Using machine learning to gain insight into quantum mechanics

Leo Arnstein
School of Physics, Engineering and Technology
University of York
Heslington, York, UK

May 12, 2025

Title: 9 words, abstract: 129 words, main text: 4993 words

Abstract

Density functional theory and potential functional theory rely on approximations to functionals to compute electronic properties of quantum systems. As theoretical treatment proves difficult, many are turning to machine learning techniques such as neural networks to derive empirical approximations. In this study, neural networks are trained on exact solutions to the time-independent Schrödinger equation to learn the mappings from both the electron density $n(x)$ and external potential $V(x)$ to kinetic energy T, total energy E, electron affinity A and fundamental gap E_g of 1D atoms. The aim was to gain physical insight into these mappings and assess the applicability of machine learning. All functionals are modelled accurately ($\leq 5.3\pm\ 4\,\mathrm{mHa}$ error), except E[n] and A[n], which exhibit larger errors ($38\pm\ 2\,\mathrm{mHa}$ and $36\pm 2\,\mathrm{mHa}$). This is attributed to their gauge-dependence, despite using a fixed gauge, highlighting a limitation of using n(x) to predict such properties.

1 Introduction

Many physical systems such as atoms, molecules and solids can be modelled as N electrons subject to a stationary external potential. Electronic structure calculations, which determine their electronic properties, play a central role in materials science, condensed matter physics and quantum chemistry [1]. They accelerate the design of novel materials and enable simulation of extreme conditions that are inaccessible in a laboratory setting [1]. However, there are few closed-form solutions to the Schrödinger equation for $N > 1$ and the computational cost of solving it numerically scales exponentially with N [2] as the many-body wave function depends on the position of each particle in the system. To tackle many-body problems requires more efficient methods such as density functional theory (DFT), which replaces the wave function as the basic variable with the electron density - a function of only three spatial coordinates regardless of the number of electrons, making it less computationally demanding. This is made possible by the first Hohenberg-Kohn (H-K) theorem, which establishes that the ground-state electron density determines the external potential up to an additive constant [3]. The ambiguity regarding the additive constant is removed by using a fixed gauge, i.e. consistently setting all potentials to a reference value at some point in space. This guarantees a 1-1 mapping from the electron density to the Hamiltonian (via the external potential) and hence all properties of the system. In other words, any property can in principle be expressed as a functional of the ground-state electron density - this is the origin of the name 'density functional theory'. The caveat is that the exact functional forms are unknown for most properties [4]; approximations are required, leading to inexact results. In the most widely used formalism, known as Kohn-Sham (KS) DFT, the density functional of an energy component termed exchange-correlation (E_{xc}) is approximated. This term accounts for quantum many-body effects, the complexity of which leads many researchers to believe that an explicit functional form cannot be found [5]. Despite significant progress in deriving useful approximations to $E_{xc}[n]$ from physical principles, common approximations suffer from limitations, such as failing to represent the derivative discontinuity that occurs at integer particle numbers [6]. This leads to systematic error in calculations of the fundamental gap, a property with important applications in opto-electronics and photovoltaics [7]. A parallel approach to DFT is potential functional theory (PFT) [8], in which the external potential plays the same role as the electron density in DFT. Although PFT could offer similar advantages to DFT, it is a less developed theory, so high-quality approximations are scarce [8]. In both of these fields, there is an ongoing search for better approximations to increase the accuracy and scope of electronic structure calculations.

Machine learning (ML) has emerged as a promising tool to aid in this search [9]. ML encompasses a range of computational methods used to automatically extract patterns from data. Using supervised learning, ML models learn input-output relationships via known pairs referred to as the training data. This allows them to predict the output for unseen inputs, acting as 'ML functionals' in DFT - machine-learned approximations to the target functional. By learning them directly from data, the challenge of analytically treating highly complex, non-analytic functionals is side-stepped. Neural networks (NNs) are a primary ML model used in this area [2], as they can approximate any function or functional to arbitrary precision [10] [11]. This includes non-smooth functionals, enabling accurate modelling of features like the derivative discontinuity [6]. Most previous work has focused on developing ML approximations to the $E_{xc}[n]$ functional [9],[12],[13],[14],[15] integrated with the KS scheme for a 'semi-empirical' approach that can outperform traditional approximations [16]. KS theory models many-body systems using fictitious, non-interacting particles, described by orbitals, that reproduce the real electron density. ML has also shown promise in orbital-free DFT (OFDFT), which avoids orbitals entirely, further reducing computational cost. Targets for ML in OFDFT include the mapping from the external potential to the electron density [17] [18] and the non-interacting kinetic energy functional $T_s[n]$ [18],[19],[20],[21]. These functionals are applied within a theoretical framework analogous to KS-DFT that enables the calculation of other properties. A more direct route employs ML to predict observables from the electron density. This approach is entirely data-driven, as the functional forms for properties that aren't energy components are largely unknown [22]. It has led to predictions of the band gap (a property of solids analogous to the fundamental gap of finite systems) that have surpassed standard DFT accuracy at a fraction of the computational cost [4],[22], exemplifying the potential for ML to succeed where traditional approaches fall short [18]. In quantum chemistry, this method has been applied to predict molecular properties such as dipole moment, solid phase heat of formation and elastic properties [23] [24] [25]. These studies employed convolutional neural networks (CNNs) - a subclass of NNs that employ convolutional filters to extract local features from the input data, effectively capturing the molecular structure encoded in the electron density.

In this study, CNNs were trained to predict electronic properties of 1D finite quantum systems, obtained from exact numerical solutions of the time-independent Schrödinger equation (TISE). The kinetic energy T, total energy E, electron affinity A and fundamental gap E_g were predicted using the electron density $n(x)$ and the external potential $V(x)$ as inputs, totalling 8 ML functionals. A primary aim was to gain physical insight into the relationships being modelled. With the exception of $T[n]$, there have been few prior attempts to machine-learn these mappings. Another objective was therefore to assess the feasibility of the approach and identify which input yields better predictions of each output. A similar study [26] found that various energy components could be determined accurately using either $n(x)$ or $V(x)$ as input to NNs. The PFT-motivated approach of using $V(x)$ could be especially advantageous, as it avoids computing the electron density. The gold standard for electronic structure calculations is chemical accuracy, defined as 1kcal/mole or $1.6\,\mathrm{mHa}$ per particle. All outputs were predicted close to or within this threshold (0.5-$5.3\,\mathrm{mHa}$) when using the external potential as input. Using the electron density, chemical accuracy was achieved for T and E_g, but errors for E and A were significantly greater at 38 and $36\,\mathrm{mHa}$ respectively. This discrepancy was attributed to E and A being gauge-dependent, meaning that they are affected by constant shifts in the external potential (this refers to the transformation $V(x) \rightarrow V(x) + c$ where c is a non-zero constant). This insight explains prior results [26], [27] and highlights a limitation of predicting certain properties from the electron density - despite its central role in DFT, the leading method for atomistic modelling.

2 Methods

2.1 Model systems

1-electron atoms and molecules were modelled with the iDEA code [6], a Python library for solving 1D quantum mechanical problems. Similar model systems have previously been used in this field [28]. Modelling in one dimension enables efficient data generation, while retaining key physics of 3D systems [28]. Given a system (defined by the number of electrons, their spin, and the external and interaction potentials discretised on finite grids), iDEA solves the TISE numerically, determining properties such as the ground-state total energy and electron density.

2.1.1 Atoms

Atoms were modelled with regularised Coulomb potentials, described by equation 1. Z is the nuclear charge in atomic units and $\alpha > 0$ is a softening parameter that removes the singularity at the origin.

$$V_{atom}(x) = -\frac{Z}{|x| + \alpha} \tag{1}$$

When modelling a specific atom, α can be calibrated to reproduce some property of the real 3D system. For example, the Hydrogen atom (depicted in figure 1) can be modelled using $Z = 1$ and $\alpha = 1$, yielding the correct ground-state energy of $-0.5\,\mathrm{Ha}$.

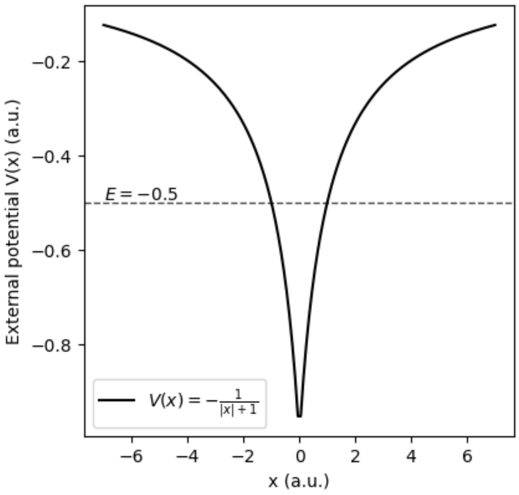

Figure 1: 1D model of the Hydrogen atom, with the external potential given in atomic units. The dashed line indicates the ground-state energy of the electron, matching the exact Hydrogen ground-state energy of $-0.5\,\mathrm{Ha}$.

However, numerical error arises due to discretisation. This error vanishes as the grid size (i.e. the domain of the x-axis covered) and density tend to infinity. As an increase in computational cost is also incurred, convergence tests are required to ensure that the solutions are sufficiently accurate without being prohibitively expensive. Figure 2(a) illustrates a convergence test: the ground-state energy of Hydrogen is calculated using grids of increasing density, approaching the exact value of $-0.5\,\mathrm{Ha}$. In this study, total energies were converged to double chemical accuracy ($0.8\,\mathrm{mHa}$ per particle) to facilitate predictions within chemical accuracy. Figure 2(b) shows that, for Hydrogen, this is reached at 7 grid points per Bohr radius. A separate test found that a system size of $10\,\mathrm{a}_0$ was sufficient to reach double chemical accuracy.

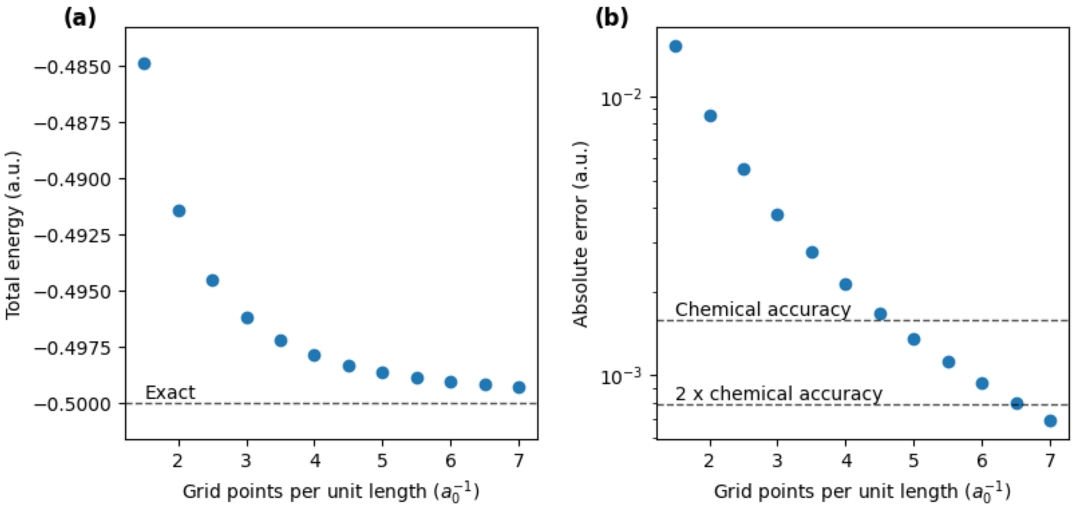

Figure 2: Convergence of the Hydrogen ground-state energy with respect to grid density. (a) shows the energy approaching the exact value of $-0.5\,\mathrm{Ha}$ as the grid becomes finer. The error at each grid resolution is plotted on a logarithmic scale in (b), indicating that chemical accuracy is reached at 5 grid points $\mathrm{a_0}^{-1}$ and double chemical accuracy at 7 grid points $\mathrm{a_0}^{-1}$.

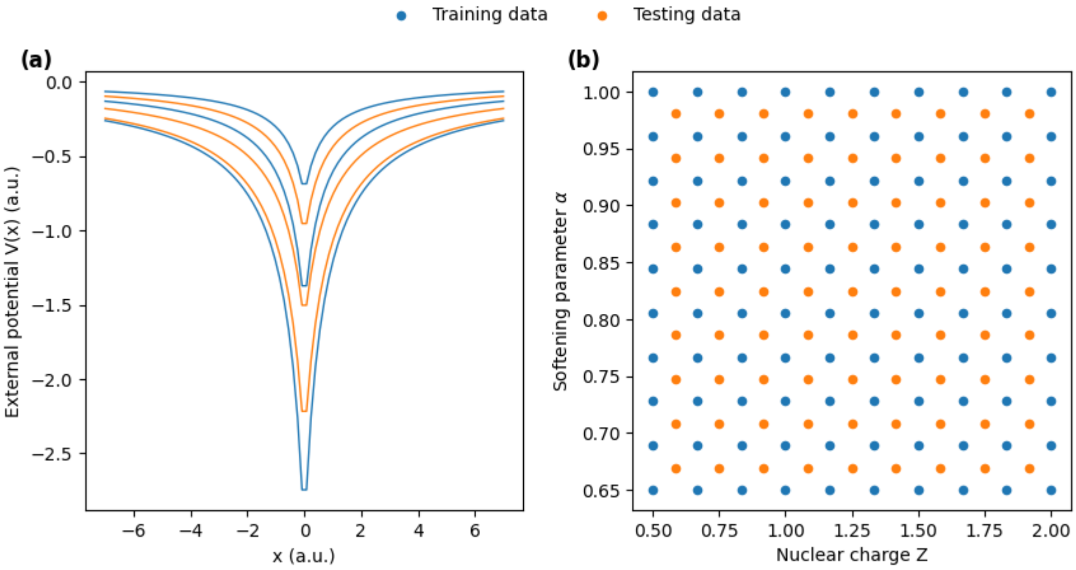

Figure 3: (a) Examples of external potentials from the main dataset used in this study - three for training (blue) and three for testing (orange). They are atomic wells centred on $x = 0$ with varying widths and depths determined by the nuclear charge Z and softening parameter α. (b) The full set of parameter combinations used to define the dataset. The training set forms a uniform 10x10 grid in parameter space, and the testing set consists of an equally-spaced 9x9 grid offset from the training points. This creates an interpolation task, as the model is evaluated on unseen combinations of the parameters that lie within the range of the training data.

ML models are trained on varied examples with the goal of interpolating to unseen data. The primary dataset used in this study comprised of atomic systems with potential wells of varying width and depth, controlled by parameters Z and α. Although the Hydrogen model was included, the remaining potentials were not intended to represent specific atoms. One reason for this was that different systems have different convergence requirements - for example, higher Z values demand finer grids to resolve their steeper potential wells. To maintain computational feasibility, Z and α were limited to the ranges 0.5-2 and 0.65-1 respectively. Ten values of each parameter were sampled uniformly from these ranges, and all possible combinations were used to create one-hundred training potentials, with the pairs of Z and α forming a grid. 81 testing potentials were defined using the midpoints of each cell in this grid, preventing overlap with the training data. This is illustrated in figure 3, along with example potentials from the training and testing set.

2.1.2 Molecules

Diatomic molecules were modelled as a sum of two atomic wells centred on $\pm r/2$, yielding an internuclear separation of r (equation 2). Figure 4 shows an example with $r = 4$. A second training set was constructed using fixed values of $Z = 1$ and $\alpha = 0.75$ and 50 values of r uniformly sampled from the range 2-6. Testing data was again defined with the midpoints of the values used for training. The parameter space was sampled densely because NNs perform better with larger training sets [29], and a greater number of testing points provides a more robust measure of performance.

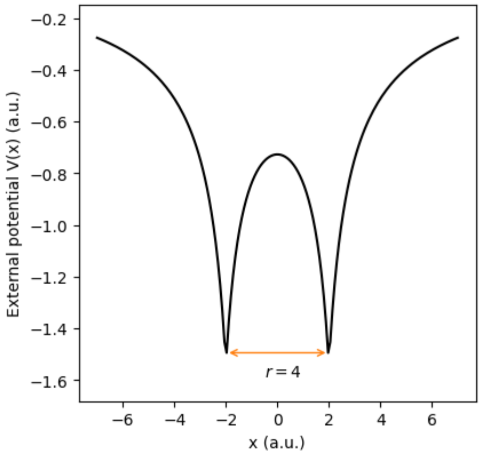

Figure 4: Example potential from the secondary dataset used in this study. It models a diatomic molecule and is defined by equation 2, with the parameter values $Z = 1$, $\alpha = 0.75$ and $r = 4$.

$$V_{molecule}(x) = -\frac{Z}{|x - r/2| + \alpha} - \frac{Z}{|x + r/2| + \alpha} \tag{2}$$

Convergence tests were undertaken only for systems with minimum and maximum values of the parameters (e.g. $r = 2$ and $r = 6$), as testing all cases was impractical. Since these systems represent the extremes, this approach was assumed to guarantee convergence of the whole dataset. In contrast to the Hydrogen model, the exact total energies of these systems is not known a priori - as a substitute, they were calculated with a highly-converged grid of length $30\,\mathrm{a_0}$ and 20 grid points per unit length. It was found that a grid of size $14\,\mathrm{a_0}$ and 12 points per unit length was sufficient to converge all atomic and molecular systems to double chemical accuracy.

In addition to the external potential, the full datasets contained the total energy and all other relevant properties of each system. The following section will detail how these properties were calculated. Each will also be classified as gauge-dependent or gauge-invariant, as this will prove essential to interpret results. For a property with value Q in a potential $V(x)$, the notation Q' will be used to denote its value in the shifted potential $V(x)+c$ where c is a non-zero constant. Q is gauge-dependent if $Q \neq Q'$ and gauge-invariant if $Q = Q'$.

2.2 Properties

2.2.1 Total energy and electron density

Solving the TISE yielded the ground-state total energy E_1 (where the subscript denotes the number of electrons, not the principal quantum number) and the ground-state electron density $n(x)$. The total energy is gauge-dependent, because shifting the external potential by c increases the potential energy of each electron by c, causing the total energy to increase by Nc. In contrast, the first H-K theorem necessitates that the electron density is not affected by the same transformation, making it gauge-invariant. These points are illustrated in figure 5, where shifting the Hydrogen potential by 0.1 Ha causes an equal increase in total energy without impacting the electron density. Including constant shifts in a dataset would prevent an ML model from learning the mapping from the electron density to the total energy (and other gauge-dependent properties), as the same input would correspond to multiple outputs. For this reason, constant shifts were ruled out by fixing the gauge. By using $1/|x|$ potentials, the gauge was fixed to $V(|x| \to \infty) = 0$, ensuring that no two potentials could differ by a constant. For example, the potential V' in figure 5(a) was excluded, as it tends to $0.1\,\mathrm{Ha}$ in the limit $|x| \to \infty$.

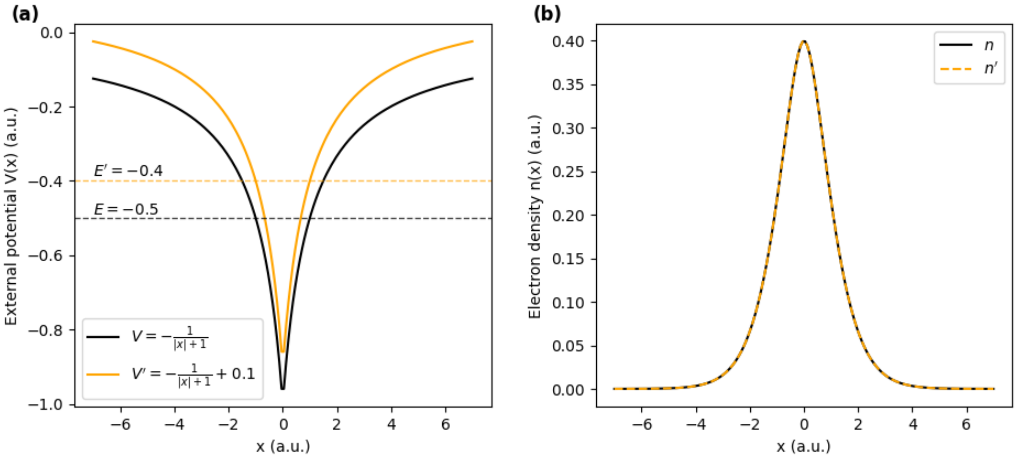

Figure 5: (a) Two potentials that differ by $c = 0.1\,\mathrm{Ha}$. The resulting total energies (indicated by dashed lines) also differ by 0.1 Ha, demonstrating that E is a gauge-dependent property. (b) The ground-state electron densities of the two systems are identical, confirming that $n(x)$ is gauge-invariant.

2.2.2 Kinetic energy

The kinetic energy is a universal functional of the electron density, meaning it does not explicitly depend on the external potential [3]. It is therefore gauge-invariant - since $n(x)$ is unchanged by a constant shift, any universal functional of $n(x)$ is also unaffected. Stated mathematically: $T' = T[n'] = T[n] = T$.

The total energy of an individual electron is the sum of its kinetic and external potential energy: $E = T + E_V$, where the external potential energy E_V is given by equation 3. This integral was approximated numerically with the rectangle rule and the kinetic energy was calculated as $T = E - E_V$.

$$E_V = \int_{-\infty}^{\infty} n(x)V(x)dx \tag{3}$$

2.2.3 Electron affinity

The electron affinity, calculated as $A = E_N - E_{N+1}$, is the energy released when adding an electron to the system. Shifting the external potential by c causes E_{N+1} to change by $(N+1)c$ and E_N by Nc. Hence, the electron affinity in the shifted potential becomes:

$A' = E_N + Nc - (E_{N+1} + (N+1)c) = E_N - E_{N+1} - c = A - c$

$A' \neq A$ so the electron affinity is gauge-dependent.

For one electron, $A = E_1 - E_2$, so E_2 was required to determine A. The TISE was solved for 2 electrons (one spin-up, one spin-down) interacting via a softened Coulomb potential defined by equation 4 (given in atomic units). Since chemical accuracy is defined per particle, the threshold for convergence of E_2 was twice as high and consequently did not require a larger or denser grid.

$$V_{int}(x,x') = \frac{1}{|x-x'|+1} \tag{4}$$

2.2.4 Fundamental gap

The fundamental gap is defined $E_g = I - A$, where I is the first ionisation energy (the energy required to remove an electron from the system). For an N-electron system, this is calculated as $I = E_{N-1} - E_N$. This is mathematically equivalent to the electron affinity for $N-1$ electrons, so by the same argument given above, $I' = I - c$. Now consider the fundamental gap subject to a constant shift:

$E'_g = I' - A' = I - c - (A - c) = I - A = E_g$; the fundamental gap is gauge-invariant.

For $N = 1$, $I = E_0 - E_1 = -E_1$ and $A = E_1 - E_2$, so E_g was calculated as $E_2 - 2E_1$.

2.3 Machine learning

2.3.1 Neural networks

NNs are functions that are often visually represented as a network of interconnected neurons structured in layers. A simple example of an NN used to model $E[V]$ is given in figure 6. It takes the external potential discretised on a 5-point grid as input, and transforms it via a sequence of operations into a prediction of the total energy. The example shown is a fully-connected NN, meaning that each neuron takes input from every neuron in the previous layer. Specifically, each input is multiplied by a weight, and the result is summed. This weighted sum is offset by a bias value and then passed through a non-linear activation function, which enables more complicated relationships to be modelled [30]. This determines the activation of the neuron, i.e. its output value. If the neuron is in a hidden layer (between the input and output layers), its activation is not directly observed, but is passed as input to the next layer. If the neuron is in the output layer, its activation represents the model's prediction.

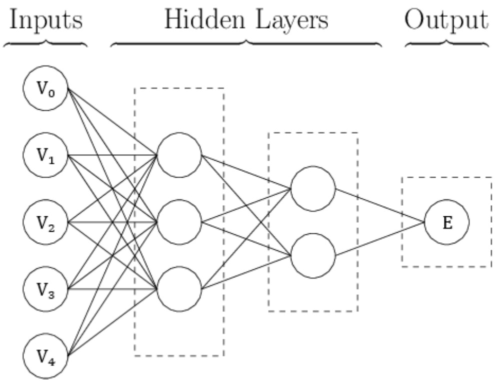

Figure 6: Schematic of an NN. The model consists of neurons (circles) arranged in layers (vertical columns of neurons). It is a fully-connected NN, because each neuron is connected to all neurons in the previous layer. The neurons in the input layer take on the values of the input sample - in this case, the external potential at 5 points in space. Each neuron in the following layer computes a weighted sum of the input values, adds a bias term, and applies an activation function. This is repeated until the output layer is reached, producing a prediction - in this case, of the total energy. Source: image adapted from [31].

The weights and biases are learnable parameters that are adjusted during training, underlying the NN's ability to learn. They are randomly initialised, leading to poor predictions at the start of training. A loss function measures the difference between a prediction and the true value. Back-propagation is then used to compute gradients of the loss function with respect to the model parameters [32], and a learning algorithm updates the parameters to reduce the loss. Each update to the parameters is referred to as a training step. Over the course of many training steps, the model's predictions iteratively improve. In addition to the learnable parameters, there are 'hyperparameters', which are fixed before training. These define the model structure (the number of neurons and hidden layers) and aspects of the learning algorithm - such as the batch size (how many training samples are processed per training step), and the learning rate (which determines the size of the step). These hyperparameters influence the training dynamics, determining whether the model converges to a minimum of the loss function and how quickly it does so [33].

2.3.2 Convolutional neural networks

CNNs build on the basic NN structure, introducing convolutional filters - small arrays of weights that slide over the input data to extract patterns or features. The number and size of the convolutional filters are hyperparameters, whereas the filter weights are learnable parameters, enabling the model to identify relevant features automatically. CNNs were originally designed for computer vision tasks, where the filters could detect visual features such as edges and objects in images. However, they have also shown exceptional performance on a range of problems involving 1D data [34]. Figure 7 represents the CNN structure used in this study. In the first convolutional layer, filters are convolved with the input data to produce feature maps, representing local, low-level features in the data. A second convolutional layer then acts on these feature maps to extract more global, higher-level patterns. This is followed by a flattening layer, which reshapes the resulting feature maps into a 1D vector. Finally, two fully-connected layers (also known as dense layers) transform this vector into a prediction. The CNN structure of input layer → convolutional layers → flattening layer → dense layers → output layer is commonly used [30]. In this study, it was found to consistently produce better results than fully-connected NNs. Throughout the study, the ReLU activation function, mean-squared error (MSE) loss function and Adam learning algorithm were used (see references [30] and [34] for more information on these and other aspects of CNNs).

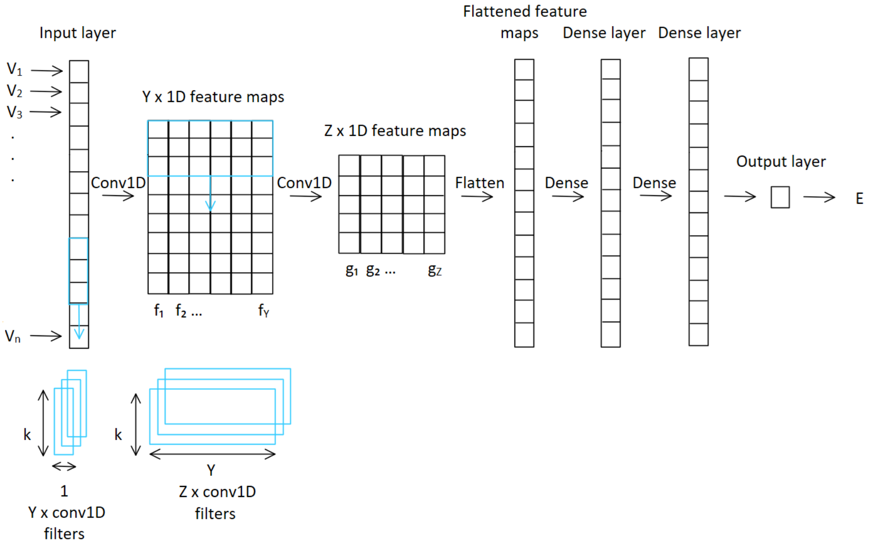

Figure 7: CNN structure used in this study. Neurons are depicted as black squares and convolutional filters are shown in light blue. The input layer is a 1D vector of length n=168, matching the dimensions of the input data. Y=10 convolutional filters of length k=8 act on the input data, each producing a 1D feature map that encodes information about local patterns in the input. These feature maps are of slightly reduced length, as the edges are excluded during convolution. In the next layer, Z=20 additional filters of size 8x10 convolve across all 10 feature maps, each outputting another 1D feature map capturing more global patterns. These are flattened into a 1D vector, and then passed through two dense (fully-connected) layers, each with 120 neurons. The output layer consists of a single neuron, to predict a scalar value.

2.3.3 Tuning, training and testing

Before training ML models, a tuning stage is used to select the optimal hyperparameters for the task. This involves training several models with different hyperparameter configurations on a subset of the training data and comparing their performance on the remaining data, termed validation data. Two tuning methods used in this study were random searches and grid searches. Both require a pre-defined search space that specifies the range of values to explore for each hyperparameter. In a grid search, the specific values for each hyperparameter are chosen explicitly and all combinations are tested, akin to the sampling process used for the parameters of the atomic potentials (see figure 3(b)). As the computational cost of grid searches scales exponentially with the number of hyperparameters included, random searches can be used to explore the search space more efficiently by sampling values at random from within the specified range [33]. Although in principle all hyperparameters should be optimised jointly, since their effects can interact, the search space may be too large for even a random search to explore thoroughly. However, through a combination of random searches and trial-and-error, it was found that several hyperparameters had limited impact on model performance, provided they were kept within reasonable ranges. For example, using anywhere between 3-6 hidden layers produced similar results. The following configuration proved effective: 4 hidden layers (2 convolutional and 2 dense), 10 and 20 filters of length 8 in the first and second convolutional layers respectively, and 120 neurons in each of the dense layers. These hyperparameters were then fixed for all models, and attention was focused on optimising the batch size and learning rate, which were identified as being the most impactful. Grid searches were used, with five learning rates (0.00032, 0.00064, 0.00128, 0.00256, 0.00512) and five batch sizes (1, 10, 25, 50, 100), resulting in 25 configurations. A randomly chosen 20% partition of the training data was reserved for validation, and one model for each hyperparameter configuration was trained for 200 training steps on the remaining 80% of the training data. After each training epoch (run through the training set), an MSE score was computed on the validation data. The training process was executed twice, and the lowest validation MSE from each run was averaged for a robust measure of performance. The results of the grid search for $A[V]$ are shown in figure 8. The red star indicates that a learning rate of 0.00256 and batch size of 50 yielded the best performance. Repeating the grid search for other functionals led to similar results, with these values being near-optimal in all cases, so they were subsequently used for all models.

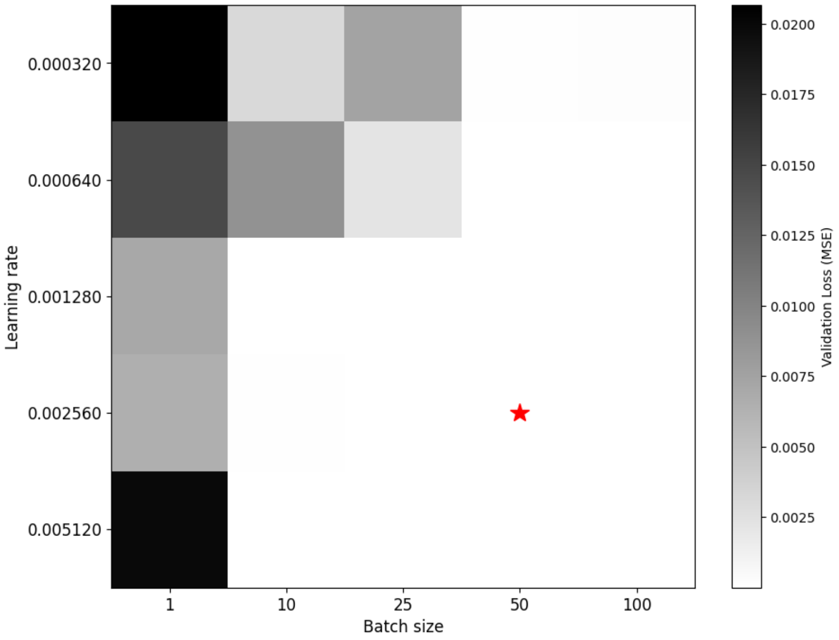

Figure 8: Results of a grid search used to determine the optimal batch size and learning rate for training a CNN to model the $A[V]$ functional. The colourbar indicates the minimum MSE validation loss, averaged over two training runs, with lighter shades corresponding to lower error. The red star marks the optimal configuration: a batch size of 50 and learning rate of 0.00256.

Once the hyperparameters were chosen, each model was trained for 300 training steps, and then assessed by its mean absolute error (MAE) on the testing set, i.e. the average absolute difference between the predicted and exact values. To mitigate the effects of random initialisation, training and testing were repeated 10 times and the MAE scores were averaged across the repeats. The standard deviation was also computed to quantify variation across runs. This process was undertaken for all 8 functionals, applied first to atomic systems, then to molecular systems.

3 Results and discussion

3.1 Atoms

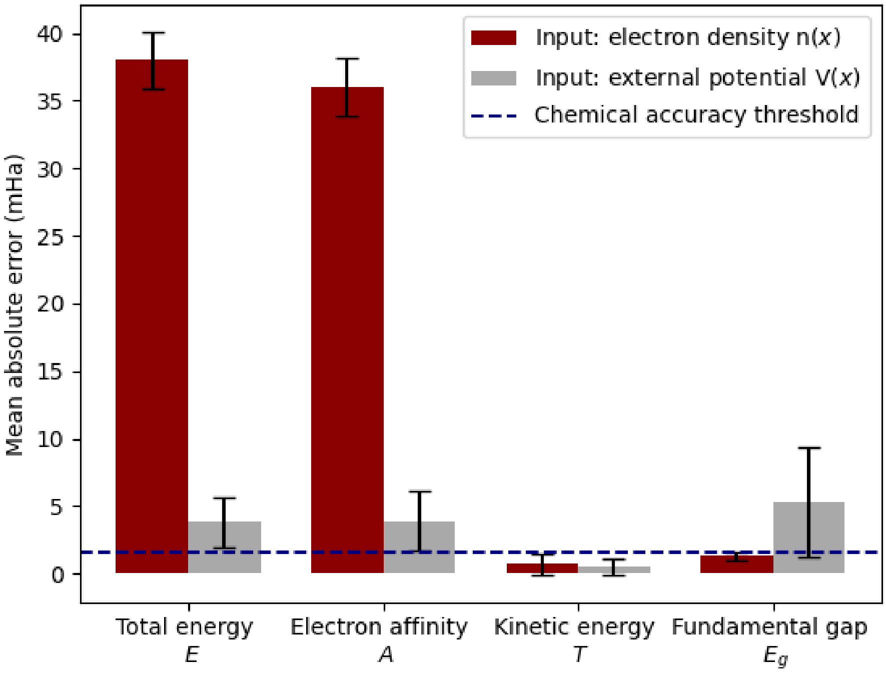

Figure 9: The average error (y-axis) predicting each output property (x-axis) of the atomic systems, using either $n(x)$ (red) or $V(x)$ (grey) as input. Errors in $E[n]$ and $A[n]$ are 1-2 orders of magnitude greater than for the remaining functionals, which were all modelled close to or within chemical accuracy (blue dashed line). Each bar represents the mean of 10 MAE scores, each an average of 81 predictions on the testing set (810 total data points). Error bars show the standard deviation over the 10 repeats. Despite some variation, the lack of overlap indicates that the larger errors in $E[n]$ and $A[n]$ were consistent across all runs.

Figure 9 shows the accuracy of each model trained and tested on the atomic dataset. The height of the bars represent the average MAE score over the 10 repeats, and the error bars indicate the standard deviation. Models that used $V(x)$ as input predicted all properties accurately, with errors in the range $0.5\text{-}5.3\,\mathrm{mHa}$. The results were less balanced when $n(x)$ was used as input - while T and E_g were predicted within chemical accuracy, the errors in E and A were substantially greater at $36\text{-}38\,\mathrm{mHa}$.

In the next section, an explanation will be proposed for this discrepancy. It will be argued that the heightened error results from mapping a gauge-invariant property ($n(x)$) to gauge-dependent properties (E and A). This also explains the results of Ryczko et al. [26], where a very similar pattern is evident. Analogous to this study, their work involved training NNs to predict a variety of electronic properties using both $n(x)$ and $V(x)$ as inputs. Crucially, the outputs included both gauge-dependent properties (total energy and external potential energy) and gauge-invariant properties (exchange, correlation and kinetic energy). Mirroring the results presented in figure 9, significantly lower accuracy was achieved on gauge-dependent density functionals than gauge-invariant ones. The authors attributed this to the external potential energy and total energy spanning larger ranges in the dataset. If this were the case, one would expect to see the same discrepancy in predictive accuracy independent of the input. However, when $V(x)$ was used as input, similar error was seen for all outputs - another striking parallel to the results presented here. Therefore, a proper account of both studies cannot be made solely with reference to the output properties, but must also explain why the discrepancy occurred specifically with the electron density input. Another previous result that may be relevant is in Ray et al. [27], where CNNs were used to map the electron density of solids to their total energy and bulk modulus. Of the two output properties, the total energy was predicted with roughly 4x lower accuracy, which may be attributable to it being gauge-dependent.

3.2 Hypothesis

To understand the potential difficulty of mapping a gauge-invariant property to a gauge-dependent one, we return to the case of the constant shift, described in section 2.2.1. If a dataset contained potentials that differed in this way, an ML model would have the impossible task of mapping the same electron density to different outputs. This difficulty can be prevented by using a fixed gauge. However, even in a fixed gauge, there is still the possibility that potentials may differ *approximately* by a constant. Since gauge-invariant properties such as the electron density are identical in two potentials that differ by a constant shift, they are likely to be *almost* identical in the case of a near-constant shift. The same does not apply to gauge-dependent properties - for example, the transformation $V(x) \to V(x) + f(x)$ where $f(x) \approx c$ is likely to cause an increase in total energy of approximately Nc. This could create a situation in which the difference in the outputs is disproportionate to that of the inputs. An example of this is shown in figure 10, which compares two of the training systems. The dashed blue line shows that, in contrast to figure 5, these potentials do not differ by a constant. Nevertheless, the outcome is very similar to figure 5, in that seemingly identical electron densities map to very different total energies. The electron densities are in fact different, although the difference is barely perceptible, whereas the difference in total energy is significant at $218\,\mathrm{mHa}$ or 22%. This suggests that the functional $E[n]$ is ill-conditioned, meaning highly sensitive to small changes in the input. The same is also true of $A[n]$ - in the example of figure 10, the electron affinities differ by $215\,\mathrm{mHa}$ or 50%. Relationships that are both highly complex and ill-conditioned can lead to numerical instability in machine learning [35] which may explain the difficulty in modelling these functionals accurately. This issue does not arise for the gauge-invariant properties: in figure 10, the kinetic energies only differ by $1.6\,\mathrm{mHa}$ or 0.76% and the fundamental gaps by $3.7\,\mathrm{mHa}$ or 0.66%. Furthermore, the potential functionals are not ill-conditioned, because the difference between the inputs is much larger than is the case for the density functionals. Hence, this interpretation explains both the increased error on gauge-dependent properties and why it occurred specifically with $n(x)$ as input.

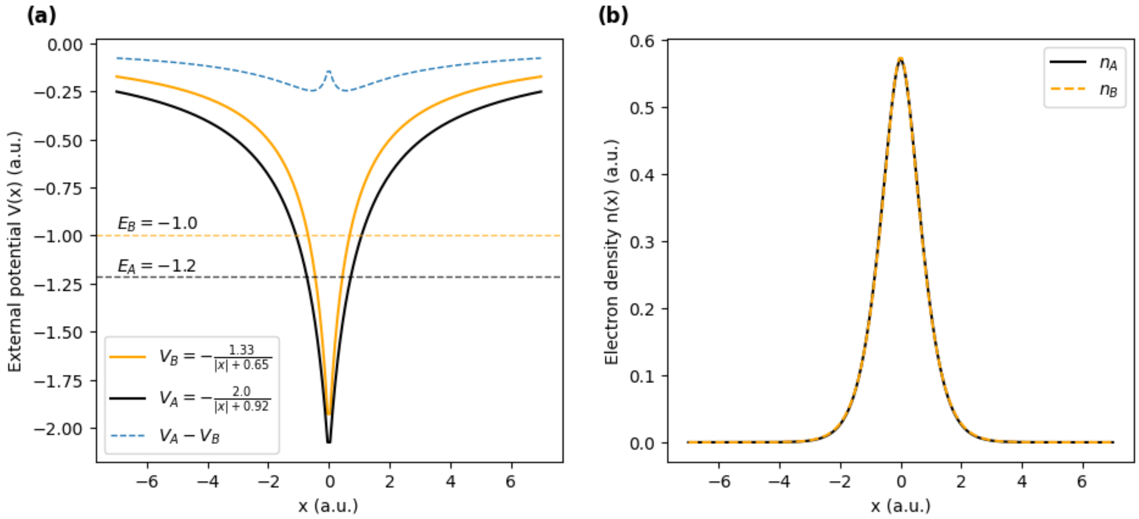

Figure 10: Two training systems that illustrate the impact of a near-constant shift in the external potential: (a) the total energies of these systems differ by over $0.2\,\mathrm{Ha}$ but (b) their electron densities are nearly identical. This reveals an ill-conditioning problem: similar inputs correspond to highly disparate outputs, making the already non-trivial mapping especially difficult for an ML model to approximate. This may explain the greater error observed when predicting E and other gauge-dependent properties from the electron density.

Figure 10 was used to illustrate the hypothesis, but this example alone is insufficient to confirm it. Similar analysis will now be extended to the rest of the training data to determine whether the greater sensitivity of $E[n]$ and $A[n]$ is a consistent pattern, and if so, whether it is a consequence of near-constant shifts in the external potential. This requires objective measures of these properties. The sensitivity of a functional is defined as the ratio of the difference of the outputs to the inputs. This is a local measure that depends on the choice of inputs. The difference in $n(x)$ is calculated with the L^2 norm given in equation 5. For the systems in figure 10, $||n_A - n_B|| = 10^{-3}$, so the sensitivity of $E[n]$ is $\frac{|E_A - E_B|}{||n_A - n_B||} = \frac{0.218}{10^{-3}} = 218$. Equation 6 is used to define the deviation from a constant shift. This represents the total variation of the difference between two potentials, which is zero when the difference is constant. In figure 10, it takes the value of 0.54.

$$||n_i - n_j|| = \sqrt{\int |n_i(x) - n_j(x)|^2 dx} \tag{5}$$

$$TV(V_i - V_j) = \int |\frac{d}{dx}(V_i(x) - V_j(x))| dx \tag{6}$$

These values were calculated for an arbitrary selection of 619 pairs of systems from the training data. In figure 11, the sensitivity of each density functional is plotted against the deviation from a constant shift for each pair. The differing y-scales of the plot make it apparent that the gauge-dependent functionals are much more sensitive on average. Moreover, it is especially pronounced for small deviations from a constant shift. This provides direct evidence of ill-conditioning in relation to near-constant shifts, specifically for gauge-dependent density functionals.

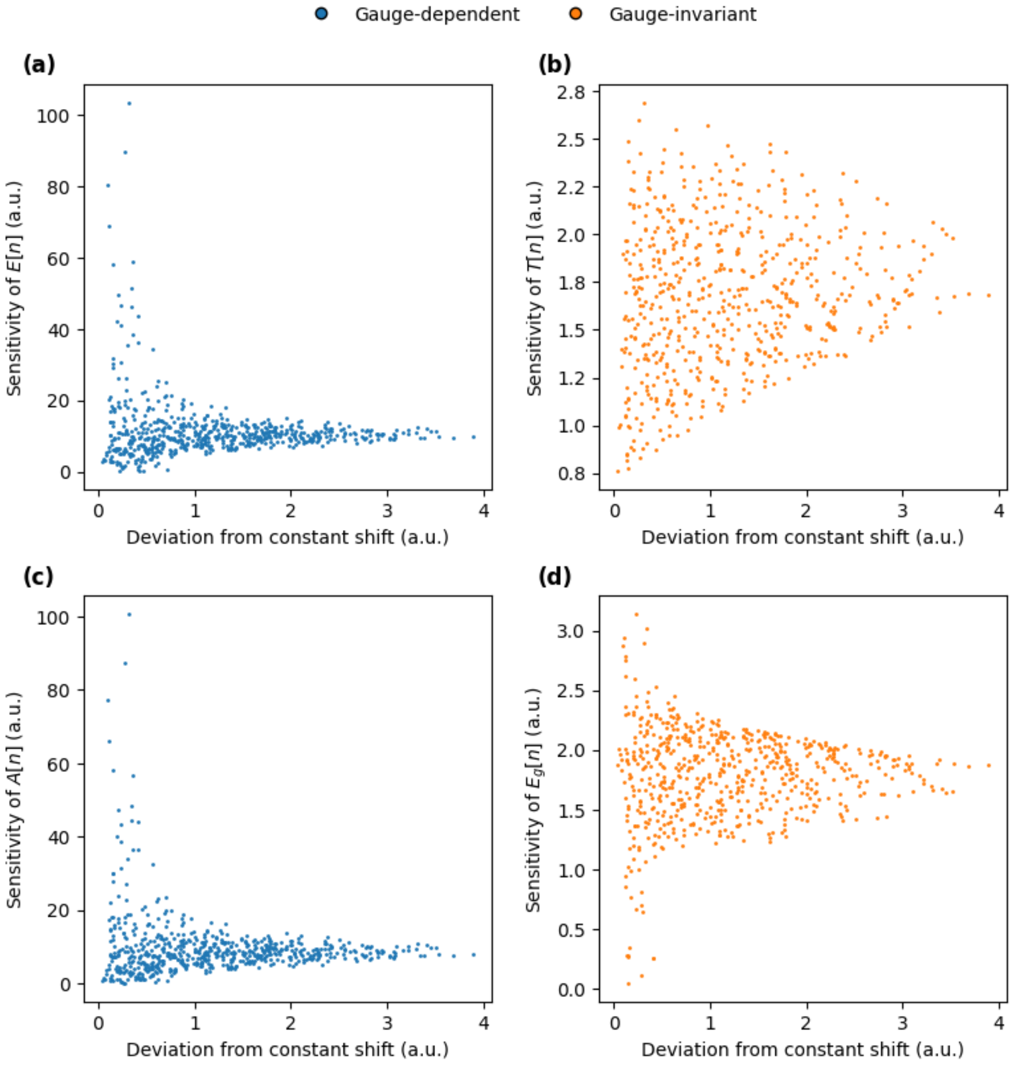

Figure 11: Sensitivity analysis of the density functionals modelled in this study, using pairs of systems from the training set. The y-axes show the ratio Δoutput/Δinput, quantifying the sensitivity of each functional. This is plotted against deviation from a constant shift to assess whether the sensitivity of gauge-dependent density functionals is accentuated by near-constant shifts. Panels (a) and (c) show sensitivities of the gauge-dependent functionals $E[n]$ and $A[n]$, while (b) and (d) show the gauge-invariant functionals $T[n]$ and $E_g[n]$. The values are consistently an order of magnitude higher for the gauge-dependent properties, indicating significantly greater sensitivity. This demonstrates that the instance of ill-conditioning shown in figure 10 is not an isolated case but is representative of the broader dataset. Vertical bands at small deviations from a constant shift in panels (a) and (b) confirm that this effect is exaggerated for near-constant shifts in the external potential.

The asymptotic behaviour in figure 11(a) and (c) is a natural consequence of mapping a gauge-invariant property to a gauge-dependent one. The sensitivity can be thought of as a finite difference estimate of the functional derivative:

$$\text{sensitivity of } Q[n] = \frac{|Q[V_i] - Q[V j]|}{||n[V_i] - n[V_j]||} = \frac{|Q[V_j + \Delta V] - Q[V_j]|}{||n[V_j + \Delta V] - n[V_j]||} \text{ where } \Delta V = V_i - V_j \tag{7}$$

In the limit that $\Delta V \rightarrow c$, the denominator tends to 0. If Q is gauge-dependent, the numerator tends to a finite value, causing the ratio to blow up to infinity. While fixing the gauge prevents the limit $\Delta V = 0$ from being reached, it can still be approached. In this limit, the functional has an extremely steep gradient - another way of describing high sensitivity. Again, the fact of having a steep gradient alone is not inherently problematic, but in combination with complicated behaviour, it makes for a challenging ML task.

3.3 Molecules

Based on the above hypothesis, it was predicted that the same difficulty learning gauge-dependent density functionals would not arise in a dataset without near-constant shifts. The molecular dataset was designed to test this prediction; by fixing Z and α and only varying r, these potentials would mostly vary in the direction opposite to a constant shift. This is illustrated in figure 12, where two molecular systems with different internuclear separations ($r = 5$ and $r = 6$) are depicted. The dashed blue line shows that the difference of the potentials is far from constant - this was confirmed quantitatively by calculating the deviation from a constant shift as 2.1, which is roughly 4x greater than in figure 10. Consequently, the electron densities of the two systems differ significantly, with an L^2 norm difference of 0.047 which is 47x greater than that of figure 10. Despite this, the total energy differs 5x less at only $40\,\mathrm{mHa}$, leading the sensitivity of $E[n]$ to be 200x lesser at 0.85. This highlights how proximity to a constant shift influences the behaviour of gauge-dependent density functionals such as $E[n]$.

To verify the lower prevalence of near-constant shifts in the molecular dataset, the deviation from a constant shift was calculated for every pair of systems in either dataset. Figure 13(a) is a kernel density estimate plot showing how the values are distributed. As intended, the deviation from a constant shift tends to be much greater for molecules. Figure 13(b) shows the impact of this: the sensitivity of $E[n]$ is greatly reduced in the molecular dataset - it is more comparable to the well-conditioned gauge-invariant functionals in the atomic dataset (figure 11(b) and (d)).

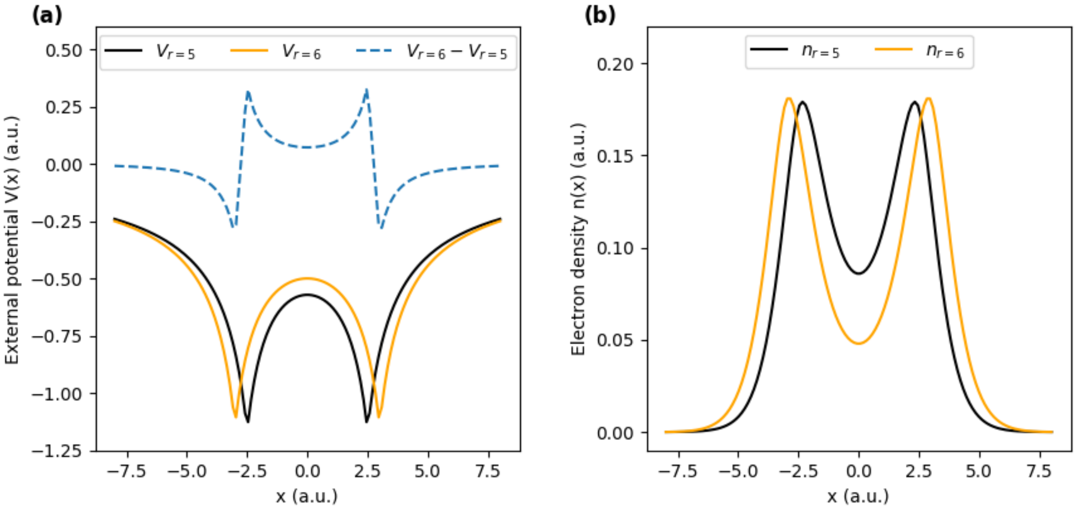

Figure 12: Example of two molecular systems with internuclear separations of $r = 5$ (black) and $r = 6$ (orange). (a) The dashed blue line shows the difference between the two external potentials, which is highly non-uniform, in contrast to the near-constant shift of figure 10. (b) This causes a large response in the electron density, illustrating the purpose of the molecular dataset - to avoid near-constant shifts and thereby mitigate the ill-conditioning observed in the atomic dataset.

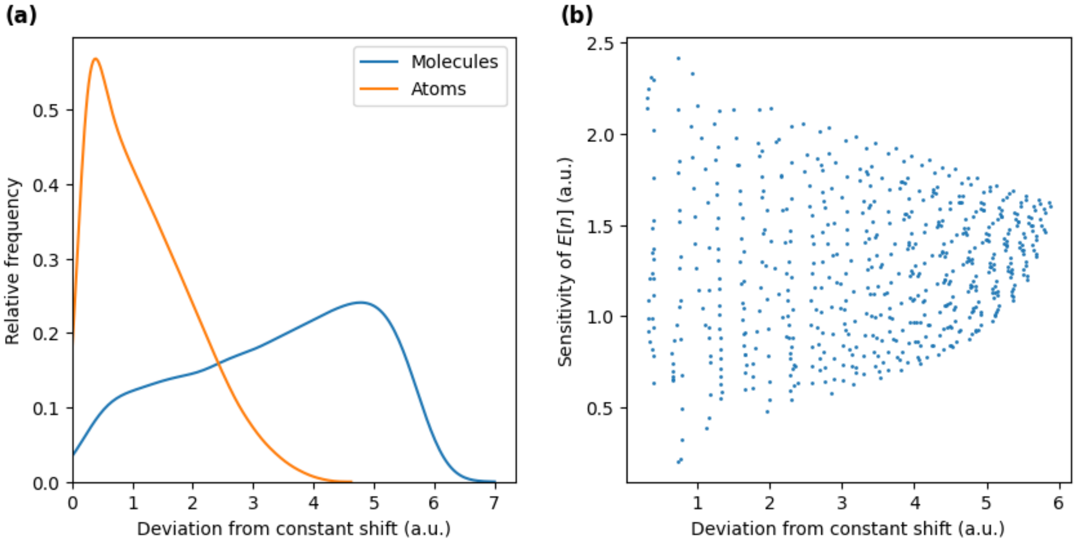

Figure 13: (a) Kernel density estimate of the distribution of deviations from a constant shift in the atomic (orange) and molecular (blue) datasets. Whereas the atomic dataset exhibits a sharp peak near zero, the molecular dataset shows a broader distribution, confirming the lower prevalence of near-constant shifts in the molecular systems.
(b) Sensitivity analysis of $E[n]$ in the molecular dataset. Compared to the equivalent plot for the atomic dataset (figure 11(a)), the sensitivity is significantly reduced and the peak at small deviations from a constant shift is no longer present. This indicates that the ill-conditioning that affected gauge-dependent density functionals in the atomic dataset is absent from the molecular dataset.

Having constructed a dataset with a lower prevalence of near-constant shifts and ill-conditioning, it was used to test the impact of these factors on ML accuracy. Figure 14 shows the accuracy of each ML functional on this dataset. The heightened error previously seen on $E[n]$ and $A[n]$ is no longer observed, with all functionals achieving chemical accuracy. This confirms the prediction, strongly supporting the proposed explanation of the previous results. The full set of results plotted in figures 9 and 14 is also given numerically in table 1.

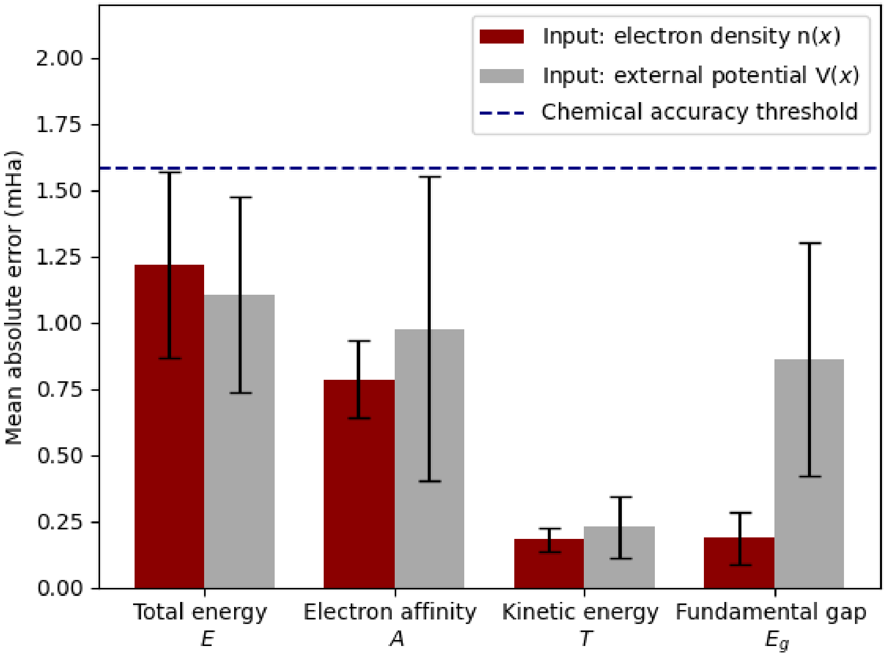

Figure 14: Equivalent plot to figure 9, showing the accuracy of each ML functional trained and tested on the molecular dataset. The heightened error previously observed for $E[n]$ and $A[n]$ is no longer present, supporting the hypothesis that these errors were caused by ill-conditioning.

Atoms

	E	A	T	E_g
$V(x)$	3.8 ± 2	3.9 ± 2	0.54 ± 0.6	5.3 ± 4
$n(x)$	38 ± 2	36 ± 2	0.72 ± 0.8	1.3 ± 0.3

Molecules

	E	A	T	E_g
$V(x)$	1.1 ± 0.4	0.98 ± 0.6	0.23 ± 0.1	0.86 ± 0.4
$n(x)$	1.2 ± 0.4	0.79 ± 0.2	0.18 ± 0.05	0.19 ± 0.1

Table 1: Mean absolute error in mHa for each functional on each dataset, with double lines separating gauge-dependent and gauge-invariant properties. This data is plotted in figures 9 and 14.

These results show that the problem is dataset-dependent, prompting further consideration of when it is likely to occur. In addition to the atomic and molecular systems detailed in this report, a variety of potentials were explored, including Gaussian dips (used in Snyder et al. [18]), sinusoids and polynomials. The difficulty in modelling gauge-dependent density functionals was a consistent pattern, with the only exception being the molecular dataset that was specifically tailored with this in mind. This suggests that the findings of this study are not limited to a restricted set of cases and may have implications for a variety of applications. This is also supported by evidence in previous work, such as the studies discussed in section 3.1 which appear to have encountered the problem identified in this study despite having used very different systems. In Ryczko et al. [26], a highly varied set of randomly-generated (RND) potentials were used to challenge NNs to generalise across diverse electronic environments. In addition to being general, this dataset bore little resemblance to the atomic dataset used in this study. Moreover, it also contained variable numbers of electrons (1, 2, 3 and 10) and the pattern persisted in all cases, providing further evidence of its generality. The data used in Ray et al. [27] was constructed from DFT calculations of realistic 3D crystals. If the greater error on $E[n]$ can be explained by the ill-conditioning hypothesis, this indicates its relevance to practical settings. A contrasting study in which $E[n]$ was modelled highly accurately with ML is Brockherde et al. [17]. Crucially, each model was trained on different geometries of a specific molecule - this mirrors the molecular dataset used in this study, so the high accuracy that was attained is consistent with our results. Considering all of the above examples, it can be inferred that the system-specific cases (e.g. modelling a single molecule) are the exception, and the issue is likely to occur when constructing a model to generalise across systems.

This finding can inform future attempts to predict gauge-dependent properties such as total energy, first ionisation energy and electron affinity using machine learning. Direct modelling of $E[n]$ can be avoided since its only gauge-dependent component (the external potential energy) can be calculated exactly, leaving only gauge-invariant components to be machine learned (the universal functional). The results of this study suggest that gauge-dependent properties in general could be predicted with greater accuracy from the external potential. Conversely, Ryczko et al. [26] found that the electron density was the more effective input for predicting both gauge-dependent and gauge-invariant properties. Further research is needed to clarify the conditions under which each input provides an advantage, particularly in the context of realistic systems. Other input quantities can also be used to predict gauge-dependent properties, such as the molecular descriptors commonly used in ML applications of quantum chemistry. This approach has been used to predict the first ionisation energy and electron affinity of molecules to an accuracy of $7\text{-}11\ \mathrm{mHa}$ [36], showing that valid alternative inputs exist.

4 Conclusions

In this study, machine learning was used to study the mapping from the electron density and external potential to various properties of 1D finite quantum systems - namely, the kinetic energy, total energy, electron affinity and fundamental gap. To assess the suitability of the method, a comparison of the accuracy achieved on each functional was made. This revealed a clear discrepancy, with gauge-dependent density functionals exhibiting significantly greater error. The source of the error was identified as being a feature of the underlying mappings, which are ill-conditioned with respect to near-constant shifts in the external potential. In combination with their complexity, this leads to numerical instability, making accurate modelling with ML difficult. This understanding led to a prediction that the error would decrease in the absence of near-constant shifts, which was confirmed using a targeted dataset.

References

[1] N. Marzari, A. Ferretti, and C. Wolverton, "Electronic-Structure Methods for Materials Design," Nature Materials, vol. 20, pp. 736–749, 2021.

[2] R. Akashi, M. Sogal, and K. Burke, "Can Machines Learn Density Functionals? Past, Present, and Future of ML in DFT," 2025.

[3] P. Hohenberg and W. Kohn, "Inhomogeneous Electron Gas," Phys. Rev., vol. 136, pp. B864–B871, Nov 1964.

[4] J. R. Moreno, J. Flick, and A. Georges, "Machine Learning Band Gaps from the Electron Density," Physical Review Materials, 2021.

[5] P. W. Ayers and R. G. Parr, "Is it Impossible to Find the Universal Density Functional? Or is it Just Well-Hidden?," Indian Journal of Chemistry Section A: Inorganic, Bio-inorganic, Physical, Theoretical and Analytical Chemistry, vol. 53A, no. 8-9, pp. 929–931, 2014.

[6] J. Gedeon, J. Schmidt, M. J. P. Hodgson, J. Wetherell, C. L. Benavides-Riveros, and M. A. L. Marques, "Machine Learning the Derivative Discontinuity of Density-Functional Theory," Mach. Learn.: Sci. Technol., vol. 3, Dec 2021.

[7] P. Borlido, J. Schmidt, A. W. Huran, F. Tran, M. A. L. Marques, and S. Botti, "Exchange-Correlation Functionals for Band Gaps of Solids: Benchmark, Reparametrization and Machine Learning," npj Computational Materials, vol. 6, no. 1, p. 96, 2020.

[8] A. Cangi, E. K. U. Gross, and K. Burke, "Potential Functionals Versus Density Functionals," Physical Review A, vol. 88, p. 062505, 2013.

[9] J. Wellendorff, K. T. Lundgaard, A. Møgelhøj, V. Petzold, D. D. Landis, J. K. Nørskov, T. Bligaard, and K. W. Jacobsen, "Density Functionals for Surface Science: Exchange-Correlation Model Development with Bayesian Error Estimation," Phys. Rev. B, vol. 85, p. 235149, Jun 2012.

[10] G. Cybenko, "Approximation by Superpositions of a Sigmoidal Function," Mathematics of Control, Signals and Systems, vol. 2, no. 4, pp. 303–314, 1989.

[11] V. E. Ismailov, "A Three Layer Neural Network Can Represent Any Multivariate Function," Journal of Mathematical Analysis and Applications, vol. 523, no. 1, p. 127096, 2023.

[12] M. Kasim and S. Vinko, "Learning the Exchange-Correlation Functional from Nature with Fully Differentiable Density Functional Theory," Physical Review Letters, vol. 127, Sept. 2021.

[13] R. Nagai, R. Akashi, and O. Sugino, "Machine-Learning-Based Exchange Correlation Functional with Physical Asymptotic Constraints," Phys. Rev. Res., vol. 4, p. 013106, Feb 2022.

[14] S. Dick and M. Fernandez-Serra, "Machine Learning Accurate Exchange and Correlation Functionals of the Electronic Density," Nature Communications, vol. 11, no. 1, p. 3509, 2020.

[15] N. Mardirossian and M. Head-Gordon, "b97x-v: A 10-Parameter, Range-Separated Hybrid, Generalized Gradient Approximation Density Functional with Nonlocal Correlation, Designed by a Survival-of-the-Fittest Strategy," Phys. Chem. Chem. Phys., vol. 16, pp. 9904–9924, 2014.

[16] J. Kirkpatrick, B. McMorrow, D. H. P. Turban, A. L. Gaunt, J. S. Spencer, A. G. D. G. Matthews, A. Obika, L. Thiry, M. Fortunato, D. Pfau, L. R. Castellanos, S. Petersen, A. W. R. Nelson, P. Kohli, P. Mori-Sánchez, D. Hassabis, and A. J. Cohen, "Pushing the Frontiers of Density Functionals by Solving the Fractional Electron Problem," Science, vol. 374, no. 6573, pp. 1385–1389, 2021.

[17] F. Brockherde, L. Vogt, L. Li, et al., "Bypassing the Kohn-Sham Equations with Machine Learning," Nat Commun, vol. 8, p. 872, 2017.

[18] J. C. Snyder, M. Rupp, K. Hansen, K.-R. Müller, and K. Burke, "Finding Density Functionals with Machine Learning," Physical Review Letters, vol. 108, Jun 2012.

[19] J. C. Snyder, M. Rupp, K. Hansen, L. Blooston, K.-R. Müller, and K. Burke, "Orbital-Free Bond Breaking via Machine Learning," The Journal of Chemical Physics, vol. 139, Dec. 2013.

[20] L. Li, J. C. Snyder, I. M. Pelaschier, et al., "Understanding Machine-Learned Density Functionals," Quantum Chem., vol. 1168, p. 819–833, 2016.

[21] K. Yao and J. Parkhill, "Kinetic Energy of Hydrocarbons as a Function of Electron Density and Convolutional Neural Networks," Journal of Chemical Theory and Computation, vol. 12, no. 3, pp. 1139–1147, 2016.

[22] B. Kolb, L. C. Lentz, and A. M. Kolpak, "Discovering Charge Density Functionals and Structure-Property Relationships with PROPhet: A General Framework for Coupling Machine Learning and First-Principles Methods," Scientific Reports, vol. 7, no. 1, p. 1192, 2017.

[23] A. D. Casey, S. F. Son, I. Bilionis, and B. C. Barnes, "Prediction of Energetic Material Properties from Electronic Structure Using 3D Convolutional Neural Networks," Journal of Chemical Information and Modeling, vol. 60, no. 10, pp. 4457–4473, 2020.

[24] Y. Zhao, K. Yuan, Y. Liu, S.-Y. Louis, M. Hu, and J. Hu, "Predicting Elastic Properties of Materials from Electronic Charge Density Using 3D Deep Convolutional Neural Networks," The Journal of Physical Chemistry C, vol. 124, p. 17262–17273, June 2020.

[25] H. Mirzaee, R. Soltanmohammadi, N. Linton, J. R. Mianroodi, L. Pastewka, and A. Hartmaier, "Elastic Constants from Charge Density Distribution in FCC High-Entropy Alloys using CNN and DFT," APL Machine Learning, vol. 2, no. 4, p. 046104, 2023.

[26] K. Ryczko, D. A. Strubbe, and I. Tamblyn, "Deep Learning and Density-Functional Theory," Phys. Rev. A, vol. 100, p. 022512, Aug 2019.

[27] P. Ray, K. Choudhary, and S. R. Kalidindi, "Lean CNNs for Mapping Electron Charge Density Fields to Material Properties," Integrating Materials and Manufacturing Innovation, vol. 14, no. 1, pp. 1–13, 2025.

[28] V. Martinetto, K. Shah, A. Cangi, and A. Pribram-Jones, "Inverting the Kohn–Sham Equations with Physics-Informed Machine Learning," Machine Learning: Science and Technology, vol. 5, p. 015050, Mar. 2024.

[29] Y. Bahri, E. Dyer, J. Kaplan, J. Lee, and U. Sharma, "Explaining Neural Scaling Laws," Proceedings of the National Academy of Sciences, vol. 121, June 2024.

[30] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions," Journal of Big Data, vol. 8, no. 1, p. 53, 2021.

[31] L. P. Cinelli, G. S. Chaves, and M. V. S. Lima, "Vessel Classification through Convolutional Neural Networks using Passive Sonar Spectrogram Images," in XXXVI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2018), 2018.

[32] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Representations by Back-Propagating Errors," Nature, vol. 323, pp. 533–536, 1986.

[33] Y. Bengio, "Practical Recommendations for Gradient-Based Training of Deep Architectures," arXiv preprint arXiv:1206.5533, 2012.

[34] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, “1D Convolutional Neural Networks and Applications: A Survey,” Mechanical Systems and Signal Processing, vol. 151, p. 107398, 2021.

[35] A. S. Krishnapriyan, A. Gholami, S. Zhe, R. M. Kirby, and M. W. Mahoney, “Characterizing Possible Failure Modes in Physics-Informed Neural Networks,” 2021.

[36] F. Pereira, “Machine Learning for the Prediction of Ionization Potential and Electron Affinity Energies Obtained by Density Functional Theory,” ChemistrySelect, vol. 8, April 2023.