Deep Density: circumventing the Kohn-Sham equations via symmetry preserving neural networks

A Preprint

Leonardo Zepeda-Núñez
Department of Mathematics
University of Wisconsin-Madison
Madison, WI 53706
zepedanunez@wisc.edu

Yixiao Chen
Program in Applied and Computational Mathematics
Princeton University
Princeton, NJ 08544
yixiaoc@princeton.edu

Jiefu Zhang
Department of Mathematics
University of California, Berkeley
Berkeley, CA 94720
jiefuzhang@berkeley.edu

Weile Jia
Department of Mathematics
University of California, Berkeley
Berkeley, CA 94720
jiaweile@berkeley.edu

Linfeng Zhang
Program in Applied and Computational Mathematics
Princeton University
Princeton, NJ 08544
linfengz@princeton.edu

Lin Lin
Department of Mathematics
University of California, Berkeley
Computational Research Division,
Lawrence Berkeley National Laboratory,
Berkeley, CA 94720, USA
linlin@math.berkeley.edu

December 2, 2019

Abstract

The recently developed Deep Potential [Phys. Rev. Lett. 120, 143001, 2018] is a powerful method to represent general inter-atomic potentials using deep neural networks. The success of Deep Potential rests on the proper treatment of locality and symmetry properties of each component of the network. In this paper, we leverage its network structure to effectively represent the mapping from the atomic configuration to the electron density in Kohn-Sham density function theory (KS-DFT). By directly targeting at the self-consistent electron density, we demonstrate that the adapted network architecture, called the Deep Density, can effectively represent the electron density as the linear combination of contributions from many local clusters. The network is constructed to satisfy the translation, rotation, and permutation symmetries, and is designed to be transferable to different system sizes. We demonstrate that using a relatively small number of training snapshots, Deep Density achieves excellent performance for one-dimensional insulating and metallic systems, as well as systems with mixed insulating and metallic characters. We also demonstrate its performance for real three-dimensional systems, including small organic molecules, as well as extended systems such as water (up to 512 molecules) and aluminum (up to 256 atoms).

1 Introduction

Kohn-Sham density function theory (KS-DFT) [1] is the most widely-used electronic structure theory because the electron density completely determines the ground state [2] and the thermal [3] properties of a quantum

many-body system. The goal of KS-DFT is to obtain a mapping of the atomic configuration to the electron density, denoted by $\varrho(\mathbf{r}, \{\mathbf{R}_I\}_{I=1}^{N_a})$. Here $\mathbf{R}_I$ is the position of the I-th nuclei, $\mathbf{r}$ is the electronic position, and N_a is the number of atoms. Notwithstanding its enormous success, KS-DFT still suffers from two significant challenges. The first challenge is its computational cost; the cost of KS-DFT calculations typically scales cubically with respect to the system size, and hence the calculations can be expensive for large systems. Despite the availability and development of linear scaling methods [4, 5], they are only applicable to treating insulating systems with relatively large energy gaps. The second, and more fundamental, challenge is its accuracy, which cannot be systematically improved due to the limitation of the available exchange-correlation functionals [6].

The recent surge in applications of machine learning methods to scientific computing problems provides an alternative route for revisiting both problems. If one can find a more effective mapping for ϱ using, e.g., a neural network, we can bypass the solution of the Kohn-Sham equations, and directly obtain the self-consistent electron density for a given atomic configuration. This may drastically reduce the computational time, particularly for large systems. We may further use such a network to encode the electron density obtained from theories that are more accurate than standard KS-DFT, such as the density matrix renormalization group [7] or the coupled-cluster theory [8]. Based on such considerations, the representation of the electron density has received much attention in the past few years [9, 10, 11, 12, 13]. These approaches are typically based on carefully hand-crafted descriptors that encode the atomic configuration, a projection of ϱ onto a basis, and a machine learning algorithm mapping the descriptors to the coefficients of the projection. For example, in [9] the authors adopted a descriptor using a fictitious potential centered at each nuclei, which is then mapped to a Fourier basis using a ridge kernel regression algorithm.

The problem of using machine learning methods to represent the electron density is closely related to the problem of finding the interatomic potential and corresponding force field for molecular dynamics simulation [14, 15, 16, 17, 18, 19, 20, 21]. In particular, the recently developed Deep Potential scheme [22, 23] has been successful in describing with high fidelity various finite-size and extended systems, including organic molecules, metals, semiconductors, and insulators.

In this work, we leverage the construction of the Deep Potential to build a neural network representation for ϱ. In contrast with related approaches, we learn the descriptor on the fly, and instead of learning the mapping to the coefficients of a basis, we evaluate the total electron density directly in a point-wise manner. The total electron density is decomposed into a linear combination of N_a components, with the I-th component describing the contribution to the electron density from the I-th atom and its neighbors. Each component is constructed to locally satisfy the necessary translation, rotation, and permutation symmetries. The number of atoms involved in each component does not scale with respect to the global system size, and hence the cost for evaluating ϱ scales linearly with respect to the system size.

By targeting the self-consistent electron density, we demonstrate that Deep Density can take advantage of the screening effect to effectively represent ϱ for both insulating and metallic systems, and thus can bypass the solution of the nonlinear Kohn-Sham equations. The resulting algorithm is not tied to a given discretization scheme nor a particular choice of basis, and it can be transferred to systems of larger sizes. The algorithm can also be implemented in an embarrassingly parallel fashion: one can evaluate different points of the density given by different configuration completely independently. The total cost of evaluating the density scales linearly with the number of atoms in the system.

We demonstrate that Deep Density can accurately predict the electron density and exhibits excellent transferability properties for one-dimensional model systems of insulating, metallic, as well as mixed insulating-metallic characters. We also report the performance of our model for three-dimensional real systems, including small molecules (C_2H_6, C_4H_{10}), as well as condensed matter systems, including water (up to 512 water molecules), and aluminum (up to 256 aluminum atoms).

The rest of the paper is organized as follows. In Section 2 we provide the framework of KS-DFT, and the map we seek to approximate. In Section 3 we provide the architecture for the neural network. In particular, we provide a succinct review of the techniques in [24] and we explain how to implement the physical ansatz and the symmetry requirements. In Section 4 we provide the numerical examples showcasing the accuracy and transferability of the algorithm. In particular, we provide the electron density for 1D models (Section 4.1) and realistic 3D systems (Section 4.2) and we compare them against the electron density produced by classical KS-DFT computations. We show that it is possible to train these models to single precision in 1D and within three digits in 3D, and that the accuracy is maintained even for test systems an order of magnitude larger. In Section 5 we provide several comments about the current work and we point to several future directions of research. Additional details for the numerical experiments are given in the appendices.

2 Preliminaries

For a system with N_e electrons at a given atomic configuration $\{\mathbf{R}_I\}_{I=1}^{N_a}$ in a d-dimensional space (i.e. $\mathbf{r}, \mathbf{R}_I \in \mathbb{R}^d$), KS-DFT solves the following nonlinear eigenvalue problem (spin omitted for simplicity)

$$H[\rho; \{\mathbf{R}_I\}]\psi_i = \varepsilon_i \psi_i, \quad i = 1, ..., N_e, \tag{1}$$

$$\int \psi_i^*(\mathbf{r})\psi_j(\mathbf{r})\,\mathrm{d}\mathbf{r} = \delta_{ij}, \quad \rho(\mathbf{r}) = \sum_{i=1}^{N_e} |\psi_i(\mathbf{r})|^2 . \tag{2}$$

The Kohn-Sham Hamiltonian is

$$H[\rho; \{\mathbf{R}_I\}] := -\frac{1}{2}\Delta_{\mathbf{r}} + V_{\text{ion}}(\mathbf{r}; \{\mathbf{R}_I\}) + V_{\text{hxc}}[\mathbf{r}; \rho]. \tag{3}$$

Here V_{ion} characterizes the interaction between electrons and nuclei, and it does not depend on the electron density. V_{hxc} is called the Hartree-exchange-correlation potential, which includes the mean-field electron-electron interaction, as well as the contribution from the exchange-correlation energy. The eigenvalues $\{\varepsilon_i\}_{i=1}^{N_e}$ are real and ordered non-decreasingly, so $\{\psi_i\}_{i=1}^{N_e}$ correspond to the eigenfunctions with lowest N_e eigenvalues. Due to the ρ-dependence of V_{hxc}, the Kohn-Sham equations needs to be solved self-consistently till convergence. We refer to [6, 25] for more details of numerical solutions of KS-DFT. For the purpose of this paper, we are interested in learning the mapping

$$\varrho : \{\mathbf{r}\} \cup \{\mathbf{R}_I\}_{I=1}^{N_a} \mapsto \rho(\mathbf{r}), \tag{4}$$

which is also denoted as $\varrho(\mathbf{r}, \{\mathbf{R}_I\}_{I=1}^{N_a}) = \rho(\mathbf{r})$.

In KS-DFT, we need to distinguish between the external potential $V_{\text{ion}}(\mathbf{r}; \{\mathbf{R}_I\})$, and the effective potential

$$V_{\text{eff}}(\mathbf{r}) = V_{\text{ion}}(\mathbf{r}; \{\mathbf{R}_I\}) + V_{\text{hxc}}[\mathbf{r}; \rho]. \tag{5}$$

We refer to the mapping from V_{eff} to ρ as the linearized Kohn-Sham map. The evaluation of the Kohn-Sham equations requires (partially) diagonalizing the Hamiltonian $H_{\text{eff}} := -\frac{1}{2}\Delta_{\mathbf{r}} + V_{\text{eff}}(\mathbf{r})$, after proper discretization. Correspondingly we refer to the mapping from V_{ion} to ρ as the self-consistent Kohn-Sham map, of which the evaluation requires solving the Kohn-Sham equations self-consistently.

For insulating systems with a positive energy gap (i.e. $\varepsilon_{N_e+1} - \varepsilon_{N_e} > 0$), it is known that the dependence of electron density at position $\mathbf{r}$ with respect to the potential at position $\mathbf{r}'$ decays exponentially with respect to $|\mathbf{r} - \mathbf{r}'|$. This is often referred to as the "nearsightedness" principle of electrons [26, 27]. More specifically, the Fréchet derivative

$$\chi_0(\mathbf{r}, \mathbf{r}') = \frac{\delta\rho(\mathbf{r})}{\delta V_{\text{eff}}(\mathbf{r}')}, \tag{6}$$

which is also called the irreducible polarizability operator in physics literature, satisfies the decay property $\chi_0(\mathbf{r}, \mathbf{r}') \sim e^{-c|\mathbf{r}-\mathbf{r}'|}$ for some constant $c > 0$. The nearsightedness principle allows the design of linear scaling methods [4, 5], which effectively truncate the global domain into many small domains to solve KS-DFT.

For metallic systems with a zero or very small energy gap, $\chi_0(\mathbf{r}, \mathbf{r}')$ only decays algebraically as $|\mathbf{r} - \mathbf{r}'| \to \infty$. Hence the nearsightedness principle does not hold anymore, and linear scaling methods become either very expensive (with a large truncation radius) or inaccurate (with a small truncation radius). On the other hand, even for metallic systems, the reducible polarizability operator, defined as

$$\chi(\mathbf{r}, \mathbf{r}') = \frac{\delta\rho(\mathbf{r})}{\delta V_{\text{ion}}(\mathbf{r}')}, \tag{7}$$

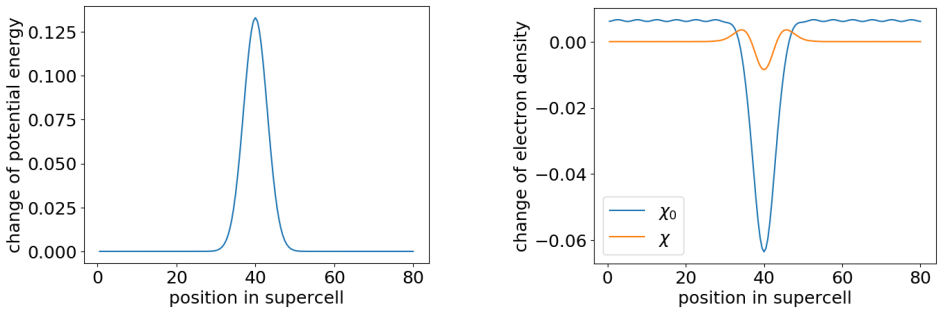
can be much more localized compared to χ_0. This is known as the screening effect [28, 29].

To illustrate the screening effect, we consider a one-dimensional periodic metallic system with 8 atoms. The details of the setup will be discussed in Section 4.1 and Appendix A. We introduce a localized and exponentially decaying perturbation of the potential δV_{ion} as

$$\delta V_{\text{ion}}(r) = \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left(-\frac{1}{2\sigma_0^2}(r - \mu_0)^2\right), \tag{8}$$

where we choose $\sigma_0 = 3.0$, and $\mu_0 = 40$ is the center of the supercell. Fig. 1(a) displays the profile of δV_{ion}. Fig. 1(b) shows that for a metallic system, the induced electron density obtained from the linearized

Kohn-Sham map, which can be approximately computed as $\chi_0 \delta V_{\text{ion}}$, has a large magnitude, and a delocalized and oscillatory tail. On the other hand, due to the screening effect, the magnitude of the induced electron density obtained from the self-consistent Kohn-Sham map, which can be approximately computed as $\chi \delta V_{\text{ion}}$, has a much smaller magnitude. Its tail is much shorter and smoother, and the support of $\chi \delta V_{\text{ion}}$ is localized around that of δV_{ion}. The screening effect indicates that Deep Density should be able to leverage such an additional level of localization which is not present in standard linear scaling solvers, and a relatively small truncation radius may be possible even for metallic systems (see the architecture in Section 3).



(a) Perturbation δV_{ion}

(b) Comparison of $\chi \delta V_{\text{ion}}$ and $\chi_0 \delta V_{\text{ion}}$

Figure 1: Illustration of the screening effect for a one-dimensional model metallic system.

3 Network architecture

3.1 Locality

Standard methods for solving KS-DFT can only be used to evaluate the linearized Kohn-Sham map, which is then used to obtain the self-consistent electron density via self-consistent field iterations. In contrast, our goal is to *directly* learn the self-consistent electron density, or the mapping $\varrho(\mathbf{r}, \{\mathbf{R}_I\}_{I=1}^{N_a})$, which already takes into account the screening effect. This allows us to construct neural networks that can be decomposed into the linear combination of local components for both insulating and metallic systems.

As illustrated in Fig. 2, we partition the electron density as

$$\varrho(\mathbf{r}, \{\mathbf{R}_I\}_{I=1}^{N_a}) = \sum_{I=1}^{N_a} \varrho^I(\mathbf{r}, \mathcal{R}^I). \tag{9}$$

Here ϱ^I characterizes the contribution to the electron density at $\mathbf{r}$ from the neighborhood of the I-th atom. The locality is imposed by building an interaction list $\mathring{\mathcal{I}}_{R_c}(I)$, defined as the set of indices J such that $|\mathbf{R}_J - \mathbf{R}_I| < R_c$. Following this notation, $\mathcal{R}^I$ denotes the set of atomic positions

$$\mathcal{R}^I = \{\mathbf{R}_J, J \in \mathring{\mathcal{I}}_{R_c}(I)\}.$$

We denote by $s(I)$ the species index (i.e. the atomic number) of the I-th atom. We may also treat the electron as a special "atom" equipped with index $J = 0$. For simplicity we define $s(0) = 0, \mathbf{R}_0 = \mathbf{r}$, and then define the extended interaction list as the index set

$$\mathcal{I}_{R_c}(I) = \{0\} \cup \mathring{\mathcal{I}}_{R_c}(I). \tag{10}$$

In other words, the electron (formally) belongs to $\mathcal{I}_{R_c}(I)$ for every atom I.

The density ϱ^I can be constructed as

$$\varrho^I(\mathbf{r}, \mathcal{R}^I) = \mathcal{N}^{\mathtt{lin},I}(\mathbf{r}, \mathcal{R}^I) e^{C_{s(I)}\left(|\mathbf{r}-\mathbf{R}_I| - D_{s(I)}\right)^2 + \mathcal{E}^I(\mathbf{r}, \mathcal{R}_I)}. \tag{11}$$

We assume $\mathcal{N}^{\mathtt{lin},I}$ takes the form

$$\mathcal{N}^{\mathtt{lin},I}_{s(I)}(\mathbf{r}, \mathcal{R}^I) = \mathcal{N}^I(\mathbf{r}, \mathcal{R}_I) + A_{s(I)}|\mathbf{r} - \mathbf{R}_I| + B_{s(I)}. \tag{12}$$

The constants A, B, C, D are trainable parameters and only depend on the species of the I-th atom.

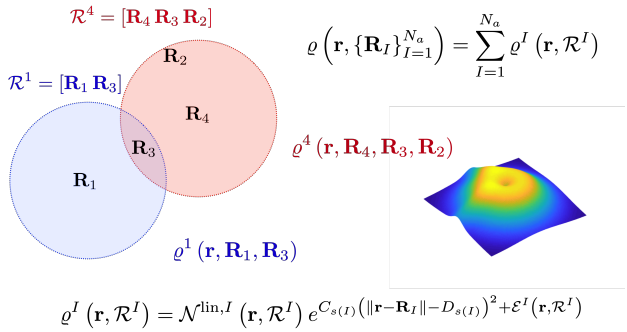


Figure 2: Illustration of the local interaction lists and local electron densities.

The ansatz is built such that the electron density decays exponentially with the distance of the electron to the center of the cluster, i.e., the I-th atom. The rate of decay is given by A, and B accounts for the possibility that the bulk of the electron density is not centered at the atom I. In addition, due to possible issues with the pseudopotential in the KS-DFT computation, the electron density can be very small in the neighborhood of the nuclei. In order to fit this the nonlinear correction, $\mathcal{E}^I(\mathbf{r}, \mathcal{R}_I)$ would need to have a large negative value. Thus, in order to bypass this issue we multiply the exponential with a linear term to capture this behavior. Moreover, to capture abrupt changes on the density we add $\mathcal{N}^I$, a non-linear correction, to this term.

In a nutshell, the exponential term mimics the behavior of the local density far from the center of the cluster, whereas $\mathcal{N}^{\mathtt{lin},I}$ captures the behavior close to center of atom I. $\mathcal{N}^I, \mathcal{E}^I$ are neural networks to be introduced later. Clearly we may absorb the constants A, B, C, D into the neural networks as well. In practice, we found that separating out such terms can reduce the training time and the test error of the network. In addition, numerically we observed that the derivative associated to the constant may widely differ from the rest of the neural network, introducing them explicitly allows us to re-scale the gradient if necessary, thus accelerating the optimization.

3.2 Symmetry

When one considers modeling the physics at different scales, it is often crucial to preserve symmetry properties. Additionally, from a learning perspective, symmetries can also significantly improve the efficiency of training in terms of the required number of parameters, the number of training steps, and the amount of training data. In this regard, as a mapping from electronic and atomic positions to the value of the electron density, $\varrho(\mathbf{r}, \{\mathbf{R}_I\}_{I=1}^{N_a})$ should be invariant under permutation of indices of identical atoms, and under a collective translation or rotation of the electron and atomic positions. As mentioned in the introduction, the problem can also be considered as finding a map from atomic positions to a electron density field represented by electron positions or other basis sets. One disadvantage is that the dimension of the output may be large: the output consists of values of the density on a list of grid points, or the coefficients corresponding to a basis set. When the output is a list of grid points, it can also be difficult to satisfy symmetry requirements. Here we follow the first approach, and we find it is much easier to define a neural network representation with a single output. Finally, to combine the locality and the symmetry requirements, we guarantee that the values of $\mathcal{N}^I$ and $\mathcal{E}^I$ in Deep Density are invariant with respect to the following symmetry operations:

1. Permutation: relabeling of indices of the same atom species in the index set $\mathcal{I}_{R_c}(I)$.
2. Translation: uniformly shifting $\mathbf{R}_I$ and the positions of all particles in $\mathcal{I}_{R_c}(I)$ by a vector.
3. Rotation: rotating all particles in $\mathcal{I}_{R_c}(I)$ around $\mathbf{R}_I$.

In order to reduce the number of parameters, all neural networks involved should share the same set of parameters if the species of the particles involved are the same.

We now illustrate the treatment of $\mathcal{N}^I$; the procedure for $\mathcal{E}^I$ is analogous. In order to satisfy permutation symmetry, we adopt a variant of the representation in [30], which states that any permutation invariant

function $f : \mathbb{R}^N \to \mathbb{R}$ can be represented in the form

$$f(\mathbf{x}) = f(x_1, \ldots, x_N) = \mathcal{F}\left(\sum_{i=1}^{N} h(x_i)\right), \tag{13}$$

where $h : \mathbb{R} \to \mathbb{R}^M$ maps each coordinate x_i into an M-dimensional feature space, and $\mathcal{F} : \mathbb{R}^M \to \mathbb{R}$ combines the features to obtain the value f. Both $\mathcal{F}$ and h can are represented by neural networks in what follows.

In this work, we follow the construction in [23], and decompose $\mathcal{N}^I$ as

$$\mathcal{N}^I(\mathbf{r}, \mathcal{R}^I) = \mathcal{F}_{s(I)} \circ \mathcal{D}^I(\mathbf{r}, \mathcal{R}^I). \tag{14}$$

Here $\mathcal{F}_{s(I)} : \mathbb{R}^{M\times M} \to \mathbb{R}$ is called a fitting network that only depends on the species index $s(I)$. $\mathcal{D}^I(\mathbf{r}, \mathcal{R}^I)$ is called a descriptor network, and $\mathcal{D}^I(\mathbf{r}, \mathcal{R}^I) \in \mathbb{R}^{M\times M}$ is a matrix of the following form

$$\mathcal{D}^I(\mathbf{r}, \mathcal{R}^I) = \left(\sum_{J\in\mathcal{I}_{R_c}(I)} h_{s(I),s(J)}(\mathbf{R}_J - \mathbf{R}_I)\right)^\top \left(\sum_{J\in\mathcal{I}_{R_c}(I)} h_{s(I),s(J)}(\mathbf{R}_J - \mathbf{R}_I)\right). \tag{15}$$

Here $h_{s(I),s(J)} : \mathbb{R}^d \to \mathbb{R}^{\tilde{d}\times M}$ is a feature mapping, which only depends on the species of the particle I and J (recall that the electron is labeled with $J = 0$ with species index 0). Then $\mathcal{D}^I \in \mathbb{R}^{M\times M}$ is a symmetric matrix that naturally satisfies the permutation and translation symmetries. As will be demonstrated below, it satisfies the rotation symmetry as well.

Now we demonstrate the construction of the mapping $h_{s(I),s(J)}$. Define $R_{JI} = |\mathbf{R}_J - \mathbf{R}_I|$, and we would like to require h to depend smoothly on $\mathbf{R}_J$ moves in and out of the index set $\mathcal{I}_{R_c}(I)$. In other words, h should continuously vanish as R_{JI} approaches R_c. We may introduce a cutoff function

$$\phi(R) = \begin{cases} \frac{1}{R+\delta}, & 0 \le R \le R_{cs} \\ \frac{1}{R+\delta}\left\{\frac{1}{2}\cos\left[\pi\frac{(R-R_{cs})}{(R_c-R_{cs})}\right] + \frac{1}{2}\right\}, & R_{cs} < R < R_c \\ 0, & R \ge R_c \end{cases} \tag{16}$$

where $0 < R_{cs} < R_c$. Note that $\phi \in C^1(\mathbb{R}^+ \cup \{0\})$ for any $\delta > 0$.

Then we may define the generalized coordinate as

$$\mathbf{d}_J^I = \begin{bmatrix} \phi(R_{JI}) \\ \frac{\phi(R_{JI})}{R_{JI}}(\mathbf{R}_J - \mathbf{R}_I) \end{bmatrix} \in \mathbb{R}^{d+1}, \quad J \in \mathring{\mathcal{I}}_{R_c}(I). \tag{17}$$

Here $\tilde{d} := d + 1$ is the dimension of the generalized coordinate. In principle we may use a different set of generalized coordinates for electrons. For simplicity, in this work we apply the same definition as in (17) to the electron, with the same truncation radius R_c. Therefore, effectively, the electron at position $\mathbf{r}$ only belongs to the extended interaction lists of its neighboring atoms.

We require h to depend only on the generalized coordinates as

$$h_{s(I),s(J)}(\mathbf{R}_J - \mathbf{R}_I) = \mathbf{d}_J^I \left[g_{s(I),s(J)}\left((\mathbf{d}_J^I)_1\right)\right]^\top \in \mathbb{R}^{\tilde{d}\times M}. \tag{18}$$

Here $g_{s(I),s(J)} : \mathbb{R} \to \mathbb{R}^M$ is a neural network that only depends on the first component of $\mathbf{d}_J^I$ (*i.e.* the radial information R_{JI}), and only depends on the species of the particles I, J. Combining Eq. (15) and (18), we have

$$\mathcal{D}^I(\mathbf{r}, \mathcal{R}^I) = \sum_{J,J'\in\mathcal{I}_{R_c}(I)} \left[g_{s(I),s(J)}\left((\mathbf{d}_J^I)_1\right)\right]\left[(\mathbf{d}_J^I)^\top(\mathbf{d}_{J'}^I)\right]\left[g_{s(I),s(J')}\left((\mathbf{d}_{J'}^I)_1\right)\right]^\top. \tag{19}$$

Both the radial information $(\mathbf{d}_J^I)_1$ and the inner product $(\mathbf{d}_J^I)^\top(\mathbf{d}_{J'}^I)$ satisfy the rotation symmetry. Therefore the descriptor $\mathcal{D}^I$ is invariant to permutation, rotation, and translation symmetry operations.

Fig. 3 provides a simplified illustration for computing $\mathcal{D}^I$, where the matrix $\mathbf{g}^I$ encodes $\left(g_{s(I),s(J)}\right)^\top$ for $J \in \mathcal{I}_{R_c}(I)$ and

$$\mathbf{h}^I = \sum_{J\in\mathcal{I}_{R_c}(I)} h_{s(I),s(J)}(\mathbf{R}_J - \mathbf{R}_I) = \sum_{J\in\mathcal{I}_{R_c}(I)} \mathbf{d}_J^I \left[g_{s(I),s(J)}\left((\mathbf{d}_J^I)_1\right)\right]^\top = \mathbf{d}^I\mathbf{g}^I. \tag{20}$$

We use a ResNet [31] architecture using dense layers to construct $\mathcal{F}_{s(I)}$, while the feature mapping $g_{s(I),s(J)}$ is a dense feed-forward neural network with a few layers.

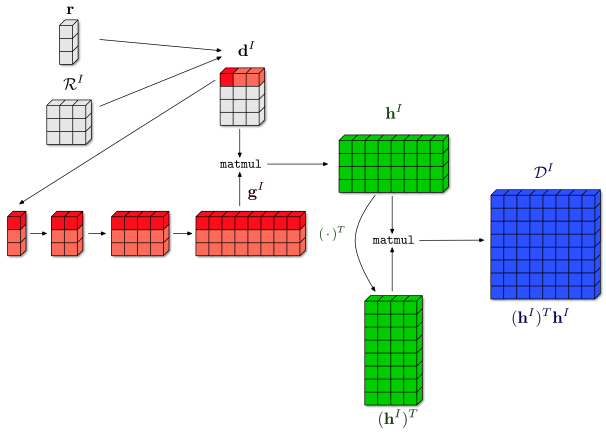


Figure 3: Schematic illustration of the computation of $\mathcal{D}^I$, where $n(I) = 3$, and there is only one atomic species (besides the electron represented in dark red).

4 Numerical examples

In this section we report the performance of Deep Density for one-dimensional model problems, as well as three dimensional real systems. The details of the setup as well as the choice of hyperparameters can be found in Appendix A and B, respectively. In all calculations the test error is measured in terms of the relative ℓ^1/ℓ^2 error, defined as:

$$\texttt{err}_{\ell^1} := \frac{\sum_i |\varrho(\mathbf{r}_i, \{\mathbf{R}_I\}) - \varrho_{NN}(\mathbf{r}_i, \{\mathbf{R}_I\})|}{\sum_i |\varrho(\mathbf{r}_i, \{\mathbf{R}_I\})|}, \tag{21}$$

$$\texttt{err}_{\ell^2} := \frac{\left(\sum_i \left[\varrho(\mathbf{r}_i, \{\mathbf{R}_I\}) - \varrho_{NN}(\mathbf{r}_i, \{\mathbf{R}_I\})\right]^2\right)^{\frac{1}{2}}}{\left(\sum_i \left[\varrho(\mathbf{r}_i, \{\mathbf{R}_I\})\right]^2\right)^{\frac{1}{2}}}. \tag{22}$$

Here the index i is taken over all the discretization points, ϱ is the electron density computed using Kohn-Sham solvers, and ϱ_{NN} is the approximation given by the neural network. In particular, the relative ℓ^1 error approximates the following quantity

$$\frac{\int |\varrho(\mathbf{r}, \{\mathbf{R}_I\}) - \varrho_{NN}(\mathbf{r}, \{\mathbf{R}_I\})| \, \mathrm{d}\mathbf{r}}{N_e} \tag{23}$$

which is the same error metric used by e.g. [32].

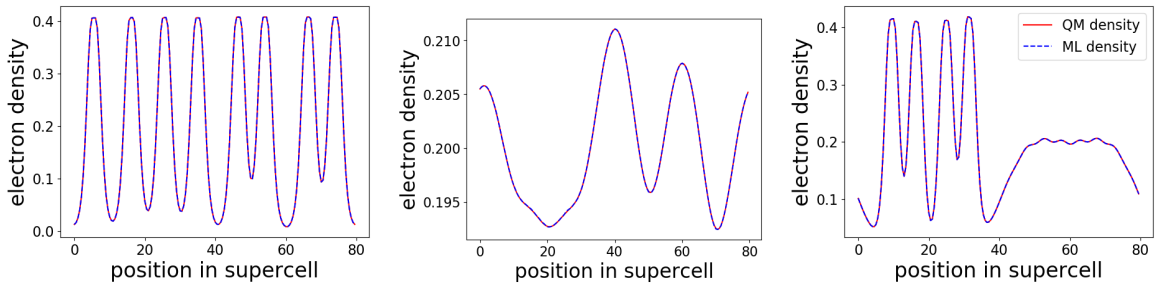
4.1 One dimensional systems

In this section we study three model systems in 1D with different characters: insulating, metallic, and mixed metallic-insulating systems. Previous study indicates that when the system is metallic or has mixed metallic-insulating characters, the self-consistent field iteration can be very difficult to converge (without a proper preconditioner) due to the small energy gaps and the associated charge sloshing behavior [33]. The details of the setup can be found in Appendix A.

We first consider a small supercell consisting of 8 atoms initially separated by 10 a.u. At the beginning of the *ab initio* molecular dynamics simulation, we perturb each of the 8 atoms by a uniform random number in $[-3, 3]$ a.u., and then let the systems evolve for 30000 time steps. In order to reduce the correlation among the snapshots and the amount of training time, we down-sample the trajectory for the first 8000 time steps by a factor 80, and we take the resulting first 100 snapshots as the training snapshots. The same procedure is applied to the validation snapshots for the next 400 time steps. We then use these 100 training snapshots and 5 validation snapshots to train the network.

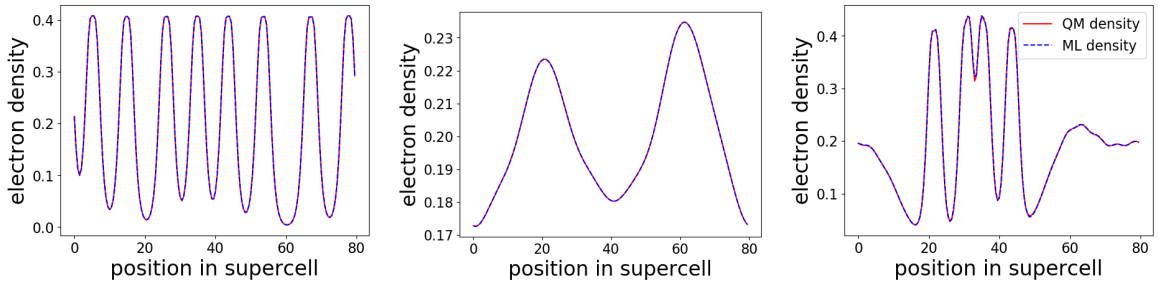
We test the trained model by comparing the predicted density and the density obtained from the KS-DFT calculation, using a snapshot which is part of the original training set at time step 2019 (before the down-sampling) in Fig. 4, as well as a snapshot that is far outside the training set at time step 29000 in Fig. 5. In

both cases, the the error of the electron density is very small, and is $0.01\% \sim 0.43\%$ measured by the relative ℓ^1 norm.



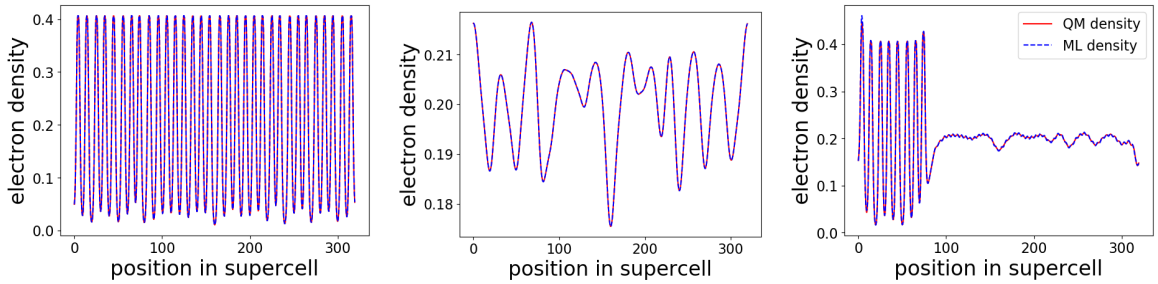
(a) the insulating system, $\texttt{err}_{\ell^1}$ = 6.27E-04 (b) the metallic system, $\texttt{err}_{\ell^1}$ = 1.21E-04 (c) the mixed metallic-insulating system, $\texttt{err}_{\ell^1}$ = 5.42E-04

Figure 4: Comparison of the electron density at time step 2019.



(a) the insulator system, $\texttt{err}_{\ell^1}$ = 8.12E-04 (b) the metallic system, $\texttt{err}_{\ell^1}$ = 1.15E-04 (c) the mixed metallic-insulating system, $\texttt{err}_{\ell^1}$ = 4.30E-03

Figure 5: Comparison of the electron density at time step 29000.



(a) the insulator system, $\texttt{err}_{\ell^1}$ = 6.86E-04; (b) the metallic system, $\texttt{err}_{\ell^1}$ = 1.49E-04; (c) the mixed metallic-insulating system, $\texttt{err}_{\ell^1}$ = 5.71E-03.

Figure 6: Transferability of the one-dimensional model, which is trained using for a system with 8 atoms and tested on a system with 32 atoms.

Our architecture constructs a local density ϱ^I for each atom as in Eq. (9). This enables us to use the trained model to predict the electron density of a larger system. We test the transferability of our model by loading parameters trained using the 8-atom systems into the model that predicts the electron density for the 32-atom systems. For the mixed metallic-insulating system, the first 8 atoms are insulator-like and the latter 24 atoms are metal-like. Our model achieves excellent transferability, and the error is $0.01\% \sim 0.57\%$, as shown in Fig. 6.

4.2 Three dimensional systems

For three-dimensional molecular and condensed matter systems, we present the following three test sets:

- organic molecules: a single ethane molecule (C_2H_6), a single isobutane molecule and a single n-butane molecule (C_4H_{10}).
- water: a set of systems composed of $32, 64, 128, 256$, and 512 water molecules in the liquid phase at T=300K.
- aluminum: a set systems composed of $32, 108$, and 256 aluminum atoms formed initially by $2 \times 2 \times 2$, $3 \times 3 \times 3$, and $4 \times 4 \times 4$ face-center cubic (fcc) unit cells, and at temperatures 300K, 600K, and 900K, respectively.

For organic molecules, the configurations are collected from the dataset provided in [34]. We take the first 101 (uncorrelated) snapshots of ethane, n-butane and isobutane from the dataset. For each molecule, we use 100 snapshots for training and one for testing.

For water and aluminum, the configurations are obtained using the DeePMD-kit package [35]. For each case, the atomic configurations are uniformly sampled from long molecular dynamics trajectories at different temperatures and different system sizes. For all simulations we perform NpT simulations at the standard pressure p=1 bar with a time step of 1 fs. The potential energy models used in the simulation are obtained with the DP-GEN scheme [24] using *ab initio* data. We take one snapshot in every 1000 time steps from the trajectory to reduce the correlation among configurations. The data sets are divided as follows: The training set was a randomly selected subset of 80 snapshots from the 100 snapshots for the smallest system. The test set was composed of the remaining 20 snapshots for the smallest systems in addition to the snapshots of the larger ones.

For all systems, we compute the electron density for each snapshot using PWDFT (which is based on planewaves and is an independent module of the DGDFT package [36]). We use the Perdew-Burke-Ernzerhof (PBE) exchange-correlation functional [37], and the SG15 Optimized Norm-Conserving Vanderbilt (ONCV) pseudopotentials [38, 39]. Other details of the setup of test systems and the training hyperparameters are given in Appendix B.

4.2.1 Small Molecules

The kinetic energy cut-off is set to 30 a.u. for both C_2H_6 and C_4H_{10}. The total number of grid points for each system is 1,906,624. For the feature networks in Eq. (18), we use a dense linear network with three layers, containing $\{5, 10, 20\}$ nodes respectively. For each fitting network we used a ResNet [40] with 3 dense layers containing 50 nodes per layer, where the skip connections are weighted by a trainable coefficient. The cutoff is the same for the 3 molecules $R_c = 4$Å. We trained the network for a few times with different random seeds and picked the one with the smallest generalization error.

Fig. 7 shows the slice with the largest error for a snapshot in the test set for both molecules. The relative test errors for the molecules are shown in Table 1. The relative ℓ^1 error in [32] for ethane and butane are 1.14% and 1.19%, respectively. Therefore our error is 6.1 and 3.8 times smaller, respectively. Finally Fig. 10 (Appendix B) summarizes the distribution of the prediction error for the three molecules. We observe that the distribution remains exceptionally close to the diagonal, thus indicating a very low error.

Molecule \error	$\texttt{err}_{\ell^2}$	$\texttt{err}_{\ell^1}$
ethane	0.172%	0.186%
isobutane	0.194%	0.222%
butane	0.289%	0.314%

Table 1: Error of the testing samples for different molecules.

4.2.2 H_2O

For water the kinetic energy cut-off is set to 40 a.u. Both the training and test systems consist of 32 water molecules. For the feature networks in Eq. (18), we use a dense linear network with four layers, each one containing $\{5, 10, 20, 40\}$ nodes respectively. The ResNet fitting network uses 5 dense layers and 50 nodes per layer, where the skip connections are weighted by a trainable coefficient. The cutoff radius is $R_c = 3.5$Å.

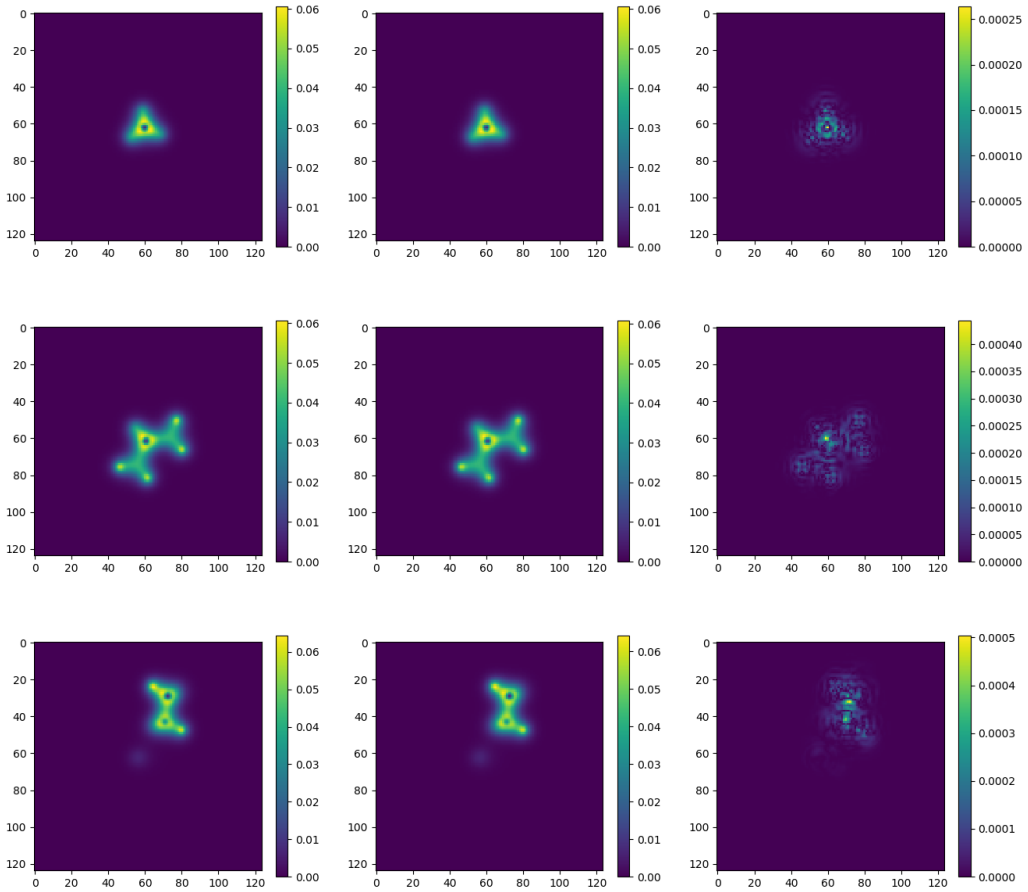


Figure 7: (left column) Slice of a snapshot of the electron density containing the largest point-wise error , (center column) slice of the density computed using the network, (right column) absolute error. Rows starting from the top : results for ethane, isobutane, and butane.

We trained the network a few times changing the random seed and we picked the one with the smallest generalization error. The error for a test snapshot with 32 atoms is showcased in Fig. 8, where we provide the slice containing the largest point-wise error.

$N_{\mathtt{mol}}(H_2O)\backslash$error	$\mathtt{err}_{\ell^2}$	$\mathtt{err}_{\ell^1}$
32	0.606%	1.080%
64	0.612%	1.021%
128	0.503%	0.891%
256	0.520%	0.903%
512	0.528%	0.921%

Table 2: Error of the testing samples for different number of water molecules.

Next we test the transferability of our model using systems with different sizes, which consists of 64, 128, 256 and 512 water molecules, respectively. The largest system has a total number of 20,213,648 grid points. The relative ℓ^2 and ℓ^1 errors are summarized in Table 2, where we can observe that inference error remains almost constant across different systems, which is around 0.5% for the ℓ^2 relative error (and 1.0% for the ℓ^1 relative error). Finally Fig. 10 (Appendix B) summarizes the distribution of the prediction error as we increase the system size. We can observe that the distribution remains very well concentrated within the diagonal thus indicating a very low error. Fig. 11 (Appendix B) shows the generalization error measured by the slice with the largest error for a snapshot in the test set. It demonstrates that the Deep Density network, learned using

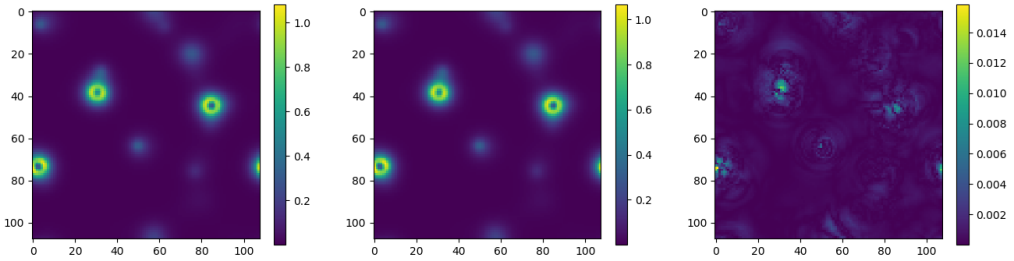


Figure 8: (left) slice of a snapshot of the electron density with 32 water molecules, (center) slice of the density computed using the network, (right) slice containing the largest point-wise absolute error for the snapshot.

a small sized system, has excellent transferability to large systems. The magnitude of the relative error agrees with that in Table 2. In this case, we observe that the error is mostly concentrated on high-gradient portions of ρ near the nuclei.

4.2.3 Aluminum

For the aluminum systems, we follow the same training pipeline as for the water case, and the kinetic energy cutoff is 20 a.u., and the largest system has total number of 1,906,624 grid points. The feature and fitting networks are chosen in a similar fashion to the water case. However, we used a large truncation radius $R_c = 6$Å. In this case, however, the initialization of constants in (11) was modified by using a larger truncation radius in order to start with a more uniform initial density .

The training stage was performed using a system with 32 atoms as explained in Appendix B.1. In Fig. 13 (Appendix B) we demonstrate that the trained model is able to efficiently recover the peaks concentrated at the center of each nuclei, which accounts mainly for the electron density associated to semi-core orbitals. Note that the ONCV pseudopotential treats all the 2s and 2p orbitals as semi-core electrons. As a result, the electron density in Al has sharp peaks and relatively large magnitudes. We test the transferability following the same procedure as for the water system. In Fig. 13 we compare the electron density provided by the network and those from KS-DFT calculations for configurations containing 108 and 256 aluminum atoms. For both cases we observe a relative ℓ^2 error below 2.5% (see Table 5). This is larger compared to that of the water system. From Fig. 13 and Table 5 we observe that the errors also grow with respect to the system size. This may be due to the quality of the training data generated by PWDFT, which only uses the Γ-point sampling of the Brillouin zone, and the system size is relatively small. To verify this, we train the network with just 4 snapshots of the $3 \times 3 \times 3$ configuration. The relative test ℓ^2 and ℓ^1 error can be improved to 0.5% and 1.2%, respectively, for a $3 \times 3 \times 3$ configuration as shown in Fig. 14. In addition, even though the absolute error of water and Al systems are comparable to each other, further inspection of Figs. 11 and 13 reveals some qualitative difference between the two systems: the errors from the Al systems appear to be much more spatially delocalized, and hence it is possible that the errors are mainly contributed by the valence electrons rather than the semi-core electrons. To verify this, we tested our method with another data set, which uses the same configurations, but with the density generated by the Vienna *ab initio* simulation package (VASP, version 5.4.4) [41, 42], which do not treat electrons at 2s and 2p orbitals as semi-core electrons. The results are reported in Appendix B.

5 Conclusion

Leveraging the success of the recently developed Deep Potential, we propose the Deep Density method to use machine learning to bypass the solution of the Kohn-Sham equations, and to obtain the self-consistent electron density in the context of *ab initio* molecular dynamics simulation. We demonstrate that the localization principle not only holds for insulating systems, but at least to some extent is also valid for metallic systems due to screening effects. Numerical results in one-dimensional systems and small molecular systems demonstrate that our construction can be very accurate, using a relatively small number of training samples. Our model can also be used to predict the electron density in the condensed phase, and can achieve excellent transferability for systems with up to 512 water molecules.

In 1D we have shown that this approach is able to efficiently compute the electron density for toy models emulating insulating, metallic and mixed system (to single precision). In 3D deep density is able to learn the electron density for realistic chemical systems. However, we also observe that the accuracy of our model deteriorates when applied to real 3D metallic systems such as aluminum. This may be caused by insufficient screening compared to toy models, but we also expect that our results may be further improved by employing different neural network architectures.

We envisage to accelerate the current algorithm. The complexity for the point-wise evaluation of the electron density is linear with respect to the systems size. However, a simple modification of the proposed approach can lead to a point-wise time evaluation that is independent of the systems size, thus producing a linear scaling algorithm. Another line of work is to improve the efficiency of training and prediction. In the current implementation, the descriptors are computed on CPUs, and this becomes a bottleneck when the electron density on millions of data points or more need to be evaluated. We expect that by employing a GPU based implementation and by computing the electron density at different grid points in an embarrassingly parallel fashion, the efficiency can be greatly improved. We expect that these improvements would make Deep Density to be a very useful tool for the analysis and prediction of electronic structures.

Acknowledgment

This work was partially supported by the Department of Energy under Grant No. DE-SC0017867 and the CAMERA program (L. L., L. Z.-N., J. Z.), as well as the Center Chemistry in Solution and at Interfaces (CSI) funded by the DOE Award DE-SC001934 (Y. C., L. Z.). We thank the Berkeley Research Computing (BRC) program at the University of California, Berkeley, the TIGRESS High Performance Computer Center at Princeton University, the National Energy Research Scientific Computing Center (NERSC), and the Google Cloud Platform (GCP) for the computational resources. We thank Roberto Car, Weinan E, Yuwei Fan, Jiequn Han, Yu-hang Tang, Han Wang, Chao Yang, Lexing Ying for valuable discussions at various stages of the project.

References

[1] W. Kohn and L. Sham, "Self-consistent equations including exchange and correlation effects," *Phys. Rev.*, vol. 140, pp. A1133–A1138, 1965.

[2] P. Hohenberg and W. Kohn, "Inhomogeneous electron gas," *Phys. Rev.*, vol. 136, pp. B864–B871, 1964.

[3] N. Mermin, "Thermal properties of the inhomogeneous electron gas," *Phys. Rev.*, vol. 137, p. A1441, 1965.

[4] S. Goedecker, "Linear scaling electronic structure methods," *Rev. Mod. Phys.*, vol. 71, pp. 1085–1123, 1999.

[5] D. R. Bowler and T. Miyazaki, "$O(N)$ methods in electronic structure calculations," *Rep. Prog. Phys.*, vol. 75, p. 036503, 2012.

[6] R. Martin, *Electronic Structure: Basic Theory and Practical Methods.* Cambridge Univ. Pr., 2008.

[7] S. R. White, "Density matrix formulation for quantum renormalization groups," *Phys. Rev. Lett.*, vol. 69, no. 19, p. 2863, 1992.

[8] R. J. Bartlett and M. Musiał, "Coupled-cluster theory in quantum chemistry," *Rev. Mod. Phys.*, vol. 79, no. 1, p. 291, 2007.

[9] F. Brockherde, L. Vogt, L. Li, M. E. Tuckerman, K. Burke, and K.-R. Müller, "Bypassing the kohn-sham equations with machine learning," *Nature Commun.*, vol. 8, no. 1, p. 872, 2017.

[10] A. Grisafi, A. Fabrizio, B. Meyer, D. M. Wilkins, C. Corminboeuf, and M. Ceriotti, "Transferable machine-learning model of the electron density," *ACS Cent. Sci.*, vol. 5, no. 1, pp. 57–64, 2019.

[11] K. Ryczko, D. A. Strubbe, and I. Tamblyn, "Deep learning and density-functional theory," *Physical Review A*, vol. 100, no. 2, p. 022512, 2019.

[12] M. Bogojeski, F. Brockherde, L. Vogt-Maranto, L. Li, M. E. Tuckerman, K. Burke, and K.-R. Müller, "Efficient prediction of 3d electron densities using machine learning," *arXiv:1811.06255*, 2018.

[13] A. Fabrizio, B. Meyer, M. Ceriotti, C. Corminboeuf, *et al.*, "Electron density learning of non-covalent systems," *Chem Sci.*, 2019.

[14] J. Behler and M. Parrinello, "Generalized neural-network representation of high-dimensional potential-energy surfaces," *Phys. Rev. Lett.*, vol. 98, no. 14, p. 146401, 2007.

[15] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, "Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons," *Physical Review Letters*, vol. 104, no. 13, p. 136403, 2010.

[16] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. VonLilienfeld, "Fast and accurate modeling of molecular atomization energies with machine learning," *Phys. Rev. Lett.*, vol. 108, no. 5, p. 058301, 2012.

[17] G. Montavon, M. Rupp, V. Gobre, A. Vazquez-Mayagoitia, K. Hansen, A. Tkatchenko, K.-R. Müller, and O. A. Von Lilienfeld, "Machine learning of molecular electronic properties in chemical compound space," *New Journal of Physics*, vol. 15, no. 9, p. 095003, 2013.

[18] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller, "Machine learning of accurate energy-conserving molecular force fields," *Science Advances*, vol. 3, no. 5, p. e1603015, 2017.

[19] K. Schütt, P.-J. Kindermans, H. E. S. Felix, S. Chmiela, A. Tkatchenko, and K.-R. Müller, "Schnet: A continuous-filter convolutional neural network for modeling quantum interactions," in *Advances in Neural Information Processing Systems*, pp. 992–1002, 2017.

[20] J. S. Smith, O. Isayev, and A. E. Roitberg, "ANI-1: an extensible neural network potential with dft accuracy at force field computational cost," *Chemical Science*, vol. 8, no. 4, pp. 3192–3203, 2017.

[21] J. Han, L. Zhang, R. Car, and W. E, "Deep potential: a general representation of a many-body potential energy surface," *Comms. Comp. Phys.*, vol. 23, no. 3, pp. 629–639, 2018.

[22] L. Zhang, J. Han, H. Wang, R. Car, and E. W., "Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics," *Phys. Rev. Lett.*, vol. 120, p. 143001, 2018.

[23] L. Zhang, J. Han, H. Wang, W. Saidi, R. Car, and W. E, "End-to-end symmetry preserving inter-atomic potential energy model for finite and extended systems," in *Advances in Neural Information Processing Systems 31*, pp. 4436–4446, 2018.

[24] L. Zhang, D.-Y. Lin, H. Wang, R. Car, and W. E, "Active learning of uniformly accurate interatomic potentials for materials simulation," *Phys. Rev. Materials*, vol. 3, no. 2, p. 023804, 2019.

[25] L. Lin, J. Lu, and L. Ying, "Numerical methods for kohn–sham density functional theory," *Acta Numer.*, vol. 28, pp. 405–539, 2019.

[26] W. Kohn, "Density functional and density matrix method scaling linearly with the number of atoms," *Phys. Rev. Lett.*, vol. 76, pp. 3168–3171, 1996.

[27] E. Prodan and W. Kohn, "Nearsightedness of electronic matter," *Proc. Natl. Acad. Sci.*, vol. 102, pp. 11635–11638, 2005.

[28] A. G. Eguiluz, "Self-consistent static-density-response function of a metal surface in density-functional theory," *Phys. Rev. B*, vol. 31, no. 6, p. 3303, 1985.

[29] X. Gonze and C. Lee, "Dynamical matrices, Born effective charges, dielectric permittivity tensors, and interatomic force constants from density-functional perturbation theory," *Phys. Rev. B*, vol. 55, p. 10355, 1997.

[30] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, "Deep sets," pp. 3391–3401, 2017.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[32] A. Grisafi, A. Fabrizio, B. Meyer, D. M. Wilkins, C. Corminboeuf, and M. Ceriotti, "Transferable machine-learning model of the electron density," *ACS Central Science*, vol. 5, no. 1, pp. 57–64, 2018.

[33] L. Lin and C. Yang, "Elliptic preconditioner for accelerating self consistent field iteration in Kohn-Sham density functional theory," *SIAM J. Sci. Comp.*, vol. 35, pp. S277–S298, 2013.

[34] L. Cheng, M. Welborn, A. S. Christensen, and T. F. Miller, "Thermalized (350k) qm7b, gdb-13, water, and short alkane quantum chemistry dataset including mob-ml features," 2019.

[35] H. Wang, L. Zhang, J. Han, and E. W., "DeePMD-kit: A deep learning package for many-body potential energy representation and molecular dynamics," *Computer Physics Communications*, vol. 228, pp. 178 – 184, 2018.

[36] W. Hu, L. Lin, and C. Yang, "DGDFT: A massively parallel method for large scale density functional theory calculations," *J. Chem. Phys.*, vol. 143, p. 124110, 2015.

[37] J. P. Perdew, K. Burke, and M. Ernzerhof, "Generalized gradient approximation made simple," *Phys. Rev. Lett.*, vol. 77, pp. 3865–3868, 1996.

[38] D. R. Hamann, "Optimized norm-conserving Vanderbilt pseudopotentials," *Phys. Rev. B*, vol. 88, p. 085117, 2013.

[39] M. Schlipf and F. Gygi, "Optimization algorithm for the generation of ONCV pseudopotentials," *Comput. Phys. Commun.*, vol. 196, pp. 36–44, 2015.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[41] G. Kresse and J. Furthmüller, "Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set," *Phys. Rev. B*, vol. 54, pp. 11169–11186, 1996.

[42] G. Kresse and J. Furthmüller, "Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set," *Phys. Rev. B*, vol. 54, no. 16, p. 11169, 1996.

[43] D. G. Anderson, "Iterative procedures for nonlinear integral equations," *J. ACM*, vol. 12, pp. 547–560, Oct. 1965.

[44] D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications*. Academic Press, 2002.

[45] D. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, may 2015.

[46] P. E. Blöchl, "Projector augmented-wave method," *Phys. Rev. B*, vol. 50, p. 17953, 1994.

[47] H. J. Monkhorst and J. D. Pack, "Special points for Brillouin-zone integrations," *Phys. Rev. B*, vol. 13, no. 12, p. 5188, 1976.

A Numerical results for 1D systems

We use a 1D reduced Hartree-Fock model similar to the one presented in [33]. This simplified model depends nonlinearly on the electron density ρ only through the Hartree interaction, and does not include the exchange-correlation functional. However, it can still qualitatively reproduce certain phenomena in 3D, such as the difference between insulating and metallic systems, and the screening effect shown in Section 2. The Hamiltonian in our 1D model is given by (we still use the notations $\mathbf{r}$ and $\mathbf{R}$ though this is a one-dimensional system)

$$H[\rho, \{\mathbf{R}_I\}_{I=1}^{N_a}] = -\frac{1}{2}\frac{d^2}{d\mathbf{r}^2} + V_{\text{hxc}}[\mathbf{r}; \rho] + V_{\text{ion}}[\mathbf{r}; \{\mathbf{R}_I\}_{I=1}^{N_a}], \tag{24}$$

$$= -\frac{1}{2}\frac{d^2}{d\mathbf{r}^2} + \int K(\mathbf{r}, \mathbf{r}')(\rho(\mathbf{r}') + m(\mathbf{r}'; \{\mathbf{R}_I\}_{I=1}^{N_a}))\, \mathrm{d}\mathbf{r}'. \tag{25}$$

Here we use a pseudopotential to represent the electron-ion interaction, and the total pseudo charge density is given by

$$m(\mathbf{r}; \{\mathbf{R}_I\}_{I=1}^{N_a}) = \sum_{I=1}^{N_a} -\frac{Z_I}{\sqrt{2\pi\sigma_I^2}} \exp\left(-\frac{1}{2\sigma_I^2}(\mathbf{r} - \mathbf{R}_I)^2\right). \tag{26}$$

Here Z_I represents the charge of I-th nucleus, and σ_I represents the width of the nuclei potential within the pseudopotential theory. σ_I is tuned so that I-th nucleus can qualitative behave as a metal or as an insulator. Since the standard Coulomb interaction diverges in 1D, we employ the Yukawa kernel

$$K(\mathbf{r}, \mathbf{r}') = \frac{2\pi}{\kappa\epsilon_0} e^{\kappa|\mathbf{r}-\mathbf{r}'|}, \tag{27}$$

where the parameters $\kappa = 0.01$ and $\epsilon_0 = 10.0$ are fixed constants throughout the experiments. Our units here are arbitrary, but will be referred to as the atomic unit (a.u.) for simplicity.

The Kohn-Sham equations are solved using the standard self-consistent field iteration [6] method. In particular, we use Anderson mixing [43] of the potential with mix dimension 10.

We study three types of systems: the insulating system, the metallic system, and the mixed metallic-insulating system. Fig. 9 displays the occupied eigenvalues and the first ten unoccupied eigenvalues for all the three

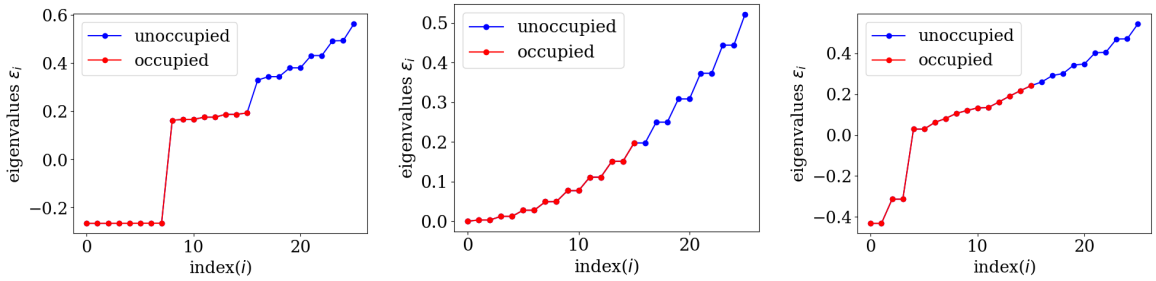


Figure 9: Eigenvalues for (a) the insulating system, $\sigma_I = 1.0$ for all I; (b) the metallic system, $\sigma = 6.0$ for all I; (c) the mixed metallic-insulating system, $\sigma_I = 1.0$ for $1 \leq I \leq 4$ and $\sigma_I = 6.0$ for $5 \leq I \leq 8$.

N_{near}	1	2	3	4	5	6	7
Insulator	1.91E-08	4.02E-08	2.26E-08	5.02E-08	1.38E-07	1.39E-07	1.13E-07
Metal	9.11E-10	1.08E-09	1.04E-09	1.10E-09	3.00E-09	7.65E-09	3.53E-09

Table 3: Validation error (MSE) for single type system with different values of N_{near}

systems. In particular, when the system is metallic or has a mixed metallic-insulating character, the self-consistent field iteration can be very difficult to converge due to the small energy gaps and charge sloshing behavior [33].

We consider a periodic system that includes 8 atoms in the unit cell, with 2 electrons per site, i.e. $Z_I = 2$ for $I = 1, ..., 8$. The atoms are 10 a.u. apart, located at $5, 15, 25, \cdots, 75$. The size of the supercell in this case is thus 80 a.u. As mentioned above, by adjusting σ_I we can obtain different qualitative behaviors. On the one hand, when $\sigma_I = 1.0$, the model qualitatively behaves as an insulating system with an energy gap of 0.136. On the other hand, if $\sigma_I = 6.0$, then the model qualitatively behaves as a metallic system and its energy gap is 5.5×10^{-8} (i.e. the system is gapless). To test the ability of the proposed architecture to deal with interactions between atoms of different species, we also introduce a mixed metallic-insulating system, which is obtained by setting $\sigma_I = 1.0$ for $I = 1, 2, 3, 4$, and $\sigma_I = 6.0$ for $I = 5, 6, 7, 8$. The energy gap in this case equals to 0.018. Fig. 9 displays the occupied eigenvalues and the first ten unoccupied eigenvalues for all the three systems.

For the 1D problem, our implementation is purely based on `python` and `tensorflow`. Furthermore, 1D model does not involve the rotational degrees of freedom. This allows us to simplify the network structure in Section 3 as below.

For simplicity of implementation, we introduce a parameter N_{near} instead of cutoff R_c, so the index set $\mathcal{I}_{N_{\text{near}}}(I)$ is decided by choosing the indices of the nearest N_{near} atoms. The model is constructed using the same ansatz as Eq. (11) where $\mathcal{N}^I$ and $\mathcal{E}^I$ are neural networks. However, the construction of the input to these networks, which are the descriptors $\mathcal{D}^I(\mathbf{r}, \mathcal{R}^I)$ defined in Section 3 as Eq. (15), is much simpler. To define the descriptors, we start with $\mathbf{d}_J^I = [R_{JI}, \frac{1}{R_{JI}}(\mathbf{R}_J - \mathbf{R}_I)]$ for $J \in \mathcal{I}_{N_{\text{near}}}(I)$ (distance information and direction information). Since we follow the form in Eq. (17) with $\mathbf{R}_J - \mathbf{R}_I$ reduced to one dimension, each $\mathbf{d}_J^I$ is in $\mathbb{R}^2$. The electron information $\mathbf{d}_0^I \in \mathbb{R}^2$ is fed to the descriptor directly, whereas the atom information $\mathbf{d}_J^I, J \neq 0$, is passed to the function $g_{s(I),s(J)}$ before being fed to the descriptor. For the insulating system and the metallic system, we only have one such function g, whereas for the mixed metallic-insulating system, we have four $g_{s(I),s(J)}$ networks because each of $s(I), s(J)$ can be one of the two types. Given that rotation symmetry in 1D is trivial, the descriptor $\mathcal{D}^I$ is formed simply by concatenating the electron information, $\mathbf{d}_0^I$, and atom information, $g_{s(I),s(J)}(\mathbf{d}_J^I)$. The output of g is of M dimension and there are N_{near} number of nearby atoms, so descriptors are $\mathcal{D}^I \in \mathbb{R}^{2+MN_{\text{near}}}$.

To treat the mixed metal-insulator system with two types of atoms, we implement a control flow so that at run time, the model knows which $g_{s(I),s(J)}$ to apply based on species of atom I and J. For simplicity we incorporate the information of the species as follows. Let range of $s(J)$ be $\{1, 2\}$. We encode the two atom types as vectors $\mathbf{v}_1 = [1, 0]^T, \mathbf{v}_2 = [0, 1]^T$. For a fixed center atom I, and adjacent atom J, we pass $\mathbf{d}_J^I$ to

N_{sample}	100	200	400
Two-atom-type	1.87E-07	6.94E-08	4.40E-08

Table 4: Validation error (MSE) for two-atom-type system with increasing training samples

$g_{s(I),s}$ for both $s = 1, 2$ and then calculate the output

$$g_{s(I),s(J)}(\mathbf{d}_J^I) = \left(\mathbf{v}_{s(J)}^T \mathbf{v}_1\right) g_{s(I),1}(\mathbf{d}_J^I) + \left(\mathbf{v}_{s(J)}^T \mathbf{v}_2\right) g_{s(I),2}(\mathbf{d}_J^I).$$

The training and test data sets are generated through molecular dynamics simulations. We use the Verlet algorithm [44] for the time propagation, where the forces are computed using the Hellmann-Feynman formula. At each time step we store the atomic configuration, $\{\mathbf{R}_I\}_{I=1}^8$, and the corresponding self-converged electron density, ρ, generated from the KS-DFT computation. For all three systems, we use the first 8000 snapshots for training and the next 400 snapshots for validation. In order to reduce the correlation of the shots and the amount of training time, we down-sample the training snapshots by a factor 80, i.e., we take 100 evenly time-spaced snapshots. The same procedure is applied to the validation snapshots. We then use these 100 training snapshots and 5 validation snapshots to train the network. For the mixed metallic-insulating system, the number of trainable parameters increases because of the four $g_{s(I),s(J)}$ networks, so we reduce the down-sampling factor to have more training snapshots (e.g. down sample the first 8000 snapshots by a factor of 20 to obtain 400 training snapshots). We also find that if we only use 100 training snapshots, the relative ℓ^1 error can increase and be higher than 1%. This indicates that the mixed insulating-metallic system is indeed more difficult and requires a larger number of training samples.

The training is performed using standard Adam optimizer [45] and a mean squared error loss. Given the simplicity and small scale of the problem we visit all the points at each snapshot, in contrast with the 3D training that will require importance sampling for efficiency consideration. The network was trained for 400 epochs, the model with lowest validation loss was saved. For each hyperparameter setting (Fixed N_{layer}, N_{nodes}, N_{near}), we run 5 experiments and report the one with lowest validation error. The validation errors are measured using mean squared error (MSE), namely

$$\frac{1}{N_{\text{validation}}} \sum_{j=1}^{N_{\text{validation}}} \left(\varrho(\mathbf{r}_j, \{R_I\}) - \varrho_{NN}(\mathbf{r}_j, \{R_I\})\right)^2. \tag{28}$$

In Table 3, we observe that the validation loss reaches well below 1E-06. Another observation is that the network is relatively insensitive to the hyperparameter N_{near} here, even when the system is gapless. Thus we fix $N_{\text{near}} = 2$ for the mixed metallic-insulating system model for simplicity.

In Table 4, N_{sample} is the number of snapshots in training, so the real amount of training data is the number of snapshots multiplied by the number of grid points for the 1D electron density. The validation loss reaches below 1E-07 once we increase the number of snapshots to 200.

B Numerical results for 3D systems

B.1 Simulation parameters

The parameters in the ansatz are initialized after a precomputation step that depends on each setup. This precomputation involves the following steps: selecting one atom for each species, sampling the density within a small radius of that atom, and computing the parameters $A_{s(I)}, B_{s(I)}, C_{s(I)}$, and $D_{s(I)}$ that best fits the sampled density, without the neural networks, using standard quasi-Newton optimization methods. The purpose of this precomputation step is to help the optimization find a suitable minimum. The weights in the Neural Network are initialized using a normalized Gaussian distribution. The objective function is the mean squared loss.

For the training stage we use the Nadam optimizer [45] with an exponential scheduling, in which for every 20000 iterations we decrease the learning rate by a factor 0.95, and we initialize the learning rate as 0.003. At each iteration we draw $n_{\mathbf{s}} = 64$ samples from the snapshots. The training is scheduled as follows: we train the network for a million iterations using only 5 snapshots, then we train the network for another million iterations using 20 snapshots and finally we train the network for two million iteration using 80 snapshots. The remaining 20 snapshots were used for testing throughout the training.

At each iteration n_s samples are extracted from the training data. Each sample represents a pixel of the images shown in Fig. 11. For the water system we have 80 training snapshots and each snapshot contain around 1.24E6 pixels, totalling roughly 100 million data points. For the aluminum system we have the same amount of training snapshots but each snapshot contain around 2.564E5 pixels, totalling roughly 21 million data points. For the small organic molecules we have roughly 125 million data point for each system. Thus we need to visit them judiciously in order to be efficient. Given that different systems may have very different localization properties we use a sampling strategy based on the norm of the density at each pixel. In particular, during the training stage the samples are drawn following the distribution $|\rho(\mathbf{r})|^\alpha$, where the value of α is tuned for each setup, in order to avoid visiting small values of the densities too often, thus improving the efficiency of training. In particular we used 1/2 for the small organic molecules, 6/5 for the water systems, and 1 for the aluminum systems.

We estimate the test error by comparing the result given by the network against the test snapshots in the small system, and we estimate the transferability of the algorithm by comparing the electron density generated by the trained model for the larger systems versus the one computed using PWDFT.

The computation of the electron density were performed at the NERSC cluster Cori, which is comprised of 2,388 dual socket nodes with 32 cores and 256 GB of RAM, whereas, the training of the models and inference steps were performed in a 16 core machine used with 64 GB of RAM and a Tesla V100 GPU with 16GB memory.

B.2 Additional plots of the organic molecules and water systems

In addition to the figures in the main text, we include Fig. 10, which depicts the performance of Deep Density for the different small molecules and different water systems. Fig. 10 represent a scatter plot in logarithmic scale of the value of the predicted density and the density computed with PWDFT for the same configuration and sampling points. We can observe that for the former the error are almost negligible, and the later the errors are higher but are still very small.

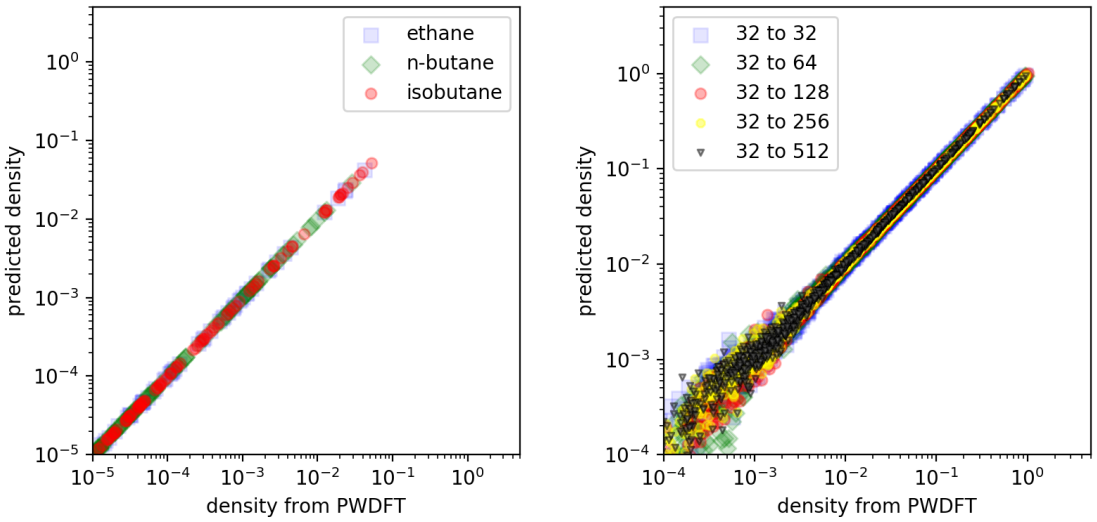


Figure 10: (left) scatter plot of the predicted and test densities for the different small organic molecules, (right) scatter plot of the predicted and test densities for different water systems.

B.3 Additional results for aluminum

Our VASP calculation also used the PBE exchange-correlation functional, but with the projector augmented wave method (PAW) [46] to handle the core electrons. In particular, the 2s and 2p orbitals are treated as core electrons, and hence there are only 3 valence electrons per atom. The kinetic energy cutoff for the plane wave expansion is set to 600 eV, and the Brillouin zone is sampled with the Monkhorst-Pack mesh [47] at the spacing $h_k = 0.08$ Å^{-1}. The order 1 Methfessel-Paxton smearing method with $\sigma = 2900$ K is adopted. The

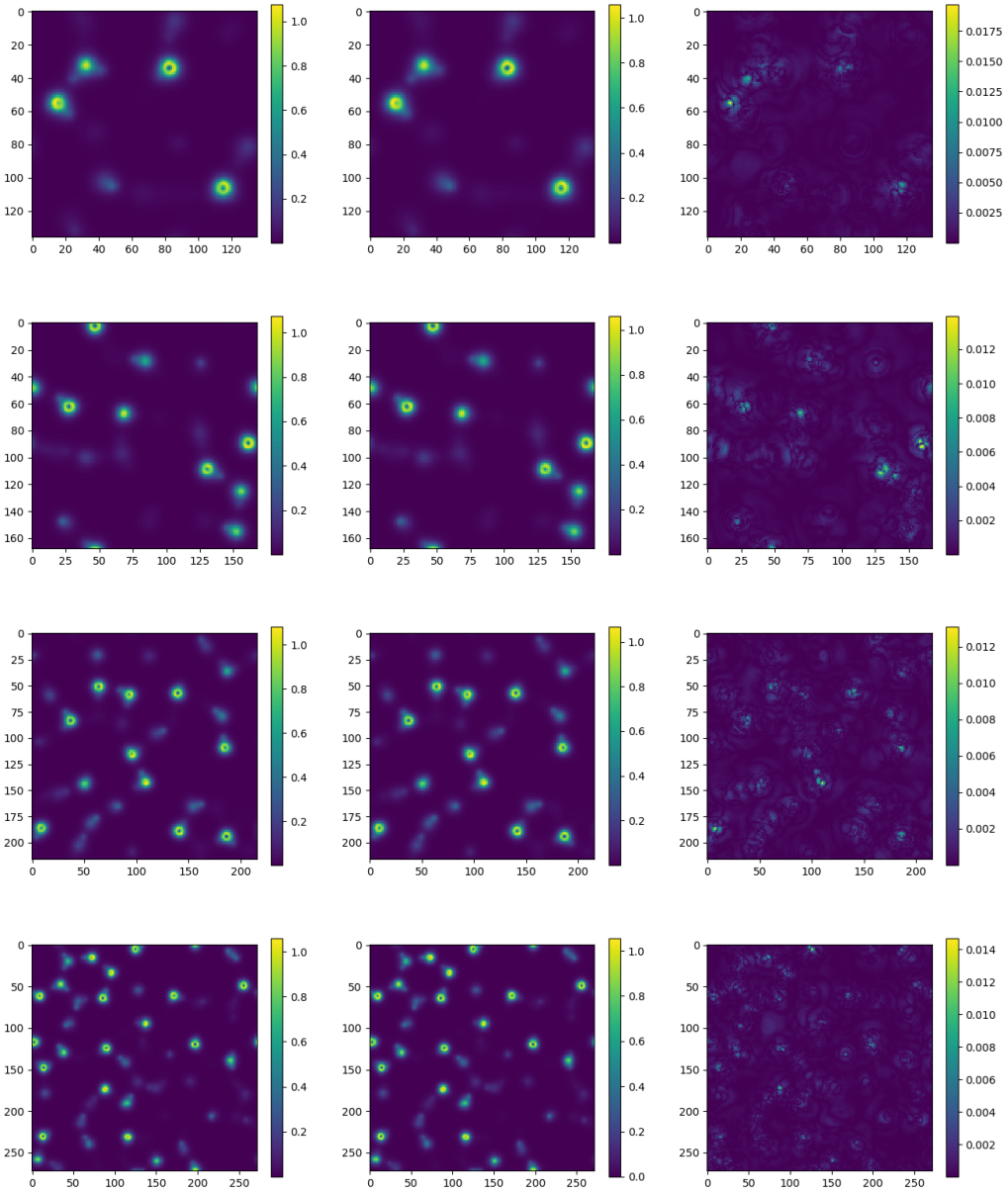


Figure 11: (left column) slice of a snapshot of the electron density, (center column) slice of the density computed using the network, (right column) slice of the absolute error with the highest point-wise absolute error. rows starting from the top : results for the system containing 64, 128, 256 and 512 water molecules

self-consistent field (SCF) iteration stops when the total energy and band structure energy differences between two consecutive steps are smaller than 10^{-6} eV. In this case the density contains only the contribution of valence electrons.

We used the same training pipeline as before. We perform training using a number of snapshots for a system containing $2 \times 2 \times 2$ unit cells, and test the network for a number of systems with $2 \times 2 \times 2$, $3 \times 3 \times 3$ and $4 \times 4 \times 4$ unit cells, respectively. The scatter plot in Fig. 12 suggests that the test error for the aluminum system is indeed much larger. Fig. 15 shows the test error using the density generated with VASP. The error is largely delocalized, which confirms the previous study with PWDFT that most error originates from valence electrons. In addition, from Table 5, we observe that the generalization error still grows, albeit slightly slower, with respect to the system size, thanks to the refined Brillouin zone sampling when generating the training data set.

N_a (Al)\ error	$\texttt{err}_{\ell^2}$ PWDFT	$\texttt{err}_{\ell^1}$ PWDFT	$\texttt{err}_{\ell^2}$ VASP	$\texttt{err}_{\ell^1}$ VASP
32	0.504%	1.400%	2.614%	2.015%
108	1.512%	3.937%	3.040%	2.529%
256	2.244%	5.801%	4.847%	4.515%

Table 5: Error of the testing samples for different number of atoms for Al. The data are generated using PWDFT (with semicore electrons) and VASP (without semicore electrons) respectively.

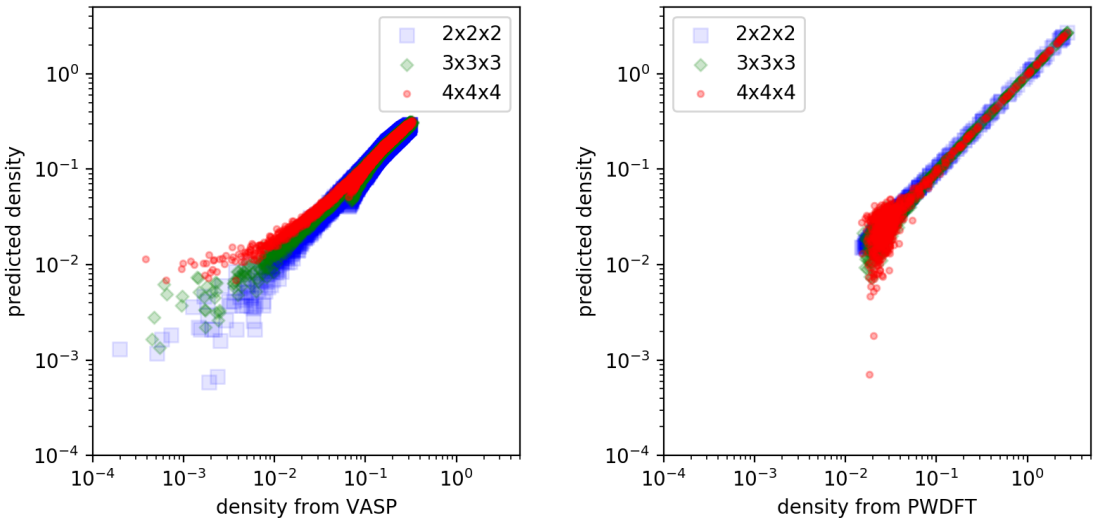


Figure 12: (left) scatter plot of the predicted and test densities generated by VASP for the different aluminum systems, (right) scatter plot of the predicted and test densities generated by PWDFT for the different aluminum systems. The magnitudes of the density from PWDFT are higher due to the inclusion of semi-core electrons.

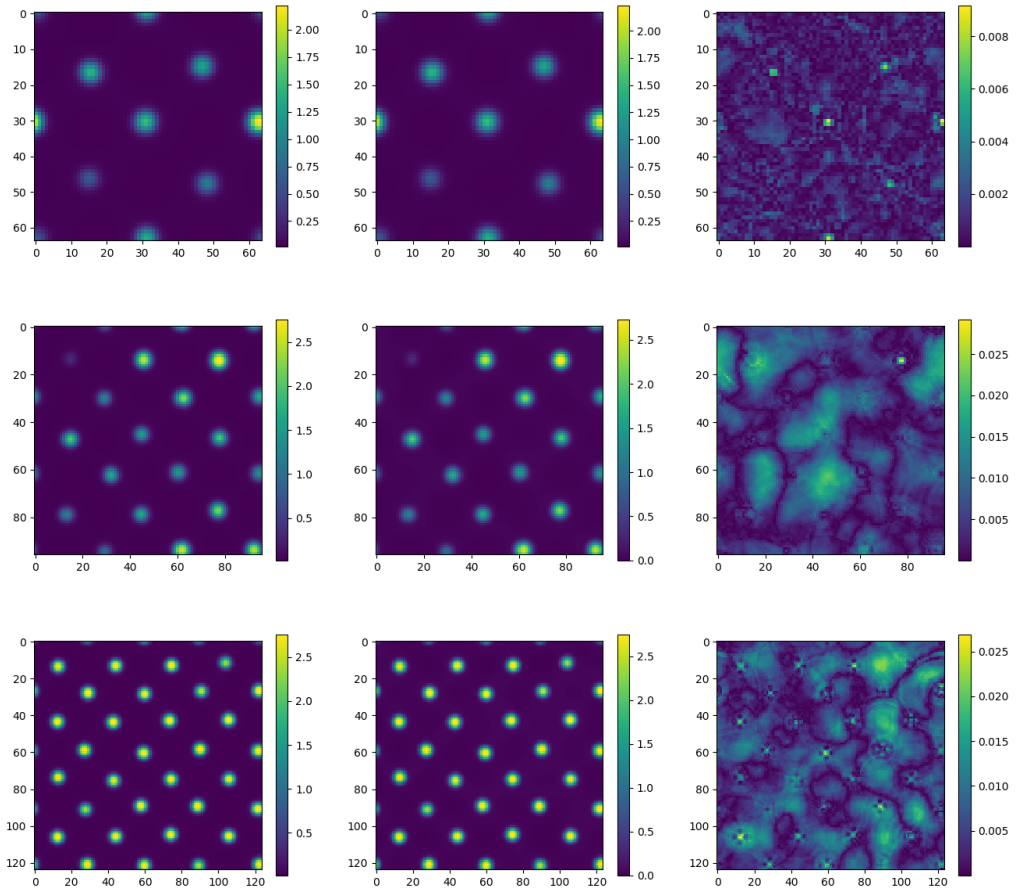


Figure 13: (left column) slice of a snapshot of the electron density, (center column) slice of the density computed using the network, (right column) slice of the absolute error with the highest point-wise absolute error. Rows starting from the top : results for the system containing 32, 108, and 256 aluminum atoms following $2 \times 2 \times 2$, $3 \times 3 \times 3$, and $4 \times 4 \times 4$ configurations respectively. The calculations are performed using PWDFT.

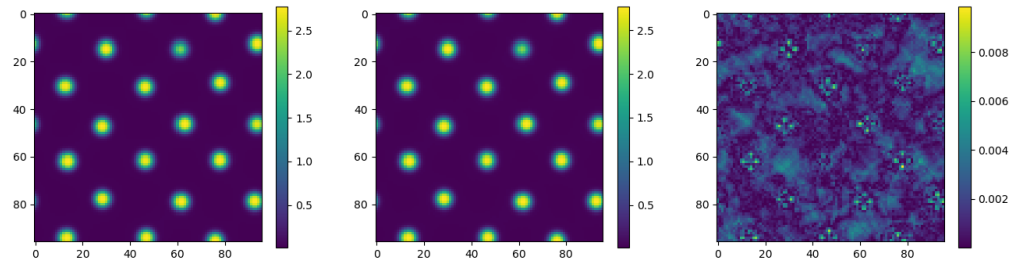


Figure 14: (left) slice of the snapshot produced by computing the electron density of 108 aluminum atoms in a $3 \times 3 \times 3$ configuration, (center) slice of the density computed using the network which was re-trained using 4 snapshots of the $3 \times 3 \times 3$ configuration, (right) slice of the absolute error with the highest point-wise absolute error. The calculations are performed using PWDFT.

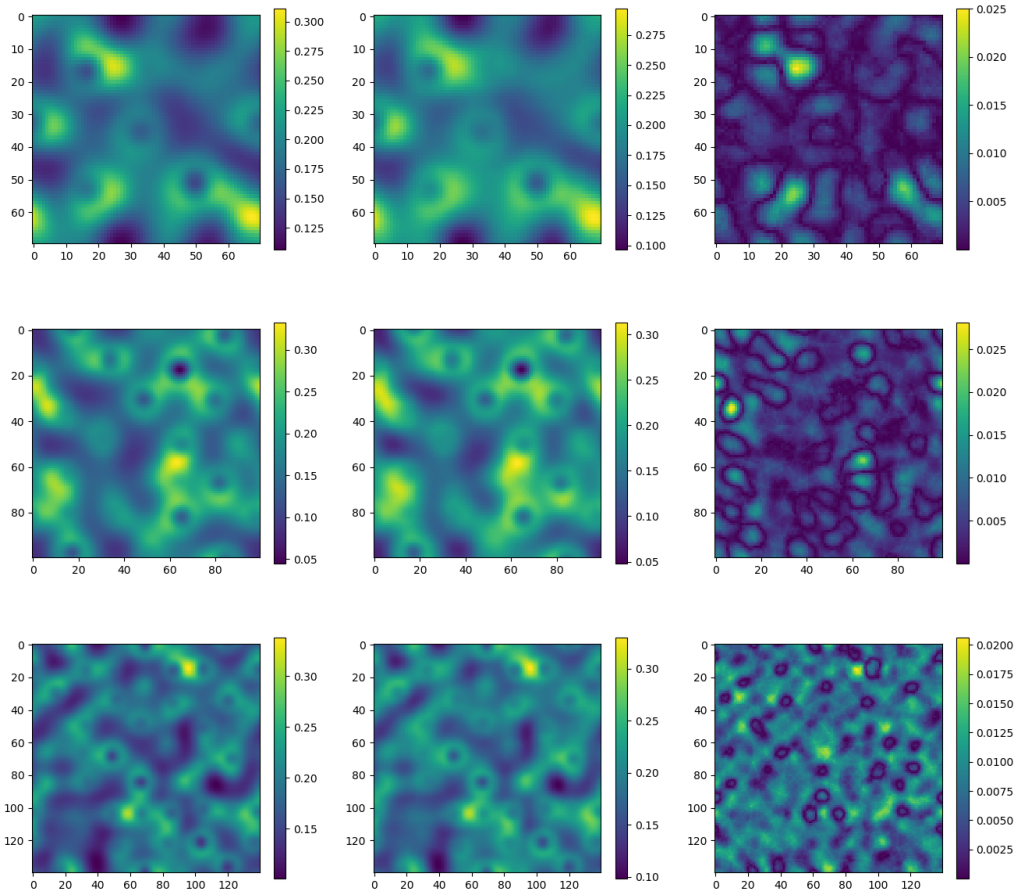


Figure 15: (left column) slice of a snapshot of the electron density from VASP, (center column) slice of the density computed using the network, (right column) slice of absolute error containing the largest point-wise error. Rows starting from the top : results for the system containing 32, 108, and 256 aluminum atoms following $2 \times 2 \times 2$, $3 \times 3 \times 3$, and $4 \times 4 \times 4$ configurations respectively.