Article

Toward Orbital-Free Density Functional Theory with Small Data Sets and Deep Learning

Kevin Ryczko,* Sebastian J. Wetzel, Roger G. Melko, and Isaac Tamblyn*

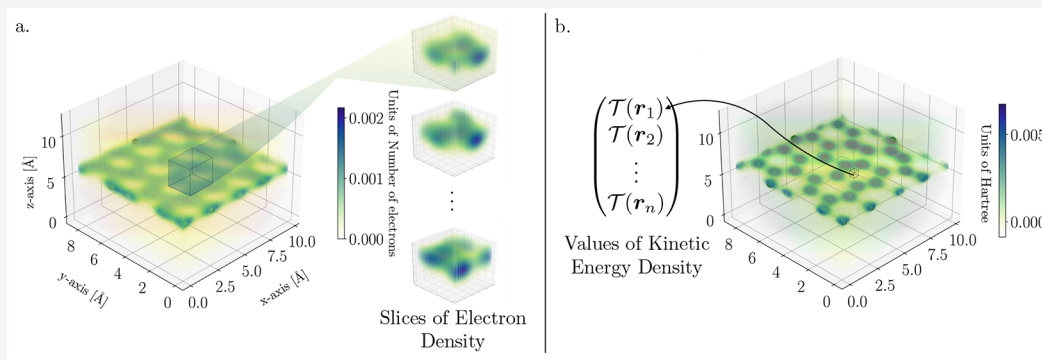Cite This: *J. Chem. Theory Comput.* 2022, 18, 1122−1128

Read Online

ACCESS | Metrics & More | Article Recommendations | SI Supporting Information

ABSTRACT: We use voxel deep neural networks to predict energy densities and functional derivatives of electron kinetic energies for the Thomas−Fermi model and Kohn−Sham density functional theory calculations. We show that the ground-state electron density can be found via direct minimization for a graphene lattice without any projection scheme using a voxel deep neural network trained with the Thomas−Fermi model. Additionally, we predict the kinetic energy of a graphene lattice within chemical accuracy after training from only two Kohn−Sham density functional theory (DFT) calculations. We identify an important sampling issue inherent in Kohn−Sham DFT calculations and propose future work to rectify this problem. Furthermore, we demonstrate an alternative, functional derivative-free, Monte Carlo based orbital-free density functional theory algorithm to calculate an accurate two-electron density in a double inverted Gaussian potential with a machine-learned kinetic energy functional.

1. INTRODUCTION

Kohn−Sham density functional theory[1] (KS-DFT) and Orbital-Free (OF) DFT[2,3] are two electronic structure methodologies to calculate properties of matter. In OF-DFT, all energy functionals depend solely on the electron density, whereas in KS-DFT, energy functionals depend on both the electron density and the set of Kohn−Sham orbitals. The explicit dependence on the electron density in OF-DFT allows for favorable, $O(N)$, computational scaling, enabling one to study large systems[4] (where N is the number of electrons). Conversely, the computational scaling of KS-DFT, $O(N^3)$, is less favorable due to the computation of a set of orbitals, rather than the electron density alone. However, the main advantage of KS-DFT implementations is that the kinetic energy is calculated via a single-particle quantum mechanical operator, leading to a more accurate approximation of the true kinetic energy functional (KEF). In OF-DFT, the kinetic energy is written as a classical, approximate functional of the electron density. The lack of knowledge of a quantum mechanical KEF reduces the accuracy and applicability of OF-DFT.

Thomas and Fermi (TF) both proposed an analytic KEF assuming a free electron gas.[5,6] They were followed by the

Thomas−Fermi−Dirac−von Weizsäcker and $X\alpha$ models[7−10] to address the failures of the TF model for atoms and molecules. Hohenberg and Kohn[11] proved the existence of a KEF that depends explicitly on the electron density of interacting electrons but never gave its exact functional form. Subsequently, Kohn and Sham[1] introduced a noninteracting, orbital-dependent KEF. This noninteracting functional is routinely used in all KS-DFT calculations.

More recently, machine learning models have been used as energy functionals.[12−17] Specifically, in refs 12 and 14, machine-learned, one-dimensional KEFs were constructed using kernel ridge regression and convolutional neural networks (CNNs). In ref 12, the authors argued that the error of a functional derivative of a machine-learned KEF (FD-KEF) was too large to be used in a direct minimization
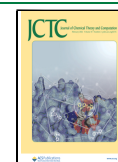
calculation. They reduced this error by projecting the functional derivative of the total energy onto a subspace found with principal component analysis. Following this report, ref 14 included the FD-KEF in a loss function to improve the predictions from the machine learning models. This improved loss function reduced the prediction error of the FD-KEF but did not eliminate it entirely. An additional projection method using a sinusoidal basis was introduced and utilized to minimize the error. The use of a sinusoidal basis eliminated the computational overhead of performing principal component analysis on the training set densities.

In addition to KEFs, machine learning models have been used as exchange-correlation functionals.[18−20] In refs 18 and 19, "slices" of the density, rather than the entire scalar field, were used as input to neural networks. It was shown that machine-learned exchange-correlation functionals could be used for a model system with a simple harmonic oscillator potential, several molecules, and a unit cell of Si, demonstrating the transferability of this methodology. Additionally, the approach drastically reduced the number of calculations needed to generate a training set.

We build on previous work which computed KEFs for one-dimensional systems and compute KEFs, FD-KEFs, electron densities, and energies in *three dimensions for a realistic system*: pristine graphene lattices. We also *eliminate the need for large data sets*. Namely, we use slices of the electron density as input to deep neural networks (DNNs), where the output is also a slice of the kinetic energy density (KED). Desired quantities are subsequently found via integration over the supercell. We call this methodology voxel DNNs (VDNNs). In Section 2, we outline the basic electronic structure, training data generation, and machine learning methodologies used. In Section 3, we outline the results of VDNNs in practice. We first investigate the Thomas−Fermi model with VDNNs as a proof of principle. The Thomas−Fermi model is simple, and both the kinetic energy and its functional derivative with respect to the electron density are analytically known for all densities. Afterward, we apply VDNNs to KS-DFT. Using VDNNs allows one to have a Kohn−Sham kinetic energy functional that explicitly depends on the electron density and enables one to insert the energy functional into OF-DFT (Figure 1). Lastly,

we show an alternative potential of our method with a demonstration of a functional derivative-free Monte Carlo (MC) based optimization for a toy, 1D model system. Direct minimization techniques have been applied in KS-DFT calculations[21,22] which avoids the self-consistent procedure, but direct minimization in OF-DFT still requires functional derivatives. Our MC based optimization eliminates the need of a functional derivative altogether. We conclude and propose future directions based on our results in Section 4.

2. METHODS

In this work, we use VDNNs to calculate KEDs ($\mathcal{T}$) and FD-KEFs ($\mathcal{F}$) of graphene lattices using OF-DFT with the Thomas−Fermi model and using KS-DFT (LDA and GGA). As discussed above, the Thomas−Fermi model serves as a preliminary experiment due to its simplicity, and KS-DFT serves as a realistic use case. We therefore first test our methodology with the Thomas−Fermi model before moving on to KS-DFT. In OF-DFT, the total energy functional is written in real space as

$$E[\rho(\boldsymbol{r})] = T[\rho(\boldsymbol{r})] + E_{\text{Hartree}}[\rho(\boldsymbol{r})] + E_{\text{ion}}[\rho(\boldsymbol{r})] + E_{\text{xc}}[\rho(\boldsymbol{r})] \quad (1)$$

where $\rho(\boldsymbol{r})$ is the electron density, and the terms in order are kinetic, Hartree, external, and exchange-correlation energies. To find the ground-state electron density, one searches for an electron density which minimizes the total energy expression under the constraint that the number of electrons, N_e, is fixed. This yields the Lagrangian

$$\mathcal{L}[\rho(\boldsymbol{r})] = E[\rho(\boldsymbol{r})] - \mu\left(\int_\Omega d\boldsymbol{r}\ \rho(\boldsymbol{r}) - N_e\right) \quad (2)$$

where μ is a Lagrange multiplier, and Ω is the volume of the supercell. Applying a functional derivative of the Lagrangian with respect to the electron density yields the Euler−Lagrange equation

$$\mathcal{F}(\boldsymbol{r}) + V_{\text{eff}}(\boldsymbol{r}) = \mu \quad (3)$$

where

$$V_{\text{eff}}(\boldsymbol{r}) = V_{\text{Hartree}}(\boldsymbol{r}) + V_{\text{ion}}(\boldsymbol{r}) + V_{\text{xc}}(\boldsymbol{r}) \quad (4)$$

and

$$\mathcal{F}(\boldsymbol{r}) = \frac{\delta T[\rho](\boldsymbol{r})}{\delta \rho(\boldsymbol{r})} \quad (5)$$

Using gradient descent, one can solve for the ground-state electron density via direct minimization

$$\phi_{n+1}(\boldsymbol{r}) = \phi_n(\boldsymbol{r}) - 2\alpha\phi_n(\boldsymbol{r})(\mathcal{F}(\boldsymbol{r}) + V_{\text{eff}}(\boldsymbol{r}) - \mu)_n \quad (6)$$

where $\phi_n(\boldsymbol{r}) = \sqrt{\rho_n(\boldsymbol{r})}$, and α is a small parameter. The use of the square root of the density ensures that the electron density remains positive during the optimization.

Using the Thomas−Fermi model and the DFTpy code,[23] we performed two direct minimization calculations for 32-atom slabs of graphene where the atoms were perturbed from their equilibrium geometry. The perturbations were generated from a normal distribution with a standard deviation of 0.1 Å. We used an energy cutoff of 45 Ha, the LDA exchange-correlation functional,[1] and norm-conserving pseudopotentials.[24] Due to the free electron gas approximation used for the kinetic energy, we maintained this approximation in our exchange-correlation functional choice. We collected ρ_{TF}, $\mathcal{T}_{\text{TF}}$, and $\mathcal{F}_{\text{TF}}$ every 10
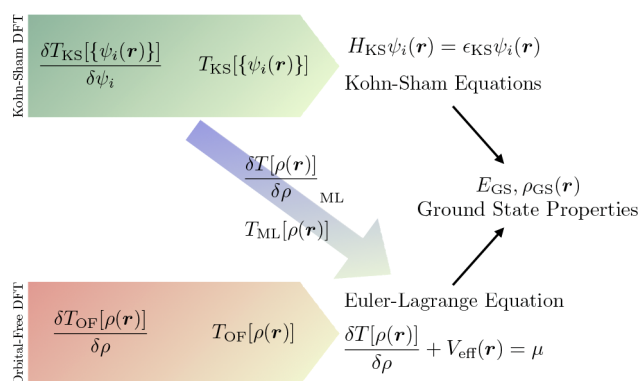
Figure 1. Our machine learning architecture, similar to refs 12 and 14, makes a connection between the Kohn−Sham density functional theory and orbital-free density functional theory. The model allows for the construction of Kohn−Sham kinetic energy functionals that explicitly depend on the electron density and, therefore, direct insertion into orbital-free density functional theory. See Section 2 for more information about the equations.
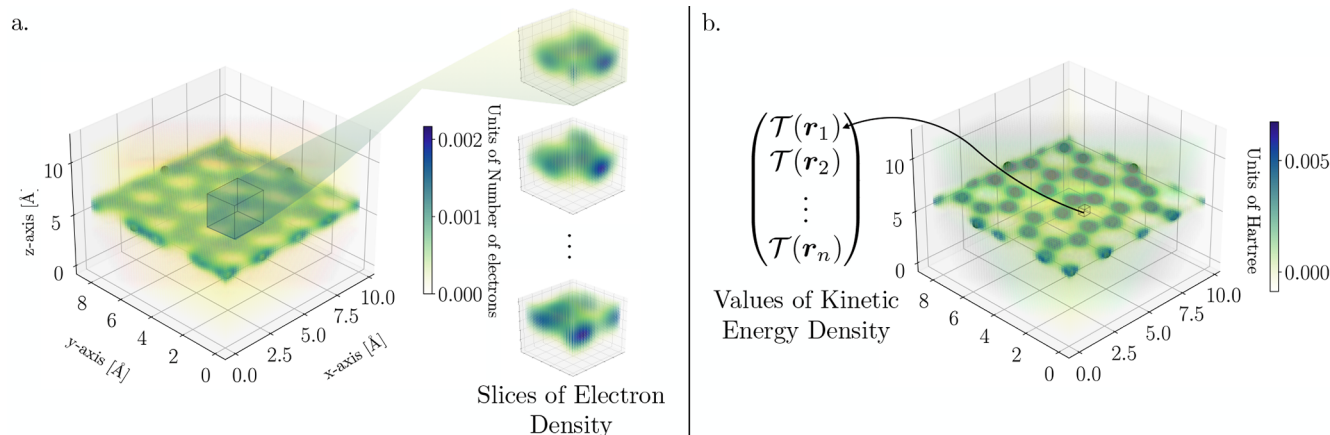
Figure 2. A visual representation of voxel deep neural networks. (a) An example electron density for a 32-atom graphene lattice supercell. The highlighted region in the electron density is a slice of the electron density centered at a particular pixel. (b) The kinetic energy density for the same 32-atom graphene lattice. The voxel deep neural network learns the mapping between the slice of electron density to the voxel of kinetic energy density.

steps (values of n in eq 6) from one of the direct minimization calculations to be used as training data. This made for a total of 173 training configurations. The second calculation was used as independent test data.

In addition to OF-DFT calculations, we used KS-DFT to investigate 32-atom graphene slabs where the atoms were perturbed in the same way as described above. In KS-DFT, the electron density is written as

$$\rho_{KS}(\boldsymbol{r}) = 2 \sum_n^{occ} \sum_{\boldsymbol{k}} w_{\boldsymbol{k}} \psi_{n,\boldsymbol{k}}^*(\boldsymbol{r}) \psi_{n,\boldsymbol{k}}(\boldsymbol{r}) \tag{7}$$

and the KED is written as

$$\mathcal{T}_{KS}(\boldsymbol{r}) = -\sum_n^{occ} \sum_{\boldsymbol{k}} w_{\boldsymbol{k}} \psi_{n,\boldsymbol{k}}^*(\boldsymbol{r}) \nabla^2 \psi_{n,\boldsymbol{k}}(\boldsymbol{r}) \tag{8}$$

In eqs 7 and 8, n is the band index, $\boldsymbol{k}$ is the k-point, $w_{\boldsymbol{k}}$ is the weighting associated with each k-point, and ψ is a Kohn−Sham orbital. In this work, we use finite differences to compute derivatives of the Kohn−Sham orbitals. To compute the Kohn−Sham orbitals, we used Abinit[25] with an energy cutoff of 45 Ha, a $4 \times 4 \times 1$ k-point grid, the PBE exchange-correlation functional,[26] and norm-conserving pseudopotentials.[24] We justify this exchange-correlation choice based on its popularity in the literature. Here, we performed a total of 200 DFT calculations where 100 of the configurations were for training and 100 were kept aside for testing. To obtain $\mathcal{F}_{KS}$ for these calculations, we used eq 3 where the potentials were evaluated using DFTpy,[23] and the chemical potentials were obtained from Abinit. It should be noted that eq 3 can only be used to define $\mathcal{F}_{KS}$ when self-consistency has been reached.[27]

To train the VDNNs, we collected slices of ρ (and $\nabla\rho$ for Kohn−Sham models) as inputs and slices of $\mathcal{T}$ and $\mathcal{F}$ as outputs. If $\tilde{\mathcal{T}}$ and $\tilde{\rho}$ are discretized forms of $\mathcal{T}$ and ρ, then a slice of ρ with dimensions (a, b, c) centered at pixels (i, j, k) is written as $\tilde{\rho}[i - a/2 : i + a/2 + 1, j - b/2 : j + b/2 + 1, k - c/2 : k + c/2 + 1]$. The addition of 1 is due to the use of odd values of a, b, and c. A slice of $\mathcal{T}$ with dimensions (a', b', c') centered at pixels (i, j, k) is similarly written as $\tilde{\mathcal{T}}[i - a'/2 : i + a'/2 + 1, j - b'/2 : j + b'/2 + 1, k - c'/2 : k + c'/2 + 1]$. We tested a variety of input sizes and used output sizes of $(1,1,1)$. This

corresponds to mapping electron density slices to values of $\mathcal{T}$, as shown in Figure 2. To avoid bias in training, we sample $\mathcal{T}$ or $\mathcal{F}$ such that a uniform distribution is produced given a target number of samples. The target number of samples was 1024^2 unless stated otherwise. Of these, 99% of them were used for training, and 1% were used for validation. Testing was done on the 100 independent DFT calculations not seen during training. Inputs were standardized and normalized such that the range of values was $\in [-1,1]$ and outputs were normalized $\in [0,1]$. We used a modified version of the deep neural network (DNN) architecture used in refs 28 and 29, which had success in predicting various energies at the DFT level with different functionals. This included 2 nonreducing convolutional layers with 64 $3 \times 3 \times 3$ kernels, 4 nonreducing convolutional layers with 16 $3 \times 3 \times 3$ kernels, 1 reducing convolutional layer with 64 $3 \times 3 \times 3$ kernels, 4 nonreducing convolutional layers with 32 $3 \times 3 \times 3$ kernels, 1 fully connected layer with 1024 neurons, and 1 fully connected layer with two outputs. We use the ELU activation function throughout due to its improved performance compared to RELU with batch normalization.[30] Since the inputs are scalar fields, a natural architectural choice is to use convolutional layers. They are designed to extract relevant features from images to make accurate predictions. The input dimensions are less than in refs 28 and 29, which is why the first two convolutional layers were changed to nonreducing layers. We note that this particular architecture choice is most likely not optimal, and one could obtain better results with another architecture choice. Models were trained for 500 epochs with learning rates of 10^{-5} and a batch size of 512. Production models were trained across 16 NVIDIA V100 GPUs with layer-wise adaptive rate scaling with clipping.[31] Training on large batch sizes leads to unfavorable results, and ref 31 has shown that layer-wise adaptive rate scaling allows one to obtain similar results to lower batch training while reducing the training time. Inference for the densities can be trivially parallelized and does not suffer from any negative large-batch effects. It was done across 64 NVIDIA V100 GPUs. Our method does not require this GPU setup but can make use of them when performing inference on large grids. Our multinode, multi-GPU training code and our multinode, multi-GPU inference code can be found here.[32]

3. RESULTS

We first discuss using VDNNs for the TF model. After training on $\mathcal{T}_{TF}$ and $\mathcal{F}_{TF}$ simultaneously, where $\mathcal{F}_{TF}$ was uniformly sampled, we study the accuracy of the model on the validation and testing data. Looking to Figure 3a,b, we plot residuals for
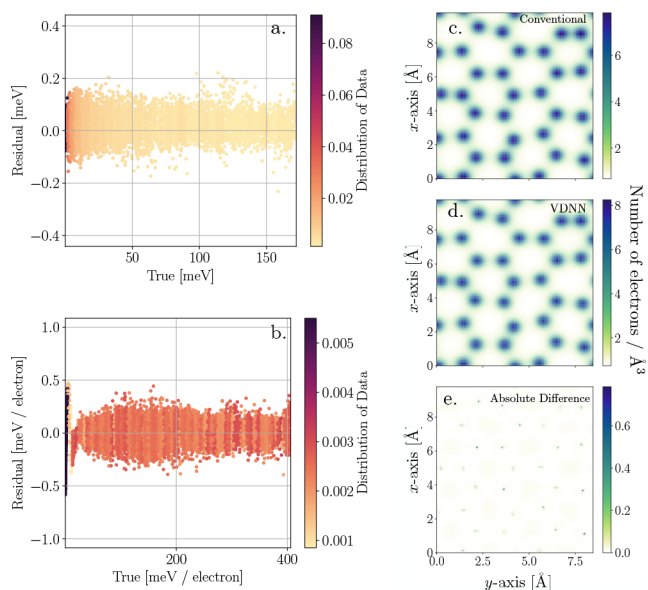

Figure 3. Thomas–Fermi model: Residual (true minus predicted) versus true for (a) $\mathcal{T}_{TF}$ and (b) $\mathcal{F}_{TF}$. (c) Thomas–Fermi electron density from a traditional direct minimization calculation and (d) electron density from a direct minimization calculation with a VDNN trained from a single OF-DFT calculation. (e) Absolute differences between the densities. VDNNs can be used in direct minimization calculations to find electron densities for the Thomas–Fermi model.

$\mathcal{T}_{TF}$ and $\mathcal{F}_{TF}$ for the validation set in units of meV and meV/electron, respectively. Density values have been multiplied by the volume such that direct integration over the numerical grid yields units of energy or energy/electron. From these plots, we can see that the residuals are a small fraction of their respective ranges. MAEs for $\mathcal{T}_{TF}$ and $\mathcal{F}_{TF}$ are 0.04 meV and 0.08 meV/ electron. RMSEs for $\mathcal{T}_{TF}$ and $\mathcal{F}_{TF}$ are 0.05 meV and 0.11 meV/electron. The error for $\mathcal{F}_{TF}$ is larger than $\mathcal{T}_{TF}$. Part of this increase in error can be attributed to the increase in the range of values (a factor of 2.67 from $\mathcal{T}_{TF}$ to $\mathcal{F}_{TF}$), which contributes to 96% of the increased error; the remaining increase in error is due to the VDNN.

We now use VDNNs to calculate an electron density and energy for the second, testing configuration via eq 3. In past reports,[12,33] it was declared unfeasible to directly solve eq 3 because the derivatives of the machine learning model had too much noise. In ref 12, noise was reduced by projecting functional derivatives onto a subspace spanned by relevant vectors via principal component analysis. A similar approach was taken in ref 33, where they projected the functional derivatives onto a subspace spanned by a sinusoidal basis. Here, without any projection scheme, we show that it is possible to use eq 6 to solve for an electron density directly. A projection scheme is not necessary since no derivatives are being taken with respect to the DNN. The VDNN directly outputs the kinetic energy and the functional derivative of the kinetic energy. We used a value of $\alpha = 10^{-3}$ and a uniform electron density as the starting guess. We renormalized the

electron density at every step to enforce charge conservation and deemed a calculation converged when the absolute change of the energy between subsequent steps was less than 10^{-4} Ha. The exact electron density and the electron density found using the VDNN are shown in Figure 3. The densities differ minimally, and the total energy difference found between the two calculations was 19.1 meV/electron. Thus, machine learning models can be used in direct minimization calculations for the Thomas–Fermi model.

We now consider $\mathcal{T}_{KS}$ and $\mathcal{F}_{KS}$. After generating a training data set that uniformly sampled $\sqrt{\rho}\mathcal{F}_{KS}$, we trained a VDNN on $\mathcal{T}_{KS}$ and $\sqrt{\rho}\mathcal{F}_{KS}$ simultaneously with ρ, $\partial_x\rho$, and $\partial_y\rho$ as inputs. We found that including gradients as input channels reduced the mean squared error on the validation set by 7%. Training on $\sqrt{\rho}\mathcal{F}_{KS}$ rather than $\mathcal{F}_{KS}$ reduced the mean squared error on the validation set by 43%. Multiplication of $\sqrt{\rho}$ eliminates $\mathcal{F}_{KS}$ where $\rho = 0$ and enhances $\mathcal{F}_{KS}$ where $\rho \neq 0$. This filter-like behavior allows for an improvement in the predictions. In Figure 4a, we plot the maximum atomic
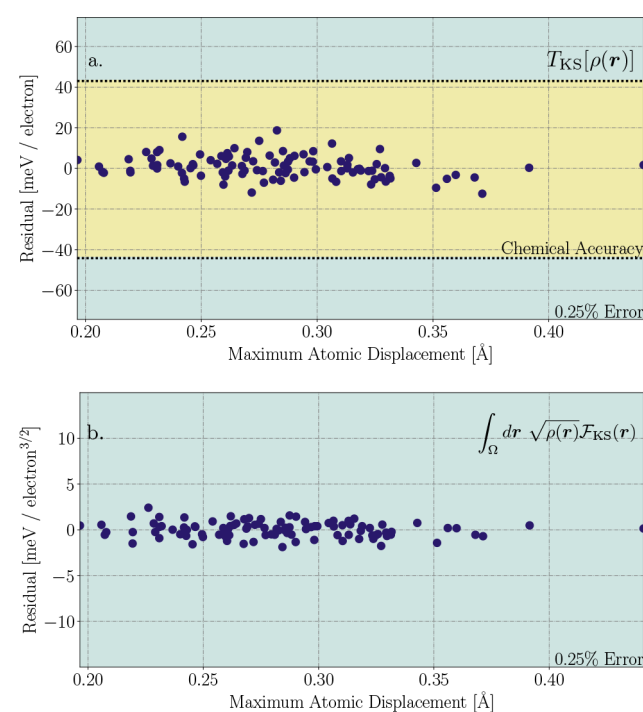
Figure 4. Residuals of (a) $T_{KS}[\rho(\mathbf{r})]$ and (b) $\int_\Omega d\mathbf{r} \sqrt{\rho(\mathbf{r})}\mathcal{F}_{KS}(\mathbf{r})$ versus maximum atomic displacement for the Kohn–Sham model. Predictions are for a test set containing 100 graphene systems with 32 atoms as described in Section 2. VDNNs allow for accurate predictions from small Kohn–Sham density functional theory data sets.

displacement versus residual energy per electron for the 100 testing atomic configurations. To determine percentage errors, the mean of the true kinetic energy values was used. From here, we see that all predictions are within chemical accuracy (43.4 meV). The MAE and RMSE were 4.3 meV/electron and 5.6 meV/electron, respectively. In Figure 4b, we plot true versus residual values for $\int_\Omega d\mathbf{r} \sqrt{\rho(\mathbf{r})}\mathcal{F}_{KS}(\mathbf{r})$. From here, we find that all values are well within 0.25% error. MAE and the RMSE were 0.67 meV/electron$^{3/2}$ and 0.83 meV/electron$^{3/2}$. VDNNs can provide all of the relevant information needed in

OF-DFT. In addition, we find there is no increase in error as the maximum atomic displacement increases. Within the range of maximum atomic displacements, the error remains constant. However, using eq 6, we were unable to obtain the correct electron density for the Kohn−Sham models. We also investigated bulk Al using a 4-atom unit cell with a lattice constant of a = 2.856 Å. We followed the previous methodological protocol while only changing the k-point grid ($8 \times 8 \times 8$ grid). We also found for this system that we could not obtain the correct electron density via direct minimization with VDNNs. These failures, however, are not due to errors of the model but due to the fact that we are only sampling $\mathcal{F}_{KS}$ for converged electron densities. In previous work,[12,33] and for the KS-DFT data, functional derivatives of the kinetic energy are collected for only converged calculations. When using eq 6, one encounters unconverged electron densities and must also know the mapping from these unconverged electron densities to their respective kinetic energy densities and functional derivatives of the kinetic energies. Although $\mathcal{T}_{KS}$ is known for all iterations in a KS-DFT calculation, $\mathcal{F}_{KS}$ is not. The lack of samples of $\mathcal{F}_{KS}$ along the optimization path prevents the insertion of more accurate, kinetic energy machine learning frameworks in OF-DFT. This highlights the need for future work in this area. Solving this challenge would significantly reduce the amount of computation for accurate electronic structure calculations. It should be noted that unconverged densities found along an optimization path are also converged densities of another external potential. If one is able to access these external potentials, the sampling problem would be solved. A promising direction could be to use the differential virial theorem as demonstrated in ref 34. As shown recently in ref 35, this problem could also be solved by considering a differential equation that includes $\mathcal{F}$ and a source function. This source function depends explicitly on the electron density, and $\mathcal{F}$ can be found once this source function is known. Unfortunately, for KS-DFT calculations, this source function is also only known for converged electron densities, but further work in this area could be promising.

We also investigated how VDNNs perform on a toy, two electron system in 1D previously investigated in refs 12 and 14. We used the same ResNet architecture and data set as described in ref 33, with a field of view of 257 voxels for the VDNN, and we compare our results to the ResNet model of ref 33. We found that using $\sqrt{\rho}\mathcal{F}$ also yielded a smaller mean squared error during training, as described above for $\mathcal{F}_{KS}$. For T, our error was ≈75 times larger than ref 33 with an MAE of 0.17 eV (3.84 kcal/mol). This large discrepancy is due to the previous models being trained directly on the energy rather than energy density. This allowed for highly accurate models with errors an order of magnitude less than chemical accuracy. For $\mathcal{F}$, we found that our error was ≈1.9 times larger with an MAE of 0.50 eV/electron (11.42 kcal/mol/electron). However, for $\mathcal{F}$, our maximum absolute error was 1.8 times smaller. In addition, when comparing the errors between T for the 1D system and the 3D system (T_{KS}), we find an increase in error by a factor of ≈20 for the 1D system. As we change the number of physical dimensions, the number of inputs to the model increases. The number of pixels in the 3D case (19^3) is ≈20 times larger compared to the 1D case (257). In 3D, the network has more information to extract features from, which leads to more accurate predictions. It should also be noted that the VDNN is capable of performing inference for an arbitrarily

sized 1D system, so long as the potentials and electron densities are similar to the training set. The existing models from refs 12 and 14 are limited to the same system sizes used during training.

An alternative approach to minimizing Section 1 that avoids computing functional derivatives is MC optimization via the Metropolis algorithm.[36] Direct minimization approaches often require information about derivatives to make a gradient based update. Gradient-free optimization is an alternative approach that does not require such information and is more compatible with machine learning methods since the computational cost associated with inference is low and derivatives can be unreliable. To showcase this potential solution, we consider two electrons in one dimension with the Thomas−Fermi model as the kinetic energy functional. Using this approximation allows us to compare our MC based optimization with a traditional, gradient based optimization. The total energy functional, excluding exchange-correlation effects, can be written as

$$E[\rho] = \frac{\pi^2}{12} \int_l dx \, \rho^3(x) + \frac{1}{2} \int_l dx \int_l dx' \frac{\rho(x')\rho(x)}{|x - x'|}$$
$$+ \int_l dx \sum_{i=1}^{2} -\alpha_i \exp(-(x - \beta_i)^2)\rho(x)$$

(9)

where l is the length of the one-dimensional cell. In Section 9, the first term is the kinetic energy of the one-dimensional Thomas−Fermi model, the second term is the one-dimensional Hartree energy, and the third term is the external energy from a toy, double inverted Gaussian potential. For the external energy, we used the following parameters: α_1 = 1.0 Ha/electron, α_1 = 2.0 Ha/electron, β_1 = −0.5 Bohr, β_2 = 1.0 Bohr. For the kinetic energy, we trained a three layer, fully connected neural network that maps a value of ρ to a value of the one-dimensional kinetic energy density. We generated 10^5 random numbers from 0 to 1, which represented values of density, and trained the network for 100 epochs with a batch size of 100 and a learning rate of 10^{-5}. We did not perform any standardization or normalization, and we used ELU activation functions throughout the network. For the MC simulation, we performed simulated annealing with a starting value of β^{-1} = 10^{-4} Ha which was decreased according to the formula β_{new}^{-1} = $\beta_{old}^{-1}/(1.0 + 2 \times 10^{-6})^n$, where n is the iteration number. After 2 million iterations, β^{-1} = 1.87×10^{-6} Ha. At each iteration, we updated all values of ρ in two steps. The first step was computing a random change

$$\Delta\rho = 1000(\rho(x) + 10\sigma)u(\sigma, x)$$

(10)

where σ is the standard deviation of a normal distribution, and $u(\sigma, x)$ is the function generated from a normal distribution centered at zero with the same shape as $\rho(x)$. The random change is then updated according to

$$\Delta\rho = \Delta\rho - \rho\frac{\langle\Delta\rho\rangle}{\langle\rho\rangle}$$

(11)

where $\langle f \rangle$ is the mean of f. We used a standard deviation of σ = 10^{-5} which was reduced during the simulation following the same protocol as β. All proposed values of ρ that were negative were set to zero, and ρ was renormalized at every step before evaluating the energy. In Figure 5, we plot ρ and the potential (Hartree + external) for a traditional direct minimization calculation, following eq 6 alongside ρ and the potential for the
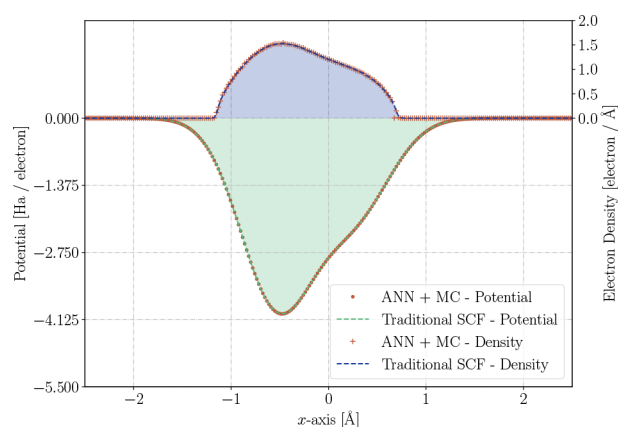
Figure 5. Comparison of electron densities and potentials between a Monte Carlo optimization (red crosses and points) and a traditional self-consistent field calculation (blue and green dashed lines) for two electrons in 1D with the external potential described in Section 9. The Monte Carlo optimization yields a very similar electron density to a self-consistent field calculation.

MC simulation. For the traditional gradient based calculation, the energy was declared converged when the difference in energy between subsequent steps was $<10^{-6}$ Ha. There is excellent agreement between the MC optimization and the gradient based optimization. The mean absolute differences between the charge densities, potentials, and total energies were 3.74×10^{-3} electron/Å, 5.61×10^{-5} meV/electron, and 1.70 meV, respectively. Future work involves implementing a three-dimensional, functional derivative-free, OF-MC algorithm capable of calculating more accurate electron densities with improved, machine-learned kinetic energy functionals.

4. CONCLUSION

We have shown that VDNNs can be used to accurately predict the kinetic energy density and the functional derivative of the kinetic energy for Kohn−Sham and Thomas−Fermi theories. This methodology drastically reduces the number of electronic structure calculations needed to generate a training set. We have shown that one can obtain an accurate charge density and total energy after training with data from only one direct minimization calculation for the Thomas−Fermi model. Similarly, we have shown that we can calculate accurate kinetic energies from only two converged calculations for Kohn−Sham density functional theory. Additionally, we show that this accuracy is held to arbitrary system size. Currently, one cannot use voxel deep neural networks to solve for Kohn−Sham electron densities via direct minimization. This is due to the fact that one only obtains the functional derivative of the kinetic energy from a Kohn−Sham calculation when convergence has been reached. However, unconverged electron densities found along an optimization path are also converged electron densities with another external potential. If these external potentials are found, the functional derivative of the kinetic energy would be known along an optimization path, and one could use a voxel deep neural network in a direct minimization calculation. In addition, we show that an alternative, functional derivative-free, Monte Carlo based orbital-free algorithm could also be used to determine ground-state electron densities.

■ ASSOCIATED CONTENT

ⓈI Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jctc.1c00812.

Hyperparameter convergence results as well as other results (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

Kevin Ryczko − *Department of Physics, University of Ottawa, Ottawa, Ontario K1N 6N5, Canada; 1QB Information Technologies (1QBit), Vancouver, British Columbia V6E 4B1, Canada; Vector Institute for Artificial Intelligence, Toronto, Ontario M5G 1M1, Canada;* ⓞ orcid.org/0000-0001-6933-3856; Email: kevin.ryczko@uottawa.ca

Isaac Tamblyn − *Department of Physics, University of Ottawa, Ottawa, Ontario K1N 6N5, Canada; Vector Institute for Artificial Intelligence, Toronto, Ontario M5G 1M1, Canada;* Email: isaac.tamblyn@uottawa.ca

Authors

Sebastian J. Wetzel − *Perimeter Institute for Theoretical Physics, Waterloo, Ontario N2L 2Y5, Canada*

Roger G. Melko − *Perimeter Institute for Theoretical Physics, Waterloo, Ontario N2L 2Y5, Canada; Department of Physics and Astronomy, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada*

Complete contact information is available at:
https://pubs.acs.org/10.1021/acs.jctc.1c00812

Notes

The authors declare no competing financial interest.

■ REFERENCES

(1) Kohn, W.; Sham, L. J. Self-consistent equations including exchange and correlation effects. *Phys. Rev.* **1965**, *140*, A1133.

(2) Wang, Y. A.; Carter, E. A. *Theoretical Methods in Condensed Phase Chemistry*; Springer: 2002; pp 117−184, DOI: 10.1007/0-306-46949-9_5.

(3) Lignères, V. L.; Carter, E. A. *Handbook of Materials Modeling*; Springer: 2005; pp 137−148, DOI: 10.1007/978-1-4020-3286-8_9.

(4) Hung, L.; Carter, E. A. Accurate simulations of metals at the mesoscale: Explicit treatment of 1 million atoms with quantum mechanics. *Chem. Phys. Lett.* **2009**, *475*, 163−170.

(5) Thomas, L. H. The calculation of atomic fields. *Proc. Cambridge Philos. Soc.* **1927**, *23*, 542−548.

(6) Fermi, E. Atti Rentes Accad. *Naz. Lincei Rend. Cl. Sci. Fis, Mater. Natur* **1927**, *6*, 602.

(7) Gombas, P. *Handbuch der Physik*; Springer: Berlin, 1956; Vol. *36*, pp 109−231, DOI: 10.1007/978-3-642-85687-7_2.

(8) March, N. The Thomas-Fermi approximation in quantum mechanics. *Adv. Phys.* **1957**, *6*, 1−101.

(9) Lieb, E. H. Erratum: Thomas-Fermi and related theories of atoms and molecules. *Rev. Mod. Phys.* **1982**, *54*, 311.

(10) Dunlap, B.; Connolly, J.; Sabin, J. On first-row diatomic molecules and local density models. *J. Chem. Phys.* **1979**, *71*, 4993−4999.

(11) Hohenberg, P.; Kohn, W. Inhomogeneous electron gas. *Phys. Rev.* **1964**, *136*, B864.

(12) Snyder, J. C.; Rupp, M.; Hansen, K.; Müller, K.-R.; Burke, K. Finding density functionals with machine learning. *Phys. Rev. Lett.* **2012**, *108*, 253002.

(13) Brockherde, F.; Vogt, L.; Li, L.; Tuckerman, M. E.; Burke, K.; Müller, K.-R. Bypassing the Kohn-Sham equations with machine learning. *Nat. Commun.* **2017**, *8*, 872.

(14) Meyer, R.; Weichselbaum, M.; Hauser, A. W. Machine Learning Approaches toward Orbital-free Density Functional Theory: Simultaneous Training on the Kinetic Energy Density Functional and Its Functional Derivative. *J. Chem. Theory Comput.* **2020**, *16*, 5685−5694. PMID: 32786898

(15) Nagai, R.; Akashi, R.; Sugino, O. Completing density functional theory by machine learning hidden messages from molecules. *npj Comput. Mater.* **2020**, *6*, 43.

(16) Kalita, B.; Li, L.; McCarty, R. J.; Burke, K. Learning to Approximate Density Functionals. *Acc. Chem. Res.* **2021**, *54*, 818−826.

(17) Schleder, G. R.; Padilha, A. C.; Acosta, C. M.; Costa, M.; Fazzio, A. From DFT to machine learning: recent approaches to materials science−a review. *JPhys. Mater.* **2019**, *2*, 032001.

(18) Zhou, Y.; Wu, J.; Chen, S.; Chen, G. Toward the Exact Exchange−Correlation Potential: A Three-Dimensional Convolutional Neural Network Construct. *J. Phys. Chem. Lett.* **2019**, *10*, 7264−7269.

(19) Ryabov, A.; Akhatov, I.; Zhilyaev, P. Neural network interpolation of exchange-correlation functional. *Sci. Rep.* **2020**, *10*, 8000.

(20) Lyon, K. *From Fundamentals to Spectroscopic Applications of Density Functional Theory* **2020**, in press.

(21) Yang, C.; Meza, J. C.; Wang, L.-W. A constrained optimization algorithm for total energy minimization in electronic structure calculations. *J. Comput. Phys.* **2006**, *217*, 709−721.

(22) Weber, V.; VandeVondele, J.; Hutter, J.; Niklasson, A. M. Direct energy functional minimization under orthogonality constraints. *J. Chem. Phys.* **2008**, *128*, 084113.

(23) Shao, X.; Jiang, K.; Mi, W.; Genova, A.; Pavanello, M. DFTpy: An efficient and object-oriented platform for orbital-free DFT simulations. *arXiv preprint arXiv:2002.02985*, 2020. https://arxiv.org/abs/2002.02985 (accessed 2022-01-05).

(24) Van Setten, M.; Giantomassi, M.; Bousquet, E.; Verstraete, M. J.; Hamann, D. R.; Gonze, X.; Rignanese, G.-M. The PseudoDojo: Training and grading a 85 element optimized norm-conserving pseudopotential table. *Comput. Phys. Commun.* **2018**, *226*, 39−54.

(25) Gonze, X.; et al. The Abinit project: Impact, environment and recent developments. *Comput. Phys. Commun.* **2020**, *248*, 107042.

(26) Perdew, J. P.; Burke, K.; Ernzerhof, M. Generalized gradient approximation made simple. *Phys. Rev. Lett.* **1996**, *77*, 3865.

(27) Liu, S.; Ayers, P. W. Functional derivative of noninteracting kinetic energy density functional. *Phys. Rev. A* **2004**, *70*, 022501.

(28) Ryczko, K.; Mills, K.; Luchak, I.; Homenick, C.; Tamblyn, I. Convolutional neural networks for atomistic systems. *Comput. Mater. Sci.* **2018**, *149*, 134−142.

(29) Ryczko, K.; Strubbe, D. A.; Tamblyn, I. Deep learning and density-functional theory. *Phys. Rev. A* **2019**, *100*, 022512.

(30) Klambauer, G.; Unterthiner, T.; Mayr, A.; Hochreiter, S. Self-normalizing neural networks. *arXiv preprint arXiv:1706.02515*, 2017. https://arxiv.org/abs/1706.02515 (accessed 2022-01-05).

(31) You, Y.; Gitman, I.; Ginsburg, B. Scaling sgd batch size to 32k for imagenet training. *arXiv preprint arXiv:1708.03888* **2017**, *6*, 12.

(32) Ryczko, K.; Tamblyn, I. Codes used in manuscript. 2021. clean.energyscience.ca/codes (accessed 2022-01-05).

(33) Meyer, R.; Weichselbaum, M.; Hauser, A. W. Machine Learning Approaches toward Orbital-free Density Functional Theory: Simultaneous Training on the Kinetic Energy Density Functional and Its Functional Derivative. *J. Chem. Theory Comput.* **2020**, *16*, 5685−5694.

(34) Ryabinkin, I. G.; Staroverov, V. N. Exact relations between the electron density and external potential for systems of interacting and noninteracting electrons. *Int. J. Quantum Chem.* **2013**, *113*, 1626−1632.

(35) Ghasemi, S. A.; Kühne, T. D. Artificial neural networks for the kinetic energy functional of non-interacting fermions. *J. Chem. Phys.* **2021**, *154*, 074107.

(36) Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E. Equation of state calculations by fast computing machines. *J. Chem. Phys.* **1953**, *21*, 1087−1092.

⊞ Recommended by ACS