the global XC approximation found with KSR. E_{nn} is the nucleus-nucleus repulsion energy. 15 sampled checkpoints are visualized in grey. Optimal checkpoint validated by $R = 3$ (black triangles) are highlighted in colors. KSR-global yields chemical accuracy (grey shadow), shown in lower panels.

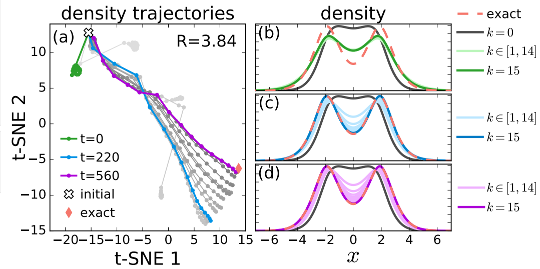

Figure 3: (a) t-SNE visualization [59] of density trajectories (grey dots) sampled by KSR during training for $R = 3.84$ from initial guess (cross) to exact density (red diamond). Darker trajectories denote later optimization steps t. Densities from each KS step in trajectories are plotted in the corresponding highlighted colors for (b) untrained $t = 0$, (c) optimal $t = 220$ in Figure 2, and (d) overfitting $t = 560$.

predicting E from geometries (direct ML) in (a) clearly fails to capture the physics. For local density approximation (LDA) and generalized gradient approximation (GGA) calculations similar to Nagai et al. [35] in (b-c), the effect of the KSR yields reasonably accurate results in the vicinity of the data, but not outside. But when a global XC functional is included in (d), chemical accuracy is achieved for all separations including the dissociation limit.

Now we dive deeper into the outstanding generalization we observed in this simple but not easy task. It is not surprising that direct ML model completely fails. Neural networks are usually underdetermined systems as there are more parameters than training examples. Regularization is crucial to improve generalization [60, 61], especially when data is limited. Most existing works regularize models with particular physics prior knowledge by imposing *constraints* via feature engineering and preprocessing [62, 63], constraints on the network [64–67] or physics-informed loss terms [68, 69]. Another regularization strategy is to generate extra data for training using prior knowledge: in image classification problems, data are augmented by operations like flipping and cropping given the prior knowledge that labels are invariant to those operations [70]. However, it is not clear how to generate extra data for physics problems solved by specific methods, e.g. electronic structure problems with KS equations. We found that training from differentiating through KS self-consistent calculations regularizes the model. Although the exact densities and energies of only two separations are given, KSR naturally samples different trajectories from an initial density to the exact density at each training step. More importantly, KSR focuses on learning an XC functional that can lead the KS self-consistent calculations to converge to the exact density from the initial density. Figure 3 visualizes the density trajectories sampled by KSR for one training separation $R = 3.84$. The functional with untrained parameters ($t = 0$) samples densities near the initial guess but soon learns to explore broadly and finds the trajectories toward the vicinity of the exact density.

In contrast, most existing ML functionals learn to predict a single step from the exact density, which is a poor surrogate for the full self-consistent calculations [71]. These standard ML models have two major shortcomings. First, the exact density is unknown for new systems, so the model is not expected to behave correctly on unseen initial densities for KS calculations. Second, even if a model is trained on many densities for single step prediction, it is not guaranteed to converge the self-consistent calculations to a good solution. Research in imitation learning shows that error accumulation from single steps quickly pushes the model out of its interpolation region [72]. On the other hand, since KSR allows the model access to all the KS iterations, it learns to optimize the entire self-consistent procedure to avoid the error accumulation from greedy optimization of single steps.

Similar results can be achieved for H_4, the one-electron self-interaction error can easily be made to vanish, and the interaction of a pair of H_2 molecules can be found without any training on this type of molecule (details in the full paper).

4 Conclusion

Differentiable programming blurs the boundary between physics computation and ML. Here we showed that treating KS self-consistent calculations as a differentiable program is a regularizer, incorporating a physics prior and resulting in a remarkable generalization of the neural XC functional trained with it. The results serve as a proof of principle to rethink physics computation in the context of the new era of computing owing to achievements in automatic differentiation software, hardware and theories. An exciting next step is to apply this idea to real molecules, as an end-to-end differentiable electronic structure method. Besides finding density functionals, all heuristics in the calculations, e.g. initial guess, density update, preconditioning, basis sets, even the entire self-consistent calculations as a meta-optimization problem [51], can be learned and optimized while keeping the rigorous physics and mathematics in the rest of the algorithm – getting the best of both worlds.

Broader Impact

This research opens a promising new direction for research in density functional theory, and provides a broadly relevant demonstration of how computational physics techniques can provide prior knowledge that greatly improves machine learning models. The demonstration of using physical computation itself as a regularizer, rather than physics-informed losses or constraints, will encourage further studies on the benefits of applying the paradigm of differentiable programming to scientific research.

As an early stage theoretical research, the ethical aspects of its outcomes are not applicable. But we would like to note one potential issue on the data – although great generalization has been shown with limited data, models trained from the Kohn-Sham regularizer are still biased to the quality of the training data. Future research should include topics such as more rigorous physics constraints and robustness against adversarial attacks.

References

[1] Atılım Günes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *The Journal of Machine Learning Research*, 18(1):5595–5637, 2017.

[2] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, and Skye Wanderman-Milne. JAX: composable transformations of Python+NumPy programs. `http://github.com/google/jax`, 2018.

[3] Michael Innes, Elliot Saba, Keno Fischer, Dhairya Gandhi, Marco Concetto Rudilosso, Neethu Mariya Joy, Tejan Karmali, Avik Pal, and Viral Shah. Fashionable modelling with flux. *CoRR*, 2018. URL `https://arxiv.org/abs/1811.01457`.

[4] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.

[5] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

[6] Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*, pages 1–12, 2017.

[7] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[8] Mike Innes, Alan Edelman, Keno Fischer, Chris Rackauckas, Elliot Saba, Viral B Shah, and Will Tebbutt. A differentiable programming system to bridge machine learning and scientific computing. July 2019.

[9] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[10] Yohai Bar-Sinai, Stephan Hoyer, Jason Hickey, and Michael P Brenner. Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, page 201814058, 2019.

[11] Stephan Hoyer, Jascha Sohl-Dickstein, and Sam Greydanus. Neural reparameterization improves structural optimization. *arXiv preprint arXiv:1909.04240*, 2019.

[12] Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019.

[13] Samuel S. Schoenholz and Ekin D. Cubuk. JAX M.D.: End-to-end differentiable, hardware accelerated, molecular dynamics in pure python. `https://github.com/google/jax-md`, `https://arxiv.org/abs/1912.04232`, 2019.

[14] Hai-Jun Liao, Jin-Guo Liu, Lei Wang, and Tao Xiang. Differentiable programming tensor networks. *Physical Review X*, 9(3):031041, 2019.

[15] Teresa Tamayo-Mendoza, Christoph Kreisbeck, Roland Lindh, and Alán Aspuru-Guzik. Automatic differentiation in quantum chemistry with applications to fully variational Hartree-Fock. *ACS Cent Sci*, 4(5):559–566, May 2018.

[16] Jan Hermann, Zeno Schätzle, and Frank Noé. Deep-neural-network solution of the electronic schrödinger equation. *Nature Chemistry*, pages 1–7, 2020.

[17] David Pfau, James S Spencer, Alexander G de G. Matthews, and W M C Foulkes. Ab-Initio solution of the Many-Electron schrödinger equation with deep neural networks. September 2019.

[18] Li Yang, Zhaoqi Leng, Guangyuan Yu, Ankit Patel, Wen-Jun Hu, and Han Pu. Deep learning-enhanced variational monte carlo method for quantum many-body physics. *Phys. Rev. Research*, 2:012039, Feb 2020. doi: 10.1103/PhysRevResearch.2.012039. URL `https://link.aps.org/doi/10.1103/PhysRevResearch.2.012039`.

[19] Walter Kohn and Lu Jeu Sham. Self-consistent equations including exchange and correlation effects. *Physical review*, 140(4A):A1133, 1965.

[20] Pierre Hohenberg and Walter Kohn. Inhomogeneous electron gas. *Physical review*, 136(3B): B864, 1964.

[21] Llewellyn H Thomas. The calculation of atomic fields. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 23, pages 542–548. Cambridge University Press, 1927.

[22] Enrico Fermi. Statistical method to determine some properties of atoms. *Rend. Accad. Naz. Lincei*, 6(602-607):5, 1927.

[23] Robert O Jones. Density functional theory: Its origins, rise to prominence, and future. *Reviews of modern physics*, 87(3):897, 2015.

[24] E. M. Stoudenmire, Lucas O. Wagner, Steven R. White, and Kieron Burke. One-dimensional continuum electronic structure with the density-matrix renormalization group and its implications for density-functional theory. *Phys. Rev. Lett.*, 109:056402, Aug 2012. doi: 10.1103/PhysRevLett.109.056402. URL `http://link.aps.org/doi/10.1103/PhysRevLett.109.056402`.

[25] Aron J. Cohen, Paula Mori-Sánchez, and Weitao Yang. Insights into current limitations of density functional theory. *Science*, 321(5890):792–794, 2008.

[26] Li Li, John C Snyder, Isabelle M Pelaschier, Jessica Huang, Uma-Naresh Niranjan, Paul Duncan, Matthias Rupp, Klaus-Robert Müller, and Kieron Burke. Understanding machine-learned density functionals. *International Journal of Quantum Chemistry*, 116(11):819–833, 2016.

[27] John C Snyder, Matthias Rupp, Katja Hansen, Klaus-Robert Müller, and Kieron Burke. Finding density functionals with machine learning. *Physical review letters*, 108(25):253002, 2012.

[28] Felix Brockherde, Leslie Vogt, Li Li, Mark E Tuckerman, Kieron Burke, and Klaus-Robert Müller. Bypassing the kohn-sham equations with machine learning. *Nature communications*, 8 (1):1–10, 2017.

[29] Li Li, Thomas E Baker, Steven R White, Kieron Burke, et al. Pure density functional for strong correlation and the thermodynamic limit from machine learning. *Physical Review B*, 94(24): 245129, 2016.

[30] Daniel S Jensen and Adam Wasserman. Numerical methods for the inverse problem of density functional theory. *International Journal of Quantum Chemistry*, 118(1):e25425, 2018.

[31] David J. Tozer, Victoria E. Ingamells, and Nicholas C. Handy. Exchange-correlation potentials. *The Journal of Chemical Physics*, 105(20):9200–9213, 1996.

[32] Jonathan Schmidt, Carlos L Benavides-Riveros, and Miguel AL Marques. Machine learning the physical nonlocal exchange–correlation functional of density-functional theory. *The journal of physical chemistry letters*, 10(20):6425–6431, 2019.

[33] Yi Zhou, Jiang Wu, Shuguang Chen, and GuanHua Chen. Toward the exact exchange–correlation potential: A three-dimensional convolutional neural network construct. *The journal of physical chemistry letters*, 10(22):7264–7269, 2019.

[34] Ryo Nagai, Ryosuke Akashi, Shu Sasaki, and Shinji Tsuneyuki. Neural-network kohn-sham exchange-correlation potential and its out-of-training transferability. *The Journal of chemical physics*, 148(24):241737, 2018.

[35] Ryo Nagai, Ryosuke Akashi, and Osamu Sugino. Completing density functional theory by machine learning hidden messages from molecules. *npj Computational Materials*, 6(1):1–8, 2020.

[36] Larry A Curtiss, Krishnan Raghavachari, Gary W Trucks, and John A Pople. Gaussian-2 theory for molecular energies of first-and second-row compounds. *The Journal of chemical physics*, 94 (11):7221–7230, 1991.

[37] John C Duchi, Michael I Jordan, Martin J Wainwright, and Andre Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015.

[38] Niru Maheswaranathan, Luke Metz, George Tucker, Dami Choi, and Jascha Sohl-Dickstein. Guided evolutionary strategies: Augmenting random search with surrogate gradients. In *International Conference on Machine Learning*, pages 4264–4273. PMLR, 2019.

[39] Luis Miguel Rios and Nikolaos V Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56(3):1247–1293, 2013.

[40] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.

[41] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in neural information processing systems*, pages 991–1001, 2017.

[42] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters*, 98(14):146401, 2007.

[43] Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O Anatole von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical review letters*, 108(5):058301, 2012.

[44] Javier Robledo Moreno, Giuseppe Carleo, and Antoine Georges. Deep learning the hohenberg-kohn maps of density functional theory. *Physical Review Letters*, 125(7):076402, 2020.

[45] Lucas O Wagner, E Miles Stoudenmire, Kieron Burke, and Steven R White. Guaranteed convergence of the kohn-sham equations. *Physical review letters*, 111(9):093003, 2013.

[46] Georg Kresse and Jürgen Furthmüller. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Physical review B*, 54(16):11169, 1996.

[47] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[48] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[49] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[50] Steven R White. Density matrix formulation for quantum renormalization groups. *Physical review letters*, 69(19):2863, 1992.

[51] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in neural information processing systems*, pages 3981–3989, 2016.

[52] Narbe Mardirossian and Martin Head-Gordon. Thirty years of density functional theory in computational chemistry: an overview and extensive assessment of 200 density functionals. *Molecular Physics*, 115(19):2315–2372, 2017.

[53] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[54] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018.

[55] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

[56] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[57] John P Perdew, Adrienn Ruzsinszky, Jianwei Sun, and Kieron Burke. Gedanken densities and exact constraints in density functional theory. *The Journal of chemical physics*, 140(18): 18A533, 2014.

[58] Thomas E Baker, E Miles Stoudenmire, Lucas O Wagner, Kieron Burke, and Steven R White. One-dimensional mimicking of electronic structure: The case for exponentials. *Physical Review B*, 91(23):235141, 2015.

[59] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[60] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[61] Jan Kukačka, Vladimir Golkov, and Daniel Cremers. Regularization for deep learning: A taxonomy. *arXiv preprint arXiv:1710.10686*, 2017.

[62] Ekin D Cubuk, Austin D Sendek, and Evan J Reed. Screening billions of candidates for solid lithium-ion conductors: A transfer learning approach for small data. *The Journal of chemical physics*, 150(21):214701, 2019.

[63] Jacob Hollingsworth, Li Li, Thomas E Baker, and Kieron Burke. Can exact conditions improve machine-learned density functionals? *The Journal of Chemical Physics*, 148(24):241743, 2018.

[64] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

[65] KT Schütt, Michael Gastegger, Alexandre Tkatchenko, K-R Müller, and Reinhard J Maurer. Unifying machine learning and quantum chemistry with a deep neural network for molecular wavefunctions. *Nature communications*, 10(1):1–10, 2019.

[66] Risi Kondor, Hy Truong Son, Horace Pan, Brandon Anderson, and Shubhendu Trivedi. Covariant compositional networks for learning graphs. *arXiv preprint arXiv:1801.02144*, 2018.

[67] Sungyong Seo and Yan Liu. Differentiable physics-informed graph networks. *arXiv preprint arXiv:1902.02950*, 2019.

[68] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[69] Rishi Sharma, Amir Barati Farimani, Joe Gomes, Peter Eastman, and Vijay Pande. Weakly-supervised deep learning of heat transport via physics informed loss. *arXiv preprint arXiv:1807.11374*, 2018.

[70] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[71] George Tucker, Andriy Mnih, Chris J Maddison, John Lawson, and Jascha Sohl-Dickstein. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. In *Advances in Neural Information Processing Systems*, pages 2627–2636, 2017.

[72] Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668, 2010.