PySCF$_{\text{IPU}}$: Repurposing Density Functional Theory to Suit Deep Learning

Alexander Mathiasen [1] Hatem Helal [1] Kerstin Klaser [1] Paul Balanca [1] Josef Dean [1] Carlo Luschi [1]
Dominique Beaini [2] Andrew Fitzgibbon [1] Dominic Masters [1]

Abstract

Density Functional Theory (DFT) accurately predicts the properties of molecules given their atom types and positions, and often serves as ground truth for molecular property prediction tasks. Neural Networks (NN) are popular tools for such tasks and are trained on DFT datasets, with the aim to approximate DFT at a fraction of the computational cost. Research in other areas of machine learning has shown that generalisation performance of NNs tends to improve with increased dataset size, however, the computational cost of DFT limits the size of DFT datasets. We present PySCF$_{\text{IPU}}$, a DFT library that allows us to iterate on both dataset generation and NN training. We create QM10X, a dataset with 10^8 conformers, in 13 hours, on which we subsequently train SchNet in 12 hours. We show that the predictions of SchNet improve solely by increasing training data without incorporating further inductive biases.

1. Introduction

Density Functional Theory (DFT) is a widely used scientific model that predicts molecular properties based on each atom's type and position (Kohn & Sham, 1965; Kohn, 1999; Pople, 1999). It is the means of choice for molecular property prediction as its accuracy is close to gold standard experimental measurement. However, DFT is computationally expensive and recently, neural networks (NN) have become a popular tool to approximate DFT at a much lower cost with comparable accuracy (Gilmer et al. (2017)).

A common strategy to improve NN model accuracy is to incorporate inductive biases, e.g. positional encodings or auxiliary losses. However, research in other areas of machine learning like vision (Zhai et al. (2022)) and large language models (Kaplan et al. (2020)) has shown that more data is crucial in order to unlock the full potential of NN training. Datasets in the molecular domain like QM9 (Ramakrishnan et al., 2014), ANI1 (Smith et al., 2017) and PCQ (Nakata & Shimazaki, 2017) contain 100k to 20M molecules, which is considered small in the context of machine learning datasets. This lack of data presents a bottleneck in molecular machine learning and prevents foundation models from better understanding chemical space. Therefore, we developed a library that allows us to decrease the time it takes to perform DFT computations to generate significantly larger datasets.

The datasets QM9, ANI1, and PCQ computed chemical properties using the DFT libraries Gaussian9 and GAMESS (Frisch et al., 2009; Barca et al., 2020). Both excel at a broad spectrum of computational chemistry tasks for systems with hundreds of atoms. However, QM9, ANI1 and PCQ contain molecules with at most twenty atoms. We introduce PySCF$_{\text{IPU}}$, a DFT library optimised for small sized chemical systems (8-12 heavy atoms) that we used to create QM10X, a dataset containing 100M examples with 10 heavy atoms.

We generated QM10X using Intelligence Processing Units (IPUs) and find that they are inherently well suited for creating DFT datasets in two distinct ways:

1. IPUs have 940MB memory with 12-65TB/s bandwidth, enough to perform small DFT computations without relying on RAM with $<$ 3TB/s bandwidth.
2. IPUs support Multiple Instruction Multiple Data (MIMD) parallelism which simplifies the computationally demanding Electron Repulsion Integrals (ERIs).

Main Contribution. PySCF$_{\text{IPU}}$ is tailored to the unique needs of generating molecular datasets. Creating a DFT dataset requires a number of distinct choices. Notable examples include DFT trade-offs (accuracy vs compute cost) and how to sample from chemical space (Dobson, 2004). In Table 1 we compare the choices made by the authors of QM9, ANI1 and PCQ. While trade-offs in DFT have been researched for decades, it remains unclear how these distinct decisions impact subsequent deep learning models. Therefore, we trained SchNet (Schütt et al. (2018)) on multiple datasets that were created based on different sets of choices.

[1] Graphcore [2] Mila - Québec AI Institute, Valence and Université de Montréal. Correspondence to: Alexander Mathiasen <alexander.mathiasen@gmail.com>.

Accepted after peer-review at the 1st workshop on Synergy of Scientific and Machine Learning Modeling, SynS & ML ICML, Honolulu, Hawaii, USA. July, 2023. Copyright Graphcore.

Table 1. The authors of previous DFT datasets made different choices. It is not clear how these design choices impact subsequent deep learning models. PySCF$_{\text{IPU}}$ allows us to investigate how choices made for the dataset creation impact deep learning, e.g., size of training set, number of conformers, basis set, and on/off-equilibrium geometry (see Section 3). † used old version with CPU bottleneck.

Dataset	**Graphs**	**Conformers**	$\frac{\text{Conformers}}{\text{Graphs}}$	**Basis Set**	**XC**	**Heavy Atoms**	**Time**	**Hardware**
QM9	133.9k	133.9k	1	6-31G(2df,p)	B3LYP	≤ 9	-	CPU
PCQ	3.38M	3.38M	1	6-31G*	B3LYP	≤ 20	>1 Year	CPU
ANI1	57462	20M	~ 348	6-31G(d)	ωB97x	≤ 8	-	CPU
QM8X (ours)	11.5k	10.53M	~ 914	6-31G	B3LYP	8	2451 Hours †	IPU
QM11X (ours)	18.38M	18.38M	1	STO3G	B3LYP	11	3667 Hours †	IPU
QM10X (ours)	105.8k	104.6M	~ 987	STO3G	B3LYP	10	2968 Hours	IPU

By publishing our code[1] we hope to enable future research into these choices by allowing researchers to iterate jointly on dataset creation and model training.

Limitations. PySCF$_{\text{IPU}}$ is an ongoing research project and still faces several challenges:

1. 940MB memory limits PySCF$_{\text{IPU}}$ to ≤ 12 heavy atoms with a less accurate electron representation (STO3G).
2. Larger numerical errors due to float32 instead of float64 (yet 7× lower than chemical accuracy 0.043eV, see Figure 1).
3. The first simulation of a molecule with N atomic orbitals takes five additional minutes due to the ahead-of-time compilation model used for IPUs (for our 104.6M dataset we spent 3.19% time compiling).

Our current implementation supports restricted Kohn-Sham, B3LYP functional (Becke, 1993; Stephens et al., 1994), no interatomic force and the basis sets {STO3G, 6-31G}.

1.1. DFT Primer.

This subsection explains **what** a typical DFT library does. The DFT inputs are atomic positions and atomic numbers which are used to compute a matrix $\mathbf{V}$. The DFT library then solves the following matrix equation for $(\mathbf{C}, \boldsymbol{\epsilon})$.

$$\big[\mathbf{V} + \mathbf{T} + J(\rho(\mathbf{C})) + K(\rho(\mathbf{C})) + V_{XC}(\rho(\mathbf{C}))\big]\mathbf{C} = \mathbf{S}\mathbf{C}\boldsymbol{\epsilon}$$

$$\rho(\mathbf{C}) = \mathbf{C}\mathbf{D}\mathbf{C}^T = \boldsymbol{\rho} \quad D_{ij} = \begin{cases} 2 & \text{if } i = j \leq N_{\text{electrons}/2} \\ 0 & \text{else} \end{cases}$$

$$J(\boldsymbol{\rho})_{kl} = \sum_{ij} I^{2e}_{ijkl}\rho_{ij} \quad K(\boldsymbol{\rho})_{il} = -\frac{1}{2}\sum_{jk} I^{2e}_{ijkl}\rho_{jk}$$

Here $\mathbf{I}^{2e} \in \mathbb{R}^{N\times N\times N\times N}$ while $\{\boldsymbol{\rho}, \mathbf{V}, \mathbf{T}, \mathbf{C}, \mathbf{S}, \boldsymbol{\epsilon}\}$ are in $\mathbb{R}^{N\times N}$, $(\mathbf{D}, \boldsymbol{\epsilon})$ are diagonal matrices and $\{\rho, J, K, V_{XC}\}$ are functions with (N, N) matrices as input and output.[2] Given a physical system, the algorithm proceeds in iterations. At iteration 0, we precompute $(\mathbf{V}, \mathbf{T}, \mathbf{S}, \mathbf{I}^{2e})$ and initialize $\mathbf{C}_1$. At iteration i, we compute $(\boldsymbol{\rho}_i = \rho(\mathbf{C}_i), V_{XC}(\boldsymbol{\rho}_i), J(\boldsymbol{\rho}_i), K(\boldsymbol{\rho}_i))$, and then solve the above equation for $\mathbf{C}_{i+1}$ (which corresponds to a generalized eigenproblem). If the iterations converge to self-consistency, one attains a solution $(\mathbf{C}, \boldsymbol{\epsilon})$ from which several properties can be computed.[3] DFT has two well-known trade-offs between accuracy and compute time: (1) V_{XC} attempts to correct the approximation error between DFT and the Schrödinger equation; hundreds of corrections exist such as LDA (Dirac, 1930), B3LYP (Becke, 1993; Stephens et al., 1994), (2) $\mathbf{C}$ represents molecular orbitals as a linear combination of Gaussian functions; many such Gaussian basis sets have been extensively studied (Pritchard et al., 2019).

2. PySCF$_{\text{IPU}}$: Hardware Accelerating DFT

While some DFT libraries support hardware acceleration, we found that QM9, PCQ and ANI1 were created on CPUs. We suspect this is due to the difficulty of hardware accelerating DFT. We developed PySCF$_{\text{IPU}}$, a hardware accelerated version of PySCF optimized to generate DFT datasets. The main differences between PySCF and PySCF$_{\text{IPU}}$ is that PySCF is a general purpose electronic structure package for CPUs, while PySCF$_{\text{IPU}}$ is specialized towards generating DFT datasets with hardware acceleration.

IPU. The IPU is a hardware accelerator with 1472 cores and 6 threads per core. All 8832 threads support MIMD parallelism. Each core has access to 639kB of 65TB/s high-bandwidth memory and 12TB/s to all other cores.

PySCF$_{\text{IPU}}$. We based PySCF$_{\text{IPU}}$ on the open-source DFT library PySCF (Sun et al., 2018). To utilize IPUs we ported PySCF from NumPy to JAX (Bradbury et al., 2018), which can target IPUs by translating JAX to XLA (Sabne, 2020). This left two remaining parts in C: [4]

1. libxc (Lehtola et al., 2018) computes V_{XC}
2. libcint (Sun, 2014) computes $(\mathbf{T}, \mathbf{V}, \mathbf{S}, \mathbf{I}^{2e})$

[1] https://github.com/graphcore-research/pyscf-ipu

[2] See Lehtola et al. (2020) for physical interpretation of tensors.

[3] The HOMO-LUMO gap measures chemical reactivity defined as $\epsilon_{\text{HOMO}} - \epsilon_{\text{LUMO}}$ where HOMO and LUMO are integers denoting the highest occupied and lowest unoccupied molecular orbitals.

[4] (Zhang & Chan, 2022) cleverly extends PySCF with AD by using JAX to pair all C calls with gradient C calls (only for CPUs).

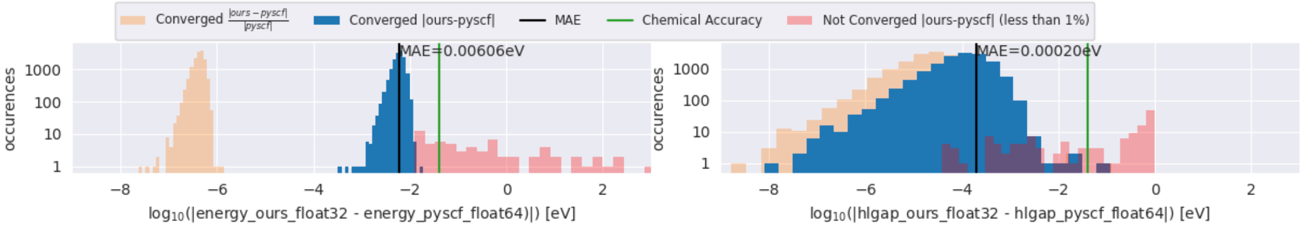

Figure 1. Histogram of numerical errors (the energies were around -10^4 and HOMO-LUMO gaps were around 10).

Libxc. We only support the B3LYP functional. The energy computation of B3LYP was implemented in JAX, which allowed us to use JAX autograd to compute the B3LYP potential. We use float32 instead of float64. JAX-XC (Zheng & Lin) recently machine translated the libxc Maple files to JAX. This may allow us support more functionals by extending JAX-XC to float32.

Libcint. We implemented the computation of $\mathbf{I}^{2e}$ by porting the libcint implementation of the Rys Quadrature algorithm from C to the IPU. The code implements a function INT2E which is called many times with different inputs. Each call can be done in parallel. However, the different calls perform different computations making it tricky to parallelise using SIMD. In contrast, using the IPUs MIMD parallelism all 8832 IPU threads can independently compute one call to INT2E in parallel. $\mathbf{I}^{2e}$ satisfies an 8x symmetry[5] which, when exploited, (1) reduces bytes needed to store $\mathbf{I}^{2e}$ by 8x and (2) gives an 8x reduction in FLOPs needed to compute $K(\boldsymbol{\rho}, \mathbf{I}^{2e})$ and $J(\boldsymbol{\rho}, \mathbf{I}^{2e})$ (usually computed using `np.einsum`). To utilize the 8x symmetry we implemented a custom einsum algorithm (the memory usage of our current implementation can be improved as outlined in Section 4).

Numerical Error of $\text{PySCF}_{\text{IPU}}$. DFT libraries usually use float64. This paragraph investigates our numerical errors due to float32 and compares them to the needs of deep learning datasets. We took a 10k uniform random sample from our QM10X dataset and recomputed DFT with PySCF in float64. The mean absolute error of $\text{PySCF}_{\text{IPU}}$ for converged[6] molecules was 0.00605eV for energies and 0.00019eV for HOMO-LUMO gap, see Figure 1. Our errors are larger than $\sim 10^{-8}$ in typical float64 DFT libraries. However, 0.006eV is $40\times$ smaller than 0.25eV from JaxSCF (Li et al., 2023).[7] Neural networks like SchNet attain 0.063eV HOMO-LUMO gap and 0.015eV for energy predictions (Schütt et al., 2018). Our numerical error for HOMO-LUMO gap is thus 300x smaller than the errors achieved by neural networks (but similar for energy). We expect further engineering will decrease numerical errors.

Generating QM10X. We used RDKit (Landrum et al., 2013) to add hydrogens to the 3M molecules Generated Data Bank with 10 $\{C, N, O, F\}$ atoms (GDB10) (Fink & Reymond, 2007; Fink et al., 2005). We selected $\sim$100k molecules with $\leq$10 hydrogens and computed 1000 conformers for each. DFT was then evaluated on each conformer using $\text{PySCF}_{\text{IPU}}$ with the DFT options B3LYP/STO3G with a 10-30k grid size.

Timing $\text{PySCF}_{\text{IPU}}$. We used 304 IPUs to create the 104.6M QM10X dataset split over 19 POD16s. Our logs recorded 2968 IPU hours of which 3.19% was spent compiling. All IPUs were done after 13 hours. The median DFT time was 101.9 ± 21.3ms. Each IPU POD16 has two physical EPYC CPUs with 240 vCPUs. On a POD16 for the example molecule FC=C1C2CN2N=C1C=O we can do 82 DFTs/sec with PySCF and 228 DFTs/sec with $\text{PySCF}_{\text{IPU}}$.

3. Generating Data To Train SchNet

Equilibrium vs Off-Equilibrium. Smith et al. (2017) discussed training neural networks on equilibrium vs offeqilibrium molecules, that is, molecules in which the forces acting on all nuclei are zero (equilibrium) or non-zero (offequilibrium). $\text{PySCF}_{\text{IPU}}$ allowed us to investigate whether it is harder to train neural networks to predict HOMO-LUMO gap for off-equilibrium molecules. We created $\text{QM9}_{F=0}$ by running $\text{PySCF}_{\text{IPU}}$ on the equilibrium atom positions provided by QM9, and then created $\text{QM9}_{F\neq 0}$ by running $\text{PySCF}_{\text{IPU}}$ on off-equilibrium atom positions from RDKit conformers. Our SchNet model attained 0.049eV on $\text{QM9}_{F=0}$, similar to 0.053eV on ”normal” QM9. Our SchNet model performed worse at 0.123eV on $\text{QM}_{F\neq 0}$, see Figure 2(b). This seems to be in line with prior work: Schütt et al. (2018) trained SchNet for 12 hours on 110k QM9 equilibrium molecules, and 12 days on the 19M offequilibrium molecules. The off-equilibrium model trained for less time was better (0.26 kcal/mol vs 0.47 kcal/mol). Both suggest off-equilibrium prediction is harder, however, more experimentation is needed to confirm this conclusion.

[5] $\mathbf{I}^{2e}_{ijkl} = \mathbf{I}^{2e}_{ijlk} = \mathbf{I}^{2e}_{jikl} = \mathbf{I}^{2e}_{jilk} = \mathbf{I}^{2e}_{jilk} = \mathbf{I}^{2e}_{klij} = \mathbf{I}^{2e}_{klji} = \mathbf{I}^{2e}_{lkij} = \mathbf{I}^{2e}_{lkji}$.

[6] Converged is defined as np.std(energies[-5:])>0.01 (reduced from 105.8M to 104.6M).

[7] Their error seem to stem from approximating $\mathbf{I}^{2e}$ using a grid which we circumvent by using the IPUs MIMD parallelism.

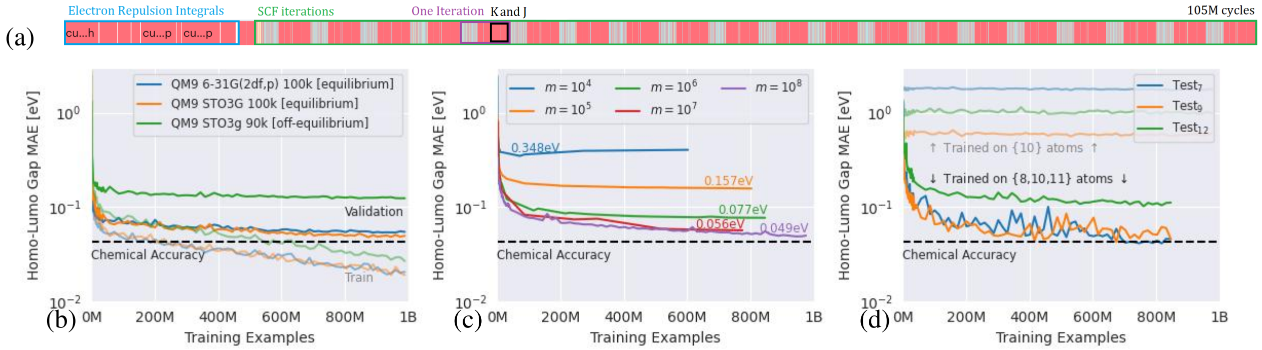

Figure 2. (a) Profile of our DFT computation for ”Fc1nocnc(=O)c1=O” (b) SchNet performs similar on QM9 in STO3G and 6-31G(d2f,p) for equilibrium molecules but worse for off-equilibrium prediction (c) SchNet improves on off-equilibrium prediction when increasing size of dataset m (d) SchNet did not generalize from $\{10\}$ atoms to $\{7, 9, 12\}$; it does generalize if we add $\{8, 11\}$ to the training data.

Training SchNet on QM10X. We trained a 9M parameter SchNet on QM10X using subsets with $m = 10^i$ conformers for $i = 4, .., 8$. The validation MAE is visualized in Figure 2(c). Increasing m from $m = 10^4$ to $m = 10^8$ improved validation MAE from 0.3484eV to 0.0486eV, comparable to the 0.049eV of SchNet on QM9$_{F=0}$. If off-equilibrium prediction is harder, as argued in the previous paragraph, it is intriguing that training a larger SchNet on more data is capable of closing the gap. Notably, the scaling strategy requires less human effort compared to augmenting neural networks with additional inductive biases.

Generalizing To Different Number of Atoms. We created five datasets with $7, 8, 9, 11, 12$ atoms to test if SchNet could generalize from 10 atoms to $\neq$ 10 atoms. The SchNet trained on 10^8 molecules with 10 atoms did not generalize to $\neq$ 10 atoms, see Figure 2(d). We augmented QM10X with 8 and 11 atom molecules: 5M DFTs for GDB8 (1000 conformers per graph) and 18.38M DFTs from QM11X (1 conformer per graph). The SchNet trained on the resulting 127M conformers with $8, 10, 11$ atoms generalized to $7, 9, 12$ atoms, see Figure 2(d).

4. Scaling to Larger Molecules

The main limitation our of implementation is the fact that it runs out of memory when $N > 70$. This limited us to use $\leq$ 12 atoms in STO3G with a small grid size (10-30k).

Current (Naive) Bottleneck. We can view $\mathbf{I}^{2e}$ as a matrix $M \in \mathbb{R}^{N^2,N^2}$ so $K(\boldsymbol{\rho}) = Mx$ where $x = \text{flatten}(\boldsymbol{\rho})$. For $N = 70$ we spend 173MB computing $v = Mx$. We represent M with the 8x symmetry as a list of matrices of size (num_integrals, integral_size). M is split over all IPU cores and never needs to move. We compute $v_i = Mx$ by copying x for every thread allowing us to compute $x_i = M_{thread_i}x$ where M_{thread_i} is the part of M stored on core i. The result is then $v = \sum_i x_i$. This uses $N^2 \cdot num\_threads$ floats (or 173MB). The memory consumption can be reduced to 173MB/N=2.47MB by splitting x into N batches each with N floats. The current strategy was a stepping stone, the main advantage is ease of implementation. Notably, prior work disregarded this 8x symmetry at the cost of a 5-8x slowdown (Zhang & Chan, 2022).

Recomputation. Instead of precomputing $\mathbf{I}^{2e}$ we can recompute the entries of $\mathbf{I}^{2e}$ whenever needed during the simultaneous computation of $K(\boldsymbol{\rho}, \mathbf{I}^{2e})$ and $J(\boldsymbol{\rho}, \mathbf{I}^{2e})$. For the case of Figure 2(a) it took 15.3M out of 105M cycles to compute $\mathbf{I}^{2e}$. Recomputing $\mathbf{I}^{2e}$ all 20 iterations would increase cycle count from 105M to an estimated 395.7M cycles (57.5ms to 216.8ms on a 1.825GHz Mk2 BOW). Finally, if we also recompute the evaluation of the atomic orbitals on the XC grid our memory consumption becomes $N^2 + \text{grid\_size} \cdot 4 + B$, where B is the memory used while (re)computing $K(\boldsymbol{\rho}, \mathbf{I}^{2e})$ and $J(\boldsymbol{\rho}, \mathbf{I}^{2e})$ (independent of N).

Using Multiple IPUs. We exemplify our plan to parallelize over 15GB SRAM in 16 IPUs following Figure 2(a). The computation of $\mathbf{I}^{2e}$ (Eletron Repulsion Integral) can be split over 16 IPUs in the same way we already utilize MIMD parallelism to split them over the 8832 threads. The computation of K and J can also be split over 16 IPUs with one REDUCE_SUM of size N^2. The remainder of the computation can be repeated independently on each IPU.

5. Discussion.

Prior work usually falls in one of the following three categories: (1) DFT libraries, (2) DFT datasets, and (3) neural networks trained on DFT datasets. This article touches on all three, demonstrating how choices made for (1) and (2) can impact (3). As the field turns towards training large foundational quantum chemical models, we hope our work will help generate sufficiently large datasets with billions (or trillions) of molecules. Furthermore, we hope to allow researchers to generate datasets specifically to fine-tune such foundational models.

6. Broader Impact.

Density Funtional Theory is used to design materials and drugs. We hope that our work on dataset creation will allow future research to create large foundational quantum chemistry models, thereby accelerating the design of new materials and drugs.

References

Barca, G. M. J., Bertoni, C., Carrington, L., Datta, D., De Silva, N., Deustua, J. E., Fedorov, D. G., Gour, J. R., Gunina, A. O., Guidez, E., Harville, T., Irle, S., Ivanic, J., Kowalski, K., Leang, S. S., Li, H., Li, W., Lutz, J. J., Magoulas, I., Mato, J., Mironov, V., Nakata, H., Pham, B. Q., Piecuch, P., Poole, D., Pruitt, S. R., Rendell, A. P., Roskop, L. B., Ruedenberg, K., Sattasathuchana, T., Schmidt, M. W., Shen, J., Slipchenko, L., Sosonkina, M., Sundriyal, V., Tiwari, A., Galvez Vallejo, J. L., Westheimer, B., Wloch, M., Xu, P., Zahariev, F., and Gordon, M. S. Recent Developments in the General Atomic and Molecular Electronic Structure System. *The Journal of Chemical Physics*, 152, 2020.

Becke, A. D. Density-functional thermochemistry. III. The role of exact exchange. *The Journal of Chemical Physics*, 98(7):5648–5652, 04 1993. ISSN 0021-9606. doi: 10.1063/1.464913. URL https://doi.org/10.1063/1.464913.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: Composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Dirac, P. A. M. Note on exchange phenomena in the thomas atom. *Mathematical Proceedings of the Cambridge Philosophical Society*, 26(3):376–385, 1930. doi: 10.1017/S0305004100016108.

Dobson, C. M. Chemical space and biology. *Nature*, 432 (70197019):824–828, Dec 2004. ISSN 1476-4687. doi: 10.1038/nature03192.

Fink, T. and Reymond, J.-L. Virtual Exploration of the Chemical Universe up to 11 Atoms of C, N, O, F: Assembly of 26.4 Million Structures (110.9 Million Stereoisomers) and Analysis for New Ring Systems, Stereochemistry, Physicochemical Properties, Compound Classes, and Drug Discovery. *Journal of chemical information and modeling*, 2007.

Fink, T., Bruggesser, H., and Reymond, J.-L. Virtual exploration of the small-molecule chemical universe below 160 daltons. *Angewandte Chemie International Edition*, 2005.

Frisch, M., Trucks, G., Schlegel, H., Scuseria, G., Robb, M., Cheeseman, J., Scalmani, G., Barone, V., Mennucci, B., Petersson, G., et al. Gaussian 09, Revision B. 01 (Wallingford, CT). *Gaussian Inc*, 2009.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural Message Passing for Quantum Chemistry. In *International conference on machine learning*. PMLR, 2017.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling Laws for Neural Language Models. *arXiv preprint arXiv:2001.08361*, 2020.

Kohn, W. Nobel Lecture: Electronic Structure of Matter—Wave Functions and Density Functionals. *Reviews of Modern Physics*, 1999.

Kohn, W. and Sham, L. J. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140: A1133–A1138, Nov 1965.

Landrum, G. et al. RDKit: A Software Suite For Cheminformatics, Computational Chemistry, and Predictive Modeling. *Greg Landrum*, 2013.

Lehtola, S., Steigemann, C., Oliveira, M. J. T., and Marques, M. A. L. Recent developments in libxc — a comprehensive library of functionals for density functional theory. *SoftwareX*, 7:1–5, Jan 2018. ISSN 2352-7110. doi: 10.1016/j.softx.2017.11.002.

Lehtola, S., Blockhuys, F., and Van Alsenoy, C. An overview of self-consistent field calculations within finite basis sets. *Molecules*, 25(5), 2020. ISSN 1420-3049. doi: 10.3390/molecules25051218. URL https://www.mdpi.com/1420-3049/25/5/1218.

Li, T., Lin, M., Hu, Z., Zheng, K., Vignale, G., Kawaguchi, K., Neto, A., Novoselov, K. S., and Yan, S. D4FT: A Deep Learning Approach to Kohn-Sham Density Functional Theory. *arXiv preprint arXiv:2303.00399*, 2023.

Nakata, M. and Shimazaki, T. PubChemQC Project: A Large-Scale First-Principles Electronic Structure Database for Data-Driven Chemistry. *Journal of Chemical Information and Modeling*, 2017.

Pople, J. A. Nobel Lecture: Quantum Chemical Models. *Reviews of Modern Physics*, 1999.

Pritchard, B. P., Altarawy, D., Didier, B., Gibson, T. D., and Windus, T. L. New basis set exchange: An open, up-to-date resource for the molecular sciences community. *Journal of Chemical Information and Modeling*, 59(11):4814–4820, 2019. doi: 10.1021/acs.jcim.9b00725. URL https://doi.org/10.1021/acs.jcim.9b00725. PMID: 31600445.

Ramakrishnan, R., Dral, P. O., Rupp, M., and Von Lilienfeld, O. A. Quantum Chemistry Structures and Properties of 134 Kilo Molecules. *Scientific data*, 2014.

Sabne, A. XLA: Compiling Machine Learning for Peak Performance. 2020.

Schütt, K., Kessel, P., Gastegger, M., Nicoli, K., Tkatchenko, A., and Muller, K.-R. SchNetPack: A Deep Learning Toolbox for Atomistic Systems. *Journal of chemical theory and computation*, 2018.

Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A., and Müller, K.-R. Schnet–A Deep Learning Architecture for Molecules and Materials. *The Journal of Chemical Physics*, 2018.

Smith, J. S., Isayev, O., and Roitberg, A. E. ANI-1, A Data Set of 20 Million Calculated Off-Equilibrium Conformations for Organic Molecules. *Scientific data*, 2017.

Stephens, P. J., Devlin, F. J., Chabalowski, C. F., and Frisch, M. J. Ab initio calculation of vibrational absorption and circular dichroism spectra using density functional force fields. *The Journal of Physical Chemistry*, 98(45): 11623–11627, 1994. doi: 10.1021/j100096a001. URL `https://doi.org/10.1021/j100096a001`.

Sun, Q. Libcint: An efficient general integral library for gaussian basis functions. (arXiv:1412.0649), Dec 2014. doi: 10.48550/arXiv.1412.0649. URL `http://arxiv.org/abs/1412.0649`. arXiv:1412.0649 [physics].

Sun, Q., Berkelbach, T. C., Blunt, N. S., Booth, G. H., Guo, S., Li, Z., Liu, J., McClain, J. D., Sayfutyarova, E. R., Sharma, S., et al. PySCF: the Python-based simulations of chemistry framework. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 2018.

Zhai, X., Kolesnikov, A., Houlsby, N., and Beyer, L. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12104–12113, 2022.

Zhang, X. and Chan, G. K.-L. Differentiable Quantum Chemistry with PySCF for Molecules and Materials at the Mean-Field Level and Beyond. *The Journal of Chemical Physics*, 2022.

Zheng, K. and Lin, M. JAX-XC: Exchange Correlation Functionals Library in Jax. In *Workshop on"Machine Learning for Materials"ICLR 2023*.