Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Ph.D. Thesis
School of Physics

Prediction of Density Matrix with Machine Learning

Candidate: Suman **Hazra**

SEPTEMBER 26, 2025

Supervised by

Prof. Stefano **Sanvito**

Declaration

I, Suman Hazra, hereby declare that this thesis is my original work and has not been submitted, either in whole or in part, as part of the requirements for a degree at this or any other university.

I consent to the deposit of this thesis in the University's open-access institutional repository or its submission to the Library on my behalf, following Irish Copyright Legislation and the terms and conditions of use and acknowledgement set by Trinity College Library.

I grant permission for the examiner to retain a copy of this thesis beyond the examination period, if they so choose, in accordance with the EU GDPR regulations (May 2018).

Signed: Suman Hazra Date: 28/09/2025

Abstract

Kohn-Sham Density functional theory (KS-DFT) has become an essential tool in computational chemistry. It provides accurate simulations of the electronic structure for a wide range of periodic or molecular systems. Nevertheless, there is computational overhead associated with the iterative self-consistent field (SCF) process, which remains a significant bottleneck for large-scale DFT calculations. The convergence of an SCF approach depends on various factors; one of them is associated with the choice of initial guesses for the density matrix (DM). If the initial guess DM is close to the final converged one then the SCF approach yields fast convergence, hence, the DFT calculations can be accelerated. However, a poor guess can lead a DFT calculation towards non-converged outcomes. This thesis aims to accelerate the SCF scheme in DFT by implementing machine learning (ML) governed approaches that predict DMs and serve the needs of better initial guesses for the SCF process. Finally, this may enable efficient, non-self-consistent molecular dynamics (MD) simulations. For this thesis, two different ML-based schemes were developed; one using neural network architecture to predict initial guesses for the DMs and another employing atomic descriptors to predict real-space electron density and then constructing KS-Hamiltonian to compute the DMs. These two schemes have been implemented with the PySCF (1, 2) electronic structure package, which always starts DFT calculation from the guess DM and uses the Gaussian type orbitals for molecular calculations. In PySCF, the standard choice of initial guess includes the superposition of the atomic densities (3, 4), one electron guess (5), atom guess (6), etc. Our neural network (NN) predicted initial guess for DM can be an excellent choice, which demonstrates a significant reduction in SCF iterations compared to any other familiar guesses, and for the DM prediction atomic positions are used as the inputs. Furthermore, we applied this ML-based approach to generate the inter-atomic forces from non-self-consistent DFT calculations for conducting MD simulations using LAMMPS (7) simulation software. The major disadvantage of this NN-predicted approach is that the density matrices are rotationally contravariant. Since our NN does not explicitly enforce this property, it may generate density matrices that are inconsistent with physically expected transformations, limiting its flexibility and generalization. As a result, we applied these NN-based predictions only to molecules oriented in a specific direction.

Alternatively, in this study, we show that an ML density-based approach offers a new way to predict the DM. Thereafter, this matrix serves as an optimized initial guess for the DFT self-consistent process, requiring only a single iteration to compute atomic forces and energies necessary for molecular dynamics simulations at various temperatures. Compared to traditional alternatives for the initial guess, our density matrix exemplifies a better choice. Furthermore, PySCF does not directly provide the Kohn-Sham Hamiltonian from the electron density, and consequently, the DM, since the DM is obtained through the diagonalization of the KS Hamiltonian, which is crucial for our study. This limitation arises because the Coulomb matrix can not be computed from the density in PySCF. Here we detail the incorporation of the Coulomb repulsion matrix over Gaussian-type orbitals in PySCF, which enables the construction of the Kohn-Sham Hamiltonian directly from the electron density. These advancements pave the way for more efficient DFT-based workflows, making high-accuracy simulations accessible for complex molecular systems and real-time MD studies via PySCF and LAMMPS.

Acknowledgements

First and foremost, I express my deepest gratitude to my supervisors, Prof. Stefano Sanvito and Dr. Urvesh Patil, for their invaluable guidance, unwavering support, and insightful advice throughout the course of my PhD. Their mentorship has been instrumental in shaping this research and in my development as a researcher. A special mention goes to Prof. Stefano Sanvito's research group, whose collaborative environment has greatly enriched my learning experience. I am also thankful for the dedicated efforts of Dr. Urvesh Patil in guiding various aspects of this work and for his invaluable contributions to this thesis. I owe my deepest gratitude to my parents, whose unwavering love, sacrifices, and encouragement have been the foundation of my achievements. Their belief in me has been my greatest source of strength and inspiration. Lastly, I thank my homeland, West Bengal, and the vibrant academic community in Ireland for inspiring me to pursue excellence and making this journey a truly rewarding experience.

I extend my sincere thanks to Trinity College Dublin for providing the resources and an inspiring academic environment to pursue my research. My heartfelt gratitude goes to the faculty and staff of the School of Physics, as well as my colleagues and friends, for their support, collaboration, and encouragement throughout this journey.

I am profoundly grateful for the financial support provided by the Ministry of Social Justice and Empowerment, the Government of India, and the Science Foundation of Ireland (SFI). I would also like to acknowledge the Embassy of India in Dublin, Ireland, for their assistance and encouragement during my studies.

This work would not have been possible without the collective contributions and support of these remarkable individuals and institutions.

Publications

- S. Hazra, U. Patil, and S. Sanvito, "Predicting the one-particle density matrix with machine learning", *Journal of Chemical Theory and Computation* **20**, 1549-9618 (2024).
- S. Hazra, U. Patil, and S. Sanvito, “A machine learning workflow to perform non-self-consistent molecular dynamics,” Not published yet, writing in progress, Year: xxxx.

Conference

- 20th International Conference on Density Functional Theory and its Applications, Paris, France, 2024. Mode of participation: **Contributed talk**
- **Contributed talk** at the Trinity College Dublin, Ireland, 2024 postgraduate seminar.

Contents

List of Figures

List of Tables

1 Introduction

Kohn–Sham density functional theory (KS-DFT) (9, 10, 11, 12, 13, 14) has emerged as one of the most powerful tools in quantum chemistry and material science, enabling accurate predictions of electronic structures for a wide range of systems. In materials science, materials can be categorized based on different criteria. One common approach is to classify them according to their chemical bonding and electronic structure. The properties of materials can be determined by how they are constituted. Primarily depends on electronic or ionic arrangements. For example, the bulk modulus of a system is defined by how much the atoms or molecules in the associated crystal change when it is subjected to pressure, temperature of a substance depends on how its electron changes their state. Thus, theoretical understanding of properties of different materials demands a deep knowledge of how the atoms, ions and electrons forming it are arranged and how their distribution responds to external influences. The theoretical observation is based on the wavefunction of the systems. The wavefunction is calculated by solving the famous Schrödinger equation (15, 16). Thus, an understanding of what a wavefunction represents and how the various properties are obtained from it is crucial from the perspective of theoretical material research. Furthermore, the Schrödinger equation is a complicated differential equation and can be solved only for a limited number of chemical systems. It is therefore equally important to understand how its solution can be generalized by introducing some reasonable approximations, and how the corresponding equations can be solved on computers with numerical stability. These theoretical methods and the associated numerical techniques have been developed ever since the Schrödinger equation was established and have evolved to become much more accurate and efficient with time. Among the various theoretical approaches, a significant preference for material scientists or computational chemists is the DFT. However, despite the various aspects of these computational techniques, the electronic-structure methods developed to calculate energetic stability, equilibrium configurations, and mechanical or electronic properties are often quite similar. Traditionally, such properties have been studied within a $T = 0$ K framework, but recent advances in various computational chemistry methods and efficient computing for accurate free energies now make it possible to extend these analyses to finite temperatures. In DFT, local density approximations (LDA) (17) and semilocal functionals, such as the family of generalized gradient approximations (GGA) (18, 19), are widely used because they offer a balance of high computational efficiency and often surprisingly good accuracy. Unlike Hartree–Fock–based methods, which can be systematically improved by enlarging the many-body wavefunction, systematic refinement of the exchange–correlation (XC) functional, though formally possible (20), is numerically impractical. As a result, there is no universal way for benchmarking the performance of different functionals. Therefore, it becomes critically important to evaluate the reliability and predictive power of XC functionals by carefully comparing their results with selected experimental data. Due to the computational efficiency and predictive accuracy, this approach has become a cornerstone for studying various molecular systems and material properties. Primarily, KS-DFT (13) is a widely used framework in quantum chemistry for multiple prospects, either for chemical or periodic systems (21, 22).

DFT has established itself as one of the most powerful and widely used methods for electronic structure calculations, bridging quantum mechanics and material science. At its core, DFT rests on the Hohenberg–Kohn theorems(9), which guarantee the existence of a universal functional for the electron density. This principle yields the exact ground-state energy of a many-electron system. Kohn and Sham's introduction of the auxiliary non-interacting reference system further revolutionized the theory, making practical calculations possible by shifting the complexity into an exchange–correlation functional. Since then, DFT has become the method of choice for predicting structural, electronic, and dynamical properties of molecules, solids, and surfaces due to its standard balance between precision and computational cost compared to wavefunction-based methods such as Configuration Interaction (23) or Coupled Cluster theory (24). The central challenge in DFT remains the construction of accurate exchange–correlation (XC) functionals. The straightforward approximations, such as LDA, captured essential physics by assuming a uniform electron gas reference, which means it only depends on electron density, but often struggled with chemical accuracy. The subsequent advancement in functionals is GGA, exemplified by functionals like PBE (Perdew–Burke–Ernzerhof) (25) and BLYP (26, 27), improved upon LDA by incorporating the gradient of the electron density; they lead to significantly better performance in molecular geometries, binding energies, and surface science applications. Despite these advances, GGAs still suffer from systematic errors, such as self-interaction errors and underestimation of band gaps. To address these limitations, the field saw the development of hybrid functionals, which mix a fraction of exact Hartree–Fock exchange with GGA correlation. Popular hybrids such as B3LYP (28, 29, 30) have become standard tools in chemistry and materials science, offering improved accuracy for thermochemistry, molecular properties, and electronic band structures, but at a higher computational cost. Beyond GGAs and hybrids, meta-GGAs and range-separated hybrid functionals represent further refinements in the DFT approach. Meta-GGAs, like SCAN (Strongly Constrained and Appropriately Normed) (31), include dependence on the kinetic energy density and satisfy more known constraints of the exact functional, enabling a higher degree of transferability across different bonding environments. Range-separated hybrids, such as $\omega B97X - V$ (32), partition the electron–electron interaction into short and long range components, which improves the description of localized and delocalized electronic states simultaneously. These developments reflect a steady progression toward functionals that are more physically grounded and systematically improvable. However, despite these advances, the “Jacob’s ladder” (33) of DFT approximations still reflects significant challenges unresolved, especially in strongly correlated systems, charge transfer excitations, and van der Waals interactions. A major breakthrough in recent years has been the development of functionals and correction schemes to explicitly capture nonlocal correlation effects, particularly dispersion interactions. Traditional LDA and GGA fail to account for long-range van der Waals forces, which are crucial for biomolecules, molecular crystals, and layered materials. The state of the art in DFT is also marked by the integration of many-body physics. Methods such as DFT+U (34, 35), DFT+DMFT (Dynamical Mean Field Theory) (36, 37), and GW+DFT (38, 39) hybrids have been developed to address cases where standard functionals fail, particularly in strongly correlated materials such as transition metal oxides and f-electron systems. These extensions combine the efficiency of DFT with explicit treatments of electron correlation beyond mean-field levels, allowing researchers to probe electronic, magnetic, and transport properties with higher reliability. In parallel to methodological advances, computational innovations have significantly enhanced the reach of DFT. Real-space grids, linear-scaling methods, and efficient pseudopotential formalisms have extended the applicability of DFT to systems with thousands of atoms. Plane-wave implementations such as VASP (40), Quantum ESPRESSO (41), remain dominant in solid-state physics, while localized basis set codes PySCF (1, 2) excel in molecular systems. Moreover, high-performance computing (HPC) architectures and GPU acceleration have enabled DFT simulations on unprecedented scales, pushing into time-resolved studies and large-scale materials or drug discovery projects. The rise of high-throughput computational materials databases such as the Materials Project, AFLOW (42), and NOMAD (43) is a direct consequence of the scalability and robustness of modern DFT implementations. Despite its successes, DFT continues to face fundamental challenges. The exact exchange–correlation functional remains unknown, and approximations inevitably introduce systematic errors. The self-interaction error, delocalization error, and underestimation of band gaps persist across many standard functionals, while treating strongly correlated systems remains a grand challenge. Furthermore, the balance between accuracy and efficiency continues to shape functional development: highly accurate methods often scale poorly, while efficient functionals may lack accuracy. Nevertheless, the rapid pace of innovation, driven by theoretical aspects, computational involvement, and machine learning integration, suggests a vibrant future for DFT. Rather than being a finished theory, DFT is evolving into a versatile framework at the intersection of physics, chemistry, biology, materials science, and computer science.

KS-DFT provides a practical route by reformulating the many-electron problem in terms of the electron density rather than the many-electron wavefunction. KS-DFT utilizes an iterative self-consistent field (SCF) to produce converged results. Hence, getting an overview of the SCF process is crucial for understanding how KS-DFT works in practice and appreciating why KS-DFT simulation can be computationally demanding. The SCF approach is an iterative algorithm used to solve the Kohn–Sham equations, which are a set of single-particle equations derived from DFT. In the KS approach, one maps the interacting many-electron system onto a non-interacting one-particle system that reproduces the same ground-state electron density. The iterative SCF approach is the heart of every DFT calculation. Without it, one cannot establish the electron density that makes the KS Hamiltonian internally consistent. In other words, this iterative scheme is the bridge between the formal equations of DFT and their practical implementation using numerical advancements. The convergence of SCF also determines the reliability of the results. If SCF fails to converge—for example, in systems with challenging electronic structures such as transition metals, strongly correlated materials, or near-degenerate states—then the predicted properties (energies, forces, band structures, etc.) will be unreliable. However, considerable effort in electronic structure software development has included various advanced SCF convergence schemes, which we will discuss later. Moreover, SCF influences not only the total energies but also DFT-derived quantities such as atomic forces and stress tensors in the case of periodic systems. This makes it indispensable for applications like molecular dynamics, geometry optimization, and materials property prediction. Crucially, every step in these workflows requires a converged SCF process. The SCF procedure is also the primary factor determining the computational cost of DFT simulations. The bottleneck arises because each SCF iteration requires diagonalization of the KS Hamiltonian (or Fock matrix), which scales cubically with system size in standard approaches. For a system with N basis functions, diagonalization is $\mathcal{O}(N^3)$, and multiple iterations (often 10–50) are needed until convergence is achieved. Thus, the overall cost is substantial, especially for large molecules or extended systems. Actually, there are several reasons that affect the computational cost of the SCF scheme. System size: As the number of atoms increases, so does the number of basis functions, making diagonalization progressively more expensive. Basis set quality: A larger and more accurate basis improves results but increases the cost of Hamiltonian construction and diagonalization. SCF convergence rate: Poor convergence behaviour, often due to challenging electronic structures, can lead to a significant number of SCF iterations, significantly raising computation time. Parallelization efficiency: Modern DFT codes attempt to mitigate the cost through distributed-memory parallelization, GPU acceleration, and linear-scaling methods. Because of this cost, researchers are actively exploring alternatives and improvements, such as machine learning surrogates to reduce the SCF iterations. The SCF process initiates with a guess solution for the density matrix (DM), called the initial DM or initial guess. The efficiency of the SCF approach significantly depends on the choice of the guess DM. In recent years, various DFT packages have incorporated different strategies within iterative SCF algorithms (SCF solvers) to accelerate the SCF process. For decades, the quantum chemistry community has explored various mathematical techniques to enhance this process, achieving various degrees of success. To date, several mathematically driven techniques have been developed and implemented to optimize this scheme. For instance, consider the update protocol for the DM in a DFT calculation, where small incremental changes to the DM are allowed between iterations to facilitate convergence. In practice, this approach requires the careful tuning of adjustable input parameters to strike a balance between minimizing the number of SCF iterations and ensuring convergence. Among the various strategies for SCF acceleration, the direct inversion of the iterative subspace (DIIS) method, introduced by Pulay (44, 45), stands out as one of the most widely adopted techniques for achieving efficient SCF convergence. From the methodological development point of view, DIIS is the most efficient SCF-solver for obtaining convergence once the solution is close to a minimum (46). Various optimization techniques have been proposed to enhance DIIS convergence, including damping (47), level shifting (48, 49), and extensions of the DIIS algorithm, such as EDIIS (46) and ADIIS (50). As an alternative to the conventional SCF process, a direct minimization protocol can also be employed. One such method is the quadratically convergent self-consistent field approach (QC-SCF) (51). However, the QC-SCF method is computationally more demanding than standard SCF schemes due to the requirement of evaluating the exact Hessian of the total energy with respect to density matrix elements. An alternative to the DIIS scheme is the second-order SCF optimization (SOSCF) algorithm (50, 52). SOSCF (Newton or quasi-Newton methods) directly minimizes the function gradients with the help of the Hessian matrix (50). Furthermore, in addition to self-consistent KS-DFT, a few different schemes have been introduced within DFT, one of them being orbital-free DFT (OFDFT)(53, 54, 55, 56, 57). In principle, the motivation for using OFDFT is to replace the expensive KS-orbital calculations and to express various DFT energy contributions in terms of the electron charge density only. The advantage of OFDFT over KS-DFT is that OFDFT is computationally less demanding and, hence, most often applicable to larger systems. However, OFDFT is less accurate in comparison to KS-DFT.

The initialization of molecular orbitals (MOs), electron density, and DMs plays an important role in the success of performing the SCF process in both HF and KS-DFT calculations. The choice of the initial guess largely determines the number of SCF iterations required, the robustness of convergence, and in some cases even the physical correctness of the solution obtained. Poor guesses can lead the result in charge mislocalization, variational collapse, or convergence to an undesired solution, while well-constructed guesses often provide rapid and stable convergence. Over the decades, different strategies have been developed and are now integrated into modern quantum chemistry and electronic structure packages, reflecting a balance between simplicity, robustness, and chemical realism. Regardless of the approach utilized to optimized MOs, an initial guess is necessary. Orbital optimization is usually simpler, the closer the initial guess is to the converged result. For a note, the optimisation problem can also be reformulated in the case of HF and KS theory (58). The most basic and historically earliest method is the core Hamiltonian (HCore) or "one-electron" guess (5), where the one-electron Hamiltonian (kinetic energy plus nuclear attraction) is diagonalized in the atomic orbital basis to obtain an initial set of orbitals. This approach is exact for single-electron systems and sometimes sufficient for very small molecules, but in larger or multi-electron systems it can produce unrealistic electron distributions by neglecting electron–electron interactions. More chemically grounded alternatives emerged in the form of superposition of atomic densities (SAD), which constructs the molecular density by summing precomputed atomic densities of isolated neutral atoms placed in the molecular coordinates. SAD captures approximate charge distributions and shell structures more effectively than HCore and has become the standard default in many codes. Refinements to SAD include the use of minimal atomic orbital projections (MINAO) (3, 4) or the superposition of atomic potentials (SAP) (59), both of which improve transferability and reliability in large or diffuse basis sets. In practice, codes such as PySCF, Gaussian, and Q-Chem expose SAD, MINAO, and SAP as default or optional initial guesses, with SAD/Harris guesses forming the backbone of robust SCF starts in most molecular applications. In the domain of density-matrix–based methods, which have grown in importance with the development of linear-scaling electronic structure approaches, the initialization problem is recast. Rather than constructing MOs explicitly, these methods directly optimize the DM subject to idempotency and electron-count constraints. Here, the initial guess can be a simple Hermitian matrix with the correct trace, which is then "purified" via iterative schemes such as McWeeny purification (60) or the Palser–Manolopoulos method (61). These purification algorithms project a noisy or approximate density matrix smoothly toward an idempotent projector, providing a physically admissible starting point without diagonalization. Such strategies are central in large-scale solvers, where direct diagonalization is prohibitive. Related approaches based on density-matrix minimization or localized subspace construction similarly provide robust density matrices that can either be used directly or serve as starting points for conventional SCF schemes. Finally, recent developments point toward machine learning–assisted initial guesses. Models such as DeepSCF (62) predict electron densities directly from atomic coordinates, once implemented with a few SCF steps, converge significantly faster than traditional methods. These approaches are especially attractive for molecular dynamics and high-throughput screening, where thousands of related structures require efficient SCF convergence. The combination of ML-predicted densities with conventional SCF polish represents a promising new frontier in electronic structure initialization. In summary, the state of the art in initializing MOS, densities, and DMs includes a broad spectrum of strategies, from simple analytic guesses like HCore and SAD, to sophisticated fragment and projection schemes, to DM purification. For molecules, SAD and MINAO/SAP-style guesses provide the most reliable starting point, while in periodic systems, initialization relies heavily on smearing and spin-polarization settings. The consensus across modern electronic-structure packages is clear: while available methods like SAD/Harris are sufficient for many cases, thoughtful initialization tailored to the system at hand remains an essential ingredient for reliable and efficient SCF convergence.

Machine learning (63, 64) is a branch of computer science and artificial intelligence that focuses on building computational approaches adaptable of learning patterns from the given data and able to determine results without being explicitly programmed for every possible situation. Unlike traditional algorithms, where rules are manually defined by a human, machine learning systems rely on programs that can automatically detect structure in large amounts of information, adapt over time, and improve their performance with learning patterns from the data. At its core, machine learning is about teaching computers to generalize from examples. When given scientific or experimental data, these algorithms construct models that capture the underlying relationships between input variables and target outputs. Once trained, the models can make predictions on new, unseen data, which makes ML highly powerful in domains where explicit rule-writing is impractical, such as image recognition, speech processing, natural language understanding, and complex scientific simulations. The foundation of machine learning lies in statistical schemes, optimization, and data representation. Algorithms learn by minimizing the error between their predictions and the actual outcomes from a reference dataset, adjusting their internal parameters iteratively to better fit the data. This process is often guided by mathematical techniques such as gradient descent (65, 66) and probability theory (67, 68) , which help identify the most accurate and efficient representations of relationships hidden in the data. Different approaches to ML exist, broadly differentiated into supervised (69), unsupervised (70), and reinforcement learning (71). In supervised learning, the algorithm is trained on labeled data, where the correct answers are known, enabling it to learn mappings between inputs and outputs. In unsupervised learning, the system learns how to uncover hidden patterns or groupings in unseen data, such as clustering similar items or reducing dimensionality. Reinforcement learning is a branch of ML that aims at how the decision-maker that performs actions (agents) can learn through trial and error, and then it gets rewards for good actions and penalties for a substandard ones. Hence, over time, it develops an approach to maximize getting rewards and finally achieve the goal. One of the reasons ML has become so game-changing is the exponential growth in data availability and computing resources. Modern methodological advancement generates vast amounts of information from sources like sensors, experiments, digital communication, and online platforms. Traditional methods of analysis often struggle to process such complex, high-dimensional datasets, but machine learning thrives in these environments. For instance, ML can analyze millions of medical images or data to detect subtle signs of various diseases (72), predict financial market trends from streams of economic indicators, or accelerate scientific discovery by simulating molecular interactions at a fraction of the cost of conventional computations (73, 74). Moreover, the rise of specialized hardware such as graphics processing units (GPUs) and tensor processing units (TPUs) has enabled ML models—especially deep learning neural networks to scale dramatically, achieving breakthroughs once thought impossible. Beyond its applications, ML also represents a transition that how we think about computation itself. Instead of designing systems that rigidly follow human-created instructions, ML emphasizes adaptability and autonomy. It provides tools for dealing with uncertainty, ambiguity, and complexity, which are hallmarks of real-world problems. This does not mean that machine learning is without challenges. Issues such as interpretability, bias, fairness, and the need for vast amounts of labeled data remain active areas of research. Nevertheless, the growing combination of ML with fields like physics (75, 76), biology (77), and engineering (78) demonstrates its importance as a general-purpose computational method. A specialized branch, deep learning, employs neural networks to capture hierarchical representations of data, making it particularly effective for complex tasks such as image and speech recognition (79). Transfer learning enhances efficiency by leveraging knowledge gained from one task to improve performance on another (80). Bayesian methods incorporate probabilities to model uncertainty (81), while ensemble methods improve accuracy by combining multiple models (82). Support Vector Machines (SVMs) further contribute by identifying optimal decision boundaries (83). By enabling computers to learn from experience and continuously improve, machine learning has become one of the defining advancements of the 21^{st} century in the domain of technologies, reshaping how one can interact with machines and how we solve problems efficiently that were once beyond our reach.

In modern research, machine learning has become a dominating tool within materials science domain, allowing for fast and accurate analysis of complicated material properties, predicting new materials with desired characteristics, and optimising fabrication procedures. With previously unseen insights into the molecular and atomic levels of material behavior, its implementation also includes improving material discovery, design, and performance prediction (84). Machine learning techniques have enormous potential to revolutionize how we comprehend, work with, and immense in materials science as they develop (85). Machine learning encompasses a variety of techniques that enable computers to learn from data and make informed decisions (86). Collectively, these approaches power innovation across diverse fields, significantly advancing the capabilities of artificial intelligence. Machine learning has revolutionized materials synthesis by enabling a wide range of applications. Predictive models allow for rapid property estimation, thereby accelerating the selection of materials for specific functions (87). By uncovering complex structure–property relationships, machine learning facilitates the targeted design of materials with desired characteristics (88). High-throughput screening further enhances efficiency by enabling the rapid evaluation of extensive material libraries (89). Process optimization techniques refine fabrication parameters to achieve optimal performance (90). Data-driven discovery accelerates innovation by identifying promising candidates within large datasets (91). Moreover, generative models contribute by creating novel materials with tailored properties, fostering creativity and exploration in materials design (92). Collectively, these applications are reshaping the field of materials science, driving rapid progress and breakthroughs. Chemists are traditionally trained to design experiments and collect data, yet they are often less familiar with modern machine learning (ML) techniques (93). Unlike the computer-aided chemical research of the 1990s, which largely depended on theoretical or empirical rules (94), contemporary ML approaches are driven by large datasets that encode essential information (95, 96). However, poor-quality datasets can introduce significant challenges, even for ML applications that should, in principle, be straightforward (97). A persistent issue in chemistry is the bias of datasets toward successful experiments, whereas both positive outcomes (e.g., desired products) and negative outcomes (e.g., failed reactions) are necessary to provide a balanced representation of chemical space. Moreover, due to the inherent complexity of chemical synthesis, reported experimental conditions are often incomplete, with critical parameters overlooked. These limitations help explain why ML applications are currently more prevalent in computational chemistry than in experimental domains.

Computational datasets, constructed reliably from quantum mechanical calculations, not only enable systematic benchmarking of physicochemical properties of molecules and materials but also support the development of advanced computational methodologies. Consequently, it is imperative that chemists acquire at least a foundational understanding of ML, as such knowledge can profoundly enhance practices ranging from data recording to the design of ML-guided experiments. Over the past decade, ML techniques have increasingly complemented traditional statistical methods in physics (98, 99, 100, 101, 102). Applications of ML span a wide range of areas, including the processing of satellite data in atmospheric physics (103), improving weather forecasting (104), predicting the dynamics of many-particle systems (99), discovering novel functional materials (105), and generating new organic molecules (106). Recent studies further demonstrate that deep learning approaches can surpass human-designed statistical methods (107), highlighting the power of ML in analyzing experimental data. Beyond data analysis, ML has also been employed to uncover new physical laws and equations. For instance, symbolic regression (108, 109) and sparse identification techniques (110) have been successfully applied to derive governing equations directly from data. Generative modeling provides yet another powerful tool, enabling the identification of the most plausible theories among competing explanations of observational data achieved purely from data without any prior assumptions about the underlying physical mechanisms. Altogether, these developments illustrate the broad potential of ML in physics, from experimental applications to theoretical discovery, thereby opening numerous opportunities for advancement.

DFT is the quantum chemistry method of modern electronic structure theory, maintaining a balance between accuracy and computational cost. However, its main bottleneck lies in the self-consistent field (SCF), which requires iterative diagonalization of the Kohn–Sham Hamiltonian or Fock matrix until convergence of the electronic density. For large or complex systems, this process becomes much more expensive. In recent years, machine learning has emerged as a powerful tool to accelerate these computational challenges (63, 64). Instead of replacing quantum mechanics, ML can produce DFT-quality data by learning patterns from high-quality data, offering ways to bypass or expedite the SCF algorithm (62, 75, 111, 112). Four major directions have crystallized: (i) ML interatomic potentials (73, 74) that nearly achieve DFT accuracy at much lower cost, (ii) ML models that learn densities (112, 113), density matrices (112, 113, 114, 115), or Hamiltonians (75) to reduce SCF steps, (iii) ML-driven approximations to exchange–correlation (XC) factors (116) or orbital-free formalism, and (iv) ML integrations for parameter prediction and workflow automation (117). The widely used direction is the development and implementation of ML interatomic potential. These ML models are trained on Ab initio quality data, in particular, using energies and atomic forces. The idea is to perform the large-scale molecular dynamics simulations with near the accuracy of quantum level simulations but orders of magnitude lower computational cost. A significant breakthrough came via Behler and Parrinello's work (118), based on a high-dimensional network potential, which maps the total energy into atomic arrangements based on local environments. Thereafter, Gaussian Approximation Potentials (GAP) (119) have been developed with kernel-based regression with smooth overlap of atomic positions (SOAP) descriptors (120), which achieved excellent precision for the diverse systems. More recently, deep learning based architectures like SchNet (121) and DeePMD (122) have enlarged these ideas utilizing continuous-filter convolutions and end-to-end differentiable frameworks, offering ML-assisted simulations of thousands of atoms on nanoseconds time scale. However, challenges for ML potentials remain in data quality, transferability, and robustness in the critical domains such as bond breaking. Another line of research to accelerate DFT calculation is direct prediction for the electronic structure itself, rather than simply learning energies and forces. ML modes are now capable of predicting electronic quantities such as electron density (113), density matrix (123), or even the Kohn-Sham Hamiltonian (75). Previously, another novel work by Brockherde and co-workers (124) demonstrated that an ML model can be used to learn the density functional directly, thereby discarding the Kohn–Sham equations. However, these kinds of "functional learning" have transferability limitations for chemically diverse systems (125). More practical approaches are available for using ML to generate high-quality initial guesses for DM to accelerate the SCF algorithm. For example, convolutional neural networks (126) and equivariant message-passing models (127) have been trained to predict the converged densities or density matrices from atomic coordinates, significantly reducing the number of SCF iterations required. Other schemes, such as the OrbNet family, leverage low-cost semi-empirical features integrated with graph neural networks to predict the DFT-level energies (128). These types of ML models behave as a surrogate DFT engine via high-throughput screening while still maintaining a strong physical foundation. The standpoint of this line of research is built on a combination of ML+DFT workflows, where ML signify a reliable efficiency and accuracy, while DFT ensures robustness through final clarification. Further advances in the line of research are the design of exchange–correlation (XC) functionals and orbital-free kinetic energy functionals (129) using machine learning. As it is known, the accuracy of DFT primarily depends on the quality of the XC functional settings. ML techniques have already been explored to learn these quantities directly from high-quality reference data. Several groups have addressed this effort using neural networks (130) or kernel methods (131), which can capture the complex correlation part beyond standard generalized gradient approximation (GGA) or hybrid DFT functionals. With these kinds of effort, ML has been applied to orbital-free DFT (53, 54, 55), which bypasses the orbitals entirely and scales linearly with system size. Here, ML models are designed to learn the kinetic energy functional, with some degree of success in homogeneous or metallic systems (132). Still, it is the fact that the orbital-free DFT assisted by ML remains far from universal applicability. Overall, while ML functionals are established for algorithmic speedups, they face a higher barrier of acceptance, since sometimes removing physical constraints can compromise predictive reliability(133). Similarly, ML models are also developed to predict dielectric screening or dispersion corrections (134). These are automating routine tasks in electronic structure frameworks. Beyond the limits of parameter prediction, ML is deeply embedded in the infrastructures that support DFT. Data platforms such as AiiDA (135) and Materials Cloud (136), established under Nicola Marzari's leadership, provide provenance tracking, well-chosen datasets, and workflow automation, hence enabling reproducible ML training and model deployment. Despite the progress of ML integrations, significant challenges remain. First, transferability is still the most desired prospect for ML models. Models trained on narrow datasets are sometimes able to predict poorly when faced with new, unseen chemical environments (137). Second, ensuring conservation of physical laws and constraints is crucial. Such as, for the ML-predicted densities must yield the correct number of electrons present in a system, and predicted energies must respect symmetries (invariant to rotations, translations, and atom indexing) and extensivity. Finally, in the specific context of fast SCF convergence, it is important to balance efficiency with accuracy. ML-predicted initial guesses for DM can reduce iteration steps, but if the ML model is trained inaccurately, it might consistently overestimate or underestimate the energies and atomic forces, resulting in failure to produce molecular dynamics trajectories. According to the modern way of research, ML is reshaping the DFT algorithm not by replacing it, but by introducing ML-enhanced acceleration to it. ML potentials are already a groundbreaking effort for molecular dynamics simulations, and emerging approaches for electron densities, KS-Hamiltonians, and SCF acceleration hold the primary goal for reducing iteration counts and enabling larger-scale molecular simulations. Finally, the development of robust data infrastructures ensures that ML strategies can be integrated well into the DFT ecosystem. Together, these efforts are pointing toward a future where ML and DFT coexist in a systematic framework, balancing computational speed, accuracy, and physical reliability.

1.1 Thesis Goals and Contributions

This thesis primarily presents two different deep-learning and machine-learning-assisted workflows that aim to accelerate self-consistent KS-DFT calculations by providing a machine-learned initial guess of the DM and the real-space charge density. It is crucial to mention here that if the guess DM is quite accurate, it means a good starting point for DFT, then a smaller number of SCF iterations are expected to converge the calculations. If the quality of DMs is far from he converged point, the entire process will demand more SCF steps, and even calculations may not achieve convergence. Throughout this study, we have used Python-based Simulations of Chemistry Framework (PySCF) (1, 2) code, which initializes the SCF procedure from a guess DM. For this reason, the DM for KS-DFT serves as the central entity in our work. For both approaches, atomic positions are utilized as the inputs. In particular, for the deep learning strategy, we utilized molecular geometry directly, and for density prediction, the real-space grid has been employed as input. density prediction strategy demands the construction of the KS-Hamiltonian and then a diagonalization operation on the Fock matrix to produce the density matrices. Finally, either the ML-predicted DM or DM from the KS-Hamiltonian then serve as the initial guess for DFT calculations for various molecular systems. The integration of these approaches with MD simulation offers an advanced, accurate ML MD framework for investigating atomic-scale processes. Integrating ML with either the electron density or the DM provides an avenue to step down the most computationally expensive task of DFT, which is the SCF cycles and the Fock matrix diagonalization. Since either the DM or the real-space electron density is able to determine the ground state energy and other observables. ML models, therefore, are directly trained to map atomic positions to those fundamental DFT properties. Once trained, such ML models can predict real-space densities or DMs orders of magnitude faster than the conventional DFT approach, while retaining near DFT-level accuracy. This also accelerates ab initio molecular dynamics simulations, where the DM or densities must be computed at every simulation step, and thus reduces computational cost without compromising the accuracy for energies and forces. At the beginning, we will start the discussion by discussing how to predict the DM with the neural network (NN). Then we will move towards the real-space density prediction due to the limited generalizability of the direct DM prediction, as explained later in this section. The first part of this research focuses on developing neural networks (NN) to predict initial guesses (ML-guesses) for the density matrix (DM), with the aim of significantly accelerating the SCF procedure in DFT or potentially bypassing it entirely. The goal is to demonstrate that the ML guess serves as an exceptional reference for performing DFT calculations efficiently. For these calculations, we utilize the PySCF (1, 2), which offers a variety of SCF solvers, including DIIS (44, 45), SOSCF (52), Damping (47), and Level Shifting (48, 49). Among these, DIIS is the default solver, and both DIIS and SOSCF algorithms have been employed in this study. PySCF also provides five to six different types of initial guess orbitals for DM, which are considered for this work to evaluate the efficiency of various guesses. However, the approach of direct DM prediction using a neural network architecture is not transferable, because of the rotational equivariance property of DM, and its dimensionality depends on the choice of basis sets. In contrast, the real-space electron charge density is a scalar field, and it is also the fundamental quantity of DFT. In our second workflow, we have developed a novel machine-learning scheme based on real-space density to construct the KS-Hamiltonian and hence the DM, by using atomic descriptor-based schemes. This is then used as the initial guess for the SCF scheme. This ML approach is finally integrated to conduct non-self-consistent molecular dynamics simulations. However, a brief discussion of these atomic descriptors will be provided in subsequent chapters.

From the perspective of MD simulation approaches, it primarily follows multiple directions, with the classical force field MD and $ab-initio$ MD approaches being the two major categories. Firstly, the force field MD simulation (FFMD) (138, 139), where the forces between atoms are obtained using some pre-defined empirical potential energy functions. These strategies approximate the inter-atomic interactions based on the parameters fitted either from the experiments or from various quantum chemistry methods. The second is AIMD (140, 141), which obtains inter-atomic forces from converged electronic structure calculations, such as DFT. In terms of accuracy, FFMD is less accurate because it depends on empirical data, which can generate a significant amount of errors in the simulation where the force field was not designed to function properly, such as bond-breaking or bond-forming even for complex electronic changes. Inversely, AIMD simulations are significantly more accurate and able to capture those crucial phenomena during an MD run. The trade-off is that FFMDs are computationally inexpensive because this approach does not require performing any iterative self-consistency to obtain atomic forces, allowing faster simulation for the larger systems for long timescales. This efficiency makes it suitable for studying large biomolecules or materials, where quantum-level accuracy is not always strictly required. In contrast, an AIMD simulation involves an iterative process to achieve self-consistency, which significantly increases its computational cost. For this reason, it is limited to small systems and shorter timescale simulations. The AIMD workflow is applicable to chemical reactions, materials under extreme conditions, and those systems where electronic effects are crucial. In summary, FFMD is fast, but limitations come into concern with the validity, whereas the AIMD strategy is accurate, but it demands more computational overhead. So, the discussion highlights the ongoing methodological challenges in MD simulation: reducing the computational cost while maintaining the quantum-level accuracy. Throughout this work, we primarily tried to implement two workflows, one is based on a deep-learning framework, and the remaining is an ML real-space density and atomic descriptor-based approach to representing an alternative avenue for MD simulation without compromising the accuracy compared to a full ab-initio run. The expected impact of this study is to significantly reduce the computational cost of DFT–based MD simulations while preserving their accuracy, which offers much larger systems and longer time scales. This has broad implications across chemistry, materials science, and condensed matter physics—where accurate and affordable simulations are crucial for understanding various properties related to diverse complex systems such as biomolecules, catalytic processes, and energy materials. Finally, from the perspective of MD, this work could help bridge the gap between classical molecular dynamics and fully quantum-mechanical treatment, opening pathways for a realistic simulation framework that was previously out of computational reach.

1.2 Summary of the Thesis

This thesis is organized into five different chapters, including this introduction section. The remainder of this thesis is structured as follows:

Chapter 2 contains a theoretical overview of the tools utilized for this thesis. It provides a discussion and mathematical overview of density functional theory (DFT), primarily focused on a few aspects of DFT, like KS-equation formulated in terms of molecular orbitals, and various energy terms expressed as the functions of the electron density. Afterwards, we discuss the Fock matrix and the matrix representation of the Coulomb interaction. This is followed by an explanation of the self-consistent field (SCF) approach and its standard implementation in DFT. Subsequently, the chapter explores theoretical aspects of the electron density and density matrix. Thereafter, it discusses DFT-related energy expression in terms of the density matrix and also mentions a few aspects of various density functionals. After that, we provide some discussion of DFT from the aspect of implementation in PySCF. Following that, the chapter introduces the mathematical representation of Jacobi-Legendre polynomial-based atomic descriptors for real space electron density, which are utilized for our ML density-operated MD workflow as explained earlier. Finally, the chapter concludes with the grid-sampling approach employed for real-space predictions.

Chapter 3 is devoted to presenting the first key findings of this thesis, which centre around the development of ML-based schemes to accelerate the SCF method. The chapter explores deep learning-based developments. In the beginning, we discuss how we prepared the DM and the associated molecular configurations to be used in training the NN framework. Then we present the performance of various traditional initial guesses implemented in PySCF in comparison to ML DM guesses to verify the accuracy and SCF acceleration, then we show the accuracy of ML DM computed atomic forces in order to perform the non-self-consistent MD. These various results are validated with the parity diagrams. Furthermore, in order to check the advantages of our ML DM, the results section also includes trajectory-oriented thermal distributions of bond length, bond angle, and the various energy components computed from *abinitio* MD and non-self-consistent MD for molecules.

Chapter 4 describes also the results related to DM from the JL charge density model for this thesis, we will discuss the mathematical formulation of the Coulomb matrix over Gaussian orbitals, its implementation in PySCF to finally obtain the DM, and the results computed from our atomic-descriptor-based workflow to conduct MD simulations. This section includes parity plots for the electron density and density matrix and parity diagrams for atomic forces. Finally, the chapter demonstrates how accurately we computed the Infrared (IR) spectra for a biomolecule using density-based ML-assisted MD (non-self-consistent MD). These results are compared with full $ab-initio$ simulations at various temperatures, underscoring the efficiency and reliability of the proposed approach. Here we have also included parity diagrams for dipole moments throughout the structures obtained from MD trajectories at various temperatures. Finally, we have compared the histogram of various properties computed from ML MD runs with the same obtained from AIMD.

Chapter 5 focuses on summarizing the conclusions drawn for all of our studies and outlines future directions for methodological development and potential applications. The conclusions are divided into two sections. The first section addresses the neural network (NN)-based approach, which includes the limitations of this scheme and proposes the optimization strategies that can be followed to overcome those challenges. The remaining part provides a similar discussion for ML-density-driven workflow. The future prospects indicate several promising directions, such as extending the Coulomb integrals for higher types of molecular orbitals, applying an ML-density assisted scheme for all electron-based DFT calculations, ML DM guess for DFT functional that incorporates electron density derivatives, etc. These machine-learning approaches aim to further enhance the applicability and efficiency of the methodologies developed in this thesis.

2 | Theoretical Background

This chapter is devoted to discussing two primary theoretical aspects, which have been utilized throughout this work. Firstly, it is the implementation of density functional theory (DFT) in PySCF and secondly, the Jacobi-Legendre charge density model (JL-CDM) (113). DFT has been very popular for calculating the electronic structure in solid-state physics since the 1970s. However, DFT was not widely used as a quantum mechanical approach of chance to study quantum chemistry until the 1990s, when the approximations used were greatly refined to better model the exchange and correlation energies. In terms of computational cost, DFT is less expensive when compared to other wave function-based traditional methods, such as exchange-only Hartree-Fock (142, 143) and post-Hartree-Fock methods (144, 145, 146) which include electron correlation. Here, first, we introduce the basic DFT equations and how the self-consistent field (SCF) scheme is implemented to obtain the ground-state electronic structure. Then we move to discuss electron density and density matrix (DM), and at the end, we close the theoretical DFT section by introducing various energy components related to DFT calculations in terms of DM.

In the next section, we introduce the theoretical background of the JLCDM, which is detailed in a recent publication *Linear Jacobi-Legendre expansion of the charge density for machine learning-accelerated electronic structure calculations* (113). This published work aims to build a linear model, which is capable of predicting the converged real-space charge density with a lower computational cost than that of DFT. This section is structured as follows: firstly, it establishes how we can define the real-space electron density in terms of invariant descriptors. Then we will discuss a number of the necessary mathematical steps that are implemented to construct the JL-charge density model. Finally, we will explain an adapted grid sampling strategy, primarily focused on training the linear charge density model, just by utilizing some fraction of total atom-centred grid points to get the accurate electron density prediction, for a given molecule. Indeed here we will discuss Gaussian sampling, which allows us to remove unnecessary redundancies while maintaining the high-quality prediction of the real space charge density.

2.1 Density Functional Theory

Density functional theory (DFT) is a powerful quantum mechanical approach used to investigate the electronic structure of many-body systems, particularly for atoms, molecules, and periodic or solid systems. The fundamental concept behind DFT is to determine the properties of many-body systems by utilizing their entire ground state density (ρ), rather than the more complex many-body wavefunction-based methods. This makes DFT computationally efficient in comparison to other wavefunction-based approaches, specifically for systems with a large number of electrons.

Modern DFT primarily lies on two theorems established by Pierre Hohenberg and Walter Kohn in 1964 (9). These theorems show us a road map by using electron density how one can understand and compute the ground state properties of quantum many-body systems, precisely electrons in atoms, molecules, and solids. The first theorem legitimizes the use of electron density $\rho(\mathbf{r})$ as a basic variable. It states: **For any system of interacting electrons in an external potential $V_{ext}(\mathbf{r})$, the external potential is uniquely determined, up to a constant, by the ground-state electron density $\rho(\mathbf{r})$.** Note that the external potential is not restricted to the Coulomb potential. The key implementations of this theorem are, the ground state electron density $\rho(\mathbf{r})$ contains all the information about the external potential and, therefore, the entire Hamiltonian. Furthermore, since the Hamiltonian is obtained, this entity determines the ground state many-body wavefunction and all physical observables. Hence, the ground-state density uniquely determines all ground-state properties, for example, the kinetic energy, potential energy, and total energy, which addresses $\rho(\mathbf{r})$ as a fundamental variable. Moreover, this theorem justifies the concept that the functional of the density, instead of many-body wavefunctions, can describe the quantum system. The second theorem provides a variational principle for the energy: **There exists a universal functional such that the total energy functional**

$$E[\rho] = F_{HK}[\rho] + \int V_{ext}(\mathbf{r})\rho(\mathbf{r})d\mathbf{r} \tag{1}$$

achieves its minimum at the true ground-state density, and the minimum value is the ground-state energy. Here, $F_{HK}[\rho]$ is the universal functional, independent of the given system. This term incorporates the kinetic energy and the electron-electron interaction. The total energy functional can be minimized with respect to the trial density $\tilde{\rho}(\mathbf{r})$, such that $\tilde{\rho}(\mathbf{r}) \geq 0$ and $\int \tilde{\rho}(\mathbf{r})d\mathbf{r} = N$, where N is the total number of electrons, to determine the ground state density $\rho(\mathbf{r})$ and the total ground state energy E. Finally, these two theorems imply, all ground-state properties of a many-electron system can be obtained by just knowing the electron density. One can formulate an exact theory in terms of $\rho(\mathbf{r})$. However, the challenge becomes accurately approximating the universal functional $F_{HK}[\rho]$. Accurate calculational implementations of the DFT are far from easy to achieve, because of the unfortunate fact that the functional $F_{HK}[\rho]$ is hard to come by in explicit form. Since the F_{HK} can not be explicitly obtained, we can define it with three components: kinetic energy of a non-interacting system $T_s[\rho]$ with the same density, classical Coulomb energy between electrons $V_H(\rho)$, and the exchange-correlation energy $E_{XC}[\rho]$, the most difficult part and least understood part. The exchange-correlation functional includes exchange energy, coming from the antisymmetric property of the many-electron wavefunction due to the Pauli exclusion principle, and correlation energy-accountable for residual interaction between electrons, which is beyond the mean-field strategy. There are several choices of exchange correlation functional, like local density approximation (LDA): this scheme assumes the exchange-correlation energy depends only on the local value of the density. Generalized Gradient Approximation (GGA) includes density gradients to capture more nonlocal behaviour. Meta-GGA and hybrid functionals, which incorporate kinetic energy densities or a fraction of exact exchange from Hartree–Fock theory. The Hohenberg-Kohn theorem and the exchange-correlation functional are the key factors of DFT, leading to immense practical value for quantum simulation. DFT method makes ab-initio calculation possible by mapping the complex interactive electronic problem onto a set of non-interacting particles moving in an effective potential (Kohn–Sham approach). This is an important fact: the DFT scales favorably with system size compared to other wavefunction-based methods (e.g, CI, CCSD). Chemists can compute reaction energies, molecular geometries, vibrational spectra, transition states, etc. Physicists use this method to investigate band structures, magnetic properties, defects in solids (147), surface science (148), and more. People from the material science domain utilise DFT to study new materials, battery components (149), catalysts (148), and quantum materials (150). The Hohenberg-Kohn theorems describe a core foundation for density-based quantum calculations, stating that the ground-state density is a sufficient variable for predicting all observables. The aspect of exchange-correlation part enables practical DFT implementation via Kohn-Sham scheme.

As we discussed about Hohenberg–Kohn (HK) theorems, its core part primarily depends on the universal functional $F_{HK}[\rho]$. The functional is universal because it is the same for all systems regardless of the external potential. Without a known form for $F_{HK}[\rho]$, the variational principle defining the second HK theorem becomes unusable in practice. According to the first HK theorem, it tells a one-to-one mapping between the ground-state $\rho(\mathbf{r})$ and the external potential $V_{ext}(\mathbf{r})$, up to a constant. This implies that, by knowing the exact ground state density, one can reconstruct the potential. However, this transformation is highly non-trivial and not known explicitly for general systems. Calculating the external potential from a given density (called "density-to-potential inversion") is extremely difficult and, in most cases, because of numerical instability, especially in the presence of electron correlation and near-degeneracy. A significant part of $F[\rho]$ is the kinetic energy functional $T[\rho]$, which is particularly hard to approximate accurately using only the $\rho(\mathbf{r})$. Unlike the potential energy or the Hartree term, which can be expressed in terms of $\rho(\mathbf{r})$, the kinetic energy requires knowledge of the wavefunction or orbitals. This limitation motivated the development of Kohn–Sham (KS) DFT, where an auxiliary system of non-interacting electrons is introduced to calculate the kinetic energy $T_s[\rho]$ exactly using orbitals, bypassing the need to approximate $T[\rho]$ directly. This is the reason Kohn-Sham (KS) DFT, not HK-DFT, is the practical DFT framework utilized as one of the modern computational chemistry methods. It is true that the HK theorems justify that the ground state electron density is the key variable in quantum mechanical calculations. As we discussed before, the pros and cons of HK theorems; however, practical DFT implementations rely on the Kohn–Sham formalism, which deals with these issues by using orbitals to compute the kinetic energy and introducing approximate exchange–correlation functionals. Finally, we can say, the HK theorems highlight the theoretical foundation of DFT but not the computational recipe for the same.

The core challenge in quantum chemistry and condensed matter physics is to describe the behavior of interacting electrons in a system accurately and efficiently. Directly solving the many-electron Schrödinger equation is computationally expensive due to the exponential scaling with the number of electrons. The KS method deals with this difficulty by mapping the interacting many-electron system onto a fictitious system of non-interacting electrons moving in an effective potential. This effective potential is constructed so that the non-interacting system yields the true ground-state electron density of the original interacting system. The KS approach makes the assumption that there exists a system of non-interacting electrons that reproduces the same ground-state electron density as the real interacting system does. This is a non-trivial assumption, but it is justified by the underlying DFT theorems. The equation of motion for electrons in the KS system satisfies single-particle Schrödinger-like equations with an effective potential $V_{eff}(\mathbf{r})$, which includes three different components: the external potential (e.g., from nuclei), the classical electrostatic potential or Coulomb potential due to electron-electron repulsion, and finally the exchange-correlation (XC) potential, which is accountable for all many-body effects beyond the Hartree approximation. The KS method supplies a practical way to approximate this universal functional by introducing the auxiliary system of non-interacting electrons and splitting the energy functional into computable terms plus the unknown exchange-correlation part. The HK theorems established that the ground-state properties of an interacting many-electron system are determined uniquely by the electron density and that the ground-state energy can be found by variational minimization of a universal density functional. The Kohn-Sham method builds a non-interacting system with the same ground-state density, transforming the many-body problem into a set of self-consistent single-particle equations. The KS method relies on the HK theorems due to its theoretical legitimacy, but moves further by providing a practical computational scheme. Primarily, the approximation for KS-DFT focused on the exchange-correlation functional, where an exact form for it remains unknown. This effective potential includes the classical Coulomb repulsion V_H, and the exchange-correlation potential V_{xc}, where the last one is accountable for capturing quantum mechanical effects. The single-particle KS-Hamiltonian depends on the electron density ρ. The density depends on the wavefunction; hence, the KS equation has to be solved iteratively through the 'self-consistent field (SCF) approach. The equation of motion of the non-interacting electron is described by the KS equation,

$$\left[-\frac{1}{2}\nabla^2 + V_{eff}(\mathbf{r})\right]\psi_i(\mathbf{r}) = \epsilon_i\psi_i(\mathbf{r}) \tag{2}$$

This is a Schrödinger-like equation, and there are many ways to solve it. For the case of DFT this many-body Schrödinger equation maps onto a single-particle problem using electron density. Here ψ_i is the KS molecular orbitals ϵ_i are the KS orbital energies, the first term $-\frac{1}{2}\nabla^2$ is the one particle non-interacting kinetic energy term, and V_{eff} is the effective potential, written as

$$V_{eff}(\mathbf{r}) = V_{ext}(\mathbf{r}) + V_H(\mathbf{r}) + V_{xc}[\rho(\mathbf{r})] \tag{3}$$

Here, $V_{ext}(\mathbf{r})$ is the external potential due to the Coulomb interaction between electrons and nuclei, we will introduce it later in this section, the second term represents classical Coulomb repulsion $V_H(\mathbf{r}) = \int \frac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}$ between two electrons. Here, the effective potential can be written in terms of electron density $\rho(\mathbf{r})$ as

$$V_{eff}(\mathbf{r}) = V_{ext}(\mathbf{r}) + \int \frac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}' + V_{xc}[\rho(\mathbf{r})] \tag{4}$$

In the result section (Chapter 5), we show the full derivation of calculating the matrix elements for the Hartree/Coulomb potential $[V_H(\mathbf{r})]$ over the Gaussian basis to implement the same in PySCF code from $\rho(\mathbf{r})$ as the need mentioned before for this thesis. $V_{xc}(\mathbf{r})$ is the exchange-correlation potential. In the KS-DFT method, the total energy of a many-electron system can be expressed as a function of electron density $\rho(\mathbf{r})$. The minimization of this energy functional, $E[\rho]$, is the key factor of DFT; it can be written as

$$E[\rho] = T_s[\rho] + E_{ext}[\rho] + E_H[\rho] + E_{xc}[\rho] \ , \tag{5}$$

where E is the ground state energy, T_s is the non-interacting kinetic energy, E_{ext} is the external energy due to the external potential, E_H is energy term responsible for Coulomb repulsion, E_{xc} the exchange-correlation energy which has the several degrees of complexity. Here is one important concept to mention: that T_s is not explicitly known as a function of $\rho(\mathbf{r})$ (there is no closed-form functional), but we still write it as a notational convention to represent total DFT energy. It contributes to the energy via the orbitals that are obtained from the density $\rho(\mathbf{r})$. Basically, it is computed through ρ, even though it is not a direct form of ρ. Again, in terms of electron density, the total energy functional in the KS formalism is

$$E[\rho] = T_s[\rho] + \int V_{ext}(\mathbf{r})\rho(\mathbf{r})d\mathbf{r} + \frac{1}{2}\int\int \frac{\rho(\mathbf{r})\rho(\mathbf{r})}{|\mathbf{r}-\mathbf{r}'|}drd\mathbf{r}' + E_{xc}[\rho] \tag{6}$$

As we know, the electron charge density is defined in Eq.[8], to that $\rho(\mathbf{r})d\mathbf{r}$ represents the probability of finding electrons in the volume element $d\mathbf{r}$ at position $\mathbf{r}$. Hence, the integral over the charge density $\rho(\mathbf{r})$ throughout the entire space determines the total number of electrons present in the system.

$$\int_{-\infty}^{+\infty} \rho(\mathbf{r})d\mathbf{r} = N\ , \tag{7}$$

where N is the total number of electrons. The ground-state minimizes $E[\rho]$ subject to the constraint N as it is defined in the above equation. To perform this constraint minimization for $E[\rho]$, we need to apply Euler-Lagrange equation as $\frac{\delta E[\rho]}{\delta\rho(\mathbf{r})} = \mu$, where μ is a Lagrange multiplier (chemical potential). But the direct minimization of the energy functional is difficult because kinetic energy term (T) is unknown, where the KS approach proposed a scheme by introducing fictitious non-interacting electrons.

As it is known for DFT, electron density is the key variable, and in the introduction section, we already mentioned that for this thesis, DM is a crucial entity. The Kohn–Sham (KS) operator ($\hat{H}_{KS}$) in density functional theory (DFT) is conceptually similar to the Fock operator ($\hat{F}$) in Hartree–Fock (HF) theory, as both serve as effective one-electron Hamiltonians used to obtain orbitals from which the electron density is built. In both HF and DFT, the Fock operator plays a fundamental role in determining the molecular orbitals and hence the associated DM. For HF, primarily the Fock operator ($\hat{F}$) includes core Hamiltonian, Coulomb, and exact exchange, whereas for the pure DFT calculation, this operator replaces this exchange term with the exchange-correlation potential. In hybrid DFT methods, a portion of the exact exchange from the Fock operator is mixed with DFT exchange–correlation functionals, providing a direct formal connection between the two. Next, we will discuss the electron density $\rho(\mathbf{r})$, density matrix ($P_{\mu\nu}$) and the Fock matrix ($F_{\mu\nu}$). Here, all the matrices are in the form of a finite-dimensional array of numbers that can represent an operator once a basis is chosen. In practice, precisely in computational methods like DFT or Hartree–Fock, we discretise the problem using a finite basis set (like atomic orbitals), and then represent operators as matrices in that basis.

2.1.1 Electron Density, Density Matrix, and Fock Operator

The single-particle electron density $\rho(\mathbf{r})$ can be defined as

$$\rho(\mathbf{r}) = \sum_{i=1}^{N_{occ}} |\psi_i(\mathbf{r})|^2 \ , \tag{8}$$

where $\psi(\mathbf{r})$ is the molecular orbitals, which is the linear combination of basis functions (atomic orbitals) ϕ and can be expressed as $\psi_i(\mathbf{r}) = \sum_\nu C^*_{\nu i}\phi^*_\nu(\mathbf{r})$, C is the molecular orbital expansion coefficient and N_{occ} referrers number of occupied orbitals. However, Eq.[8] can be written in terms of molecular orbitals

$$\begin{aligned} \rho(\mathbf{r}) = \sum_{i=1}^{N_{occ}} \psi_i^*(\mathbf{r})\psi_i(\mathbf{r}) = & \\ &= \sum_{i}^{N_{occ}} \sum_{\nu} C^*_{\nu i}\phi^*_\nu(\mathbf{r}) \sum_{\mu} C_{\mu i}\phi_\mu(\mathbf{r}) \\ &= \sum_{\mu\nu} \left[\sum_{i=1}^{N} C_{\mu i} C^*_{\nu i} \right] \phi^*_\nu(\mathbf{r})\phi_\mu(\mathbf{r}) \\ &= \sum_{\mu\nu} P_{\mu\nu}\phi^*_\nu(\mathbf{r})\phi_\mu(\mathbf{r}) \ . \end{aligned} \tag{9}$$

Since we always deal with real values of the wavefunction means $\phi^*_\nu = \phi_\nu$, hence Eq.[9] can be written as

$$\rho(\mathbf{r}) = \sum_{\mu} \sum_{\nu} P_{\mu\nu}\phi_\nu(\mathbf{r})\phi_\mu(\mathbf{r}) \ , \tag{10}$$

where the one-particle density matrix takes the following form

$$P_{\mu\nu} = \sum_{i=1}^{N_{occ}} C_{\mu i} C_{\nu i} \ . \tag{11}$$

Finally, the total density matrix can be written as

$$P_{total} = P_{\mu\nu} S_{\mu\nu} \ , \tag{12}$$

here $S_{\mu\nu}$ is the overlap matrix, defined as

$$S_{\mu\nu} = \int \phi_\mu(\mathbf{r}) \phi_\nu(\mathbf{r}) d\mathbf{r} \ . \tag{13}$$

S is essential where density with non-orthogonal basis sets, where the basis functions are not mutually orthogonal. In practical calculations, it ensures that molecular orbitals are properly normalized and orthogonalized. Two important properties that need to be satisfied by the DM are, (1)the trace of the total DM should be equal to the number of electrons N present in the system, i.e. $Tr[P_{total}] = N$; (2) P_{total} should obey a condition called idempotency, which needs $P_{total}^2 = P_{total}$. Other properties of the total DM are that it is Hermitian and that in all representations the diagonal elements are always real and lie in the interval [0, 1]. When we consider the spin system then total electron density is constructed by summing up the up-spin (α_{up}) and down-spin (β_{down}) densities together as $\rho(\mathbf{r}) = \rho_{\alpha_{up}}(\mathbf{r}) + \rho_{\beta_{down}}(\mathbf{r})$. The spin-$-\sigma$ electron density can be computed as

$$\begin{aligned} \rho_\sigma(\mathbf{r}) &= \sum_{i=1}^{N_\sigma} |\psi_i^\sigma(\mathbf{r})|^2 \\ &= \sum_{i=1}^{N_\sigma} \sum_{\mu\nu} C_{\mu i}^\sigma C_{\nu i}^\sigma \phi_\mu(\mathbf{r}) \phi_\nu(\mathbf{r}) \\ &= \sum_{\mu\nu} P_{\mu\nu}^\sigma \phi_\mu(\mathbf{r}) \phi_\nu(\mathbf{r}) \ . \end{aligned} \tag{14}$$

in which N_σ is the total number of spin-(σ) electrons present in the system. Therefore the density matrix for the spin system P^σ can be defined as

$$P^{\sigma}_{\mu\nu} = \sum_{i=1}^{N_\sigma} C^{\sigma}_{\mu i} C^{\sigma}_{\nu i} \,. \tag{15}$$

It is clear from Eq.[15], that the density matrices are mirror symmetric, $P^{\sigma}_{\mu\nu} = P^{\sigma}_{\nu\mu}$. As outlined before, the total electron density is obtained from the sum of the α and β densities. Hence, a total density matrix is given by

$$P_{\mu\nu} = P^{\alpha_{up}}_{\mu\nu} + P^{\beta_{down}}_{\mu\nu} \,, \tag{16}$$

through this equation, the total electron density can be evaluated using a relation obtained in Eq.[14]. However, for our study, we have not considered any spin-polarized calculations.

The Fock matrix consists of multiple components, as defined by the Kohn–Sham (KS) equation in a chosen basis set, such as Gaussian-type basis functions. The Fock matrix F in DFT is represented as the sum of various terms as

$$F_{\mu\nu} = T_{\mu\nu} + V^{ne}_{\mu\nu} + J_{\mu\nu} + V^{XC}_{\mu\nu} \,, \tag{17}$$

where the matrix element $T_{\mu\nu}$ is the kinetic energy term and can be written as

$$T_{\mu\nu} = \int \phi_\mu(\mathbf{r}) \left(-\frac{1}{2}\nabla^2\right) \phi_\nu(\mathbf{r}) d\mathbf{r} \,, \tag{18}$$

$V^{ne}_{\mu\nu}$ is the matrix for the nuclear-electron interaction term, which represents the attraction between electrons and nuclei as

$$\begin{aligned} V^{ne}_{\mu\nu} &= \int \phi_\mu(\mathbf{r}) V_{ext}(\mathbf{r}) \phi_\nu(\mathbf{r}) d\mathbf{r} = \\ &= \int \phi_\mu(\mathbf{r}) \left(\sum_A \frac{-Z_A}{\mathbf{r} - \mathbf{R}_A} \right) \phi_\nu(\mathbf{r}) d\mathbf{r} \,, \end{aligned} \tag{19}$$

where $V_{ext}(\mathbf{r}) = \sum_A \frac{-Z_A}{\mathbf{r}-\mathbf{R}_A}$ is the external potential with Z_A be the charge of nucleus A, and $\mathbf{R}_A$ is its position. In Eq.(17) $J_{\mu\nu}$ represents each matrix element for classical electron-electron repulsion (Coulomb matrix). Mathematically, it can be represented as

$$J_{\mu\nu} = \int \phi_\mu(\mathbf{r}) V_H(\mathbf{r}) \phi_\nu(\mathbf{r}) d\mathbf{r} \tag{20}$$

Finally, $V^{XC}_{\mu\nu}$ is the exchange-correlation matrix, which is responsible for quantum mechanical exchange and correlation phenomena. In general, it can be represented as

$$V^{XC}_{\mu\nu} = \int \phi_\mu(\mathbf{r}) V_{xc}(\mathbf{r}) \phi_\nu(\mathbf{r}) d\mathbf{r} \,, \tag{21}$$

where the exchange-correlation potential is usually expressed as $V_{xc}(\mathbf{r}) = \frac{\partial E_{xc}[\rho(\mathbf{r})]}{\partial \rho(\mathbf{r})}$. In practice, E_{xc} is approximated. These can be divided into several classes, like the local density functional (LDA), where the E_{xc} depends only on the electron density. The generalized gradient approximation (GGA), where E_{xc} depends on the density gradient $|\nabla\rho(\mathbf{r})|$. The hybrid functional, where the fraction of exact exchange is used. The meta-GGA functional, where the E_{xc} includes kinetic energy density $\sum_i |\nabla\psi_i(\mathbf{r})|^2$, and the density Laplacian $\nabla^2\rho(\mathbf{r})$, etc. Any standard DFT algorithm targets the minimization of total energy for a given basis set (151), a task that leads to the following equation

$$FC = SC\epsilon \,, \tag{22}$$

where F is the Fock matrix, C is the orbital coefficients, ϵ is a diagonal matrix that represents the corresponding energy eigenvalues, and S is the overlap matrix. Finally, Eq.[22] has to solve self-consistently, starting from some initial guess density. In Density Functional Theory, the SCF cycle is an iterative process used to solve the Kohn–Sham equations and obtain the ground-state electron density of a system. It starts with an initial guess for the density matrix, often generated from superimposed atomic densities or a previous calculation. Using this density matrix, the Kohn–Sham potential or the Fock matrix is constructed, which includes contributions from the external potential (due to nuclei) and non-interacting kinetic energy component, the Hartree (electron–electron Coulomb) potential, and the exchange–correlation potential. This potential is then used to solve the Kohn–Sham equations—essentially a set of single-particle Schrödinger equations to obtain a new set of orbitals or density matrix and corresponding electron density. The newly computed density is then compared to the previous one: if the difference is within a predefined convergence threshold value, the SCF cycle stops, and the ground-state properties can be evaluated. If not, the density is updated, often using mixing schemes like Pulay (44, 45, 50) or simple linear mixing, and the process repeats. This iterative loop continues until self-consistency is achieved between the input and output densities, hence the name "self-consistent field." Computationally, the SCF scheme involves the following steps: (1) **system initialization**: define the system parameters for the DFT calculation, including atomic positions, choice of basis functions, and the selected DFT functional; (2) **density matrix guess**: Provide an initial guess for the density matrix; (3) **Fock matrix construction and diagonalization**: construct the Fock matrix by following Eq.(17) and perform matrix diagonalization to obtain energy eigenvalues and orbital coefficients using the Eq.(22); (4) **convergence check**: Check the convergence criteria if $|\rho_{new}(\mathbf{r}) - \rho_{old}(\mathbf{r})| < \epsilon$, then the convergence is achieved, otherwise it is not; (5) **density matrix update**: update the density matrix based on the new results. (6) **repeat if necessary**: if convergence is not achieved, repeat the process from steps (2) to (5) until convergence is obtained. In the SCF procedure of Kohn–Sham density DFT, several key quantities must be initialized and updated at each iteration. Initially, one must define the atomic positions, number of electrons, and the basis set or real-space grid used to represent orbitals and operators. An initial guess for the density matrix, hence the electron density $\rho(\mathbf{r})$, is required to construct the Kohn–Sham potential, which includes several components, as we mentioned before. In practice, the initial guess can be made for the density matrix $P_{\mu\nu}$, particularly when working in an atomic orbital basis, since the density matrix is used to build the Fock (or Kohn–Sham) matrix through integrals involving electron repulsion and the effective potential. For each SCF iteration, the Fock matrix is constructed from the current density matrix, and then diagonalized to obtain updated Kohn–Sham orbitals and orbital energies. These orbitals are then used to recompute the density matrix and the electron density. Hence, the electron density and the density matrix are both important entities. $\rho(\mathbf{r})$ is used to evaluate the Hartree and exchange potential in real space, where the density matrix is the key representation for constructing the Fock matrix for a finite basis set. At each SCF iteration, the updated density matrix is mixed with the previous one to aid convergence, and this cycle continues until the input and output densities agree within a predefined threshold. Other quantities that need to be maintained or updated include the orbital occupation numbers, total energy, and convergence metrics. In most of the DFT packages, the matrix responsible for the classical Coulomb repulsion integral $J_{\mu\nu}$ is directly computed by utilizing the initially provided guess for DM. The exchange-correlation matrix ($V_{\mu\nu}^{XC}$) is derived using the chosen exchange-correlation (xc) functional applied to the electron density obtained from the guess DM using the library libxc (152). Furthermore, the core Hamiltonian (H_{core}) consist of the kinetic energy term and the nuclear attraction term. These core terms are obtained from the information provided by the given basis function and the positions of the nuclei. In the PySCF package, there are several options available to perform the SCF process for a DFT calculation, like direct inversion in the iterative subspace (DIIS) (44, 45), a general second-order solver (SOS) called the co-iterative augmented hessian (CIAH) method (52).

The Direct Inversion in the Iterative Subspace (DIIS) scheme, also known as Pulay (44, 45) mixing, is a widely used method to accelerate the convergence of SCF calculations in quantum chemistry, such as those in Hartree–Fock or Kohn–Sham Density Functional Theory. In SCF iterations, the goal is to find a stable electron density or Fock matrix that is self-consistent, but iterative approaches can converge slowly or become unstable, especially for complex systems. DIIS addresses this by linearly combining a set of previously computed Fock matrices using coefficients that minimize the associated error vectors, which measure the deviation from self-consistency. Essentially, DIIS constructs an extrapolated solution that lies in the subspace spanned by prior iterations and best satisfies the SCF equations. This method greatly improves both the robustness and can accelerate the convergence, making it a standard acceleration technique in modern electronic structure codes. At each step of the SCF iteration, an approximate error vector e_i, associated with the respective variable value p_i, is calculated. Once enough iterations are obtained, a linear combination of the last m error vectors is formed.

$$e_{m+1} = \sum_{i=1}^{m} c_i e_i. \tag{23}$$

The DIIS method primarily aims to minimize the norm of e_{m+1}, by maintaining the constraint that the coefficients sum to one. The DIIS method tries to find a suitable combination of past error vectors in such a way that the new error e_{m+1} is as small as possible. To do this, it builds a new trial Fock matrix as a weighted sum of previous ones, and it requires that the weights or coefficients add up to one. If we write the trial vector as the sum of the exact solution p^f and an error vector. In the DIIS approximation, we can write

$$p = p^f \sum_i c_i + \sum_i c_i e_i. \tag{24}$$

Then the scheme minimizes the second term, while it is clear that the sum coefficients must be equal to one if we want to find the exact solution. The minimization is done by a Lagrange multiplier technique. Introducing an undetermined multiplier λ, a Lagrangian is constructed as

$$\begin{aligned}\mathcal{L} &= ||e_{m+1}||^2 - 2\lambda\left(\sum_i c_i - 1\right), \\ &= \sum_{ij} c_j B_{ji} c_i - 2\lambda\left(\sum_i c_i - 1\right), \text{where } B_{ij} = \langle e_j|e_i\rangle\end{aligned} \tag{25}$$

Now equating zero to the derivatives of L with respect to the coefficients

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial c_k} &= \sum_j B_{kj} c_j + \sum_i c_i B_{ik} - 2\lambda = \\ &= 2\left(\sum_j B_{kj} c_j - \lambda\right) = 0\end{aligned} \tag{26}$$

Since B is symmetric (i.e., $B_{ij} = B_{ji}$), and the derivatives with respect to the multiplier lead

$$\frac{\partial \mathcal{L}}{\partial \lambda} = -2\left(\sum_i c_i - 1\right) = 0 \tag{27}$$

system of $(m+1)$ linear equations that can be solved for the m coefficients and the Lagrange multiplier.

$$\begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1m} & -1 \\ B_{21} & B_{22} & \cdots & B_{2m} & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ B_{m1} & B_{m2} & \cdots & B_{mm} & -1 \\ -1 & -1 & \cdots & -1 & 0 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{28}$$

Solving this equation gives optimal c_i, which are then used to extrapolate, meaning coefficients are then used to update the variable as

$$p_{m+1} = \sum_{i=1} c_i p_i. \tag{29}$$

However, A second-order or Newtonian solver (52) refers to a method that uses both first and second derivatives, primarily the gradient and the Hessian, to accelerate convergence in iterative procedures for SCF calculations. Unlike first-order methods like DIIS, which rely only on residual minimization and linear extrapolation, Newton-type solvers focus on finding the next update by solving equations involving the Hessian matrix, which captures how the output (e.g., the Fock matrix) changes with respect to the input. This approach allows the algorithm to account for the curvature of the error surface and can lead to faster and more robust convergence, particularly in difficult cases where the DIIS approach may struggle. However, calculating or approximating the Hessian is computationally expensive, so Newtonian methods (SOS or SOSCF approach) are used selectively or with approximations in large-scale quantum chemistry calculations. As we know that in KS-DFT, the total ground state energy E is a functional of electron density, which can also be written in terms of the orbital coefficient matrix C. To optimize molecular orbitals, we consider unitary transformations of the occupied and virtual molecular orbitals:

$$C \rightarrow C^{'} = Ce^{\kappa}. \tag{30}$$

where κ anti-symmetric matrix that generates the orbital rotations:

$$\kappa_{\mu\nu} = -\kappa_{\nu\mu}. \tag{31}$$

for the SOS scheme, the goal is to search optimal κ that minimize the total energy E (52) via energy expansion (Taylor Series).

$$E(\boldsymbol{\kappa}) \approx E(0) + \sum_{\mu\nu} g_{\mu\nu}\,\kappa_{\mu\nu} + \frac{1}{2}\sum_{\mu\nu,\lambda\sigma} \kappa_{\mu\nu}\,H_{\mu\nu,\lambda\sigma}\,\kappa_{\mu\nu}. \tag{32}$$

where $g_{\mu\nu} = \frac{\partial E}{\partial \kappa_{\mu\nu}}$ and $H_{\mu\nu,\lambda\sigma} = \frac{\partial^2 E}{\partial \kappa_{\mu\nu} \partial \kappa_{\lambda\sigma}}$ which is approximate Hessian. Then it follows the Newton-like equation (matrix equation) for orbital rotation:

$$[H_{\mu\nu,\lambda\sigma}].[\kappa_{\lambda\sigma}] = -[g_{\mu\nu}] \tag{33}$$

Once κ is obtained, the n^{th} orbital coefficient C^n can be updated as

$$C^{n+1} = C^n e^{\kappa}. \tag{34}$$

here, e^{κ} ensures unitarity of the transformation (since κ is anti-symmetric). Practically, it computes the Fock matrix (F) and hence the density matrix (P), then computes the gradient of energy and approximated Hessian (H), then solves the Newton-like equation for κ, updates C using the exponential of κ until it achieves convergence. These SCF solvers, SOS and DIIS, are well implemented in the PySCF package.

2.1.2 DM-Driven Energy in PySCF

The DFT-HF total ground-energy expression (151) from the PySCF package using the density matrix P can be defined as

$$\begin{aligned} E_{tot} = \sum_{i<j} \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|} + \sum_{\mu\nu} P_{\mu\nu} H^{core}_{\mu\nu} + \frac{1}{2} \sum_{\mu\nu,\lambda\sigma} P_{\mu\nu} (\mu\nu|\lambda\sigma) P_{\sigma\lambda} - \\ - \frac{a}{2} \sum_{\mu\nu,\lambda\sigma} P_{\mu\nu} (\mu\lambda|\nu\sigma) P_{\sigma\lambda} + b \int f(\mathbf{r}) d\mathbf{r} \, , \end{aligned} \tag{35}$$

where the first term represents nuclear repulsion energy (E_{nuc}) for two different atoms Z_i and Z_j centred at $\mathbf{R}_1$ and $\mathbf{R}_2$, notably this term is not included in SCF process because it is the function of atomic position $\mathbf{R}$ only and also does not depend on $\rho(\mathbf{r})$.

where $(\mu\nu|\lambda\sigma)$ are the two-electron repulsive Coulomb integrals. Explicitly this can be defined as

$$(\mu\nu|\lambda\sigma) = \int \int \frac{\phi_\mu(\mathbf{r}_1)\phi_\nu(\mathbf{r}_1)\phi_\lambda(\mathbf{r}_2)\phi_\sigma(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1 \, d\mathbf{r}_2 \, . \tag{36}$$

Then $(\mu\nu|\lambda\sigma)$ is the electron-electron repulsion, and another term responsible for

Hartree exchange (in the fourth term), $(\mu\lambda|\nu\sigma)$ can be written as

$$(\mu\lambda|\nu\sigma) = \int \frac{\phi_\mu(\mathbf{r}_1)\phi_\lambda(\mathbf{r}_1)\phi_\nu(\mathbf{r}_2)\phi_\sigma(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1\, d\mathbf{r}_2 \, . \tag{37}$$

The key difference between Eq.(36) and Eq.(37) is the exchange integral uses $(\mu\lambda|\nu\sigma)$ instead of $(\mu\nu|\lambda\sigma)$, it is clear that here the second and third indices are swapped, which ensures the Pauli exclusion principle is accounted to preserve the antisymmetric nature of the exchange term in Eq.(37). Here a and b are constants which define some fraction of the Hartree-Fock exchange and the weights of the density functional approximation, respectively. The choices $a = 1$ and $b = 0$ refer to exchange only (HF) calculation, whereas $a = 0$ and $b = 1$ yield actual density functional calculation without exact exchange, like the Perdew–Burke–Ernzerhof (PBE) functional (25). In contrast, $a \neq 0$ and $b \neq 0$ correspond to a hybrid functional. In quantum chemistry, the most famous example is the B3LYP functional(28, 29, 30). The second term in Eq.[35] referred to as the one-electron term (E_{one}), which is associated with the core Hamiltonian, includes the kinetic energy of the electrons and the Coulomb attraction term for N nuclei in the system, and the matrix element can be evaluated as

$$H_{\mu\nu}^{core} = \int \phi_\mu(\mathbf{r}) \left[\frac{1}{2}\nabla^2 + V_{ext}(\mathbf{r})\right] \phi_\nu(\mathbf{r}) d\mathbf{r} \tag{38}$$

However, the core Hamiltonian does not include electron-electron interactions. These contributions are described by the third and fourth terms in Eq.[35], which describes the classical Coulomb and the quantum mechanical "exchange" energy, and are referred to as E_J and E_K , respectively. E_J can be computed directly derived from the electron charge density $\rho(\mathbf{r})$ as

$$E_J = \frac{1}{2}\int\int \frac{\rho(\mathbf{r}_1)\rho(\mathbf{r}_2)}{\mathbf{r}_1 - \mathbf{r}_2} d\mathbf{r}_1 d\mathbf{r}_2 = \frac{1}{2}\sum_{\mu\nu,\lambda\sigma} P_{\mu\nu}(\mu\nu|\lambda\sigma)P_{\sigma\lambda} \, , \tag{39}$$

whereas the expression for the exchange energy contribution E_K follows Eq.(37). The last term of Eq.[35] known as quantum mechanical exchange energy $E_{XC} = E_X + E_C$, describes the DFT exchange (E_X) and DFT correlation (E_C) contributions, these are similar to E_K, arise from electronic interactions, because for HF there is no concept of (E_C). This E_{XC} term can be expressed as

$$E_{xc} = \int d\mathbf{r} f(\mathbf{r}) = \int \rho(\mathbf{r})\epsilon^{XC}(\mathbf{r})d\mathbf{r}\ . \tag{40}$$

Here, the density $\rho(\mathbf{r})$ is contracted from the converged DM by following Eq.(9). ϵ^{XC} is the exchange-correlation energy density per electron. Here $f(\mathbf{r})$ is a function of electron density. When this function only depends on $\rho(\mathbf{r})$, then the XC functional is called the LDA (17) functional. This $f(\mathbf{r})$ also depends on the derivative of electron density $[\nabla\rho(\mathbf{r})]$ as $f[\rho(\mathbf{r}), \nabla\rho(\mathbf{r})]$ for GGA functional. Furthermore, it also depends on the kinetic energy density $\tau(\mathbf{r})$ and $\nabla^2\rho(\mathbf{r})$ for meta-GGA functional. The accuracy of DFT calculations depends on which rung of Jacob's Ladder (33) is used to describe the exchange-correlation effects.

2.1.3 DFT in PySCF

PySCF is a high-performance computing (HPC)-friendly Python-based software for quantum chemistry calculations, including the DFT. PySCF is particularly known for its flexibility and efficiency in both small and large-scale electronic structure calculations (52). It also supports restricted and unrestricted calculations from the perspective of HF and KS-DFT calculations. This code offers a variety of methods to compute molecular properties such as total energies, atomic forces, dipole moment, etc. All of these calculations rely on the choice of basis sets and exchange-correlation functionals to solve the Kohn-Sham equations in DFT. PySCF performs DFT calculations for all electrons in a given system, which means including both the valence electrons and the core electrons. However, it also uses the effective Core potential (ECP) (153, 154, 155, 156, 157) to perform DFT calculations. ECPs are particularly useful for heavy atoms where the number of core electrons is large, and the computational cost can become smaller in comparison to all-electron calculations. ECPs reduce the number of basis functions required to describe the system, focusing only on the valence electrons. In PySCF, for molecular DFT calculations, it employs Cartesian Gaussian-type basis functions. This is the reason for this study: we used Gaussian-type orbitals to construct the Coulomb matrix from electron density. This package also supports a variety of basic sets. A few examples are $STO-3G$, $6-31G$, and $6-311G$. It also supports polarization and diffuse functions, like $6-31G^*$, $aug-cc-pVTZ$, etc. As we described before, the exchange-correlation energy E_{xc} is approximated by density functional approximation, and they are themselves divided into several classes. PySCF also includes several options for exchange-correlation functionals. Here, the exchange-correlation functional can also be customized and utilized for DFT calculations. Moreover, this Python-based code is highly modular and flexible, enabling easy integration with other computational chemistry and machine learning packages. Moreover, this code provides an option to initiate or restart the DFT calculations by using the different initial guesses for DM, as in this study, we provided the ML DM guess in PySCF. Furthermore, PySCF can also able to save the DM from its calculations and allows the restarting of DFT calculations using the stored DM for a given molecule. This flexibility, specifically for DM guesses, makes it an excellent choice for extending or enhancing various workflows, whether it involves coupling with molecular dynamics engines or applying machine learning models to predict electronic structures. Here it is also worth mentioning that PySCF also provide an option to perform molecular dynamics simulations.

2.2 Introduction of Machine Learning

In modern computational science, machine learning has emerged as a transformative tool across disciplines, enabling computers to learn patterns from data and make predictions or decisions without being explicitly programmed for specific tasks. In the context of scientific computing, particularly in areas like quantum chemistry and materials science, ML methods are now being used to accelerate simulations, predict molecular properties, and model complex physical systems with remarkable accuracy and efficiency. At the core of many ML applications lies the concept of a neural network, a computational model inspired by the structure of the human brain. A neural network consists of interconnected layers of nodes (also called neurons), where each connection has an associated weight. These weights are adjusted during training so that the network learns to approximate a desired function, such as mapping molecular structures to their potential energy or predicting the DM or electron density from atomic configurations.

2.2.1 Neural Network

A neural network (158, 159) is typically built with an input layer, one or more hidden layers, and an output layer. The input layer receives the input data (such as molecular geometry), and the output layer generates the prediction (such as ground state energy, DM or charge density). The hidden layers contain the neurons that process inputs using weighted sums followed by non-linear activation functions. During the training process, the network learns by minimizing a loss function, which measures the difference between the predicted output and the known reference results. To perform this optimization, a technique called backpropagation is used in conjunction with gradient descent algorithms (65, 66), which iteratively update the weights of the network to reduce the loss. The success of a machine learning model relies on how the available data is prepared. In practice, the entire dataset is divided into three subsets: the training set, validation set, and test set. The training set is used to teach the model; this is where the model learns the underlying patterns and adjusts its internal parameters. The validation set is used to tune hyperparameters (like learning rate, number of neurons, or regularization strength) and monitor the model's performance during training to prevent overfitting, a situation where the model learns too well from the training data but performs poorly on new, unseen data. The test set, on the other hand, is used only after training and validation are complete. It provides an unbiased evaluation of the final model's performance in terms of accurate prediction capability. Training of ML model is an iterative process. It begins with initializing the model's weights randomly. During each iteration or epoch, a batch of training data is fed into the network, and the predictions are compared to the known targets using the loss function. The gradients of this loss with respect to each weight are computed using backpropagation, and the weights are updated accordingly. This process is repeated over many epochs until the model converges to a solution that performs well on the training data without overfitting. During training, the model's performance on the validation set is continuously monitored. If the validation error starts to increase while the training error decreases, it indicates overfitting. Once the model achieves satisfactory performance on the validation set, it is tested on the held-out test set to ensure it can generalize to new data. Test performance of an ML model involves checking the prediction accuracy on new data that the model has never seen before. The choice of evaluation metrics depends on the few measures. For regression problems, such as predicting energy or density or atomic forces, common metrics include mean absolute error (MAE), root mean squared error (RMSE), and R^2 score of a given ML model. If the model performs well on the test set, it is considered to have good generalization capabilities. However, in real-world implementation, additional robustness checks, such as cross-validation, uncertainty quantification, or extrapolation tests, may be necessary to ensure reliability across diverse inputs.

In summary, machine learning serves as an advanced tool for learning complex patterns from data, especially in domains where traditional simulations are computationally expensive. Neural networks, as a flexible function, can model nonlinear relationships between input data and the target properties. The process of training, validation, and test sets ensures that models learn effectively and generalize well. The advantages of machine learning lie primarily in its speed, scalability, and flexibility. Once trained, ML models can make predictions orders of magnitude faster than traditional computational methods, enabling high-throughput screening or real-time decision-making. They excel at capturing complex, nonlinear relationships without requiring explicit physical models, which can be invaluable when the underlying equations are unknown or expensive to solve. Moreover, ML can integrate data sources—experimental measurements, simulation outputs, and theoretical calculations—into a unified predictive framework. However, these strengths come with some downsides. ML models depend heavily on the quality and quantity of training data, and they may fail to predict accurately beyond the domain represented in the dataset. They can also behave as "black boxes," providing a small amount of insight compared to explicit physical models. Overfitting, bias from unbalanced datasets, and the computational cost of generating high-quality training data are also challenging. Compared with other computational approaches, such as first-principles simulations (DFT) or empirical force fields, machine learning offers a standard ground, where it provides a balance between accuracy and scalability. First-principles methods are highly accurate and physically interpretable but computationally expensive, limiting their use in large-scale simulations.

2.3 Jacobi-Legendre Descriptor Based Density Model

In this section, our main aim is to discuss some methodological aspects of the Jacobi-Legendre charge density model (JLCDM) (113, 160, 161). As outlined before, this JLCDM model was implemented to predict converged real-space DFT charge density, at a fraction of DFT computational cost. This is achieved through a scalar-field adapted-JL approach, utilizing a cluster expansion of a grid-point representation of the electron density. The electron charge density, $\rho(\mathbf{r})$ at any gird point r_g can be represented in terms of many-body contributions as

$$\rho(\mathbf{r_g}) = \rho^{(1)}(\mathbf{r}_g) + \rho^{(2)}(\mathbf{r}_g) + \rho^{(3)}(\mathbf{r}_g) + ... + \rho^{(n)}(\mathbf{r}_g) \tag{41}$$

where ρ^n stands for n^{th} body term of the series. Thus, $\rho^{(1)}$ represents individual atomic contributions ($one - body$ term) to the density at each grid point $\mathbf{r}_g$, $\rho^{(2)}$ is for each set atom pairs (*two − body* term), $\rho^{(3)}$ is responsible for atom triplets (*three − body* term), etc. Considering the definition of all the above terms Eq.(41) can take the following form

$$\rho(r_g) = \sum_i \rho_i^{(1)}(\mathbf{r}_g) + \sum_{i \neq j} \rho_{ij}^{(2)}(\mathbf{r}_g) + \sum_{\substack{i \neq j \\ i \neq k \\ j \neq k}} \rho_{ijk}^{(3)}(\mathbf{r}_g) + ... \tag{42}$$

here the indexes $i, j, k...$ run over the neighbouring atoms with in a cutoff distance r_{cut} from the grid point r_g The atomic configurations required for each term in the expansion are represented using a local descriptor, referred to here as a "fingerprint." These fingerprints must satisfy the following properties: (i) invariance under translations, (ii) invariance under global rotations of the atomic positions relative to the grid point, (iii) invariance under changes in the coordinate system, and (iv) invariance under permutations of atomic indices.

However, the one-body, two-body, and three-body terms can be expanded to include different polynomials. For instance, using the distance between the grid points and atomic neighbour, from Eq.(42) the one-body term can be written as

$$\rho_i^1(\mathbf{r}_g) = \sum_{n=1}^{n_{max}} a_n^{z_i} \tilde{P}_n^{(\alpha,\beta)} \left(cos \left[\pi \frac{r_{ig} - r_{min}}{r_{cut} - r_{min}} \right] \right) \tag{43}$$

Where the indexes i run over the neighboring atoms within a cutoff distance r_{cut} from the grid point $\mathbf{r}_g$, $a_n^{z_i}$ is the expansion coefficient and

$$\tilde{P}_n^{(\alpha,\beta)}(x) = \begin{cases} P_n^{(\alpha,\beta)}(x) - P_n^{(\alpha,\beta)}(-1), & \text{for } -1 \leq x \leq 1 \\ 0, & \text{for x} \leq -1 \end{cases}$$

where $P_n^{(\alpha,\beta)}$ is the **Jacobi polynomial** (162) of the order n. Furthermore, $r_{ig} = |\mathbf{r}_i - \mathbf{r}_g|$ encodes the distance between the grid point $\mathbf{r}_g$ and the position of i^{th} atom at $\mathbf{r}_i$, r_{cut} is the cutoff radius. r_{min} refers minimum possible distance that varies in the range of

$(-\infty, r_{cut})$, for our case we set $r_{min} = 0.0$. Here, α and β are accountable for the shape of the polynomials with always finite values $\alpha, \beta > -1$. However, different values of α, β yields different types of polynomials, like $(\alpha, \beta) = (0, 0)$ represents Legendre polynomial, etc. It is established that to make this approach general leave the choice of (α, β) depending on the hyperparameter optimization. Similarly, following the Eq.(42) the two-body contribution, $\rho_i^{(2)}$ can also be written as a function of the distance between two points r_{ig}, r_{jg}, and the cosine of the angle situated at point g by the unit vector $\hat{\mathbf{r}}_{ig}$ and $\hat{\mathbf{r}}_{jg}$.

$$\rho_{ij}^{2}(\mathbf{r}_g) = \sum_{n_1,n_2=1}^{n_{\max}} \sum_{l=0}^{l_{\max}} a_{n_1,n_2,l}^{z_i,z_j} \tilde{P}_{n_1 ig}^{(\alpha,\beta)} \tilde{P}_{n_2 jg}^{(\alpha,\beta)} \tilde{Q}_l^{ijg} \tag{44}$$

Here $\tilde{Q}_l = Q_l(\hat{\mathbf{r}}_{ig} \cdot \hat{\mathbf{r}}_{jg})$, Q_l is the **Legendre polynomials** (162), evaluated on the scaler product between $\hat{\mathbf{r}}_{ig}$ and $\hat{\mathbf{r}}_{jg}$, as it naturally describes the angular dependence in spherical harmonics for expanding the scalar products between two real space versos in three dimensions. Here, $\hat{r}_{pg} = \frac{\mathbf{r}_p - \mathbf{r}_g}{r_{pg}}$, and l defines the Legendre expansion degree with sum running in the interval $[0, l_{max}]$. The expansion coefficient $a_{n_1,n_2,l}^{z_i,z_j}$, as it is for the individual atomic contributions, here those coefficients depend on the pair of atomic species considered. Thus, in a similar manner, the contribution for an atom-triple $\rho_i^{(3)}$ can be derived using the same recipe. Recent publications (113, 160) provide further details and implementation in the JL scheme for each expansion. Finally, adding all these components together provides a linear representation of electron charge density in terms of expansion coefficients. Therefore, employing linear regression allows us to learn the charge density. The final expression for charge density to be learned has been detailed in (113). Now, the JL-Charge density Model (JLCDM) symbolically can be written as the scalar product of two terms

$$\rho(\mathbf{r}_g) = \sum_{v}^{n_{body}} a^{(v)} \cdot J^{(v)} \tag{45}$$

where a is the expansion coefficient of the body-order vectors and J contains all the descriptors. The linear nature of this model enables a small number of expansion coefficients to fit, in comparison with the Neural-Network (NN) based approaches, which demand larger amounts of weights. As the final step, the expansion coefficients a can be obtained by minimizing a loss function, primarily the similar strategy utilized in the recent work (113). All the grid points defining the charge density can be used as input data.

It is crucial to mention that we have employed the Ridge regression linear model from the scikit-learn library (163) to fit the real-space density. This includes Linear least squares with l_2 regularization. It solves the matrix equation

$$\hat{\beta} = (X^T X + \alpha I)^{-1} X^T y \tag{46}$$

to determine the coefficient $(\hat{\beta})$ values, nevertheless, this is the expansion coefficient (a^v) for the JLCDM model. Where X is the matrix containing all feature vectors, which is, in this case, all the descriptors, α is the regularization parameter, for this study it was set to zero $(\alpha = 0.0)$, which means no regularization or ordinary least square (OLS) regression fitting, and I is the identity matrix. Finally, y is the target variable and β is the coefficient to determine. So, finally, the equation is

$$\hat{\beta} = (X^T X)^{-1} X^T y \tag{47}$$

If X is ill-conditioned, as it is in this case, computing $(X^T X)^{-1}$ term directly is unstable. Then the algorithm suggested us singular value decomposition (SVD) scheme as it is implemented in the scikit-learn library (163). Using the SVD strategy the linear regression problem can be computed by using the pseudo-inverse as

$$\hat{B} = X^+ y \tag{48}$$

where X^+ is the pseudo-inverse of X, obtained using the SVD scheme.

Since the electron density is defined in the grid, its size significantly varies with the choice of molecular systems and basis functions employed. As the system sizes increase and more orbitals are included in the basis, the number of grid points also increases. This leads to a considerable increase in computational overheads due to the large grid-point mesh. Indeed, the strategy used for sampling the grid points will be crucial for the performance of the model, as will be discussed in the next section.

3 | Results and Discussion: Prediction of Density Matrix

This chapter presents the key findings of this thesis and the associated discussions on predicting the density matrix (DM) using a deep learning-based framework to accelerate the self-consistent (SCF) process in DFT calculations. In this approach, the predicted DM serves as the initial guess density for a few molecules. As mentioned before, our main goal is to implement an ML-predicted DM as the initial guesses for DFT calculations, where the ML model utilizes only chemical and structural information about the molecular systems. All the electronic structure calculations conducted in this study to produce the training data and the reference results have been generated with PySCF (1, 2). Despite the success of DFT, the method faces significant challenges, specifically for systems requiring high computational accuracy. The iterative nature of the SCF process limits its scalability, presenting challenges for applications involving large molecular systems or real-time dynamics. In principle, as described earlier, to solve the eigenvalue equation appearing in DFT, we need to provide an initial guess for the DM. Then, the KS-equation we solved through an iterative process until it reached convergence (global minimum for the energy) or the stationary point. Beyond this stationary point, the electron density for a given structure will not change (164). For a larger or more complex system, the SCF scheme often experiences difficulties in converging. As a matter of fact, in our study, the SCF approach encountered convergence issues for one metallic molecule. In the context of self-consistent KS-DFT, a crucial step within the SCF scheme involves the diagonalization of the KS-Hamiltonian. Diagonalization yields the coefficients of the DFT orbitals along with the associated ground-state energies. The electron density is the key quantity in DFT and is subsequently constructed using the converged KS-DFT orbitals or their coefficients. So, the quality of the initial DM guess plays a critical role in accelerating the convergence of the self-consistent field (SCF) procedure. Orbital optimization becomes significantly more efficient when the initial orbital guess is closer to the converged solution. This can be stated in another way: a more accurate initial guess of DM demands fewer SCF iterations to converge DFT calculations. In this work, the performance of our ML-predicted DM will be evaluated against other conventional DM initial guesses; therefore, it is important to introduce them here. Several strategies can be employed to generate the initial guess, each with its strengths and limitations. One common approach is to use the superposition of atomic densities, in PySCF it is called ′minao′ initial guess (3, 4). For this technique, the guess solution is obtained by projecting the minimal basis of the first contracted function in the cc-pVTZ or cc-PVTZ-pp basis set onto the orbital basis function and then forming the DM. The guess orbitals are computed by diagonalizing the Fock matrix that arises from the spin-restricted guess DM. An alternative method is the one-electron (1e) guess (5), derived from the diagonalization of the core Hamiltonian ($H_0 = T + V$), where T represents the kinetic energy and V is the nuclear attraction potential. This method neglects electron-electron interactions and omits the significant screening effects of core electrons on the nuclear charge. Consequently, it lacks adaptability for capturing the shell structure of atoms accurately. Another widely used method is the atomic guess (6), called ′atom′ as this is PySCF named, where the density matrix is constructed by superposing atomic Hartree–Fock (HF) density matrices. These atomic HF calculations are spin-restricted, employing spherically averaged fractional occupations. The ground-state configurations are determined through fully numerical calculations at the complete basis set limit. This approach provides a robust initial guess for molecular systems, leveraging high-quality atomic data to improve the convergence. This specific initial guess for the DM is implemented in PySCF to perform DFT calculations only. Another approach is the Huckel method, which generates on-the-fly atomic Hartree-Fock calculations that are performed analogously to the 'atom' option. Initially designed to describe only valence orbitals, the method has been extended to include core orbitals in later adaptations. The extended Huckel approach is relatively simple to implement and is applicable to both all-electron and effective core potential calculations. This makes it a versatile choice in computational chemistry workflows. Each of these initial guess methods has its place in quantum chemistry, and their effectiveness depends on the specific requirements of the calculation and the type of system to investigate. Here, we use a dense neural network (NN) model together with global structural descriptors to generate an initial DM, where the model predicts independent matrix elements of the DM. NN architectures, various types of descriptors and the composition of the training and test sets characterize the models. All of these will be detailed here.

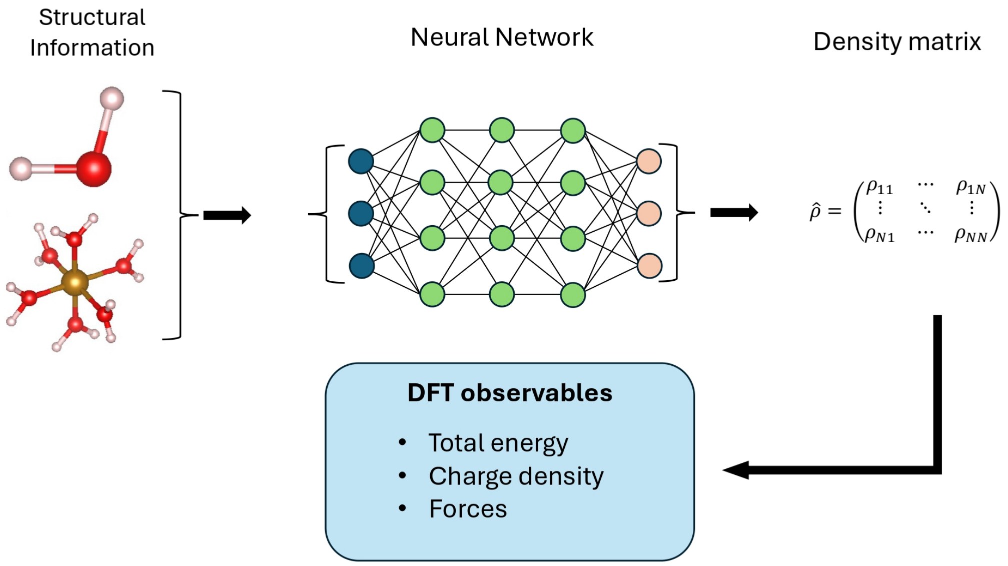

Figure 3.1: In this figure, we show the workflow for the neural network predicted density matrix. The neural network model receives inputs from the molecular structures, and then it predicts the DM. Further, those predicted DMs are utilized as the initial guess for DFT calculations. As a result, we obtain various DFT observables, total energy, forces, etc.

Here, Fig.3.1 schematically presents the workflow implemented to predict the DM using NN for various molecules. This workflow comprises four key components: firstly, the structural information of various molecules is provided; next, this information is used as inputs for an NN model, which has been trained on DFT converged DM data; subsequently, the NN model predicts the DM, which means DM is the output of the NN model; finally those predicted DM implemented into PySCF used as the initial guesses to perform DFT calculations for a given molecule. Although efficient in accelerating convergence, this DM approach is not transferable, since the density matrix is rotationally contravariant. This means that its value changes when the rotations are applied to the molecule. The rotation of the coordinate systems changes the density matrix. Moreover, the DM depends on the choice of basis functions, which are primarily atom-centred and may not hold the same symmetry as the system itself. For example, if a DM expanded over a local orbital basis like Gaussian-type orbitals, then a rotation can mix those basis functions in a different way, yielding a different DM with different elements. This is an intrinsic property of the DMs that limits the flexibility and applicability of the machine learning-based DM workflow. In addition, the size of DMs scales quadratically with the number of basis functions used $[\mathcal{O}(N^2)]$, leading to the more computationally extensive tasks for obtaining the ML DM predictions as the system size increases.

3.1 Machine Learning Methods for DM Predictions

A key component in the success of ML models, particularly in the scientific domain, is the choice of descriptors (112, 113, 125, 128). Descriptors are numerical representations of the chemical system that capture essential features relevant to the prediction task. In the context of molecular modelling, descriptors can include atomic numbers, bond lengths, bond angles, charges, or more sophisticated representations. The goal of a good descriptor is to represent the physical system in a way that is invariant to translation, rotation, and permutation of atoms, while still being unique enough to distinguish between different atoms with the same or different chemical species. The better the descriptors are able to capture the underlying physics or chemistry of the system, the more accurately the model can learn and generalize. Descriptors bridge the gap between raw scientific data and the machine learning models. For instance,

Table 3.1: Table summarizing the structure and performance of the final neural networks trained for the three molecules. Here we report the number of features defining the input, N_{in}, the dimension of the DM, D_{DM}, the number of the network hidden layers, N_{hi}, the total number of neurons forming each hidden layer, N_{nu}, and the total number of weights, N_{w}. Then, we report the mean absolute error (MAE), the largest error on the matrix elements, $\delta\rho_{\text{max}}$, the root-mean-square error (RMSE), and the R^2 coefficient of the DMs. All errors refer to the test sets and they are in atomic units (a.u).

Molecule	N_{in}	D_{DM}	N_{hi}	N_{nu}	(N_{w})	MAE	$\delta\rho_{\text{max}}$	RMSE	R^2
H_2O	3	24×24	2	$18, 32$	662	0.0002	0.0057	0.0003	0.9999
S_2O	3	50×50	2	$18, 28$	583	0.0002	0.0112	0.0003	0.9999
$[Fe(H_2O)_6]^{2+}$	6	187×187	2	$16, 32$	640	0.0002	0.0283	0.0005	0.9993

in predicting properties like total energy or forces, a neural network does not operate directly on atomic coordinates. Instead, the coordinates are processed into descriptors that map local environments of atoms, which are then used as input to the model. In some advanced approaches, such as graph neural networks (GNNs) (79, 112). In this study, the initial guess for the DM is predicted utilizing an NN model, where the input features are the independent Cartesian coordinates of the molecules. In Table-3.1, we summarized the test set errors corresponding to the used NN model for each molecule, associated with ML-predicted DM elements. By utilizing Cartesian coordinates with a dense neural network, we effectively map an equivariant DM using an invariant model. To address this problem, we manually remove the rotational and translational degrees of freedom from the molecules by fixing them in the 2D plane, ensuring invariance. It is true that this approach is not universally applicable, but it serves as a practical solution for our preliminary investigation. A more robust and elegant solution would involve employing a fully equivariant representation of the molecular structures (165); however, this would introduce substantial complexity, which we aim to avoid at this stage.

The neural network architecture was optimized by varying the number and size of hidden layers to minimize the mean absolute error (MAE). The optimal configuration of the NN for each molecule is detailed in Table [3.1]. We can say, our NN architecture, which can naturally deal with molecular structures, and the approach yields high accuracy (Table [3.1]) with MAE $\approx 10^{-4}$ a.u in DM elements. In all cases, the exponential linear unit (ELU) activation function was employed. It is known, that one can exactly specify the output dimension of the NN-model as per the requirement. In this study, the output of our NN model is the density matrix, whose dimension typically depends on the choice of basis sets. In Fig.(3.2), we show the learning curves associated with the performance of NN models for training and the validation data for three molecules. Panel (a) is for H_2O, panel in the middle (b) represents the same for S_2O, and panel (c) for $[Fe(H_2O)_6]^{2+}$. We can say from these plots that for three molecules, errors are pretty low with respect to the fitting epoch, when the training and validation have been performed. Furthermore, it is also true that we can see from Fig.(3.2) that more fluctuation is recorded in the validation set, but the order of magnitude is lower than 10^{-4}. The hyperparameters were set through manual empirical tuning. We tried various configurations of learning rate, hidden layer sizes, and activation functions were checked iteratively, and the settings that yielded the lowest validation loss while maintaining training stability were utilized. To be specific, the Adam optimizer (166) with its default learning rate has been considered, as it provided stable convergence. The hidden layer sizes are already mentioned in Table-3.1. L2 regularization was tested with various strengths but was finally set to zero, as no performance improvement was observed.

Since the DM is a square matrix and its upper and lower triangles are related by transposition ($P_{\mu\nu} = P_{\nu\mu}$), instead of predicting the entire DM, we can either predict the upper or the lower triangular portion. Let us take an example. From Table-3.1, we can see the dimension of the DM of H_2O, is (24×24). Now, instead of predicting 576 DM elements, we can only predict 276 elements with 24 diagonal values. After doing this, we can easily reconstruct its original dimension before treating it as an initial guess of DM for DFT calculations. We have implemented this approach for NN-based DM prediction for all the molecules studied here. In addition, to reduce the number of elements to predict further we precisely investigated which elements in the DM are relevant to initiate a DFT calculation. In other words, we can establish which matrix elements are larger in magnitude ($\sim 10^{-3}$) inside a density matrix for a given

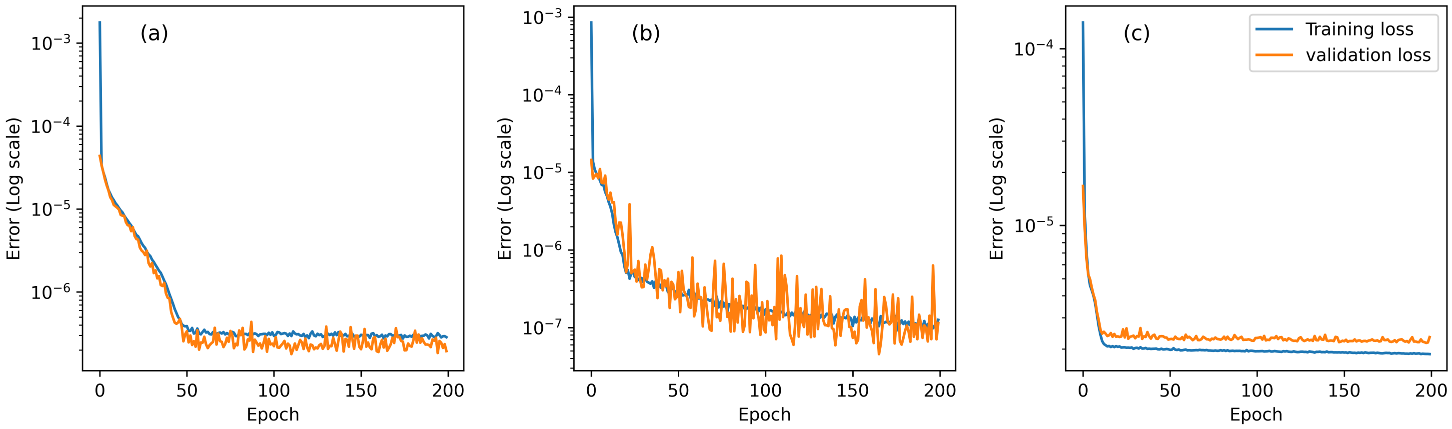

Figure 3.2: In this figure, we show Learning curves for NN models. Plot (a) represents the learning pattern of the ML model, employed to train the H_2O molecule, plot (b) is the same for the S_2O molecule, and finally the plot (c) shows the learning curve for $[Fe(H_2O)_6]^{2+}$ molecule, for the training and validation loss. Along the x-axis, these graphs plot epoch (iteration steps) during the NN training, and the y-axis addresses the values for the corresponding loss. Note that the loss values are on the logarithmic scale.

molecule. This protocol can also be applied in either the upper or lower triangle of the DM. A diagram-based example for the S_2O molecule is presented in Fig.-3.3. It showcases the orbital-wise elements of a density matrix for S_2O. In both panels, the x and y axes contain the various atomic orbitals coming from the used basis function. The magnitude of the DM elements is represented in colour codes. We have used only the lower triangle's white and light blue parts to train the DM [(b)]. Again from Table-

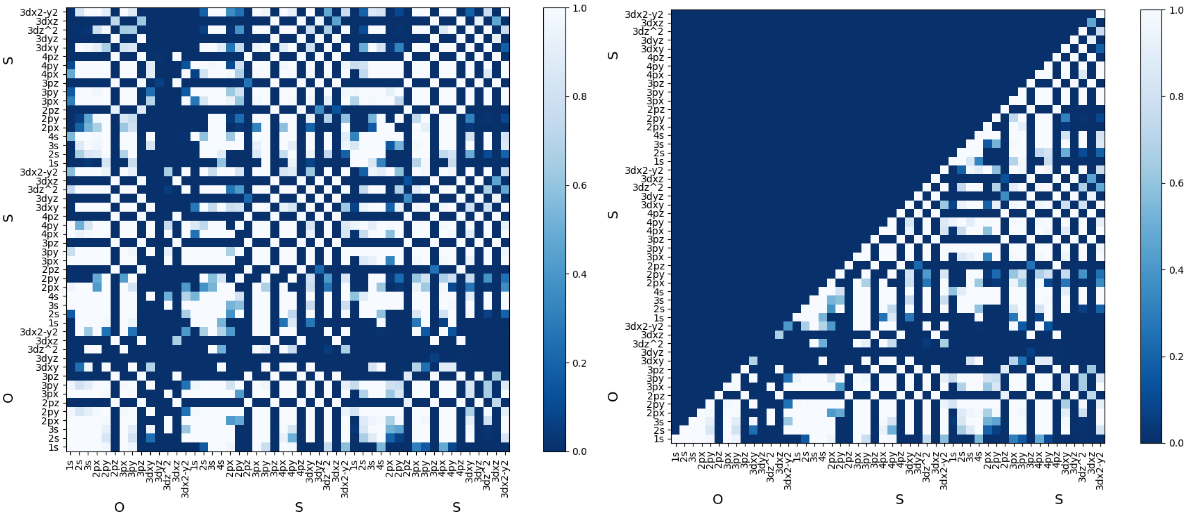

Figure 3.3: Left Side (a): Orbital-wise impactful elements inside a converged DM. Right Side (b): Lower triangular part of that DM, for a given atomic position for the S_2O molecule. This analysis has been done to eliminate small values from the DM

3.1, it can be noticed that for the S_2O molecule, the dimension of a DM is (50×50), which means 2500 elements for a given geometry, but by applying this IOB approach, that size has been reduced to 800 elements. Finally, to preserve its original dimension and to construct the initial guesses of DM, those blue spots were filled with 0.

For the context of DFT, the difference between an invariant and equivariant model becomes important when the context involves DM, and how it behaves under the symmetry operation, such as translations and rotations defined for a molecular system. DM is the fundamental object for electronic structure calculations; it encodes electron density for a given system, which is the key observable for DFT, and from DM one can also compute dipole moments, atomic forces and other properties. Once a rigid transformation (say a rotation or translation) is applied to the atomic coordinates of a molecule, the physical meaning of such a system will not change; it is simply being described in a different coordinate representation. Therefore, any ML model predicting the DM must hold those symmetries. An invariant model would be one where the prediction of scalar observables derived from the DM remains unchanged under such transformations. For instance, the total electron density integrated over space, or the total energy computed from the DM, also the bond lengths and bond angles should not depend on whether the molecule is rotated or translated; these are invariant properties. In contrast, an equivariant model comes into play when the outputs are themselves geometric quantities that transform in a well-defined way under coordinate changes. The DM itself, though often expressed in a basis of atomic orbitals, transforms covariantly under rotations of the underlying basis or molecular geometry. Mathematically, for a transformation T for a model f. invariance means $f(Tx) = f(x)$, whereas equivariance means $f(Tx) = Tf(x)$. Similarly, vector quantities derived from the DM, like the dipole moment or forces on atoms, must rotate consistently with the molecule. That is, if the molecule is rotated, the forces predicted from the DM must also rotate in the same way; this is the essence of equivariance. Thus, when designing ML models to predict DMs in DFT workflows, one must ensure that the model respects equivariance for vector quantities outputs and invariance for scalar quantities. By doing so, the learned ML model preserves the same physical symmetries as the underlying quantum system, leading to results that are not only accurate but also physically consistent.

3.2 Preparation of the Datasets for DM

The PySCF package implements all-electron DFT formalism with several quantum-chemistry methods, such as Hartree-Fock over the Gaussian basis set, GW approximation, coupled-cluster theory, etc. For our NN-based DM approach, we have used the cc-pVDZ basis (167) as it is implemented in PySCF, which utilizes double-ζ polarized orbitals for the valence electron. For this study, we have considered three different molecules, namely, H_2O, S_2O, and $[Fe(H_2O)_6]^{2+}$. The given basis set for the relevant chemical species contains 5 basis functions for H, 14 for O, 18 for S, and 43 for Fe, the molecular structures are shown in Fig -3.4. Among those molecules, H_2O has a simple electronic structure and typically does not experience convergence issues. Since its electron charge density closely resembles a superposition of atomic densities. For a strict test, we consider S_2O. Notably, both H_2O and S_2O can be described utilizing only three structural features, two bond lengths and one bond angle. As a further test, we analyze one more complex structure, $[Fe(H_2O)_6]^{2+}$, which also allows us to examine our ML DM approach on a system with metal bonding. All DFT calculations (all electrons) are performed using the BLYP functional, which combines Becke's (28) generalized gradient approximation for the exchange energy with the Lee–Yang–Parr correlation energy (29), as implemented in the LIBXC library (152) via PySCF. For creating the training, validation, and test sets, the SCF cycle is converged using the DIIS (44, 45) scheme for H_2O and S_2O. However, for $[Fe(H_2O)_6]^{2+}$, a second-order solver (SOS) (51, 52) is employed due its to more challenging convergence behaviour, because small HOMO-LUMO gaps or near-degenerate states (such as transition metal complexes) can suffer from slow convergence or even divergence, where the second-order methods stabilize the SCF process by better predicting the direction and step size needed to reach self-consistency. This was the case for $[Fe(H_2O)_6]^{2+}$.

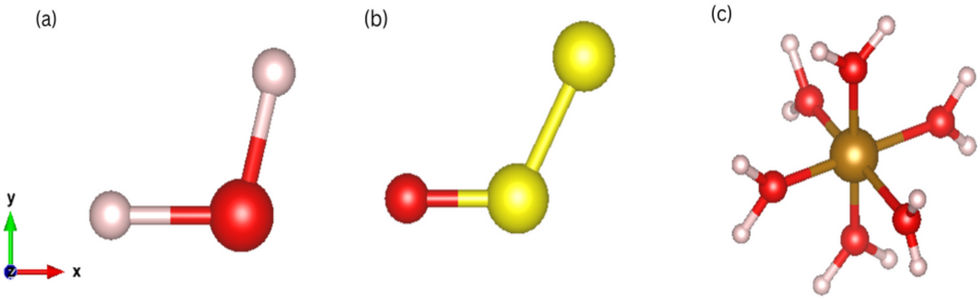

Figure 3.4: The molecules investigated in this study were placed in their equilibrium positions along the Cartesian axes: (a) H_2O, (b) S_2O, and (c) $[Fe(H_2O)_6]^{2+}$. Color code: H = white, O = red, Fe = dark golden, S = yellow, x-axis = red, y-axis = green, z-axis = blue.

Additionally, when analyzing the convergence history of different initial density matrices (DM) guesses, both DIIS and SOS SCF algorithms are considered. To eliminate the translational degrees of freedom, we fix a specific atom at the origin—oxygen for H_2O, the central sulfur for S_2O, and the Fe^{2+} cation for $[Fe(H_2O)_6]^{2+}$. Rotational degrees of freedom are addressed by applying suitable constraints. For the triatomic molecules H_2O and S_2O, we place one atom along the negative x-axis and a second atom within the x–y plane, thus reducing the description to three independent coordinates. While for $[Fe(H_2O)_6]^{2+}$, we restrict the oxygen atoms of the H_2O ligands to lie along the three Cartesian axes and consider only variations in the $Fe - O$ bond lengths, treating the water molecules as rigid units. This alignment gives six independent coordinates for the system (see Fig.-3.4). Since we performed the molecular dynamics simulation with ML-predicted DM guess for H_2O and S_2O molecules, the rotation operator has been applied to the new geometries obtained from molecular dynamics steps, and the inverse-rotational operator on the ML DM guess based DFT-inter-atomic forces for the next molecular dynamics steps to keep the molecules fixed in xy-plane during the simulations. For H_2O and S_2O, these configurations were derived from ab initio Born–Oppenheimer molecular dynamics (BOMD) trajectories conducted at 150 K, utilizing PySCF. Specifically, simulations were run for 117 ps and 130 ps for H_2O and S_2O, respectively, using the **Nosé-Hoover** thermostat (168) via the pyLAMMPS API, as implemented in the LAMMPS package (7). The datasets for the model training consist of 9,000 configurations for training, 800 for validation, and 1,000 for testing. For

$[Fe(H_2O)_6]^{2+}$, we introduce random restricted displacements of the H_2O molecules, allowing the Fe–O bond length to vary within 10% of its equilibrium value (2.0525 Å). This approach yields training-set mean absolute errors (MAEs) on the order of 10^{-3} atomic units (au). Notably, the largest elements of the DM are primarily found along the diagonal, reaching values close to unity, while many off-diagonal elements remain small. For example, in the case of H_2O, 8.15% of the DM elements (P) have magnitudes in the range $0.1 < |P| < 1$, 43.75% fall within $10^{-3} < |P| < 0.1$, and 48.1% have magnitude less than 10^{-3}.

3.3 Results

In the results section, we will present the accurate outcomes of the NN predictions, which are DM. Next, we will see how this ML DM approach is responsible for accelerating the SCF scheme for DFT calculations, including in terms of energy and density convergence, in comparison with other DM guesses. This yields the one SCF iteration with ML DM guess that is capable of maintaining DFT quality results. Further, we also investigated the quality of atomic forces coming from the ML DM guess just after applying one SCF iteration. Finally, this thesis shows the comparison of the MD simulations, as we deployed the workflow (Fig.[3.1]) to compute the simulation trajectories through ML DM based non-self-consistent (one SCF) and AIMD schemes.

3.3.1 Parity Plots for NN Predicted DM

Fig-3.5 presents the parity plots in logarithmic scale for the DM elements of H_2O [see panels (a) & (d)], similarly for the S_2O molecule [panels (b) & (e)], and finally for $[Fe(H_2O)_6]^{2+}$ [panels (c) & (f)]. The upper panels are associated with the training set, while the lower panels represent the test set, with the mean absolute error (MAE) values reported in each case. As shown, there is an excellent agreement between the ML-predicted DM and the fully converged DFT DM, with most points closely aligned along the parity line. This is true for matrix elements greater than approximately 10^{-2},

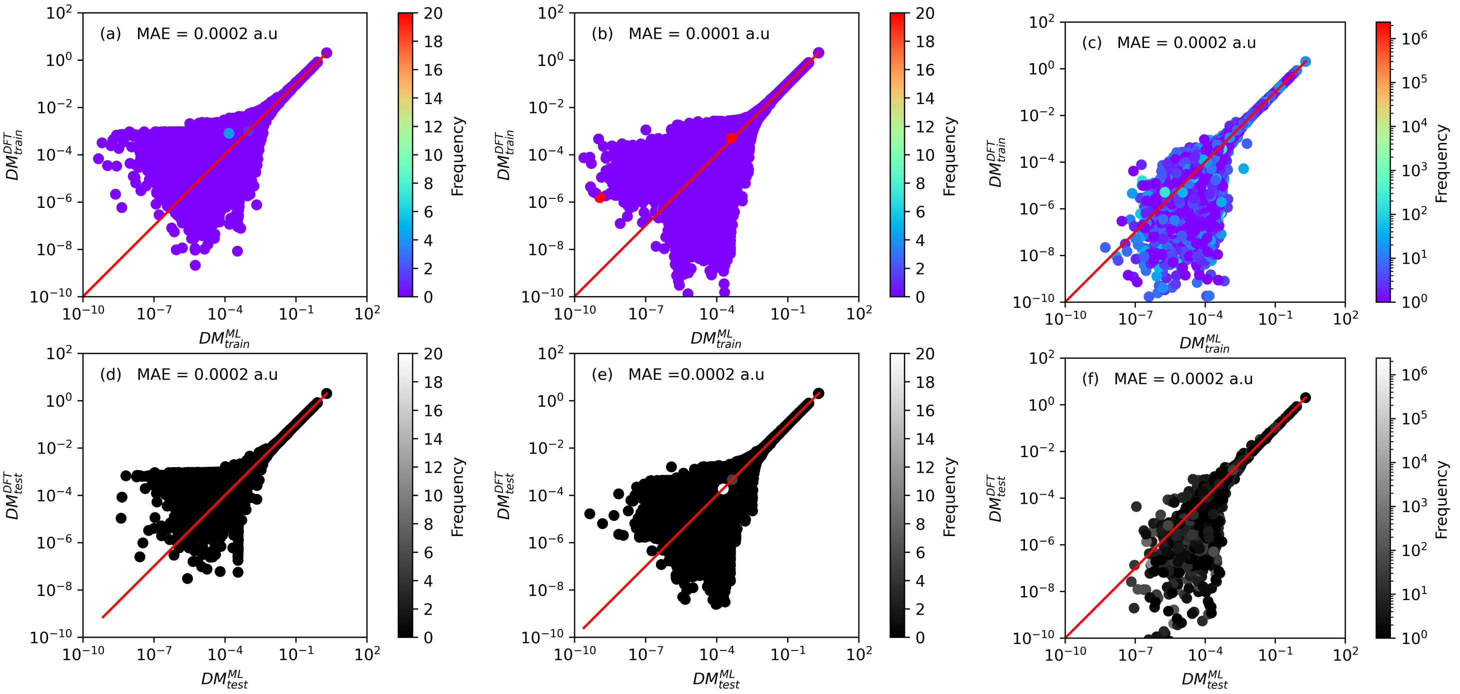

Figure 3.5: Parity plots for H_2O [panels (a) and (d)], S_2O [panels (b) and (e)], and $[Fe(H_2O)_6]^{2+}$ [panels (c) and (f)]. The upper panels are for the training set and the lower ones for the test set. Each graph also reports the MAE achieved. Note that all the parity plots are in logarithmic scale (we plot $|\rho_{ij}|$) and that deviations are only found for the smaller matrix elements. The colour code describes the density of the given DM matrix-element values.

primarily for those that significantly influence the key observables, such as the total ground state energy and atomic forces. However, significant relative errors are observed for smaller matrix elements, where the DFT data tend to be noisy, making it difficult for the model to learn effectively. This error distribution aligns with the low MAE values reported across all cases, as detailed in Table-3.1. An exception is noticed for a few points in the training set of $[Fe(H_2O)_6]^{2+}$, where more pronounced deviations are reported. These deviations indicate cases, where the DFT calculations itself struggled to converge. In such instances, the DM used in the training was obtained after the maximum number of default self-consistent iterations allowed [50] in PySCF, rather than being fully converged calculations. These problematic training points were subsequently discarded. In contrast, the test set includes only fully converged structures. It is important to mention that the numerical construction of the DM with our ML scheme does not strictly enforce the idempotency. Since this property enforces a nonlinear relation between matrix elements, this inclusion into the NN model is challenging. However, this limitation is mitigated by the high numerical accuracy achieved, as demonstrated in the subsequent sections.

3.3.2 Accuracy of ML-DM as the Initial Guess for SCF

We now evaluate the accuracy of the DM predicted by the NN model, when used as a starting point for DFT calculations. This test is conducted over 1000 new configurations for each molecule. Fig-3.6 depicts the average number of SCF iterations required to reach the convergence and their corresponding variance. Convergence here is defined as achieving an energy difference between two successive SCF iterations below 10^{-9} (Ha), establishing a stringent criterion.

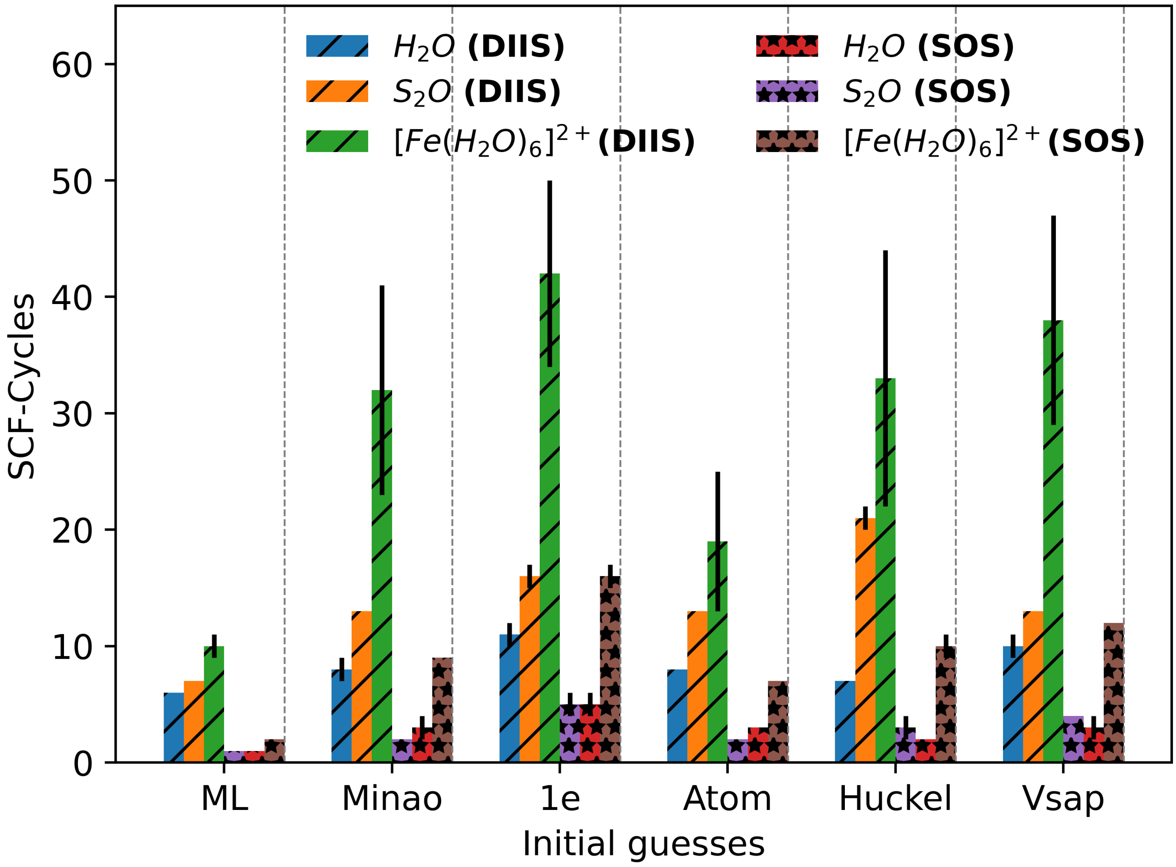

Figure 3.6: Average total number of SCF steps taken to achieve convergence for various initial DMs and mixing schemes. Here the convergence criterion is on the total energy between two successive SCF iterations that should be lower than 10^{-9} Ha. The black lines around the mean indicate the variance. Variances lower than one iteration are not depicted.

Two different sets of calculations are performed for this investigation, with the SCF steps driven by either the DIIS algorithm or the SOS mixing scheme. The convergence parameters in both cases are set to the PySCF default values. Overall, the SOS DM-update strategy proves to be significantly more efficient than the simpler DIIS scheme, reducing the total number of iterations by approximately a factor of three, regardless of the molecule or the initial DM. However, this advantage is partially offset by the SOS approach being more computationally intensive, as each iteration requires more time. Additionally, for $[Fe(H_2O)_6]^{2+}$ using the DIIS solver, there are instances where 50 iterations are insufficient to achieve convergence. This behaviour is somewhat expected given the electronic structure of the $[Fe(H_2O)_6]^{2+}$ cation. Specifically, $[Fe(H_2O)_6]^{2+}$ is a spin-crossover molecule that undergoes a temperature-driven transition from a low-spin to a high-spin state, caused by distortions in the octahedral coordination shell of the Fe^{2+} ion. While this phenomenon is only partially captured by DFT (169), a multi-determinant theory would provide a more accurate description (170). Consequently, it is unsurprising that some highly distorted configurations pose challenges for convergence in DFT calculations. However, determining the exact ground-state calculations for $[Fe(H_2O)_6]^{2+}$ is not the focus here; rather, the molecule represents a test case for "difficult" convergence scenarios. In cases where convergence is not reached, the SCF cycle is terminated after 50 iterations. Irrespective of these differences, the convergence speed concerning various initial DM strategies is fairly consistent across both mixing schemes. Therefore, we primarily focus on the results obtained with the SOS algorithm in our discussion. As expected, the H_2O and S_2O molecules converge significantly faster than $[Fe(H_2O)_6]^{2+}$, which is consistent with their simpler covalent bonding structures. As we discussed earlier, the "1e" option for DM provides the least effective starting point, thus even for the SOS mixing scheme the '1e' DM guess requires 5 iterations for H_2O and S_2O and approximately 16 for $[Fe(H_2O)_6]^{2+}$ to converge. Other conventionally constructed starting DMs show comparable performance, with the covalently bonded molecules converging within 3–4 SCF steps and $[Fe(H_2O)_6]^{2+}$ in approximately 10. In contrast, our ML-predicted DM significantly outperforms all other methods. For H_2O and S_2O, convergence is achieved in a single SCF iteration, while $[Fe(H_2O)_6]^{2+}$ requires only two iterations. This results in a speedup of the SCF cycle by a factor ranging from 3 to 5, depending on the molecule. It is worth mentioning that the acceleration is less pronounced when the DIIS mixing scheme is deployed, particularly for the covalently bonded molecules, where the improvement over other methods is only marginal. This appears to stem from the inefficiency of the DIIS scheme, which requires 6–7 iterations to converge, even when starting with the ML-predicted DM. Although it is not entirely clear why the SOS algorithm provides a greater convergence speedup with ML-generated initial DMs, it is worth mentioning that the DIIS scheme generally may not fully utilize the initial DM guess. Instead, their optimization process can lead the electron density to explore broader regions of the variational space, potentially diminishing the benefit of a high-quality starting point. it is then expected that such limitation penalizes more DMs close to the fully converged one than more inaccurate ones.

3.3.3 Energy and Density Convergence

In this section, we look into the convergence behaviour of the ground state energies and electron density for various initial DM guesses for three molecules, introduced in the previous sections. We start with considering H_2O, then $[Fe(H_2O)_6]^{2+}$ and finally we move to S_2O.

In Fig-3.7 we report the variations in total energy for the ground state as a function of the SCF iteration number, n. Here, for H_2O we have used the DIIS [panel (a)] and SOS [panel (b)] mixing schemes. Additionally, panel (c) displays the norm of the difference between the converged DM P_{ij} and the DM at each SCF step P_{ij}^{n}, for the DIIS mixing scheme for H_2O. While the specifics of each SCF cycle may vary depending on the molecular geometry, the figure represents a typical example. In general, all initial DMs differ to some extent from the final ground-state DM, and the most significant deviations observed, as expected, one for the “1e” initialization (where the total energy difference at $n = 0$ exceeds 8 Ha and is not shown here). Convergence

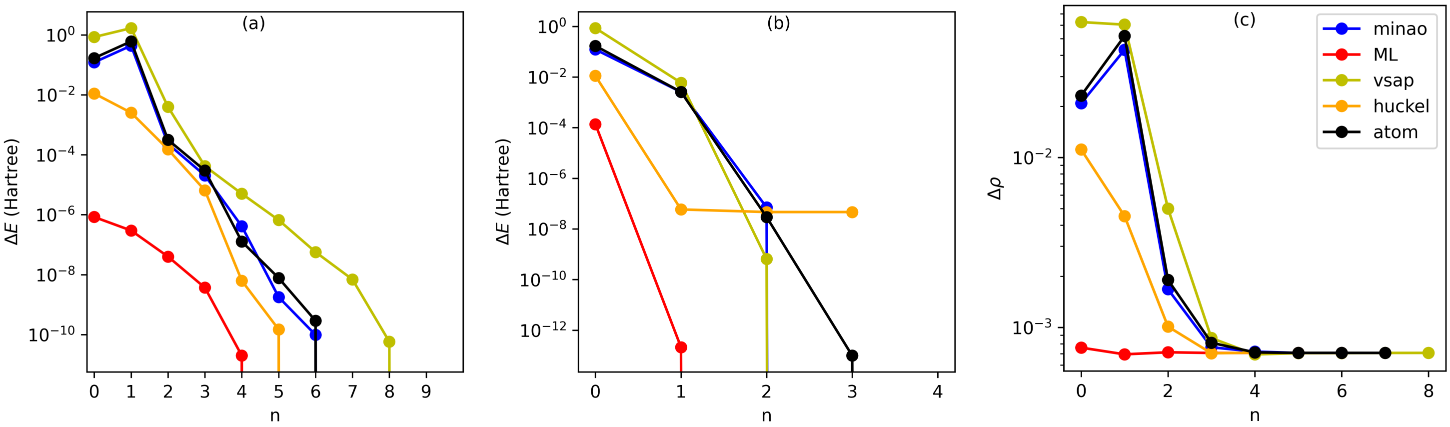

Figure 3.7: Analysis of the SCF cycle for H_2O. In panels (a) and (b) we show the total energy (measured with respect to the ground-state energy) as a function of the iteration number, n, for convergence driven by the DIIS and SOS mixing scheme, respectively. In panel (c) we present the norm of the difference between the ground-state converged DM, ρ^{GS}, and that computed at the n-th iteration, ρ^n. In this case, we follow the DIIS-driven SCF cycle. For ease of visualization in all plots, the y axis is on a logarithmic scale.

is monotonic when the SOS solver is utilized but it can exhibit oscillations with DIIS, accounting for the higher number of iterations typically required by DIIS. Remarkably, the ML-generated DM is exceptionally close to the final ground-state DM, guaranteeing monotonic convergence in all cases. For H_2O, the computed total energies at $n = 0$ are, on average, within 10^{-4} Ha from the converged ground state value, and the percentage variation in the DM after the first iteration is only 0.196%. The percentage variation for any two arbitrary square matrices A and B, which have N number of elements, is as follows

$$\frac{\frac{\sum |A-B|}{N}}{\text{mean}\left[\frac{A+B}{2}\right]} \times 100\% \ ,$$

Here A is our ML density-based DM with one SCF iteration and B is the DM obtained from the fully converged DFT calculation, for the same uracil structure. This indicates that the ML DM provides an excellent agreement with the ground-state DM. For comparison, the next to best initial DM is generated using a restricted Hartree-Fock calculation, as the "huckel" option in PySCF, which yields an initial total energy error of approximately 10^{-2} Ha. In contrast, all other DM initialization approaches produce initial errors exceeding 0.1 Ha.

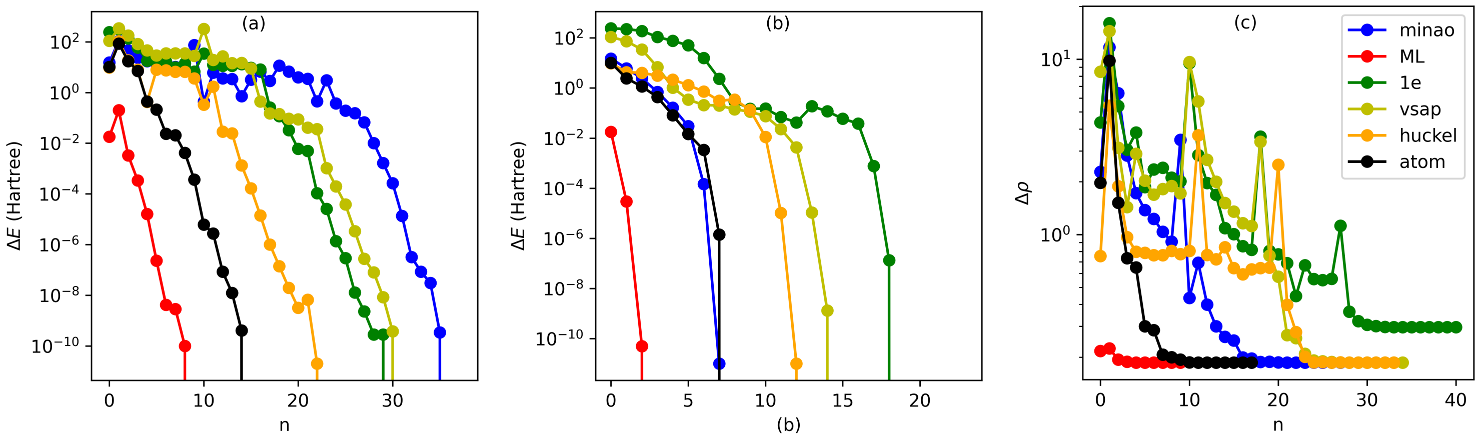

Figure 3.8: Analysis of the SCF cycle for $[Fe(H_2O)_6]^{2+}$. In panels (a) and (b) we show the total energy (measured with respect to the ground-state energy) as a function of the iteration number, n, for convergence driven by the DIIS and SOS mixing scheme, respectively. In panel (c) we present the norm of the difference between the ground-state converged DM, ρ^{GS}, and that computed at the n-th iteration, ρ^n. In this case, we follow the SH: DIIS-driven SCF cycle. For ease of visualization in all plots, the y axis is on a logarithmic scale.

It is clear that the convergence path becomes oscillatory when using the DIIS SCF solver for $[Fe(H_2O)_6]^{2+}$ (see Fig-3.8). In this situation, the ground state energy and the DM fluctuations are much more pronounced, with "spikes" appearing at certain self-consistent steps. These spikes result from fluctuations in the atomic orbital occupations across different SCF iterations. By contrast, the SOS mixing scheme effectively suppresses these large fluctuations, restoring a monotonic progression toward the ground-state solution. Notably for the $[Fe(H_2O)_6]^{2+}$ cation, the ML-predicted DM guess provides much more accurate values, in fact sufficiently accurate to suppress convergence oscillation, regardless of the SCF algorithms. In the first iteration, the ML DM provides the energy, which is about 3×10^{-5} Ha away from the converged result (in percentage it is 1.7×10^{-6}%), while the percentage variation for the DM at the first SCF step is 0.77% with respect to the converged DM.

In Fig-3.9, we present the total energy (with respect to the ground-state energy) as a function of the iteration number, n, for the S_2O molecule, obtained through the DIIS [panel (a)] and SOS [panel (b)] mixing scheme. Furthermore, in panel (c) we present the norm of the difference between the converged DM, and that at each iteration, also

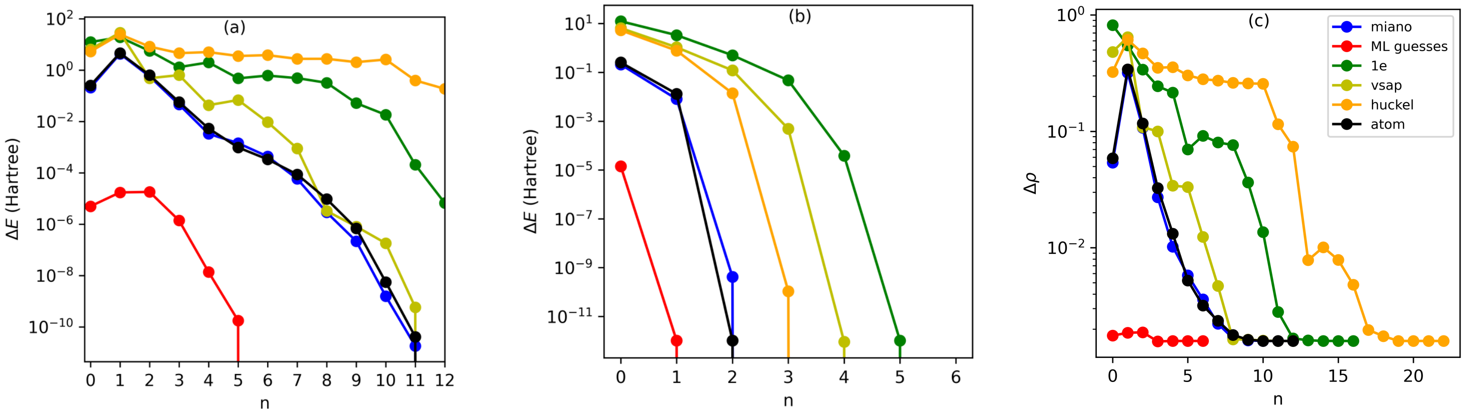

Figure 3.9: Analysis of the SCF cycle for S_2O. In panels (a) and (b) we show the total energy (measured with respect to the ground-state energy) as a function of the iteration number, n, for convergence driven by the DIIS and SOS mixing scheme, respectively. In panel (c) we present the norm of the difference between the ground-state converged DM, P^{GS}, and that computed at the n-th iteration, P^n. In this case, we follow the DIIS-driven SCF cycle. For ease of visualization in all plots, the y-axis is on a logarithmic scale.

implementing the DIIS scheme. Here we can see due to the ML DM guess, the changes either in energy or in DM between $n = 0$ and final SCF steps are significantly smaller, compared to other DM guesses. For example, in the ML DM guess, the difference in energy (ΔE) is $\sim 10^{-7}$ Ha, but for the 'minao' guess it is ~ 0.99 Ha for the DIIS scheme. Whereas for the SOS mixing scheme, (ΔE) for ML-DM guess is $\sim 10^{-6}$ Ha, and for 'minao' it is approximately 0.09 Ha. This indicates ML DM guess is an excellent choice to accelerate the self-consistent process as it is shown for the other two molecules.

3.3.4 Nonself-Consistent Atomic Forces

It has already been examined through the previous sections that our ML DM guess demands an extremely small number of SCF iterations to achieve convergence compared to other conventional initial DM guesses. Even without any SCF cycle, it can already provide an accurate result for the ground-state DFT energy. In this section, we inspect another observable predicted from the ML DM approach, namely the atomic forces. we mainly focus on the forces for a single S_2O molecule and then we summarize the same for H_2O, which provides qualitatively similar results.

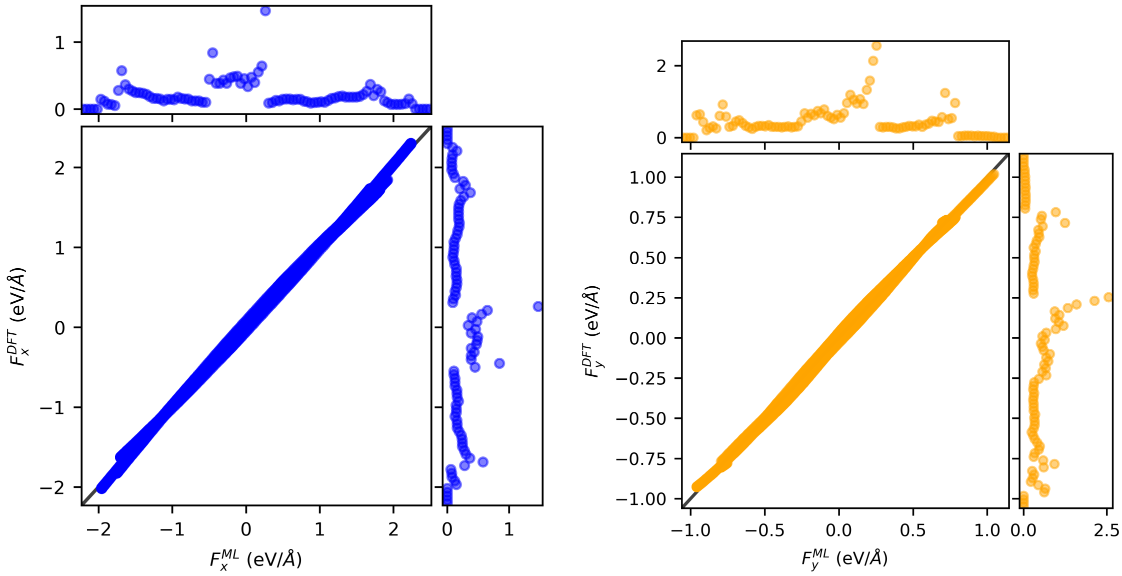

Figure 3.10: Parity plot for the $\alpha = x$ and y components of the atomic forces computed by using the ML DM, F_α^{ML}, with one SCF cycle, against the fully converged DFT ones, F_α^{DFT}. Data are here presented for a set of 1000 S_2O molecules extracted from the same molecular dynamics trajectory used to generate the training set. The upper panel is for the forces x component, while the lower panel is for the y component. The histograms on the side describe the frequency of the forces in the test set.

Fig.-3.10 presents the parity plots for the x component (left panel) and y component (right panel) of atomic forces, for the S_2O molecule placed in the x-y plane. These calculations are performed for a set of 1000 distorted molecules computed from the AIMD trajectory used to create the training set, but not included in the neural network's construction. As the molecules are inherently aligned in the x-y plane, there are no forces along the z-axis. The parity plot compares the fully converged DFT forces (y-axis) with the predictions from the ML DM without any SCF iterations (x-axis). Points lying on the parity line indicate exact predictions. Additionally, the diagrams include histograms showing the distribution of atomic forces. Along the molecular dynamics trajectory forces can reach up to 2 eV/Å, although they are generally varying within the range of ±0.25 eV/Å. The mapping between the ML-DM-predicted forces and the DFT ones is exceptionally accurate, with the majority of points aligning closely along the parity line. This is further evidenced by the nearly identical force distributions. Interestingly, there is no systematic bias in the error distribution across the range of force magnitudes explored. This contrasts with typical ML force fields, where the largest errors are often observed for smaller forces. The MAE is calculated as 126 meV/Å and 62 meV/Å for the x and y components, respectively. This error can be further reduced by addressing the fact that the ML-generated DM does not necessarily correspond to an integer number of electrons, resulting in the ML-DM computed ground-state energies being consistently lower than the DFT results. One can correct this problem by rescaling the ML-generated DM by a factor N_e/N_e^{ML}, where N_e is the total number of electrons and N_e^{ML} is the number of electrons computed using the ML DM guess without using SCF approach, $N_e^{ML} = Tr[P.S]$, where S is the overlap matrix. For S_2O the difference between N_e^{ML} and N_e is less than 0.1% but this correction with the addition of one SCF iteration significantly reduces the MAE to 62 meV/Å and 22 meV/Å for the x and y components, respectively. In particular, adding one SCF strategy conveys the thoughts of the Harris functional (171, 172), which is an approximate density functional involved in electronic structure calculations in the context of the DFT framework. The main idea here is that the Harris functional is a non-self-consistent approach to the total energy functional, instead of allowing a full SCF solution. It calculates energy from train input density. The parity plots in Fig-3.10 are based on forces computed after applying this DM rescaling. The errors are significantly lower than those typically observed in state-of-the-art force fields (160, 173, 174). While a comprehensive comparison between these two methods is challenging—requiring consistent analysis with the same molecules, training set size, and other factors—these results clearly demonstrate that predicting the DM for use in non-self-consistent DFT is a viable alternative to constructing a force field. Specifically, the forces derived from the ML-predicted DM can serve as a reliable tool for performing molecular dynamics. This aspect is further explored in the following section.

Fig -3.11 demonstrates parity plots for the x (left panel) and y (right panel) components of the atomic forces acting on the H_2O molecule confined in the x-y plane. These forces are evaluated for 1000 distorted H_2O configurations sampled from the MD trajectory used to generate the training dataset, but excluded from the ML training process. By design, the molecule remains aligned in the x-y plane, hence there are

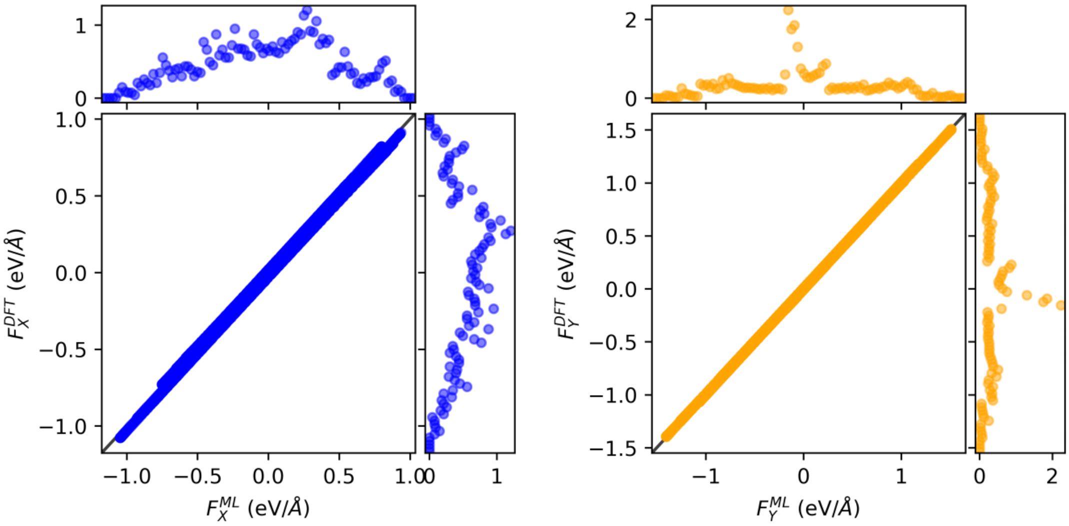

Figure 3.11: Parity plot for the $\alpha = x$, and y component of the atomic forces computed by using the ML DM, F_{α}^{ML}, with one SCF cycle, against the fully converged DFT ones, F_{α}^{DFT}. Data are presented here for a set of 1000 H_2O molecules extracted from the same molecular dynamics trajectory used to generate the training set. The right panel is for the forces x component, while the lower panel is for the y component. The histograms on the side describe the frequency of the forces in the test set.

no forces along the z-direction. The parity plots compare fully converged DFT forces (y-axis) with those predicted by the ML-generated density matrix (DM) without any SCF iteration (x-axis). Here one can see the points lying on the parity line indicate perfect agreement with the fully converged results. For H_2O we also applied the same rescaling scheme on DM and included an additional SCF step to obtain the atomic forces. Finally, we found MAE is 19 meV/Å for the x-component of the forces and 3 meV/Å for the y-component. The plots also include histograms illustrating the distributions of the atomic forces.

3.3.5 ML Guess Operated Nonself-Consistent MD

Finally, we test how the predicted DM can be used to drive molecular dynamics (MD) trajectory. Specifically, we perform $ab-initio$ MD using the ML-predicted DM instead of one obtained from the SCF scheme. This demands careful handling, as the neural networks are trained to predict the DM only for molecules with specific spatial orientations. To address this, we adopt the following workflow: (1) A molecule placed at an arbitrary position is shifted to the origin and rotated to match the orientation required for the trained NN model; (2) the DM is then predicted using the NN model and implemented as the initial guess for a static DFT calculation (non-self-consistent) to evaluate energy and forces via PySCF; (3) then the molecule is rotated and translated back to its initial position, and the same transformation is operated to the forces; (4) the atomic forces supplied to the MD package, which updates atomic structural information accordingly; (5) these steps are then repeated for subsequent MD steps. The MD approach is performed with the LAMMPS package (7). Although our ML DM non-self-consistent workflow is more intricate than the standard MD, it enables the usage of simple structural descriptors to determine the DM, a task which is inherently translationally invariant and rotationally covariant. While translational invariance could be addressed using local structural descriptors and rotational covariance through a covariant model, here we chose to keep the NN model simple to focus on exhibiting the suitability of predicting the DM with ML. By following the steps summarized above we perform the MD simulation with ML DM approach. Specifically, from the ML-predicted DM we compute the atomic forces after rescaling the DM by the factor N_e/N_e^{ML} and adding a single SCF step thereafter. The reason is that, without any iteration, the total ground state energy is marginally lower than the converged one. The MD simulations are performed for H_2O and S_2O molecules at 150 K for 0.14 and 0.12 nanoseconds, respectively. In both cases, we used the aforementioned rotational scheme to discard the need for an equivariant model. The MD trajectory computed from the ML DM approach is then compared with fully converged $ab-initio$ AIMD trajectories and with those computed utilizing only a single SCF step driven by the DIIS algorithm, starting from the PySCF default DM guess option 'minao'.

The MD trajectories are then analyzed and compared through the thermal distributions of the bond length and bond angle, as shown in Fig-3.12 for the two molecules. For H_2O [panels (c) and (d)], the distributions exhibit no major differences across the various methods, a feature that is expected due to the rapid convergence of the SCF cycle in this case. The average O-H bond length is approximately 0.9837 Å, and the av-

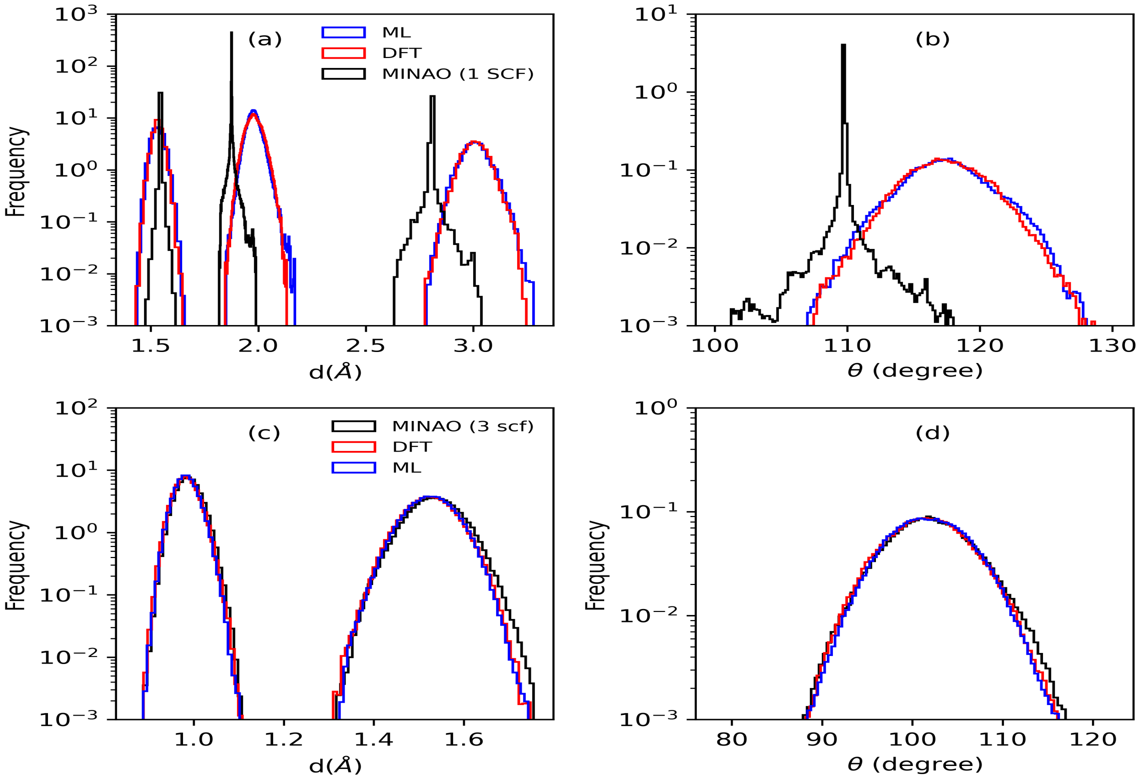

Figure 3.12: Histograms of the bond lengths, d, and bond angles, θ, along the molecular dynamics trajectories for S_2O [panels (a) and (b)] and H_2O [panels (c) and (d)]. There are two distinct bond lengths for H_2O, namely O-H and H-H , while there are three for S_2O, namely two S-O and one S-S .

erage bond angle is 101.83°, which is consistent with the static DFT BLYP results, and the experimental values (0.9578 Å and 104.47°) (175). The S_2O case presents a notable difference. Panels (a) and (b) of Fig-3.12 demonstrate that the ML-DM approach yields an outstanding approximation of the fully DFT-converged result, leading to thermal distributions of bond lengths and angles that are highly comparable to those obtained from fully $ab - initio$ molecular dynamics. In this case, three bond lengths are examined: the S-O bond, the S-S bond, and the second S-O distance between the two most peripheral atoms. The centres of their distributions are in line with the reported experimental values of 1.4650 Å (S-O), 1.8834 (Å S-S), and 3.2505 (Å S-S), (175) as does the bond angle, 117.876°). The remaining discrepancies are attributed to the choice of DFT exchange-correlation energy. However, this consistency is not maintained when molecular dynamics simulations are performed with a single SCF step initialized from PySCF's "minao" DM guess. This is because the distributions generated by the "minao"

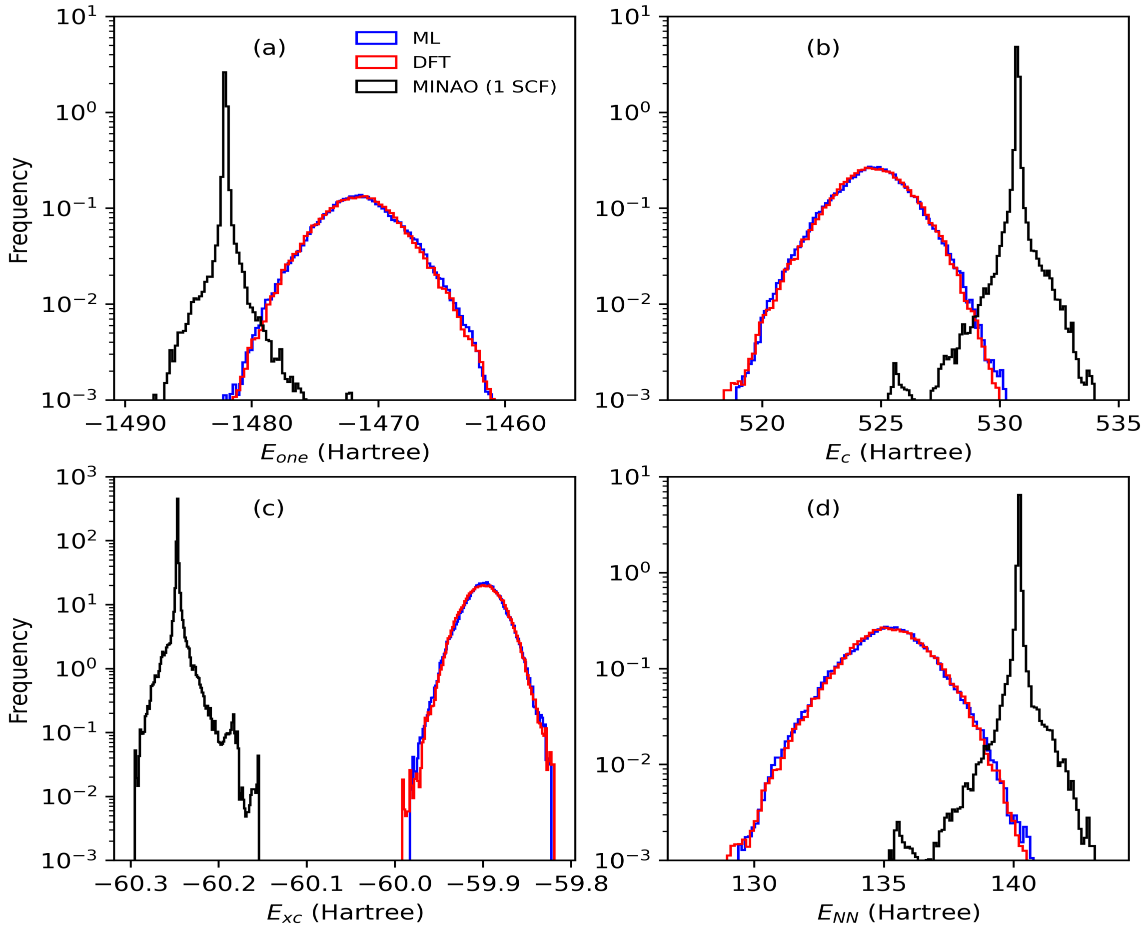

Figure 3.13: Histogram of the various energy components along the different molecular dynamics trajectories for the S_2O molecule. Here we separate the total energy into one-electron, H_{one}, Coulomb (Hartree), H_{C}, exchange-correlation, H_{XC}, and nucleus-nucleus, H_{NN}, components.

guess with one SCF cycle deviate significantly from the fully converged DFT results. Fig-3.13 represents the various components of total ground state energy: one-electron H_{core}, Coulomb repulsion H_c, exchange-correlation H_{xc} part, and nucleus-nucleus H_{NN} interaction terms. As anticipated, the distributions generated from MD simulations driven by 'minao' DM guess with one self-consistent cycle, that deviates significantly from those of self-consistent DFT due to inaccurately predicted average structures. In contrast, our ML DM not only captures the structural features accurately but also provides a reliable description of all energy components. Finally in Fig-3.14 we show the histogram of various ground state energy components for H_2O molecule computed from various MD trajectories, as one-electron H_{core}, Coulomb repulsion H_c, exchange-

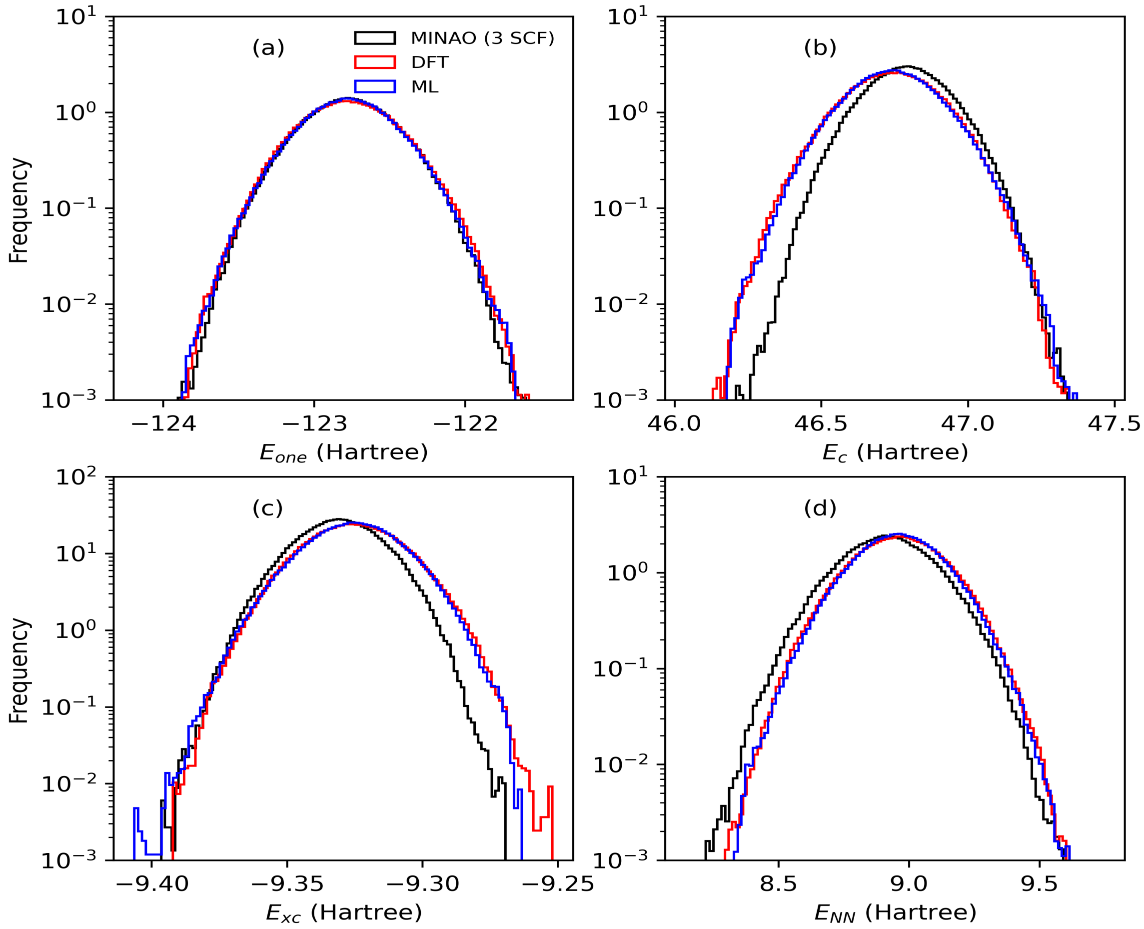

Figure 3.14: Histogram of the various energy components along the different molecular dynamics trajectories for the H_2O molecule. Here we separate the total energy into one-electron, H_{one}, Coulomb (Hartree), H_{C}, exchange-correlation, H_{XC}, and nucleus-nucleus, H_{NN}, components.

correlation H_{xc} part, and nucleus-nucleus H_{NN} interaction terms. Here we consider MD simulation with 'minao' initial density, where the DFT forces have been obtained using three self-consistent cycles. As expected, the ML DM approach consistently outperforms the 'minao' guess across all energy components except for the one-electron distribution. For the one-electron energy, the differences in distribution among the three MD approaches are not significant.

3.3.6 Limitations of NN Predicted ML DM Approach

The limitation of our NN-based approach is that the number of matrix elements to predict scales quadratically with the number of basis functions used in the calculation, making it progressively more computationally expensive as the system size grows. In practice, many of the matrix elements are relatively small, and they can be safely neglected when evaluating the DM for accurate non-self-consistent electronic structure calculations or molecular dynamics simulations. Additionally, efficiency can be improved by constructing the ML-predicted DM over a small basis function and subsequently utilizing it for calculations employing larger ones (176).

3.3.7 Optimizing ML Strategies for Density Matrix Construction

It is important to emphasize that the suitability of an ML-based strategy for predicting DM compared to alternative methods depends on various factors, including the characteristics of the target system and the adopted workflow. Specifically, three key considerations play a determining role: (1) the size of the required training set, i.e., the number of DFT calculations needed to construct the ML model; (2) the scalability of the neural network as the DM's dimensionality increases, with more complex networks potentially required for larger or more intricate systems; and (3) the workflow in which the ML model is applied, particularly the number of calculations needed after the model is built. These factors collectively define the "computational economy" of an ML approach, necessitating a thorough evaluation before selecting a specific computational strategy. Our approach becomes increasingly advantageous as the scaling of the overarching electronic structure method with system size grows more prohibitive. In such cases, the quadratic scaling of DM construction is outweighed by the high computational cost of lengthy self-consistent cycles, enabling substantial reductions in computational overhead. This is particularly relevant for methods involving non-local exchange-correlation functionals.

4 | Results and Discussion

MD from ML charge density

This chapter primarily describes how we calculate two-electrons-based Coulomb integrals in order to obtain the Coulomb matrix $J_{\mu\nu}$ from density, then the implementation of the JL-charge density model to predict the real-space electron densities and hence compute the KS-Hamiltonian and DMs, and finally the atomic forces for MD simulations. PySCF begins a DFT calculation with an initial guess for the DM. In PySCF, the matrix components relevant to determining the KS-Hamiltonian include $J_{\mu\nu}$, which describes the Coulomb repulsion potential V_H and primarily depends on the initial guess for the density matrix (DM) rather than directly on the electron density. In contrast, the matrix responsible for the exchange-correlation term, $V_{\mu\nu}^{XC}$, can be directly computed from the density. Therefore, the construction of KS-Hamiltonian for obtaining DM is not feasible from density for our ML-based MD workflow, where the electron density is the key variable from the perspective of ML predictions. This is the reason, here we have introduced an avenue to compute the J matrix over Gaussian-type orbitals (GTO) from the real space electron density. This workflow involves the construction of the Kohn-Sham (KS) Hamiltonian from the predicted electron density, followed by a matrix diagonalization process to compute the DM. This is then employed as the initial guess to expedite the SCF process in DFT calculations. Finally, the entire non-self-consistent workflow is utilized to obtain atomic forces and ground-state energy to conduct MD simulations. In brief, here the proposed ML-driven charge density-based approach develops an alternative avenue to perform non-self-consistent MD simulations, achieving accuracy comparable to $ab - initio$ MD (AIMD) while reducing the computational cost.

Recent advancements, based on machine learning (ML), have introduced transformative approaches to accelerate electronic structure calculation methods, including DFT. By leveraging ML models to predict key quantities such as the DM (123, 177), the electron densities (112, 113) or the potential energy surfaces (73, 74), it is possible to reduce the computational overhead of material science investigation while maintaining high accuracy. As an example, there is a developed method using a deep-learning neural network strategy to represent the DFT Hamiltonian, called DeepH (75). Furthermore, there are several approaches where deep-learning methods are well integrated with ab initio calculations (76, 111, 178). One crucial problem of DFT is that it is still computationally rather demanding and not often applicable to those systems where the number of atoms is more than a thousand. To overcome this drawback, deep-learning methods can be trained from DFT results, and the constructed ML model can be implemented to bypass computationally extensive steps. However, it is well-established that DFT can also be utilized to compute the atomic forces for performing molecular dynamics (MD) simulations, called $ab - initio$ molecular dynamics (AIMD). This MD simulation is a tool utilized to simulate and study the atomistic movement and interaction in molecules or solids over time, enabling the prediction of physical, chemical, and biological properties. MD simulations are utilized for a wide range of applications in a variety of fields, such as physics (179, 180), chemistry (181, 182, 183), biology (77, 184, 185), material science, and drug discovery (186, 187). In recent years, the interest has migrated to some extent, where the machine learning potential (MLP) has become a prominent approach from the aspect of machine-learning MD (ML MD) simulation (160, 188, 189, 190, 191) towards accurately reproducing the potential energy surface (PES) (192) and the associated atomic forces. MLPs rely upon a training process, which uses various atomic descriptors to train the ML model on atomic configurations and the associated atomic forces and energies (160).

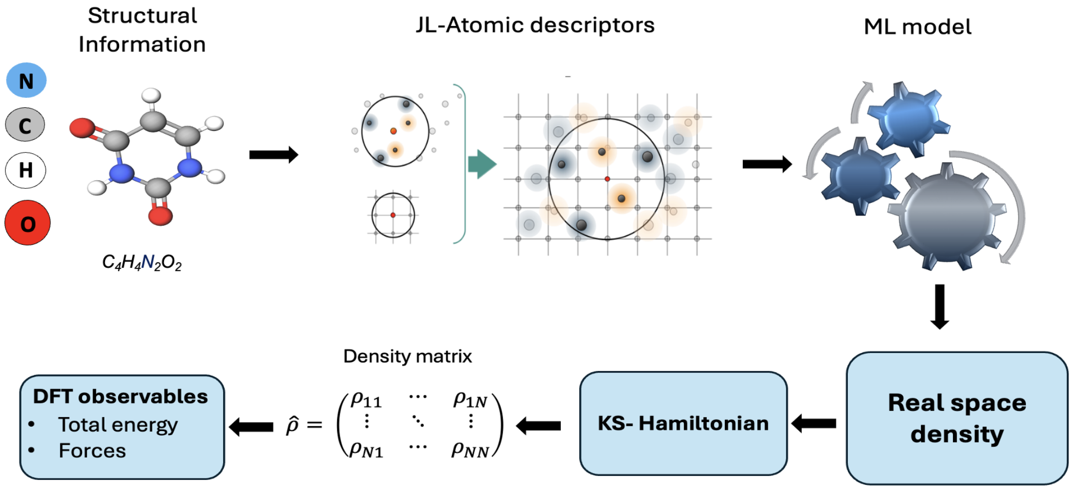

Figure 4.1: The workflow to compute DM using Jacobi-Legendre (JL) charge density model. Here, the JL-atomic descriptor takes atomic positions from a molecule as input. Using this information, the JL-charge density model predicts the electron density. The KS-Hamiltonian is then constructed from the predicted density, and after diagonalization, it yields the DM. These DMs are used as the initial guess for DFT calculations. As a result, various DFT observables, such as total energy and forces, are obtained

In Fig.-4.1, we show the workflow introduced to predict the real-space density and finally obtain the DM for molecular systems. The primary input for this workflow remains the structural information of the molecule. For this workflow, we have provided the grid data related to the given structure, then first those data passed through an atomic descriptor scheme, which describes the atomic environment inside a molecule, and then to the charge density model to obtain the final prediction for real-space electron density. It is crucial to mention, that for this specific study, we only have employed the Jacobi-Legendre (JL) charge density model (113) to obtain the prediction on charge densities. This thesis is not related to the development of the JL-change density model; the mathematical and implementation details can be found here (113, 161). Next, a linear model is trained on the grid data, which describes electron density using the JL descriptors, where the DFT-converged atom-centered real-space electron densities are provided as outputs. The predicted electron densities are then utilized to construct the Kohn–Sham (KS) Hamiltonian and the corresponding DM through the developed Python module. As the final step, these DMs are used as initial guesses for DFT calculations, enabling the computation of atomic forces non-self-consistently to perform molecular dynamics (MD) simulations. In summary, this workflow can be divided into three primary components as follows. (1) prediction of real-space electron density using the JL-density model; (2) Construction of the KS-Hamiltonian from the predicted density to obtain the DM; (3) execution of DFT calculations with the predicted DMs as initial guesses to compute forces and generate MD trajectories (ML-MD). However, to validate our workflow, we applied this density-based ML-MD approach to compute the dipole moment at each step of the MD run, thereby computing infrared (IR) spectra for a molecule in the gas phase. The results were then compared with fully $ab-initio$ calculations at various temperatures.

4.1 Two Electron Coulomb Integrals

We utilized the PySCF package to implement our ML-based methods; however, it does not provide the capability to construct the KS-Hamiltonian or the DM directly from the electron density. This limitation arises because PySCF does not offer direct access to the matrix elements associated with the classical Coulomb interaction. Instead, it allows for restarting a DFT calculation by accessing only the guess DM, similar to the approach used in the $SIESTA$ (193) code. Using PySCF the the Coulomb matrix describing the classical Coulomb interaction can not be constructed from the electron density. However, one can obtain other Hamiltonian terms like the one-electron matrix, and from the electron density, the exchange-correlation matrix. There are a few methods dedicated to calculating the Coulomb matrix for performing $ab-initio$ quantum mechanical calculations using pseudopotentials and plane a plane-wave basis set (40). This scheme is well-implemented in a popular electronic structure calculation package called Vienna Ab initio Simulation Package (VASP) (194, 195). It is a common fact that the most time-consuming part of electronic structure calculations using approximate DFT is the evaluation of Coulomb and exchange-correlation terms. The approach proposed in (196) demonstrates a fast and sufficiently accurate evaluation of Coulomb interaction using the expansion of molecular electronic density in atom-centred auxiliary basis sets. L. Gagliardi, N.C. Handy, et al (197) showed the resolution of the identity approximation for the Coulomb energy in DFT to improve the computational efficiency of large-scale calculations. In our work, we have developed a Python module to compute the Coulomb matrix over Gaussian-type orbitals (GTOs), since PySCF performs DFT calculations for molecules on GTO basis functions, using the real-space electron density as the primary variable. Thereafter, one separate module was constructed using all terms required to build the KS-Hamiltonian (detailed discussion will follow in the result section) for obtaining DM. However, few advanced numerical libraries are also able to compute these kinds of two-electron integrals. Such as libint (198), a widely used C++ library for evaluating Gaussian integrals in quantum chemistry. This library efficiently handles repulsion integrals (ERIs), overlap, kinetic, nuclear attraction integrals, etc, and it is implemented in various quantum chemistry packages, like Psi4 (199), MPQC (200), GAMESS (200), etc. Another known numerical library is libcint (201), developed by Qiming Sun and well implemented with the PySCF framework. This is a C library for Gaussian integrals. It supports Cartesian and spherical Gaussian-type orbitals for various integrations and derivatives. It is often lighter than libint. Both integral engines compute primitive integrals over Gaussian basis functions for obtaining the Coulomb matrix, but do not directly work on electron density; they involve DMs, as $J_{\mu\nu} = \sum_{\lambda\sigma} P_{\lambda\sigma} \left(\mu\nu|\lambda\sigma\right)$. For our second workflow, real-space electron density is the accessible entity from the ML prediction, so each components in the KS-Hamiltonian have to be represented as a function of electron density.

Furthermore, density fitting (DF) (202, 203), also known as the resolution of identity, is an efficient strategy in electronic structure theory, which approximates the product of two basis functions (pair densities) by expanding them in an auxiliary basis set. In practice, this reduces the four-centre two-electron integrals , which are central to Hartree–Fock and DFT, into a combination of three-centre and two-centre integrals involving the auxiliary functions. Since these two-electron integrals describe the Coulomb interaction between charge distributions Eq.(39), DF is most naturally connected to the Coulomb integral. By fitting orbital pair densities in an auxiliary space, DF provides an accurate and low-cost way of evaluating the Coulomb matrix elements needed in building the Fock or Kohn–Sham Hamiltonian. Thus, density fitting can be viewed as an approximation scheme that accelerates the computation of Coulomb-type integrals while maintaining near-exact accuracy for an appropriate choice of auxiliary basis functions. Electrons interact with each other due to their electric charges. The Coulomb potential describes this interaction, which depends on the product of two point charges and is inversely proportional to the square of the distance between them. Mathematically, the Coulomb potential $V_H(\mathbf{r})$ at a point $\mathbf{r}$ in space due to a charge distribution coming from all other electrons $\rho(\mathbf{r}^{'})$ is given by

$$V_H(\mathbf{r}) = \int \frac{\rho(\mathbf{r}^{'})}{\mathbf{r}-\mathbf{r}^{'}} d\mathbf{r}^{'} \tag{1}$$

In DFT, the Coulomb potential is expressed in terms of electron density, which is the key quantity of interest. In practice, the Coulomb potential is often discretized over a grid in space for a DFT calculation. Overall, the Coulomb interaction is a fundamental concept in KS-DFT, as it describes the electron-electron interaction and plays a crucial role in the electronic structure and properties investigations of atoms, solids, and molecular systems. As any standard DFT code demands builds the KS-Hamiltonian. The Coulomb potential is then written as a matrix on a given basis called the Coulomb matrix. The matrix element of the Coulomb matrix, computed between two basis functions, can take the following form

$$\begin{aligned} \langle \phi_\mu | v_{\mathrm{H}} | \phi_\nu \rangle &= \int \int \frac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}^{'}|} \phi_\mu^*(\mathbf{r}) \phi_\nu(\mathbf{r}) d\mathbf{r} d\mathbf{r}' = \\ &= \int \rho(\mathbf{r}') \left[\int \frac{\phi_\mu^*(\mathbf{r}) \phi_\nu(\mathbf{r})}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r} \right] d\mathbf{r}' . \end{aligned} \tag{2}$$

Where $\phi_\mu^*(\mathbf{r})$ (complex conjugate) and $\phi_\nu(\mathbf{r})$ are the basis basis function. On a note, in this development, $\phi_\mu^*(\mathbf{r})$ is a real-valued function; we have only considered

Gaussian-type orbitals (GTO) to construct the Coulomb matrix from the charge density. From Eq.2. This expression describes the electrostatic interaction of electrons. However, its evaluation is computationally expensive due to its long-range nature and slow decay. In brief, the Coulomb potential extends across the entire system, meaning every electron interacts with all others. Additionally, it decays as $\frac{1}{|\mathbf{r}-\mathbf{r}'|}$ making distant interactions still significant and expensive to compute. For example, if two charges are apart by 1 Å the interaction strength is 1/1, if they are apart by 10 Å, the interaction is 1/10, and if they are 100 Å away the interaction is 1/100- which still not zero.

Next, the discussion will broadly cover how mathematically we implemented the molecular integral evaluation to construct the Coulomb matrix from the electron charge density over the Gaussian-type orbitals (GTO) in the real space grid. This mathematical derivation demands a few more things such as the GTO normalization factor, the Gaussian product theorem, and some other approximations. Before beginning that derivation, we introduce a few steps to clarify a few more mathematical functions.

4.2 GTO Normalization

As we described before, in all of our DFT calculations or in ML approaches, we have used PySCF. In this package, Gaussian-type orbitals are implemented to perform the molecular DFT calculations. Therefore, we have considered Gaussian-type orbitals to calculate the Coulomb matrix from the real-space electron density. There are two different types of Gaussian functions: Cartesian and Spherical harmonics. Here we will only discuss its Cartesian form. For the mathematical derivation, first, we introduce the general form of the molecular integral relevant to the Coulomb matrix using primitive Gaussian functions. Then the final form of the same matrix is obtained via the contracted Gaussian functions, which are the linear combinations of multiple primitive Gaussian functions. The functional expression of a primitive Cartesian Gaussian centred at the origin A =follows the given form (204)

$$\begin{aligned}\phi_1(\mathbf{r}) &= (x-A_x)^l (y-A_y)^m (z-A_z)^n e^{-\alpha|\mathbf{r}-\mathbf{A}|^2} = \\ &= x_A^l y_A^m z_A^n e^{-\alpha \mathbf{r_A}^2} \ .\end{aligned} \tag{3}$$

Eq.3 is a basic function of angular momentum l, m, and n, with an orbital exponent α, and it is not normalized. The normalization condition for a given ϕ_1 can be written as

$$\begin{aligned}1. &= N^2 \int_{-\infty}^{+\infty} (x_A^l y_A^m z_A^n \exp^{-\alpha \mathbf{r}_A^2})(x_A^l y_A^m z_A^n e^{-\alpha \mathbf{r}_A^2}) dr \\ &= N^2 \int_{-\infty}^{+\infty} x_A^{2l} y_A^{2m} z_A^{2n} e^{-2\alpha \mathbf{r}_A^2} dr \\ &= N^2 \int_{-\infty}^{+\infty} x_A^{2l} e^{-2\alpha x_A^2} dx \int_{-\infty}^{+\infty} y_A^{2m} e^{-2\alpha y_A^2} dy \int_{-\infty}^{+\infty} z_A^{2n} e^{-2\alpha z_A^2} dz \\ &= N^2 I_x^A I_y^B I_z^C \ .\end{aligned} \tag{4}$$

Now, if we apply the standard Gamma function integration, then we find

$$I_x^A = \int_{-\infty}^{+\infty} x_A^{2l} e^{-2\alpha x_A^2} dx = \frac{(2l-1)!!\sqrt{\pi}}{(4\alpha)^l \sqrt{2\alpha}} \ , \tag{5}$$

and the same expressions can be obtained when the integrations are performed for y and z. Then, by using these results in Eq.4 one can obtain the normalization factor N as

$$N = \left[\left(\frac{2}{\pi}\right)^{3/4} \frac{2^{(l+m+n)} \alpha^{(2l+2m+2n+3)/4}}{[(2l-1)!!(2m-1)!!(2n-1)!!]^{1/2}} \right] \ . \tag{6}$$

This result is completely general for uncontracted primitive Cartesian Gaussian functions ϕ_1.

4.3 The Gaussian Product Theorem

Here, we need to introduce the Gaussian product theorem, because this theorem is useful to compute the Coulomb matrix using Gaussian orbitals. The theorem said the product of two different Gaussian functions G_1 and G_2 centred on the two different points can be expressed as a summation of Gaussian functions centred at the midpoint. That makes it efficient for evaluating integrals involving Gaussian-type orbitals. If two Gaussian functions G_1 and G_2 are centred on **A** and **B** then according to the product theorem it can be written as

$$
\begin{aligned}
G_1G_2 &= G_1(\mathbf{r},\alpha_1,\mathbf{A},l_1,m_1,n_1)G_2(\mathbf{r},\alpha_2,\mathbf{B},l_2,m_2,n_2) \\
&= e^{-\alpha_1\alpha_2(\overline{\mathbf{AB}})/\gamma}\times \\
&\left[\sum_{i=0}^{l_1+l_2} f_i(l_1,l_2,\overline{\mathbf{AP}}_x,\overline{\mathbf{BP}}_x)x_P^i e^{-\gamma x_P^2}\right]\times \\
&\left[\sum_{j=0}^{m_1+m_2} f_j(m_1,m_2,\overline{\mathbf{AP}}_y,\overline{\mathbf{BP}}_y)y_P^j e^{-\gamma y_P^2}\right]\times \\
&\left[\sum_{k=0}^{n_1+n_2} f_k(n_1,n_2,\overline{\mathbf{AP}}_z,\overline{\mathbf{BP}}_z)z_P^k e^{-\gamma z_P^2}\right].
\end{aligned}
\tag{7}
$$

We can prove it in the following way. Firstly, we need to define the multiplicands as

$$G_1 = G_1(\mathbf{r},\alpha_1,\mathbf{A},l_1,m_1,n_1) = x_A^{l_1}y_A^{m_1}z_A^{n_1}e^{-\alpha_1\mathbf{r}_A^2}\ ; \tag{8}$$

$$G_2 = G_2(\mathbf{r},\alpha_2,\mathbf{B},l_2,m_2,n_2) = x_B^{l_2}y_B^{m_2}z_B^{n_2}e^{-\alpha_2\mathbf{r}_B^2}\ , \tag{9}$$

where $\mathbf{r_A}$ can be defined as $\mathbf{r}_A = \mathbf{r} - \mathbf{A}$ and $|\mathbf{r}_A| = \sqrt{(x-A_x)^2+(y-A_y)^2+(z-A_z)^2}$, the difference between grid points (x,y,z) and the atomic positions(A_x,A_y,A_z). Next, we will prove the theorem and yield Eq (7). To analyse it let us set all the angular momentum values to be 0 and hence the form of Eq.(8) and Eq.(9) as follows

$$G_1 = e^{-\alpha_1\mathbf{r}_A^2}; \tag{10}$$

$$G_1 = e^{-\alpha_2\mathbf{r}_B^2} \tag{11}$$

As per the Gauss product theorem, we can write the product $G_1 \cdot G_2$ as the third

Gaussian with a constant K, i.e. $G_1.G_2 = G_3$,

$$\begin{aligned} e^{-\alpha_1 \mathbf{r}_A^2} e^{-\alpha_2 \mathbf{r}_B^2} &= K e^{-\gamma \mathbf{r}_P^2} \\ &= K e^{-[\gamma(\mathbf{r}-\mathbf{P})^2]} \\ &= K e^{-[\gamma(\mathbf{r}^2+\mathbf{P}^2-2\mathbf{r}\mathbf{P})]} \, . \end{aligned} \tag{12}$$

Again, by using the definition of $\mathbf{r}_A$ and $\mathbf{r}_B$

$$\begin{aligned} e^{-\alpha_1 \mathbf{r}_A^2} e^{-\alpha_2 \mathbf{r}_A^2} &= e^{-\alpha_1(\mathbf{r}-\mathbf{B})^2-\alpha_2(\mathbf{r}-\mathbf{B})^2} \\ &= e^{-(\alpha_1+\alpha_2)\mathbf{r}^2+2(\alpha_1\mathbf{A}+\alpha_2\mathbf{B})\mathbf{r}-\alpha_1\mathbf{A}.\mathbf{A}-\alpha_2\mathbf{B}.\mathbf{B}} \, . \end{aligned} \tag{13}$$

Now if we compare each terms in Eq.(12) and Eq.(13), then we will obtain the following expressions $\gamma = (\alpha_1 + \alpha_2)$, $\mathbf{P} = \frac{\alpha_1 A + \alpha_2 \mathbf{B}}{\gamma}$, and $K = e^{-\alpha_1 \mathbf{A}^2 - \alpha_2 \mathbf{B}^2 + \gamma \mathbf{P}^2}$. So, by using the expression of $\mathbf{P}$ and γ we can write

$$\gamma \mathbf{P}^2 = \frac{(\alpha_1^2 \mathbf{A}^2 + 2\alpha_1\alpha_2 \mathbf{A} \cdot \mathbf{B} + \alpha_2^2 \mathbf{B}^2)}{(\alpha_1 + \alpha_2)} \, . \tag{14}$$

Hence, in the expression of K after substituting the final form of $(\gamma \mathbf{P}^2)$ from the above equation we can obtain

$$K = e^{-[\alpha_1\alpha_2(\overline{\mathbf{AB}})^2/\gamma]} \, . \tag{15}$$

Hence, if we define $\overline{|\mathbf{AB}|} = |\mathbf{A} - \mathbf{B}| = \sqrt{(A_x - B_x)^2 + (A_y - B_y)^2 + (A_z - B_z)^2}$ for the two s types orbitals, then the Gaussian product will take the following form

$$e^{-\alpha_1 \mathbf{r}_A^2} e^{-\alpha_2 \mathbf{r}_B^2} = e^{-[\alpha_1\alpha_2(\overline{\mathbf{AB}})^2/\gamma]} e^{-[\gamma(\mathbf{r}-\mathbf{P})^2]} \, . \tag{16}$$

Furthermore, for any arbitrary angular momentum, the product of two different Gaussian functions centred at two separate points A and B can be written as

$$G_1 G_2 = x_A^{l_1} y_A^{m_1} z_A^{n_1} x_B^{l_2} y_B^{m_2} z_B^{n_2} e^{-[\alpha_1\alpha_2(\overline{\mathbf{AB}})^2/\gamma]} e^{-[\gamma(\mathbf{r}-\mathbf{P})^2]} \, . \tag{17}$$

Where Eq.(16) has been implemented in Eq.(17) to take care of the product of two exponentials. Now, we need to play with the term $x_A^{l_1} x_B^{l_2}$ to construct Eq.(7)

$$\begin{aligned}
x_A^{l_1} x_B^{l_2} &= (x - A_x)^{l_1} (x - B_x)^{l_2} \\
&= [(x - P_x) + (P_x - A_x)]^{l_1} [(x - P_x) + (P_x - B_x)]^{l_2} \\
&= (x_p - \overline{\mathbf{AP}}_x)^{l_1} (x_p - \overline{\mathbf{BP}}_x)^{l_2} \\
&= \sum_{i=0}^{l_1} x_P^i (\overline{\mathbf{AP}})_x^{(l_1-i)} C(l_1, i) \sum_{j=0}^{l_2} x_P^j (\overline{\mathbf{BP}})_x^{(l_2-j)} C(l_2, j) \\
&= \sum_{k=0}^{l_1+l_2} x_P^k \left[\sum_{i=0}^{k} \sum_{i=0}^{l_2} x_P^i x_P^j (\overline{\mathbf{AP}})_x^{(l_1-i)} C(l_1, i) (\overline{\mathbf{BP}})_x^{(l_2-j)} C(l_2, j) \right] \\
&= \sum_{k=0}^{l_1+l_2} x_P^k f_k(l_1, l_2, \overline{\mathbf{AP}}_x, \overline{\mathbf{BP}}_x) \ .
\end{aligned} \tag{18}$$

Here, $x_p = (x - P_x)$, $C(l_1, i) = \frac{l_1!}{i!(l_1-i)!}$, and $\overline{\mathbf{AP}}_x = (\mathbf{A} - \mathbf{P})_x$. Similarly,

$$y_A^{m_1} y_B^{m_2} = \sum_{k=0}^{m_1+m_2} y_P^k f_k(m_1, m_2, \overline{\mathbf{AP}}_y, \overline{\mathbf{BP}}_y) \ ; \tag{19}$$

$$z_A^{n_1} z_B^{n_2} = \sum_{k=0}^{n_1+n_2} z_P^k f_k(n_1, n_2, \overline{\mathbf{AP}}_z, \overline{\mathbf{BP}}_z) \ . \tag{20}$$

Here we have used standard binomial expansion of $(x_p - \overline{\mathbf{AP}}_x)^{l_1}$ and other similar terms for m and n angular momentum. Finally, the coefficient of x_p^k, y_p^k, and z_p^k can be written as

$$f_l^x(l_1^l, l_2^l, \overline{\mathbf{AP}}_x, \overline{\mathbf{BP}}_x) = \sum_{i=0}^{k} \sum_{i=0}^{l_2} x_P^i x_P^j (\overline{\mathbf{AP}})_x^{(l_1-i)} C(l_1, i) (\overline{\mathbf{BP}})_x^{(l_2-j)} C(l_2, j) \tag{21}$$

Since l, m, n and x, y, and z are the dummy indices and variables, one can construct other coefficients by replacing those indexes and coordinates in Eq.(21). Here, $k = (l_1 + l_2)$. Now one can obtain Eq(7) by substituting Eq.(18,19, 20) into Eq.(17), where

f_l, f_m, and f_n follows Eq.(21)

4.4 Calculation for Coulomb Matrix

Eq.(2) we can write as

$$\langle \phi_\mu | V_H | \phi_\nu \rangle = \int \rho(\mathbf{r}^{'}) \left[\int \frac{\phi_\mu^*(\mathbf{r}) \phi_\nu(\mathbf{r})}{|\mathbf{r} - \mathbf{r}^{'}|} d\mathbf{r} \right] d\mathbf{r}^{'} \ . \tag{22}$$

However, from Eq.(22) we can say the expression inside the square bracket does not depend on the real-space charge density. This is the standard expression for the Coulomb matrix and since it is integrable over a product space, it allows for the change in the order of integration between $d\mathbf{r}$ and $d\mathbf{r}^{'}$. The main idea here is to compute this part which exists inside the bracket in Eq. (22), once it is done then obtaining matrix elements is straightforward, which demands a numerical integration with the electron density. Here we always consider the real basis function that $\phi = \phi^*$. It is crucial to mention here, first, we derive the expression which is inside the square bracket of Eq.(22) for two uncontracted or primitive Gaussian basis functions ϕ_1 and ϕ_2, then we finally obtain the Coulomb matrix for the contracted Gaussian function ϕ_μ. Hence the expression we can write from Eq.(22)

$$\phi_{12} = \int \frac{\phi_1^*(\mathbf{r}) \phi_2(\mathbf{r})}{|\mathbf{r} - \mathbf{r}^{'}|} d\mathbf{r} \ . \tag{23}$$

Now By applying the Gaussian product theorem, we can use the mathematical expression from Eq.(7) in Eq.(23), and hence obtain

$$
\begin{aligned}
\phi_{12} = N_{1,2} \int e^{-\left[\alpha_1\alpha_2 \times (\overline{\mathbf{AB}})/\gamma\right]} |\mathbf{r} - \mathbf{r}^{'}|^{-1} \times \\
\times \left[\sum_{i=0}^{l=l_1+l_2} f_i^x(l_1, l_2, \overline{\mathbf{AP}}_x, \overline{\mathbf{BP}}_x) x_P^i e^{-\gamma x_P^2} \right] \times \\
\times \left[\sum_{j=0}^{m=m_1+m_2} f_j^y(m_1, m_2, \overline{\mathbf{AP}}_y, \overline{\mathbf{BP}}_y) y_P^j e^{-\gamma y_P^2} \right] \times \\
\times \left[\sum_{k=0}^{n=n_1+n_2} f_k^z(n_1, n_2, \overline{\mathbf{AP}}_z, \overline{\mathbf{BP}}_z) z_P^k e^{-\gamma z_P^2} \right] d\mathbf{r} = \\
N_{1,2} \left[K \sum_l \sum_m \sum_n f_l^x f_m^y f_n^z \int e^{-\gamma \mathbf{r}_P^2} \frac{x_P^l y_P^m z_P^n}{|\mathbf{r} - \mathbf{r}^{'}|} d\mathbf{r} \right] .
\end{aligned}
\tag{24}
$$

Here $K = e^{-\left[\alpha_1\alpha_2(\overline{\mathbf{AB}})/\gamma\right]}$ and $N_{1,2} = N_1 N_2$ is normalization constant for the primitive Gaussian function given in the Eq.(6). From Eq.(24) it can be observed the usual Gaussian-like term $x_P^l y_P^m z_P^n$ is straightforward to compute, because the integration in the x, y, and z components are separable. In contrast, the term in the denominator $|\mathbf{r} - \mathbf{r}^{'}|$ is not separable along the independent coordinates. This is the reason for making this molecular integral more complex over $\frac{1}{|\mathbf{r}-\mathbf{r}^{'}|}$. In our work, we have implemented the most commonly used scheme to deal with $\frac{1}{|\mathbf{r}-\mathbf{r}^{'}|}$, namely the Laplace transformations, which transforms $\frac{1}{r}$ into some exponential form. As a result, the entire expression can be represented as a Gaussian product. If we apply the standard Laplace transformation we will obtain

$$
\frac{1.}{\mathbf{r} - \mathbf{r}^{'}} = \frac{1}{\sqrt{\pi}} \int_0^\infty e^{-s|\mathbf{r}-\mathbf{r}^{'}|^2} s^{-\frac{1}{2}} ds \; . \tag{25}
$$

Hence from the Eq.(24) we can write

$$\begin{aligned}\phi_{12} = \frac{1}{\sqrt{\pi}} N_{1,2} \Bigg[K \sum_l \sum_m \sum_n f_l^x f_m^y f_n^z \\ \times \int_{-\infty}^{+\infty} \int_0^{\infty} e^{-\gamma \mathbf{r}_P^2} x_P^l y_P^m z_P^n e^{-s|\mathbf{r}-\mathbf{r}'|^2} s^{-\frac{1}{2}} d\mathbf{r} ds \Bigg] = \\ = \frac{1}{\sqrt{\pi}} N_{1,2} \Bigg[K \sum_l \sum_m \sum_n f_l^x f_m^y f_n^z I_{xyz}^{lmn} \Bigg] ,\end{aligned} \tag{26}$$

where

$$I_{lmn}^{xyz} = \int_{-\infty}^{+\infty} \int_0^{\infty} ds \left[x_P^l y_P^m z_P^n \right] \left[e^{-\gamma \mathbf{r}_P^2} e^{-s|\mathbf{r}-\mathbf{r}'|^2} \right] s^{-\frac{1}{2}} d\mathbf{r} \, . \tag{27}$$

Here our main focus is on the integration Eq.(27). If we apply again the Gaussian product theorem for the two exponents present in the above equation, we can write

$$\begin{aligned} e^{-\gamma \mathbf{r}_P^2} e^{-s|\mathbf{r}-\mathbf{r}'|^2} &= e^{-\gamma|\mathbf{r}-\mathbf{P}_r|^2} \times e^{-s|\mathbf{r}-\mathbf{r}'|^2} \\ &= e^{-\frac{\gamma s|\mathbf{r}' - \mathbf{P}_r|^2}{(\gamma+s)}} e^{-(\gamma+s)|\mathbf{r}-\mathbf{D}_r|^2} \, , \end{aligned} \tag{28}$$

Where, the definition of $\mathbf{D}_r$ follows as $\mathbf{D}_r = \frac{\gamma \mathbf{P}_r + s\mathbf{r}'}{\gamma+s}$. From Eq.(27) by using before mentioned definition we can expand $(x_p^l y_p^m z_p^n)$ term into $(x - P_x)^l (y - P_y)^m (z - P_z)^n$. Furthermore, it can be seen in Eq.(28), that it is a function of a new variable D. So, the use of Eq.(28) in Eq.(27) will make Eq.(27) the function of two different variables together which is $\mathbf{P}_r$ and $\mathbf{D}_r$. Hence, we can follow the steps that we made for Eq.(18), Eq.(19), and for Eq.(20)

$$\begin{aligned} (x - P_x)^l &= (x_D - \overline{\mathbf{PD}}_x)^l = \\ &= \sum_{l'=0}^{l} (x_D)^{l'} \overline{\mathbf{PD}}_x^{(l-l')} C(l, l') \, , \end{aligned} \tag{29}$$

Where, $x_D = x - D_x$ and $\overline{\mathbf{PD}_x} = (\mathbf{P} - \mathbf{D})_x$, similarly

$$(y - P_y)^m = \sum_{m'=0}^{m} (y_D)^{m'} \overline{\mathbf{PD}}_y^{(m-m')} C(m, m') \, , \tag{30}$$

$$(z-P_z)^n = \sum_{n'=0}^{n} (z_D)^{n'} \overline{\mathbf{PD}}_z^{(n-n')} C(n,n') \,. \tag{31}$$

Here, y_D, z_D, and $\overline{PD_y}$, $\overline{PD_z}$ follow the before mentioned definitions. Now substituting all of these values in Eq.(27) will give

$$\begin{aligned}
I_{lmn}^{xyz} = & \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} \int_{0}^{\infty} \sum_{l'=0}^{l} \sum_{m'=0}^{m} \sum_{n'=0}^{n} C(l,l')C(m,m')C(n,n') \overline{\mathbf{PD}}_x^{(l-l')} \overline{\mathbf{PD}}_y^{(m-m')} \overline{\mathbf{PD}}_z^{(n-n')} \times \\
& \times (x_D)^{l'} (y_D)^{m'} (z_D)^{n'} e^{-\frac{\gamma s|\mathbf{r}'-\mathbf{P}_r|^2}{(\gamma+s)}} e^{-(\gamma+s)|\mathbf{r}-\mathbf{D}_r|^2} s^{-\frac{1}{2}} d\mathbf{r} ds = \\
& = \frac{1}{\sqrt{\pi}} \sum_{l'=0}^{l/2} \sum_{m'=0}^{m/2} \sum_{n'=0}^{n/2} C(l,2l')C(m,2m')C(n,2n') \int_{-\infty}^{+\infty} \int_{0}^{\infty} \overline{\mathbf{PD}}_x^{(l-2l')} \overline{\mathbf{PD}}_y^{(m-2m')} \overline{\mathbf{PD}}_z^{(n-2n')} \times \\
& \times (x_D)^{2l'} (y_D)^{2m'} (z_D)^{2n'} e^{-\frac{\gamma s|\mathbf{r}'-\mathbf{P}_r|^2}{(\gamma+s)}} e^{-(\gamma+s)|\mathbf{r}-\mathbf{D}_r|^2} s^{-\frac{1}{2}} d\mathbf{r} ds = \\
& = \frac{1}{\sqrt{\pi}} \sum_{l'=0}^{l/2} \sum_{m'=0}^{m/2} \sum_{n'=0}^{n/2} C(l,2l')C(m,2m')C(n,2n') \int_{0}^{\infty} \overline{\mathbf{PD}}_x^{(l-2l')} \overline{\mathbf{PD}}_y^{(m-2m')} \overline{\mathbf{PD}}_z^{(n-2n')} \times \\
& \times e^{-\frac{\gamma s|\mathbf{r}'-\mathbf{P}_r|^2}{(\gamma+s)}} s^{-\frac{1}{2}} \left[\int_{-\infty}^{+\infty} (x-D_x)^{2l'} e^{-(\gamma+s)(x-D_x)^2} dx \right] \left[\int_{-\infty}^{+\infty} (y-D_y)^{2m'} e^{-(\gamma+s)(y-D_y)^2} dy \right] \times \\
& \times \left[\int_{-\infty}^{+\infty} (z-D_z)^{2n'} e^{-(\gamma+s)(z-D_z)^2} dz \right] ds \,.
\end{aligned} \tag{32}$$

From the definition of the gamma function integral, we can write

$$\int_{-\infty}^{+\infty} (x-D_x)^{2l'} e^{-(\gamma+s)(x-D_x)^2} dx = \frac{(2l'-1)!!\sqrt{\pi}}{2^{l'}(\gamma+s)^{l'}\sqrt{\gamma+s}} \,. \tag{33}$$

By substituting this expression for the given three integrals in Eq.(32) and after rearranging we can obtain the following

$$\begin{aligned}
I_{lmn}^{xyz} = & \frac{1}{\sqrt{\pi}} \sum_{l'=0}^{l/2} \sum_{m'=0}^{m/2} \sum_{n'=0}^{n/2} C(l,2l')C(m,2m')C(n,2n') \frac{\pi^{3/2}(2l'-1)!!(2m'-1)!!(2n'-1)!!}{2^{l'+m'+n'}} \\
& \overline{\mathbf{PD}}_x^{(l-2l')} \overline{\mathbf{PD}}_y^{(m-2m')} \overline{\mathbf{PD}}_z^{(n-2n')} \int_{0}^{\infty} \frac{s^{-1/2} e^{-\frac{\gamma s|\mathbf{r}'-\mathbf{P}_r|^2}{(\gamma+s)}}}{(\gamma+s)^{l'+m'+n'}(\gamma+s)^{3/2}} ds \,.
\end{aligned} \tag{34}$$

Again we have introduced

$$\begin{aligned}
\overline{\mathbf{PD}}_x^{(l-2l^{'})} &= \left[P_x - \frac{\gamma P_x + s x^{'}}{\gamma + s}\right]^{l-2l^{'}} = \\
&= \left[\frac{P_x(\gamma + s) - (\gamma P_x + s x^{'})}{\gamma + s}\right]^{l-2l^{'}} = \\
&= \left[\frac{s}{\gamma + s}\right]^{l-2l^{'}} (P_x - x^{'})^{l-2l^{'}} .
\end{aligned} \tag{35}$$

Similarly

$$\overline{\mathbf{PD}}_y^{(m-2m^{'})} = \left[\frac{s}{\gamma + s}\right]^{m-2m^{'}} (P_y - y^{'})^{m-2m^{'}} , \tag{36}$$

$$\overline{\mathbf{PD}}_z^{(n-2n^{'})} = \left[\frac{s}{\gamma + s}\right]^{n-2n^{'}} (P_z - z^{'})^{n-2n^{'}} . \tag{37}$$

Hence by using the form of $\overline{\mathbf{PD}}_x^{(l-2l^{'})}$, $\overline{\mathbf{PD}}_y^{(m-2m^{'})}$, $\overline{\mathbf{PD}}_z^{(n-2n^{'})}$ in Eq.(34) we write

$$\begin{aligned}
I_{lmn}^{xvz} = {} & \frac{1}{\sqrt{\pi}} \sum_{l^{'}=0}^{l/2} \sum_{m^{'}=0}^{m/2} \sum_{n^{'}=0}^{n/2} C(l, 2l^{'}) C(m, 2m^{'}) C(n, 2n^{'}) \frac{\pi^{3/2} (2l^{'} - 1)!!(2m^{'} - 1)!!(2n^{'} - 1)!!}{2^{l^{'}+m^{'}+n^{'}}} \times \\
& \times \left[\frac{s}{\gamma + s}\right]^{l-2l^{'}} (P_x - x^{'})^{l-2l^{'}} \left[\frac{s}{\gamma + s}\right]^{m-2m^{'}} (P_y - y^{'})^{m-2m^{'}} \left[\frac{s}{\gamma + s}\right]^{n-2n^{'}} (P_z - z^{'})^{n-2n^{'}} \times \\
& \times \int_0^\infty \frac{s^{-1/2} e^{-\frac{\gamma s |\mathbf{r}^{'} - \mathbf{P}_r|^2}{(\gamma + s)}}}{(\gamma + s)^{l^{'}+m^{'}+n^{'}} (\gamma + s)^{3/2}} .
\end{aligned} \tag{38}$$

After simplifying the present integral can be written in the more coupled form

$$
\begin{aligned}
I^{xyz}_{lmn} = & \frac{1}{\sqrt{\pi}} \sum_{l^{'}=0}^{l/2} \sum_{m^{'}=0}^{m/2} \sum_{n^{'}=0}^{n/2} C(l,2l^{'})C(m,2m^{'})C(n,2n^{'}) \times \\
& \times \frac{\pi^{3/2}(2l^{'}-1)!!(2m^{'}-1)!!(2n^{'}-1)!!}{2^{l^{'}+m^{'}+n^{'}}} \times (P_x - x^{'})^{l-2l^{'}} (P_y - y^{'})^{m-2m^{'}} (P_z - z^{'})^{n-2n^{'}} \times \\
& \times \int_0^{\infty} \frac{s^{(l+m+n)-2(l^{'}+m^{'}+n^{'})} s^{-1/2} e^{-\frac{\gamma s |\mathbf{r}^{'}-\mathbf{P}_r|^2}{(\gamma+s)}}}{(\gamma+s)^{(l+m+n)-(l^{'}+m^{'}+n^{'})}(\gamma+s)^{3/2}} = \\
= & \frac{1}{\sqrt{\pi}} \sum_{l^{'}=0}^{l/2} \sum_{m^{'}=0}^{m/2} \sum_{n^{'}=0}^{n/2} C(l,2l^{'})C(m,2m^{'})C(n,2n^{'}) \times \\
& \times \frac{\pi^{3/2}(2l^{'}-1)!!(2m^{'}-1)!!(2n^{'}-1)!!}{2^{l^{'}+m^{'}+n^{'}}} \times (P_x - x^{'})^{l-2l^{'}} (P_y - y^{'})^{m-2m^{'}} (P_z - z^{'})^{n-2n^{'}} \times I^{'} ,
\end{aligned}
\tag{39}
$$

where

$$
\begin{aligned}
I^{'} = & \int_0^{\infty} \frac{s^{L-2L^{'}} s^{-1/2} e^{-\frac{\gamma s |\mathbf{r}^{'}-\mathbf{P}_r|^2}{(\gamma+s)}}}{(\gamma+s)^{L-L^{'}}(\gamma+s)^{3/2}} ds = \\
= & \sum_{h=0}^{L^{'}} \frac{(-1)^h}{\gamma^{L-L^{'}+h+1.5}} \times C(L^{'},h) \times \\
& \times \frac{\Gamma\left[L-2L^{'}+h+0.5\right] \Gamma_{inc}\left[(L-2L^{'}+h+0.5), \gamma(\mathbf{r}^{'}-\mathbf{P}_r)^2\right]}{|\mathbf{r}^{'}-\mathbf{P}_r|^{2(L-2L^{'}+h)+1}} .
\end{aligned}
\tag{40}
$$

In the expression above we have introduced $L = l + m + n$, $L^{'} = l^{'} + m^{'} + n^{'}$, Γ is the gamma function and Γ_{inc} is the incomplete gamma function. A full analytical solution of the above integration can be found in the appendix section. Finally by using Eq.(40)

and Eq.(39) in Eq.(26) the expression of the Coulomb matrix can be written as

$$\begin{aligned}\phi_{12} = N_{1,2}K &\sum_{l=0}^{(l_1+l_2)} \sum_{m=0}^{(m_1+m_2)} \sum_{n=0}^{(n_1+n_2)} f_l^x f_m^y f_n^z \sum_{l'=0}^{l/2} \sum_{m'=0}^{m/2} \sum_{n'=0}^{n/2} C(l,2l')C(m,2m')C(n,2n') \\ &\times \frac{\pi(2l'-1)!!(2m'-1)!!(2n'-1)!!}{2^{L'}} \times (P_x - x')^{l-2l'}(P_y - y')^{m-2m'}(P_z - z')^{n-2n'} \\ &\sum_{h=0}^{L'} \frac{(-1)^h}{\gamma^{L-L'+h+1.5}} \times C(L',h) \\ &\times \frac{\Gamma\left[L-2L'+h+0.5\right]\Gamma_{inc}\left[(L-2L'+h+0.5),\gamma(\mathbf{r}'-\mathbf{P}_r)^2\right]}{|\mathbf{r}'-\mathbf{P}_r|^{2(L-2L'+h)+1}} \,.\end{aligned} \tag{41}$$

Above expression depends on the angular orbitals l, m, n, and the shifted coordinates $P_x = \frac{(\alpha_1 A_x + \alpha_2 B_x)}{\alpha_1+\alpha_2}$, $P_y = \frac{(\alpha_1 A_y + \alpha_2 B_y)}{\alpha_1+\alpha_2}$, $P_z = \frac{(\alpha_1 A_z + \alpha_2 B_z)}{\alpha_1+\alpha_2}$. $A_{x,y,z}$ and $B_{x,y,z}$ are the atomic positions. Furthermore, A contracted Gaussian-type orbital ϕ_μ is a linear combination of multiple primitive Gaussian-type orbitals. The general form of a contracted Gaussian in Cartesian coordinates is as follows

$$\phi_\mu(\mathbf{r}) = \sum_{i=1}^{N_{pre}^{\mu}} \chi_i \phi_i(\mathbf{r}) \,, \tag{42}$$

where N_{pre}^{μ} represents the number of primitive Gaussian functions in the contraction, χ_i for a given basis set, as available in PySCF. Finally, the expression for the Coulomb matrix for two contracted Gaussian-type basis functions ϕ_μ and ϕ_ν can be written as

$$\langle \phi_\mu | V_H | \phi_\mu \rangle = \int \rho(\mathbf{r}') \left[\sum_{i=1}^{N_{pre}^{\mu}} \sum_{j=1}^{N_{pre}^{\nu}} \chi_i \chi_j \phi_{ij} \right] d\mathbf{r}' \,. \tag{43}$$

Eq.(43) represents the finally implemented form of the Coulomb matrix in PySCF. One can access the Coulomb matrix responsible for the electron-electron repulsion by providing accurate electron density and orbital information in Eq.(43). For our study, we only implemented this Coulomb matrix computation scheme for the Gaussian basis function up to p orbitals. The entire module is written in Python and integrated with the PySCF electronic structure package. All the analytical derivations to establish Eq.(43) have been computationally implemented in the stated-above Python module. From Eq.(43) it is clear that the evaluation of the matrix elements demands one more integral over $\mathbf{r}^{'}$. This integral is computed numerically via Eq.(44) by applying the quadrature formula with the implementation of incomplete gamma function integral from a standard python-friendly numerical library SciPy (205). The integral is then approximated as

$$\int_a^b f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i) \; . \tag{44}$$

where n is the number of grid points, w denotes the associated weights for gird, and the function f includes all the terms present in Eq.(43) for individual matrix entry, which is the function of $\mathbf{r}^{'}$, evaluated at each grid point. In practice, the atom-centered grid and the weights are obtained from PySCF for a given molecule and a basis. Since the choice of basis function determines the number of orbitals and consequently the total number of matrix elements, so, if any basis function ϕ contains N_{orb} number of orbitals then Coulomb matrix elements $(J_{\mu\nu})$ are governed by the following equation

$$J_{\mu\nu} = \sum_{i=1}^{n} w_i \rho(\mathbf{r}_i^{'}) \left[\sum_{i=1}^{N_{pre}^{\mu}} \sum_{j=1}^{N_{pre}^{\nu}} \chi_i \chi_j \phi_{ij} \right] \; , \tag{45}$$

with the dimension of $(N_{orb} \times N_{orb})$, where the real space electron density, atomic grids, and the associated weights were supplied to compute Eq.(45) for each element of the Coulomb matrix. We obtained the entire matrix responsible for classical Coulomb interaction from the electron density from Eq.(45). It is evident that Eq(45) becomes computationally expensive as the basis or orbital size grows. To minimize this expense, we compute only the upper triangular part $(J_{\mu\nu}^{upper})$ of the symmetric square matrix $(J_{\mu\nu})$. The complete matrix is then constructed by adding its lower triangular part, which is obtained from the transpose of $J_{\mu\nu}^{upper}$ without the diagonal elements, therefore the final version of $J_{\mu\nu}$ matrix follows $J_{\mu\nu} = J_{\mu\nu}^{upper} + [J_{\mu\nu}^{upper} i \neq j]^T$. Moreover, to further enhance the efficiency of our Coulomb matrix evaluation, we have also implemented multi-processing parallelization to compute the Gaussian-type integration responsible for the Coulomb repulsion. However, this Python module is also capable of calculating the Coulomb matrix responsible for spin-polarized systems involving up-spin and down-spin densities. In the following section, we will discuss the construction of Kohn-Sham Hamiltonian from various components as described earlier, but it is crucial to mention here, that the entire module offers the production of DM from real-space electron density for a given GTO basis function (up to p orbitals) for molecules including open-shell systems. This module also tested for varieties of GTO functions and pure density functionals e.g. PBE (25), BLYP (26, 27), etc.

To check the accuracy for developed Python module for the Coulomb matrix a few tests have been conducted for a set of molecules H_2O, S_2O, $C_9H_8O_4$, and $C_4H_4N_2O_2$. For the current discussion, we will consider only one molecule, $C_9H_8O_4$, to analyse the accuracy and the computational efficiency of the molecular integrals and hence the matrix defined in Eq.(45). Fig.4.2 shows the parity plots (in logarithmic scale) of the Coulomb matrix for the Aspirin molecule for computed with $6-31+g$ Gaussian basis set (206, 207). From this plot, we can see an excellent agreement between the matrix elements calculated from the electron density as from Eq.(45), and the PySCF-generated initial DM-based Coulomb matrix. Both the matrices have been obtained from a converged DFT charge density. Here most of the points are speared over the parity lines with a reported mean absolute error (MAE) value in the order of 10^{-6} in a.u. Furthermore, the Python module that we have developed, achieves computational efficiency through the implementation of parallel processing.

Here, Fig.-4.3 presents a measure of the computational efficiency of our Coulomb matrix evaluation scheme for a single aspirin ($C_9H_8O_4$) molecule. From Eq.(45), the final integration needs to be performed over the $\mathbf{r}'$ grid, which means it is a grid-based numerical calculation, and the computational cost will increase as the system size

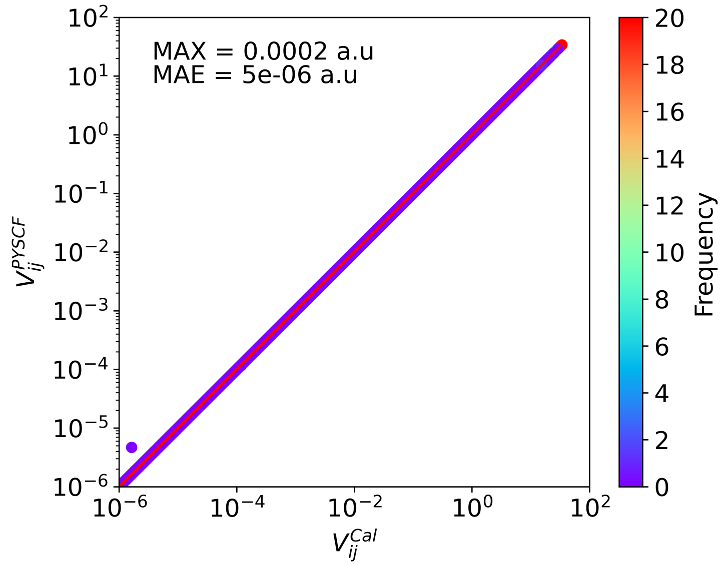

Figure 4.2: Parity plots for the Coulomb matrix for a single $C_9H_8O_4$ molecule. Along the X-axis we present calculated Coulomb matrix elements ($V_{ij}^{Cal} = J_{\mu\nu}^{Cal}$) obtained from the electron density. The same quantity ($V_{ij}^{PySCF} = J_{\mu\nu}^{PySCF}$) computed from PYSCF is shown along the Y-axis. Note this parity plot is in logarithmic scale (we plot $|V_{ij}| = |J_{\mu\nu}|$). The colour code describes the density of the given V_{ij} matrix elements. Maximum (MAX) and mean absolute (MAE) deviations are indicated in the plot.

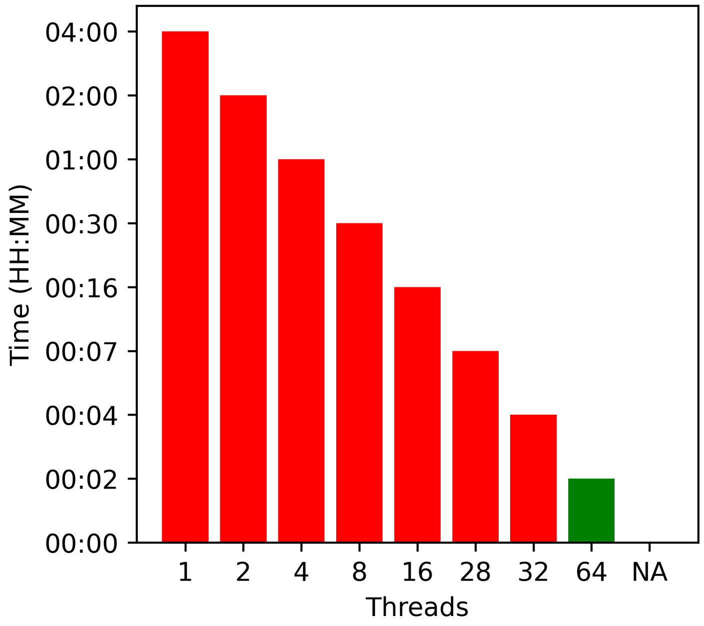

Figure 4.3: Evaluation of Coulomb matrix for $C_9H_8O_4$ over the different number of CPU-threads by utilizing all electron charge density. The X-axis follows the number of threads whereas the Y-axis presents computation time for each thread configuration.

grows, and for aspirin molecule considered the grid size is approximately 2.54×10^5. This means that each matrix element ($J_{\mu\nu}$) has to be integrated out through 2.54×10^5 mesh grid points. As already mentioned, we have used $6-31+g$ basis set, which yields the total number of matrix elements is 185 × 185. Furthermore, since, as we discussed before, the Coulomb matrix is symmetric, we have explained, computed only the upper diagonal of the Coulomb matrix (17205 elements) and then constructed the full one. In -Fig.4.3 the x-axis represents the number of threads responsible for computing the Coulomb matrix. The Y-axis shows the corresponding time consumption in the ($hours : minutes$) format. When the number of threads is set to 1, evaluation takes 4 hours, for two threads it takes 2 hours, for 4 threads it takes 1 hour, for 16 threads it demands 16 minutes and finally, for 64 threads (green bar) the calculation of the Coulomb matrix finishes within 2 minutes. It is evident from the given evaluation pattern that significant efficiency is achieved as we increase the number of CPUs.

4.5 DM from JL-Charge Density Model

This is our second approach, where we already discussed how we constructed DM from the electron density $\rho(\mathbf{r})$. Our scheme here directly deals with the real-space representation of the ρ, defined on a Cartesian grid. This is obvious that ρ is available in any DFT code. Its key advantage lies in the fact that the electron density at a given grid point is rotationally invariant with respect to the external potential, i.e., the positions of the surrounding nuclei. This allows one to build ML models that predict point by point, using as descriptors the local atomic environment within a chosen cutoff radius. The effectiveness of this grid-based strategy depends strongly on the representation of the local environment and the learning method. While a single DFT calculation can produce tens of millions of grid points, suggesting that generating large training datasets is straightforward, the data often contain redundancy and limited diversity, as only a narrow range of external potentials is sampled. To address this, multiple configurations of the same system are typically used. In practice, this usually leads to very large neural networks with millions of trainable parameters, which are computationally heavy and tend to have limited transferability, as we discussed before for DMs (Chapter 3). Our primary objective is to reformulate the grid-based strategy into a lightweight framework that can be applied broadly across DFT calculations. This is accomplished by substantially reducing computational overhead while maintaining very high accuracy. Specifically, we introduce a grid-centred representation of atomic structures constructed from Jacobi and Legendre polynomials, which have previously been employed in the development of efficient ML force fields. The JL representation enables a linear regression formulation for the charge density, in which many-body contributions of varying orders are explicitly separated. The outcome is a highly compact model requiring only a small number of coefficients, trained on a limited subset of the available grid points. For brevity, we refer to this family of models as Jacobi–Legendre charge density models.

The Jacobi–Legendre Charge Density Model differs from neural networks in how it handles symmetry. Since the charge density is a scalar field, the problem is inherently rotationally and translationally invariant, meaning JLCDM only needs to respect invariance. This is achieved analytically through Jacobi and Legendre polynomials, which encode the local environment in a symmetry-preserving way. In contrast, neural networks often rely on equivariant architectures to learn how vector or tensor quantities transform under rotations, making them flexible but computationally heavy, which we have used for DM prediction, as discussed in the previous chapter. JLCDM's invariance-based design keeps the model compact, lightweight, and transferable, while avoiding the overhead of large equivariant neural networks. In our deep learning-based DM prediction approach, we demonstrated that using a neural network (NN) prediction ML DM guess can accelerate the SCF algorithm, without sacrificing the accuracy, subject to reducing the rotational degrees of freedom for DM. However, this deep learning machine-learning scheme demands a significant number of molecular structures to train the NN model. In contrast, the linear ML charge density model (JLCDM) offers an alternative pathway for predicting the real-space charge density, requiring only a few molecular structures for training. Additionally, the atomic descriptor-based JLCDM method generalizes the workflow by removing the constraint of rotational invariance for DM. As outlined earlier, except for the Coulomb matrix (J), all the other matrix components necessary to construct the Fock matrix can be obtained from PySCF by providing the accurate basis information and real-space electron density. Thus, finally, the implementation of Eq.(45) with PySCF provides the option to construct the Fock matrix defined in Eq.(17) using the real-space electron density. Hence, the DM can then be computed through diagonalization of the KS-Hamiltonian (Fock matrix). In brief, this approach enables restarting DFT calculation from the real-space electron density. For this approach, it is evident that the accuracy of the real-space electron density is crucial. If the predicted charge density is sufficiently accurate, the resulting DM will be close to the converged DM. Thereafter, this matrix can then serve as an efficient initial DM guess for the DFT algorithm, allowing non-self-consistent density functional calculation without compromising reliability. In this study, ML-predicted density has been used to build the KS-Hamiltonian and hence the DM. In the next section, we will demonstrate the prediction of real-space electron density using the linear Jacobi-Legendre (JL) atomic descriptors. The efficiency and the accuracy of our scheme are demonstrated via the Uracil molecule. In particular, we show that the SCF scheme can be bypassed to compute DM, total energies, and the atomic forces via the PySCF package. In order to validate our entire approach based on the machine-learned real-space electron density, we conduct three different evaluations. Firstly, we assess the accuracy of the predicted density and the resulting DM using parity plots. Then by using these DMs as initial DM guesses, we compare the accuracy of non-self-consistent atomic forces with the same derived from fully converged DFT calculations. Finally, we evaluate the performance of the non-self-consistent MD run for the various observables, like the HOMO-LUMO gap, the dipole moments, and infrared spectra. In particular, we compare results where the MD is driven by the forces associated with the machine-learned DM (ML DM), with those obtained from fully self-consistent AIMD simulations at different temperatures.

4.6 Preparation of the Data Set for ML MD

We have applied our machine learning-derived density matrix (ML DM) approach to a single, isolated uracil ($C_4H_4N_2O_2$) molecule. All the electronic structure calculations, including training and test data generation and $ab-initio$ molecular dynamics (AIMD), were carried out using DFT. The second-order mixing scheme (SOS) (51, 52) as implemented in PySCF was utilized for density functional calculations, while LAMMPS (7) was used for AIMD simulations. We used the $'6-311++G'$ Gaussian type basis functions (206, 207), as implemented in PySCF, and the Local-density approximation (LDA, VWN) (17). Additionally, for all of our calculations, we utilized the Compact Effective Core Potential (ECP) (153, 154, 155, 156, 157) with a relativistic nodal boundary condition, as included in PySCF for carbon (C), nitrogen (N), and oxygen (O) atoms. In order to train the ML models, data for uracil [$C_4H_4N_2O_2$] molecule were extracted from the various MD trajectories initiated with random initial velocities and simulated at 100 K and 300 K up to 3.5 picoseconds (ps), with a time-step of 10^{-4} ps. To ensure randomness, the fully converged electron densities of the 46 uncorrelated structures from both the 100 K and 300 K simulations were included in the training set for two distinct ML-charge density models addressing two different MD simulation temperatures. Note that the ML model trained at 300 K was also used to simulate the 200 K MD trajectory. To evaluate the test performance of the JL-charge density model 50 different arbitrary snapshots from both 100 K and 300 K simulations were obtained from the 20.0 ps long AIMD trajectories, precisely between 3.0 ps and 6.0 ps, whereas, for the 200 K investigation, the time frame was set between 5.0 and8.0 ps, using the time-step as 10^{-3} ps. To prepare the training data, we computed the DM across the MD trajectory, and then the electron density for selected structures is computed through PySCF. As the DM to the real-space electron density conversion algorithm implemented in PySCF. Which follows: (1) the converged DM, which we computed across the AIMD trajectory; (2) orbital information, obtained from PySCF; (3) the corresponding grid representations, also obtained from PySCF. All the MD runs

Table 4.1: Table summarizing the optimized parameters used for the JL charge density model. Here we report the number of different atoms present in the molecule, # Species, σ is the shape of the probability distribution used for grid sampling, single (1b) and pair atomic (2b) contributions to the electron density, *Body*, cut-off radius from a specific grid point, r_{cut}, the term responsible for angles situated for two vectors in the grid space,l_{max}, r_{min} is the distance shift parameter, α, β are the parameters accountable for the shape of polynomials, and #features is the number of features was required to feed the ML model associated with each grid point. The unit of r_{cut} is in Å

Molecule	# Species	σ	Body	r_{cut}	n_{max}	l_{max}	r_{min}	α	β	# features
$C_4H_4N_2O_2$	4	90.0	$1b$	2.90	60	-	0.0	1.0	1.0	6180
-	-	-	$2b$	2.90	12	5	0.0	1.0	1.0	

Table 4.2: Table demonstrating the performance of the JL-charge density model and the associated DM for 50 random Uracil structures from the test set. Here MAE is the mean absolute error associated with predicted density and the corresponding DM (includes one SCF iteration), δ_{max} is for the highest error in the predicted density, and DM, the root-mean-square error is (RMSE), and the coefficient of determination R^2 of the density, for two different temperatures. Here, errors are specified for the test sets and the units are in atomic units (a.u)

Observables	Temp.(K)	MAE	δ_{max}	RSME	R^2
Real-space Density (ρ)	100	1.10×10^{-3}	0.0835	2.79×10^{-3}	0.9997
Real-space Density (ρ)	200	1.50×10^{-3}	0.1077	3.70×10^{-3}	0.9995
Real-space Density (ρ)	300	1.60×10^{-3}	0.1012	3.89×10^{-3}	0.9994
Density Matrix (DM)	100	5.75×10^{-5}	0.1022	4.27×10^{-4}	-
Density Matrix (DM)	200	2.10×10^{-4}	0.4933	2.05×10^{-3}	-
Density Matrix (DM)	300	3.70×10^{-4}	0.5274	2.64×10^{-3}	-

are performed using the **Nosé–Hoover** thermostat (168) in the NVT ensemble.

To examine the test performance of JLCDM, the predicted electron density was provided to PySCF for constructing the KS-Hamiltonian. The KS-Hamiltonian matrix was then diagonalized to yield the DM, using the algorithm available in the PySCF numerical library. The hyperparameter of the JLCDM has been optimized by varying its values, resulting in minimizing the mean absolute errors (MAE) and the maximum errors associated with the electron density. The optimal set of JL parameters for the Uracil molecule is reported in Table 4.1. The different types of errors related to the density prediction and the DM construction are summarized in Table 4.2. Note that we only obtained the R^2 coefficient for the electron density, not for the DM, because, in this study, the only entity predicted by the ML model is the electron charge density; then, we obtain the DM from the approach we discussed before. For the errors reported in Table-4.2 primarily based on ML density, and then we also computed errors related to the constructed DM from the JL-charge density model. Furthermore, to compute the DM, we added one SCF iteration, because, as we have mentioned before, incorporating a single SCF step in the NN-based ML DM approach helps prevent the issue of obtaining a slightly lower ground-state energy compared to the fully converged DFT result. Notably, Table-4.2 demonstrates a strong agreement of the ML predicted density with the DFT results, also indicating that parity fitting for density matrices will have an R^2 close to 1.

4.6.1 Grid-point sampling strategy

Referring to one recent publication (113), a strategy was introduced for sampling grid points to effectively minimize redundancy. It is often observed in molecules that numerous grid points are outside the molecule and are distributed across the grid. Our approach involves selectively utilizing a small fraction of the total grid, which diversely includes essential information about the electron density. The sampling scheme follows the following probability distribution $\exp\left[-\frac{(1/\rho(\mathbf{r}))^2}{2\sigma^2}\right]$, where the parameter σ controls shape of the probability distribution. Larger values of σ are responsible for border and flattened distributions, while the small values make the distributions sharp. In practice, using our chosen basis function, the charge density of the $C_4H_4N_2O_2$ molecule is represented across $152,320$ atom-centred grid points, with the corresponding density matrix having the dimension of (152×152). However, by following the aforementioned grid sampling strategy, we trained our charge density machine learning model on $10,012$ grid points, which is only 6.57% of the total grid data. Since all the DFT calculations utilized the ECP potential, our sampling scheme includes more grid points away from the nucleus and fewer points towards the nucleus. However, a few grid points where the atoms are present in $C_4H_4N_2O_2$ were added to the sampling approach to ensure the proper inclusion of the atomic positions and to capture the higher density values. Thereafter, we supplied the model with multiple batches of grid data, each containing 10,012 points, to obtain predictions for the entire mesh grid.

4.7 Parity Plots for the Electron Density and DM

In Fig-4.4 we show the parity plots along with their colour bars to show the quantitative information about the elements for the electron density and the associated DM for two different AIMD runs. Panel (a), (c), and (e) show the ML-predicted real-space charge density from 100 K, 200 K, and 300 K MD simulations, respectively. Panel (b) and (d) display the corresponding DMs, obtained from the MD trajectories. Each plot reports the mean absolute error (MAE) values. The parity plots bring a direct comparison between the ML-predicted electron density (x-axis) and fully converged DFT electron density (y-axis) [panel(a, c and e)].

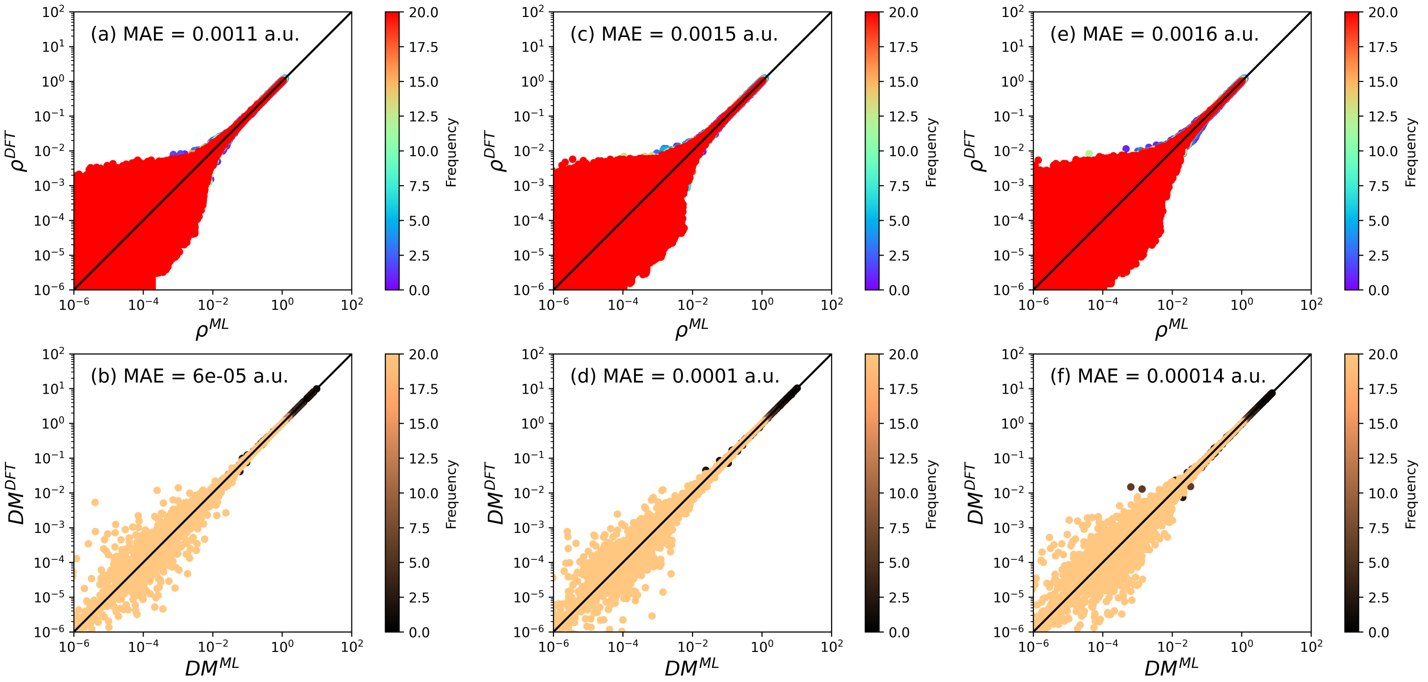

Figure 4.4: Parity plots for the real-space electron charge density [panels (a), (c), and (e)] and the associated DM [panels (b), (d), and (f)] for the $C_4H_4N_2O_2$ molecule over AIMD trajectories at three different temperatures from the test set. Panels (a) and (b) represent density and DM at 100 K, respectively. Panels (c) and (d) show the same observables at 200 K, while panels (e) and (f) depict the ML-predicted electron density corresponding DM at 300 K. Note that, these parity plots are in logarithmic scale, we show the absolute value for DM elements ($|DM_{i,j}|$). The colour bar represents quantitative information about elements present in the given electron density and DM. Each plot reports MAE values.

For DMs, the same comparisons are applicable along the x-axis and y-axis [panel(b, d, and f)], where the ML DMs were constructed from the predicted charge density as mentioned in previous sections. In this case, one self-consistent field (SCF) iteration was included to obtain the DM from the Fock matrix; the reason behind adding one SCF step is already discussed before and in the ML DM prediction section. It is evident from these parity plots that most of the data points larger than $\sim 10^{-2}$ are aligned along the parity line for both the densities and the DMs. The reported amount of deviations (MAE) and other error metrics [see Table 4.2] for DM compared to the benchmark results are sufficiently low to restart a DFT calculation without sacrificing accuracy (177). In terms of the efficiency of this approach, the average energy is 3.08×10^{-5} Ha (Hartree is the default energy unit in PySCF) away from the converged results for 100 K, while for 200 K, run it is 5.97×10^{-5} Ha, and the amount is 0.0013 Ha for 300 K. Moreover, the average percentage variation of the DMs at the first iteration with respect to the ground-state energy is 0.98%, 1.77%, for 100 K and 200 K trajectories, respectively, while for the 300 K simulation mean % difference between converged DM and ML DM (with one SCF) is 2.32%. The percentage variation formula for two different matrices is defined before.

4.8 Non-self-consistent Forces from ML DM

In the previous section, we have demonstrated how our machine-learned real-space charge density agrees well with the fully converged DFT results. This finding motivates our further investigation: determining the accuracy of the predicted atomic forces computed from the ML DM. In Fig.-4.5, the parity plot diagrams for the comparison of the atomic forces between those obtained from ML MD with the fully converged ones from AIMD, at two different temperatures. Here, panels (a) and (b) represent the forces along the x-direction for simulations at 100 K, and 200 K, respectively. Panels (c) and (d) show the forces along the y-direction for 100 K and 200 K simulations, while panels (e) and (f) present the forces along the z-direction for 100 K and 200 K simulations, respectively. As detailed in the methodological section, these are the computed values for 50 distorted random structures of uracil, extracted from a 20 ps long MD trajectory at 100 K and 200 K, where those structures are not included in the training set. For the simulation time, It is also important to mention that in a previous study they have shown that relatively short ab initio MD trajectories are sufficient for reliable vibrational spectra of nucleobases: using a MD trajectory of about 7 ps for uracil in aqueous solution, finally the study was able to compute the IR spectrum from the Fourier transform of the dipole–dipole autocorrelation function with good agreement to experiment (208).

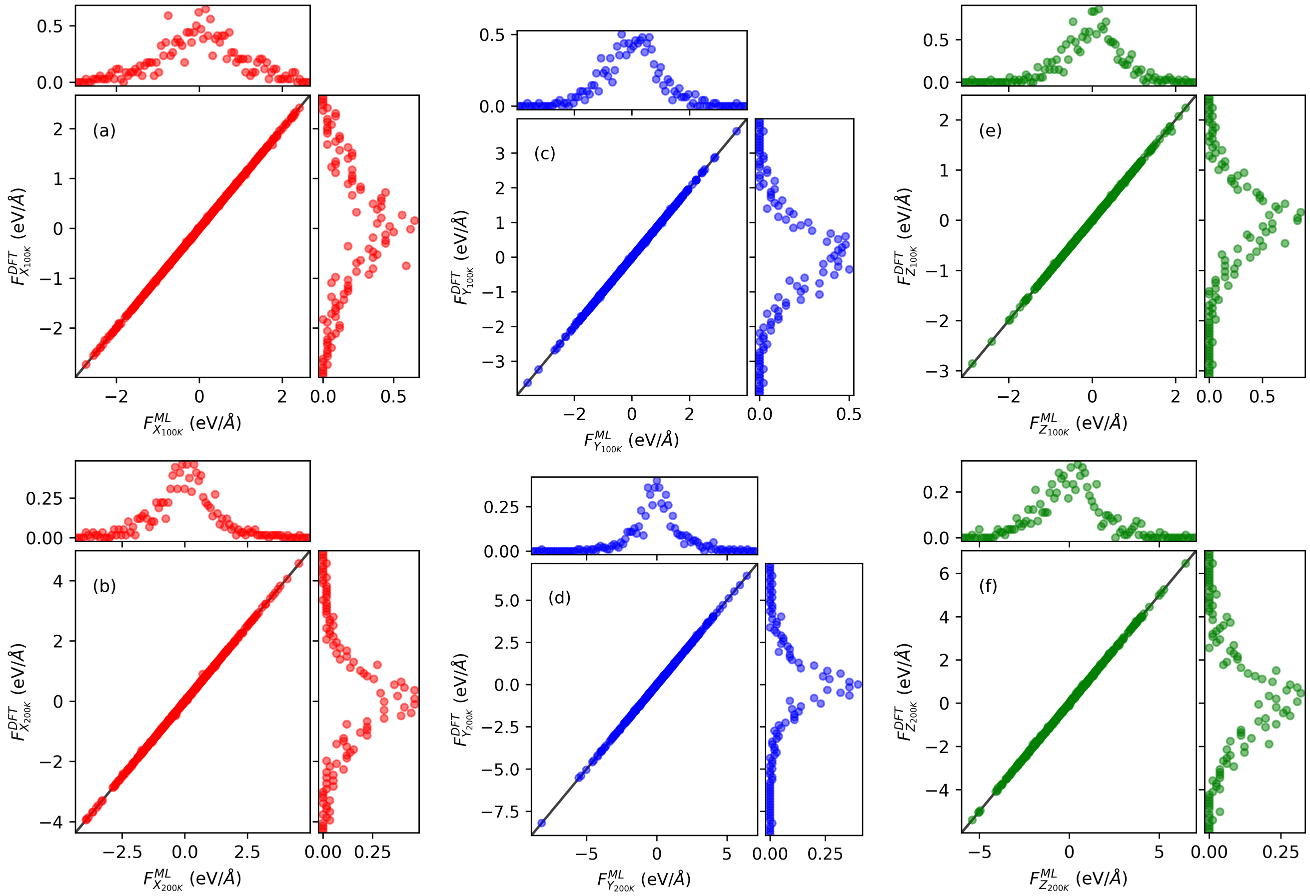

Figure 4.5: Parity plots for the various atomic forces, $\alpha = x, y, z$ component. Here, F^{ML}_{α} represents forces associated with non-self-consistent DFT calculations, while F^{DFT}_{α} denotes the forces obtained from fully self-consistent KS-DFT calculations, for 50 random distorted uracil configurations from the test sets. The upper panels (a), (c), and (e) display the x,y, and z components of the forces computed from the 100 K simulation, and the lower panels (b), (d), and (f) depict the atomic forces extracted from the 200 K run along x,y, and z directions, respectively. The parity plots also show the corresponding histogram describing the distribution of the forces.

For each parity plot, the x-axis shows the forces calculated from the ML DM guess, while the y-axis is for the atomic forces obtained from fully converged DFT calculations. It is noticeable that most points in Fig.-4.5 are situated closely along the parity line. Additionally, these graphs include histograms showing the force distributions for each component. Note that, by using the ML DM scheme, the maximum deviation with the DFT forces can be found as 91.89 meV/Å for the x component, 69.00 meV/Å for y, and 134.90 meV/Å for z at 100 K simulation, while 189.50 meV/Å along x, 98.70 meV/Å for y, and 129.10 meV/Å along z at 200 K. From these plots, it is evident that the atomic forces obtained from ML DM guess show an excellent agreement with those coming from the fully self-consistent KS scheme. The MAE of all other forces are very low. Specifically, for the x-components the MAE is 2.80 meV/Å at 100 K and 5.2 meV/Å at 200 K. For the y-components, the MAE is 3.30 meV/Å at 100 K run, while for 200 K simulation, it is 5.6 meV/Å , and for the z-components, it is 2.10 meV/Å for MD trajectory at 100 K and 8.6 meV/Å at 200 K.

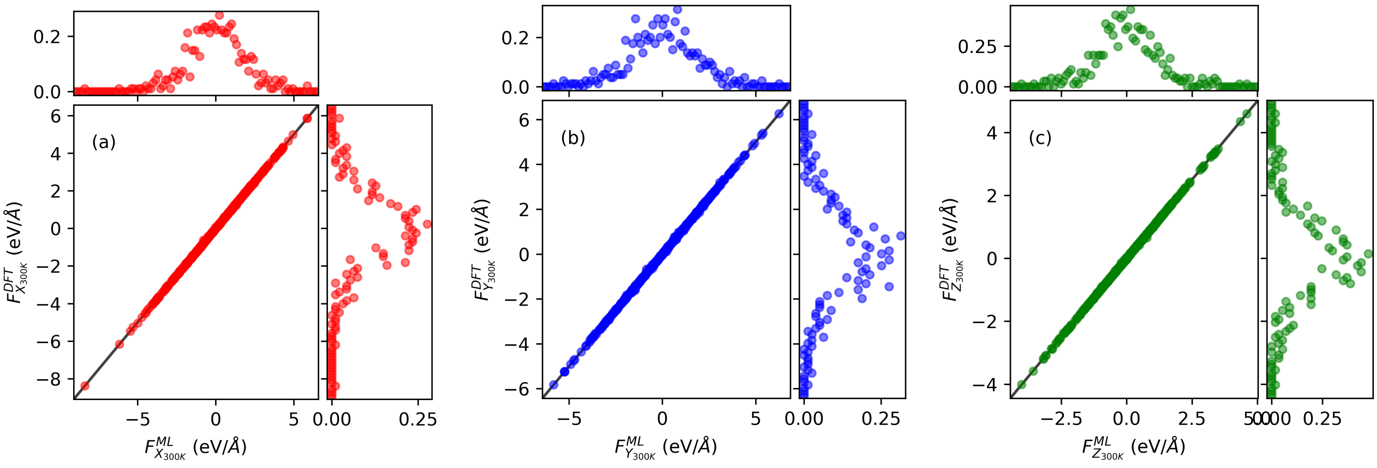

Figure 4.6: Parity plots for the atomic forces, $\alpha = x, y, z$ component, for 300 K MD simulation. Here, F^{ML}_{α} represents forces associated with non-self-consistent DFT calculations, while F^{DFT}_{α} denotes the forces obtained from fully self-consistent KS-DFT calculations, for 50 random distorted Uracil configurations from the test sets. Panels (a), (b), and (c) display the x,y, and z components of the forces, respectively. The parity plots also include the corresponding histogram detailing the distribution of the forces.

Fig.-4.6 depicts parity plots of the atomic forces computed within 300 K MD trajectory using ML DM non-self-consistent DFT and fully converged DFT results. Panels (a) represent forces along the x-direction, panel (b) those values in the y-direction, and panel (c) the z-component. For each panel, the x-axis contains the force values calculated using the ML DM guess, while the y-axis represents the forces obtained from the AIMD run for the same structures. The ML DM scheme exhibits maximum error with the DFT forces of 177.80 meV/Å along x, 180.20 meV/Å for y, and 118.30 meV/Å along the z-component. However, the MAE observed is 10.7 meV/Å for the y-components of forces [panel (b)], while for the x-components, the MAE deviation is 7.8 meV/Å , and for the Z component it is 5.10 meV/Å. Those errors are extremely low, which means that the ML DM applied non-self-consistent approach can produce the DFT quality atomic forces. Moreover, in the previous chapter, we have detailed the results for atomic forces obtained from NN-based ML DM, and the magnitude of errors is not far from these results. Therefore we can implement this JL-density-based ML DM workflow to conduct the non-self-consistent MD simulation (ML MD). This approach eliminates the need for full self-consistent KS DFT calculations for an MD simulation. This state-of-the-art workflow gives motivation to inspect the next section.

4.9 Non-Self-Consistent Molecular Dynamics

In this section, we discuss the performance of ML DM's initial guess to drive MD trajectories at two different temperatures: 100 K and 200 K. As mentioned in the method section, for the 200 K non-self-consistent MD simulation, the ML-charge density model (JLCDM) was trained using data obtained from MD trajectories at 300 K. To verify the accuracy of the ML MD approach, we compare here the infrared (IR) spectra of the uracil molecule obtained from the fully converged $ab-inito$ run with the simulation performed using the ML-based DM guesses at the different temperatures specified above. To compute the IR spectra, we perform a Fourier transformation (FT) on the scalar product between the dipole moment vectors computed at time t and $(t+\delta t)$ from the MD trajectories. Here δt denotes the time step (10^{-3} ps) using the MD simulations. However, the dipole moments $\boldsymbol{\mu}(t)$ in each MD step were computed using the PySCF package, which demands orbital information, DM and nuclear positions at each step of the MD run. The following equations for the dipole moments implementation in PySCF describe the $\boldsymbol{\mu}$ along x, y, and z directions

$$\mu_x = -\sum_{\alpha}\sum_{\beta} P_{\alpha\beta}\langle\alpha|x|\beta\rangle + \sum_{A} Q_A X_A \tag{46}$$

$$\mu_y = -\sum_{\alpha}\sum_{\beta} P_{\alpha\beta}\langle\alpha|y|\beta\rangle + \sum_{A} Q_A Y_A \tag{47}$$

$$\mu_z = -\sum_{\alpha}\sum_{\beta} P_{\alpha\beta}\langle\alpha|z|\beta\rangle + \sum_{A} Q_A Z_A \tag{48}$$

where, μ_x, μ_y, μ_z are the x y and z components of dipole moment, $P_{\alpha\beta}$ is the DM, the integral represents the expectation values of the position operator $\langle\alpha|r(x, y, x)|\beta\rangle$ with respect to the occupied orbitals α and β. Finally Q_A are the nuclear charges and X_A,Y_A, and Z_A are the nuclear positions. Furthermore, mathematically, the integral form of the dipole autocorrelation functions for FT is as follows

$$\tilde{\mu}(\omega) = \int_{-\infty}^{+\infty} \langle \boldsymbol{\mu}(t) \cdot \boldsymbol{\mu}(t + \delta t)\rangle e^{-i\omega\delta t} d\delta t \tag{49}$$

Subsequently, the intensity (infrared absorption intensity) of the IR spectra at temperature T (209) can be determined by

$$I(\omega) \propto |\tilde{\mu}(\omega)|^2 \left[1.0 - \exp^{\frac{-\hbar\omega}{K_B T}}\right] \;, \tag{50}$$

Where, ω is the angular frequency of the IR radiation, $\tilde{\mu}(\omega)$ is the FT of the autocorrelation function from Eq.(49), $\hbar$ is the reduced plank's constant, ω is the angular frequency of molecular vibration, K_B is Boltzmann's constant, T is the temperature of simulation. Finally, the corresponding wavenumber (ν), written as $\nu = \frac{\omega}{c}$, where c is the speed of light. The frequency of absorption corresponds to the vibrational frequency of molecular bonds and is typically measured in wavenumber ν with unit (cm^{-1}). The factor $\left[1.0 - \exp^{\frac{-\hbar\omega}{K_B T}}\right]$ accounts for the Bose-Einstein distribution in the context of vibrational mode. It reflects that at higher temperatures more vibrational modes are present leading to increased absorption. In Fig.-4.7 we present the IR spectra for the uracil molecule calculated using two MD trajectories at different temperatures. Panels (a) and (b) represent the spectra obtained from a 20-ps-long 100 K simulation, while panels (c) and (d) show the results for the same observable, from a 20-ps-long 200 K MD run. In all panels, the x-axis represents the IR frequency and the y-axis indicates the corresponding relative intensities as described by Eq.(50). For the 100 K trajectory, panel (a) shows the fully converged self-consistent DFT spectra, while panel (b) illustrates the non-self-consistent spectra driven by the ML DM guesses. From panels (a) and (b) it is appreciable that there is no significant difference between the results observed from two different workflows. The various peaks concerning the vibrational frequencies spotted from the ML results are closely aligned with the DFT ones. The highest peak for relative intensity was observed from the fully self-consistent $ab - initio$ MD at 1736.37 (cm^{-1}). In comparison, the ML workflow has captured the highest peak at 1733.04 (cm^{-1}), which shows good agreement with the simulation performed using ML-driven DM guesses non-self-consistently. This highest peak is coming due to $C = O$ bond stretching and $N - H$ or $C - H$ bond bending (8). In addition to the largest peak, the smaller peaks around the frequencies of 500 (cm^{-1}), 1000 - 1500 (cm^{-1}), and 3500 (cm^{-1}) are also situated similarly in both obtained from the fully converged DFT [panel(a)] and the accelerated ML DM simulation [panel (b)]. Similarly, the remaining panels [(c) and (c)] illustrate the spectra results from the 200 K simulation. Panel (c) represents vibrational spectra obtained from ML-MD trajectory whereas on the other hand panel (d) shows the same IR spectra but computed through AIMD simulation. It is also clear that for a 200 K run, the ML approach is compatible as a substitute for AIMD workflow.

In this run, using the AIMD approach, the maximum peak was observed at 1716.49 (cm^{-1}) whereas the ML MD approach put this at 1718.22 (cm^{-1}). It is clear that the spectra obtained from non-self-consistent workflow at 200 K exhibits more noise than the 100 K run. This trend is also observed in the AIMD simulations. The high-temperature MD experiences larger thermal fluctuations, anharmonic atomic vibrations, faster dynamics, etc, which lead to greater atomic movement and less stable configurations.

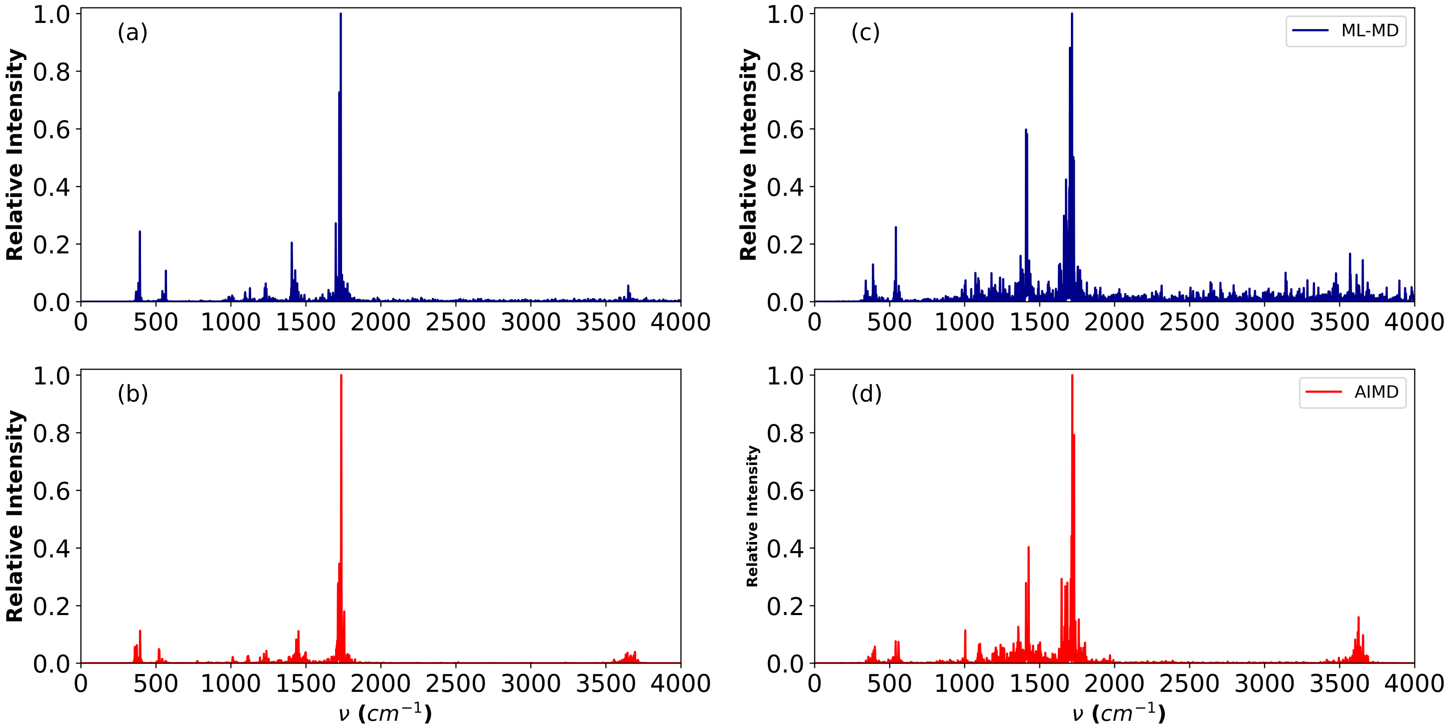

Figure 4.7: IR spectra plots obtained from various MD trajectories. Panel (a) and panel (b) represent the IR spectra calculated from ML DM-driven non-self-consistent trajectory (ML-MD) and AIMD run, respectively, at 100 K. Similarly, panel (c) and Panel (d) display the IR spectrum computed from ML-MD run and the fully self-consistent AIMD trajectory, respectively, at 200 K. In all four panels, the x-axis is for the vibrational frequencies in the unit of (cm^{-1}), while the y-axis reports the associated relative intensities. For better visualization, we shift the magnitude of intensities (y-axis) to the range of [0, 1.0] interval.

These factors introduced more variability in the spectra we obtained for 200 K, indicating the common characteristic of high-temperature MD trajectories. In contrast, for 100 K atomic motion is less pronounced which provides greater stability, leading to clearer, more defined spectra. Furthermore, to make a comparison between the results for the IR spectra reported from our fully converged effective core potentials (ECP) based DFT study and one previously reported DFT results for uracil, we show in Fig-4.8 the IR spectra reproduced from a paper where an electron-based density functionals investigation in contrast to the experiment has been reported (8). However, here our main motive is to compare the results between DFT and ML DM workflows not with the experiments, thus we did not include experimental data in Fig.-4.8. For this result, they performed DFT calculations with the Becke exchange (210) and double-zeta plus polarization orbital basis sets (211). Here, the infrared absorption intensities have been calculated by taking the numerical derivatives of the dipole moment and transforming them to the corresponding ones with respect to the normal modes (212). Therefore, one can see our full self-consistent ECP-DFT study and the strategy followed by reference do not produce the same results for the IR spectra. It is evident, that some peaks are missing from our ECP-based fully self-consistent DFT computed results in between the frequency range of 700 (cm^{-1}) to 1000 (cm^{-1}) [Fig.(4.9), panel (b)]. However, rather than the peaks between 700 (cm^{-1}) to 1000 (cm^{-1}) frequency region, all the other peaks are present in our calculated IR-spectra using AIMD or via ML MD schemes [Fig-50].

This range (cm^{-1}) to 1000 (cm^{-1}) is accountable for depicting out-of-plane vibrations and contributes to the formation of the ring structure for the uracil molecule. Since our ECP-based DFT approach fails to produce those peaks, it is quite obvious that they will also be absent from the ML DM-based non-self-consistent approach. Therefore, we have conducted an all-electron DFT MD simulation using the 'BLYP' functional (26, 27) at 100 K to investigate these discrepancies. In this simulation, we have observed that in the before-specified frequency zone, a few previously missing peaks are present, see Fig.-4.9, panel (a). This suggests that there is (1) either an issue with the ECP potential implemented in PySCF; (2) that more advanced DFT function-

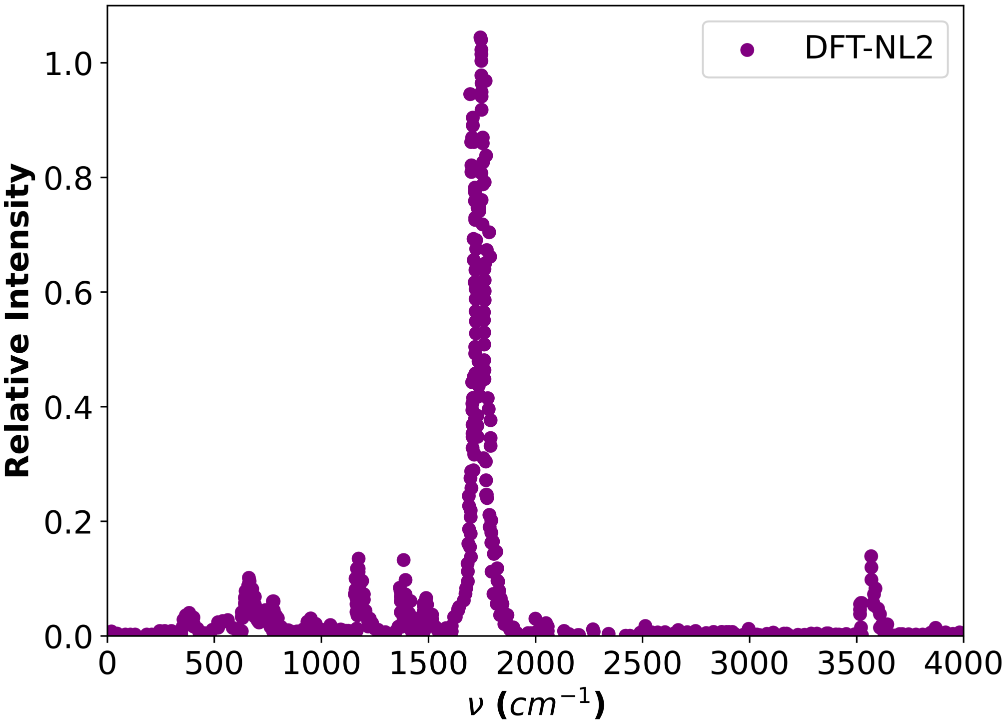

Figure 4.8: IR spectra plot, reproduced from the ref.(8). These spectra have been calculated using DFT with NL2 (non-local) density functionals. The x-axis is for the vibrational frequencies in units of (cm^{-1}), while the y-axis reports associated relative intensities.

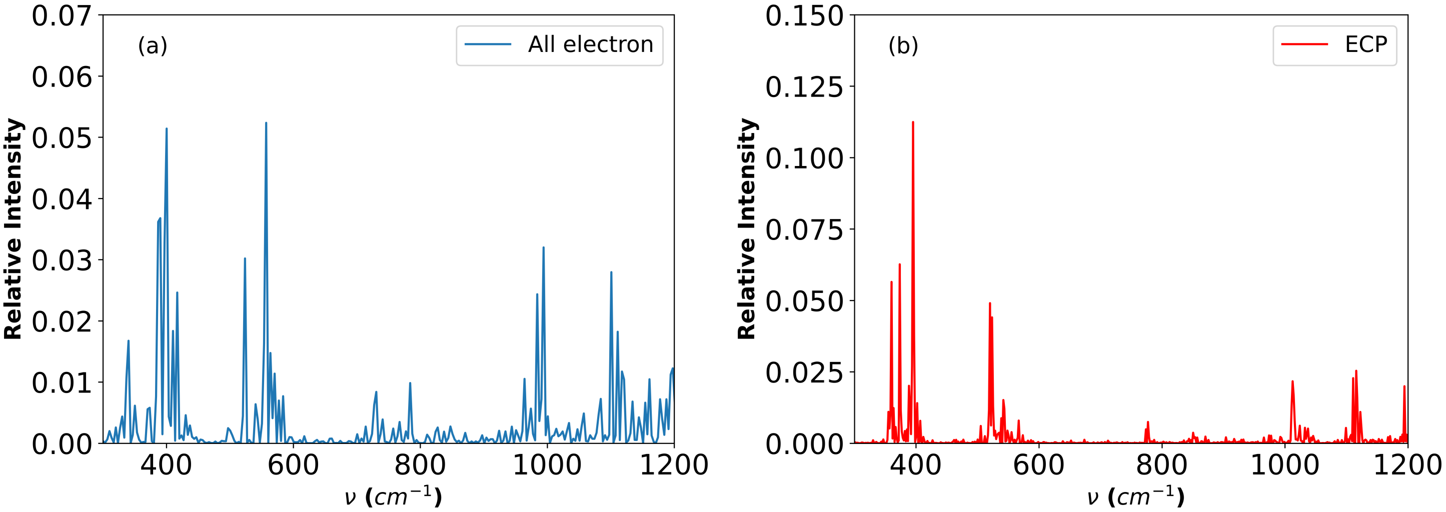

Figure 4.9: IR spectra obtained from two different AIMD trajectories are shown here, The left-hand side plot from all-electron DFT with BLYP functional. The other one is obtained from effective core potential (ECP) operated DFT with LDA functional. For both panels, the x-axis represents the vibrational frequencies in units of (cm^{-1}) while the y-axis depicts the relative intensities. Both the MD simulations were conducted at 100 K.

als can capture those peaks precisely; (3) that those peaks may merge with others in the frequency spectrum; (4) higher temperature MD may able to capture those peaks. Since our ML DM approach for non-self-consistent MD maps well to the fully self-consistent results, these deviations are expected. Because the ML method can not perform better than the DFT approach it is based on. Further investigation related to these missing peaks is beyond this study's scope.

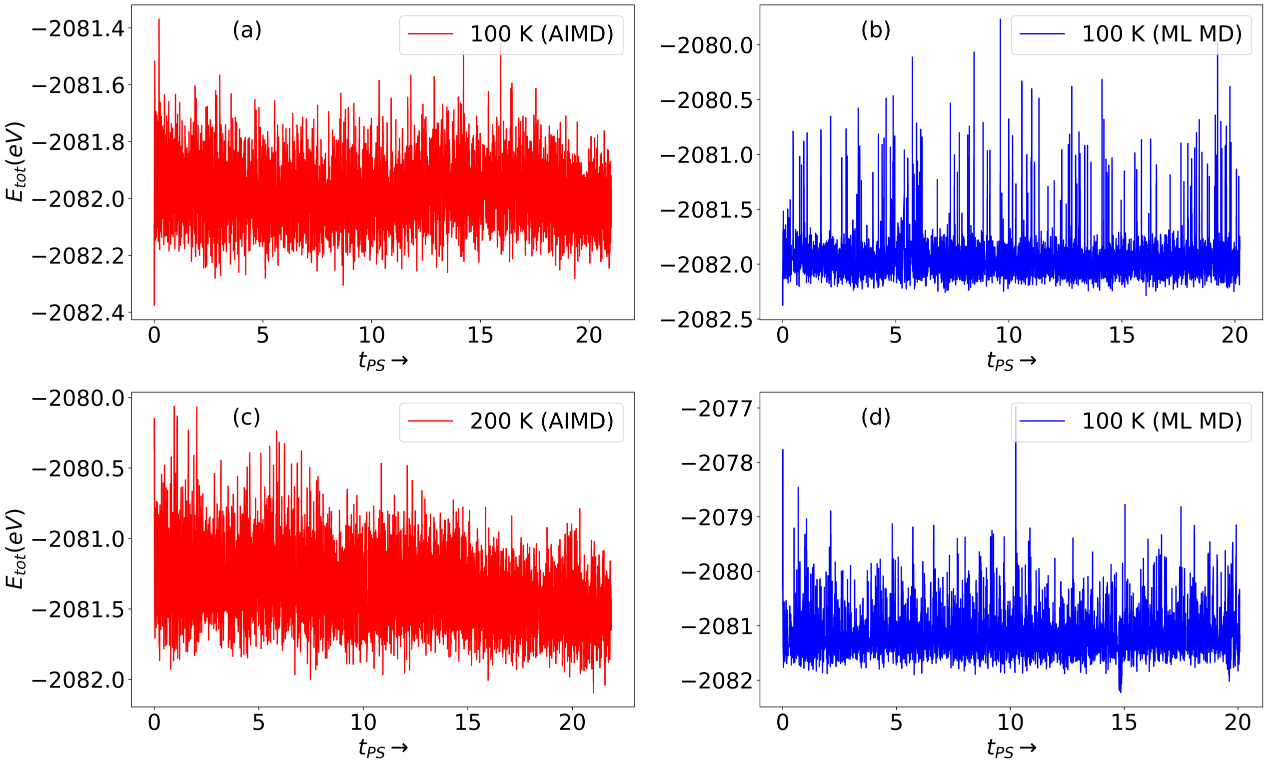

Figure 4.10: In this figure, we plot the total energy (eV) behaviour with respect to simulation time (t_{ps})for one isolated uracil molecule from 100 K and 200 K trajectories, obtained from both AIMD and ML MD schemes. Planes (a) and (c) were computed using the AIMD trajectory at 100 K and 200 K, respectively. While Panels (b) and (d) represent the same, they were calculated from the ML MD approach. Along the x-axis, it plots the simulation time and along the y-axis, energy values.

Fig.(4.10) justifying the total energy is conserved for obtaining the integration of Newton's equations in the NVE ensemble. In an NVE ensemble, the total energy should fluctuate around its mean value. So, from this figure, we can see the same either from a fully-converged trajectory or from the non-self-consistent run. The AIMD runs (red curves, left panels) fluctuation is in a narrow band, approximately between $(0.5 - 1eV)$, but the mean value is the same across the 20 (PS) trajectories. In comparison with ML MD runs (blue curves, right panels) show greater fluctuations, meaning more spikes towards higher energy scale, because it is not the fully converged solution, and the ML approach can not perform exactly the same as AIMD or better than it; it accumulates prediction and other numerical errors. It is also a fact that there is no systematic drift upward or downward, and the average stays flat.

Besides these simulations, we have investigated the 300 K case using our ML DM non-self-consistent approach. However, the ML-based charge-density model often failed to accurately predict the real-space charge density. As a result, the Fock matrix and hence the corresponding DM guesses deviated significantly from the converged results. Allowing partial mixing (one SCF iteration) with these DMs for the ML MD simulation consistently resulted in higher HOMO energies than LUMO energies, indicating orbital swapping, which is not feasible for a stable structure. As another outcome, rapid fluctuations in the magnitude of dipole moments were observed, leading to unexpected noise across the IR-frequency region.

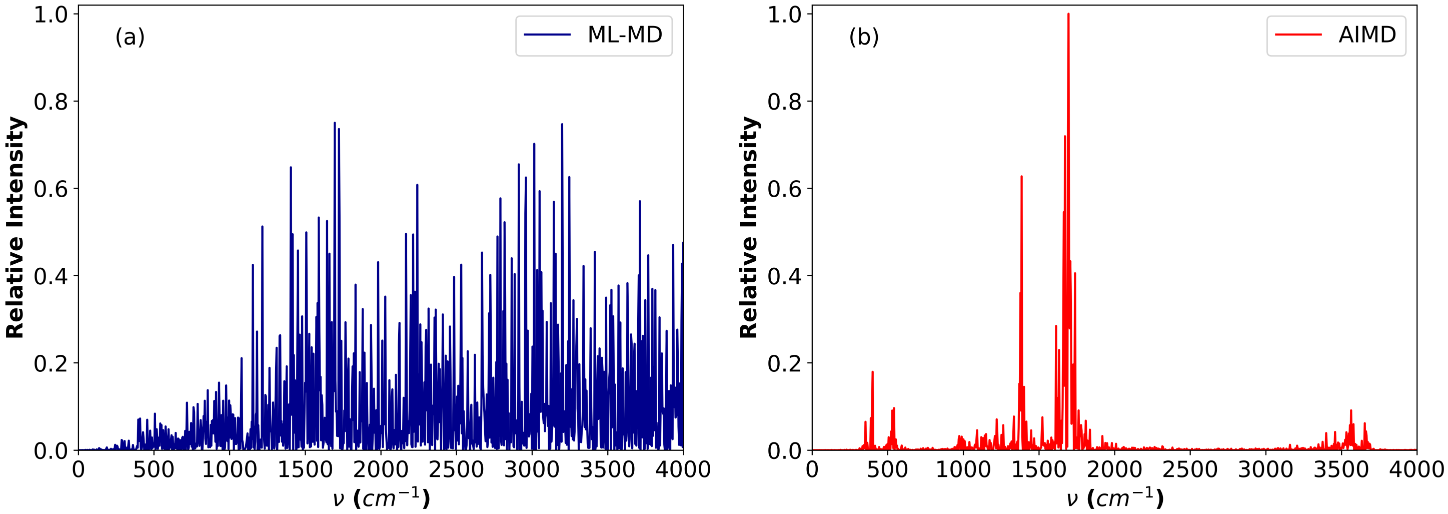

Figure 4.11: IR spectra for the Uracil molecule obtained from the 300 K MD trajectory. Panel (a) shows spectra computed from the ML-driven non-self-consistent MD, while panel (b) is for the AIMD simulation. For both plots, the x-axis represents the vibrational frequency and the y-axis defines the relative intensities.

In Fig.-4.11 we show the IR spectra for the Uracil molecule calculated from 10 ps long MD simulation at 300 K. In both panels, the y-axis is for the relative IR intensities and the x-axis corresponds to the associated frequencies. Panel (a) shows that the ML-driven non-self-consistent workflow is not capable of achieving AIMD-quality results as shown in panel (b). The ML workflow introduces significant noise in the spectral region, making it more difficult to distinguish individual peaks compared with the results produced by the AIMD scheme. The highest peak was found at 1698.69 (cm^{-1}) from AIMD, while the ML DM scheme moved it to 4466.89 (cm^{-1}), indicating substantial errors in the ML DM generated results. In the result and discussion section, we have already highlighted that the 300 K simulation run by the ML DM-driven non-self-consistent scheme frequently encounters inconsistencies in the HOMO-LUMO energy gap, across the trajectory. Fig.-4.12 illustrates this problematic phenomenon observed in the 300 K trajectory, where HOMO energy $>$ LUMO energy, indicating unfeasible configurations. In Fig-4.12 we demonstrate the rapid fluctuations in various properties, such as the HOMO-LUMO energy gap, dipole moments and total energies, as the functions of simulation time for 300 K MD run using the ML-driven approach. Here, panel (a) represents changes in the HOMO-LUMO gap, panel (b) shows the prompt variation in the total dipole moment magnitude ($|\mu|$), and panel (c) illustrates the fluctuation in total ground state energies ($|E_{tot}|$).

These changes are shown as a function of the simulation time, over the interval of 6.26 ps to 6.38 ps accounting for 100 successive Uracil structures. From panels (a, b, and c) three distinct regions can be identified: the first, where the problem does not exist around 6.20 ps to 6.30 ps; second, the mid-region between 6.31 ps and 6.33 ps, where the inconsistencies occur, as we consistently find positive HOMO-LUMO energy gaps, significantly larger dipole moments, and a maximum change in total energies of 340.14 (meV); finally, from 6.34 ps onward, the trajectory returns to normal. It is worth mentioning that the same HOMO-LUMO issue has arisen multiple times throughout the ML DM-driven MD trajectory, starting from the initial stages. Moreover, after 12 ps, the trajectory consistently encountered unphysical Uracil structures with these problematic phases. Furthermore, to investigate the HOMO-LUMO discrepancy due to the ML-predicted real space charge density, we show the HOMO and LUMO orbitals. Panels (d, f) display the 2D heat maps of the HOMO for a single Uracil structure ob-

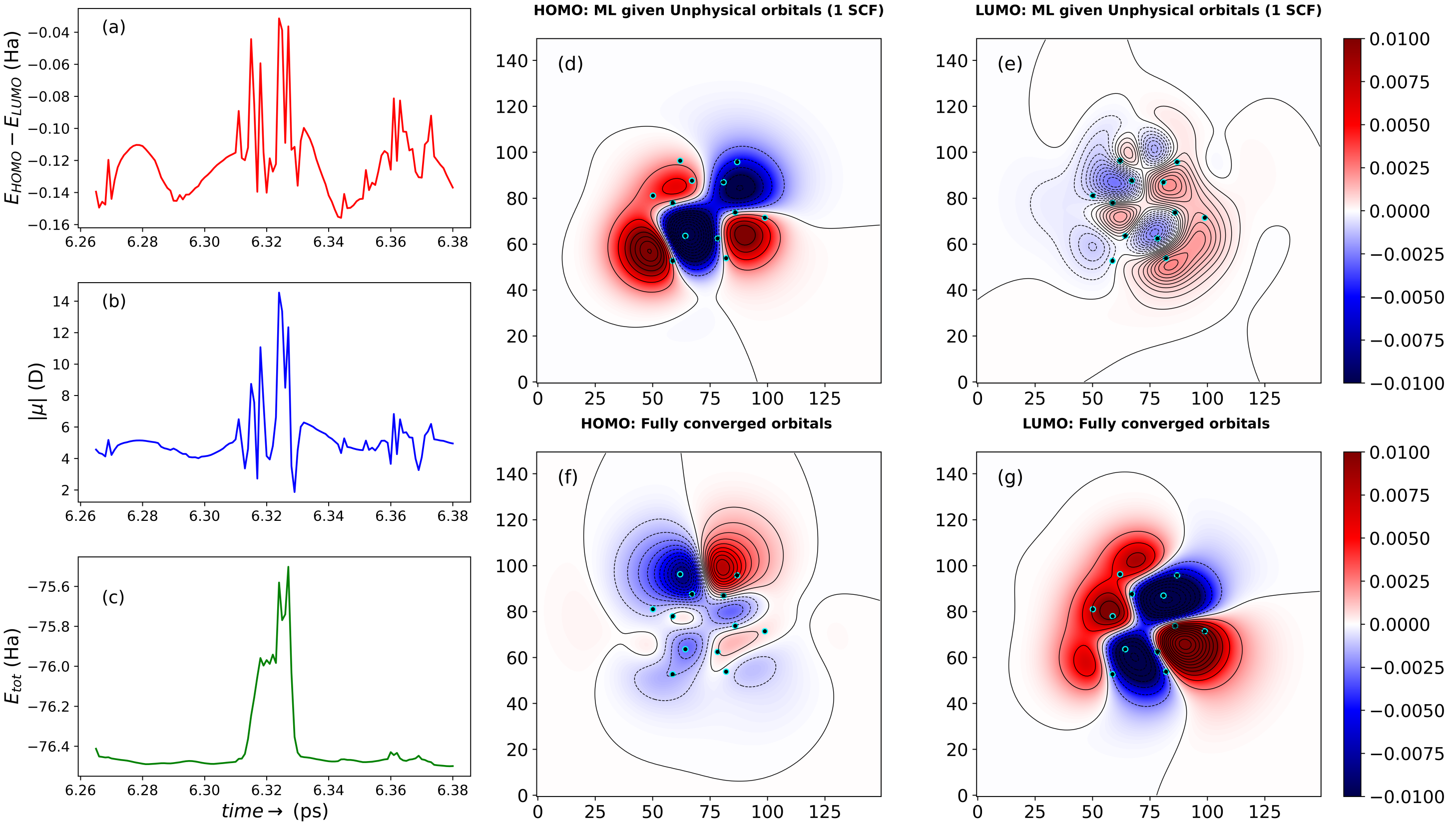

Figure 4.12: Plots for various properties [(a), (b), and (c)] with their normalized values, obtained from ML DM-driven non-self-consistent MD simulations of Uracil structures at 300 K. Panel (a) illustrates the variation of the HOMO-LUMO energy gap, panel (b) displays the corresponding changes in dipole moments, and panel (c) shows the variation in total energies, as the function of simulation time along the x-axis. In panel (d), we show a 2D heat map for an impractical HOMO orbital, generated from ML DM guess, while panel (f) represents the same orbital from the fully converged solution. Panels (e) and (g) provide the associated representation for the LUMO orbitals of the same uracil structure. Colour bars indicate quantitative information about the elements in the given molecular orbitals.

tained from the ML DM-based non-self-consistent MD trajectory, and the fully converged DFT result, respectively. Similarly, panels (e) and (g) display the LUMO for the same structure from the two approaches. This uracil structure corresponds to that non-self-consistent MD trajectory where the HOMO-LUMO inconsistency occurred as indicated in panel (a) within the time window 6.31 ps to 6.33 ps. It is evident, that panels (d, f) and panels (e, f) do not depict the same molecular orbitals, implying that the ML-charge density model fails to accurately predict the corresponding real-space electron density, resulting in the appearance of HOMO-LUMO inconsistency. Moreover, partial orbital swapping is visible between panels (d) and (g), showing that the

HOMO from the ML DM approach became the LUMO after full convergence. In Fig.-4.13 we also show 2D heat maps of the HOMO and LUMO, associated with a single Uracil structure that corresponds to the stable region in the ML-MD trajectory, where the HOMO-LUMO shape remains consistent [see panel (a), extreme right around the region 6.38 ps]. Panels (h) and (j) represent HOMO from ML-operated workflow, and the fully converged DFT solution, respectively. While panels (i) and (k) display the plots for LUMO, from ML given one SCF solution and fully converged DFT orbitals, respectively. From these plots, we can argue the ML-given orbitals either HOMO or LUMO always yield fully converged orbitals, as it is expected for a stable structure.

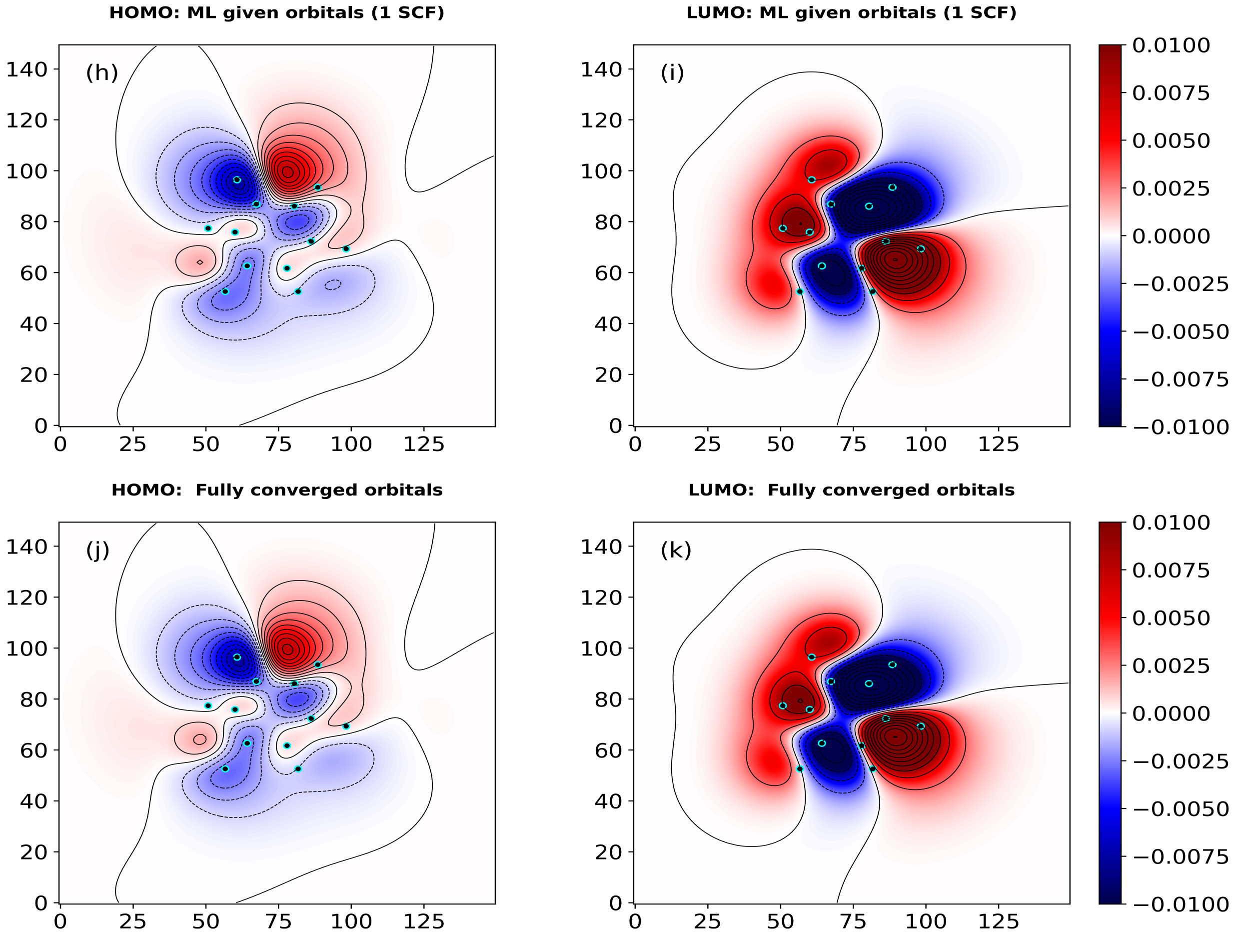

Figure 4.13: 2D heat map for HOMO and LUMO orbitals of a single uracil molecule, obtain from the region without any HOMO-LUMO discrepancies, at 300 K. Panels (h) and (j) represent HOMO orbitals followed by ML MD and fully converged DFT schemes, respectively. Panels (i) and (k) display the corresponding LUMO representations.

In the IR spectra, a small shift between DFT and ML results is present in the vibrational frequencies for the 100 K and 200 K runs. In the present work, we conducted six different MD trajectories: two at 100 K, two at 200 K, and the remaining at 300 K. For each temperature two kinds of approach were followed, one was fully self-consistent $ab-initio$ MD, and the other one was non-self-consistent MD (ML-MD). It is clear that among these three sets of MD, simulations at 100 K and 200 K, the ML MD workflow produced promising results compared to the AIMD approach. These simulations are initiated with an identical set of dynamical parameters, such as the initial velocities, and the same initial structure of the uracil molecule. The deviation for IR spectra arises because it is an obvious fact that we did not apply the ML DM approach to the same uracil configurations generated from the AIMD trajectory, we employed the ML MD approach for a different trajectory but with the same initial parameters, and this explanation is also applicable for ML MD trajectory. Since the structures are not identical, hence the dipole moments will not be the same, although we have calculated the MD simulations using the same initial parameters. Furthermore, to investigate those small shifts we compute the dipole moments by employing ML DM-based non-self-consistent DFT and the fully converged self-consistent DFT to the same uracil configurations generated from both AIMD and ML MD simulations, respectively, as we performed for atomic forces, but here we consider the structures from both MD approaches.

We show the party plots for the dipole moments (μ) in Fig.-4.14 and Fig.-4.15 computed using the MD trajectories at 100 K, and 200 K, respectively, for $1,000$ random uracil structures. For the 100 K simulation, those $1,000$ structures were sampled across the trajectory within the time window of 5.0 ps to 6.8 ps, while for the 200 K analysis, those were extracted between the 3.0 ps to 6 ps time window. From both plots [Fig 4.14 and 4.15], panels (a), (c), and (e) show the parity plots for the x,y and z components of μ, respectively. In these plots, the x-axis is the absolute value of μ for uracil, computed from the non-self-consistent MD trajectory. The y-axis depicts the $|\mu|$ obtained from fully self-consistent DFT calculations, where DFT was applied to the same uracil configurations generated from the non-self-consistent MD run. Then, panel (b), (d), and

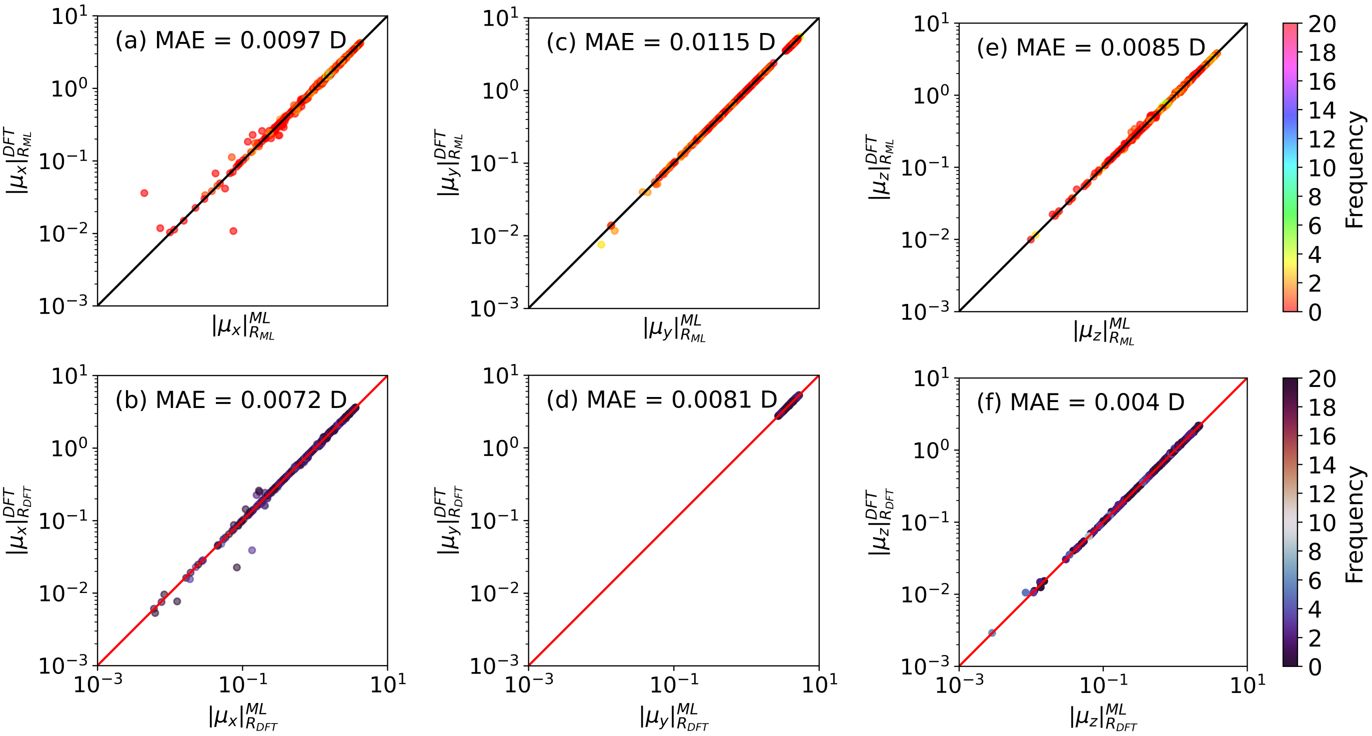

Figure 4.14: Parity plots for the $\alpha = x, y, z$ component of the dipole moments for 1000 random uracil structures obtained from 100 K molecular dynamics simulations. Here $|\mu_\alpha|^{DFT}_{R_{ML}}$ is the dipole moment calculated using fully converged DFT for the uracil structures generated through the non-self-consistent MD simulation, $|\mu_\alpha|^{ML}_{R_{DFT}}$ indicates dipole moment calculated using ML DM DFT calculations for the uracil structures generated from the fully self-consistent $ab-initio$ MD run. $|\mu_\alpha|^{ML}_{R_{ML}}$ denotes the dipole moments obtained from the ML DM non-self-consistent MD simulation, and $|\mu_\alpha|^{DFT}_{R_{DFT}}$ indicates the dipole moments calculated using fully converged $ab-initio$ MD run. Each graph report also reports the MAE achieved in the Debye (D) unit. All the plots are on a logarithmic scale and the color codes describe the frequencies of the dipole moment values.

(f) also from Fig.[4.14 and 4.15] show the parity plots for the x,y, and z components of dipole moments, respectively. In all panels, the x-axis is the absolute value of μ, calculated using ML-DM. In contrast, the y-axis refers to $|\mu|$, for those same uracil structures obtained from the AIMD simulations. The value of the MAE is reported in all cases. For the 100 K investigation, the maximum deviations from panels (a), (c) and (e) in Fig.-4.14 are obtained to be 0.0988 (D), 0.0991 (D), and 0.0949 (D) respectively. While, the maximum errors from panels (b), (d), and (f) were 0.0990 (D), 0.0956 (D), and 0.0744 (D), respectively. Similarly, for the 200 K run [Fig.(4.15)], the largest deviations are obtained to be 0.0997 (D) for panel (a), 0.0996 (D), and 0.0976 (D) for panel (e). While, the greatest deviations were found to be 0.0994 (D) for panel (b), 0.0990 (D) for panel (d), and 0.0966 (D) for panel (f). From panels (a) and panels (b) in Fig. (4.14 and 4.15), we can observe that, with a few exceptions, most points are closely aligned along the parity lines. The few points deviating from the parity lines are smaller in magnitude, typically less than $\sim 10^{-1}$. This implies that significant relative errors are primarily associated with smaller dipole moment values. The parity plots for the y components of the dipole moments shown in panels (c) and (d) of Fig.-4.14, indicate that the results from the ML-driven non-self-consistent MD are well in agreement with those obtained from the fully DFT converged MD simulations. Whereas, for the y components in panels (c) and (d) of Fig.-4.15, most points lie along the parity lines with only a few scatter points. The order of magnitude for these deviations is less than $\sim 10^{-2}$ in most cases. Similarly, for the z components of dipole moments [depicted in panels (e) and (f) for both the Fig.-4.14 and Fig.-4.15], ML-MD results also demonstrate an excellent agreement with the fully self-consistent $ab-initio$ calculations.

Finally, in Fig-4.16 we present the histograms illustrating the distributions of various observables, including total energies [panel (a)], the HOMO-LUMO energy gaps and total dipole moment's magnitude [panel (c)] obtained from the 100 K MD simulations with various approaches. Here two different workflows are utilized to compute the thermal distributions: 1) fully self-consistent AIMD, and 2), ML DM-driven non-self-consistent simulations.

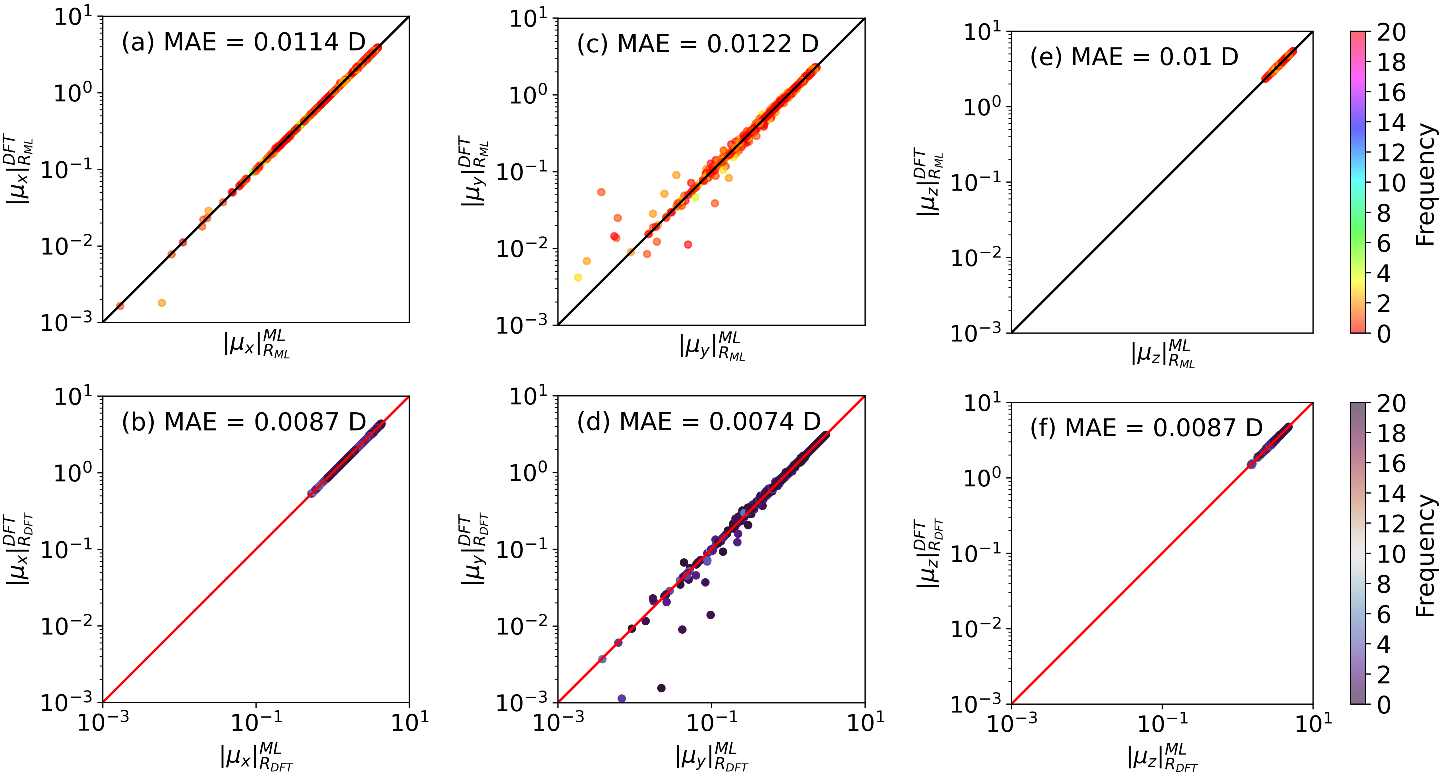

Figure 4.15: Parity plots for the $\alpha = x, y, z$ component of the dipole moments for 1000 random uracil structures obtained from 200 K molecular dynamics simulations. Here $|\mu_\alpha|^{DFT}_{R_{ML}}$ is the dipole moment calculated using fully converged DFT for the uracil structures generated through the non-self-consistent MD simulation, $|\mu_\alpha|^{ML}_{R_{DFT}}$ indicates the dipole moment calculated using ML DM DFT calculations for the uracil structures generated from the fully self-consistent $ab-initio$ MD run. $|\mu_\alpha|^{ML}_{R_{ML}}$ denotes the dipole moments obtained from the ML DM non-self-consistent MD simulation, and $|\mu_\alpha|^{DFT}_{R_{DFT}}$ indicates the dipole moments calculated using fully converged $ab-initio$ MD run. Each graph report also reports the MAE achieved in Debye (D). All the plots are on a logarithmic scale and the color codes describe the frequencies of the dipole moment values.

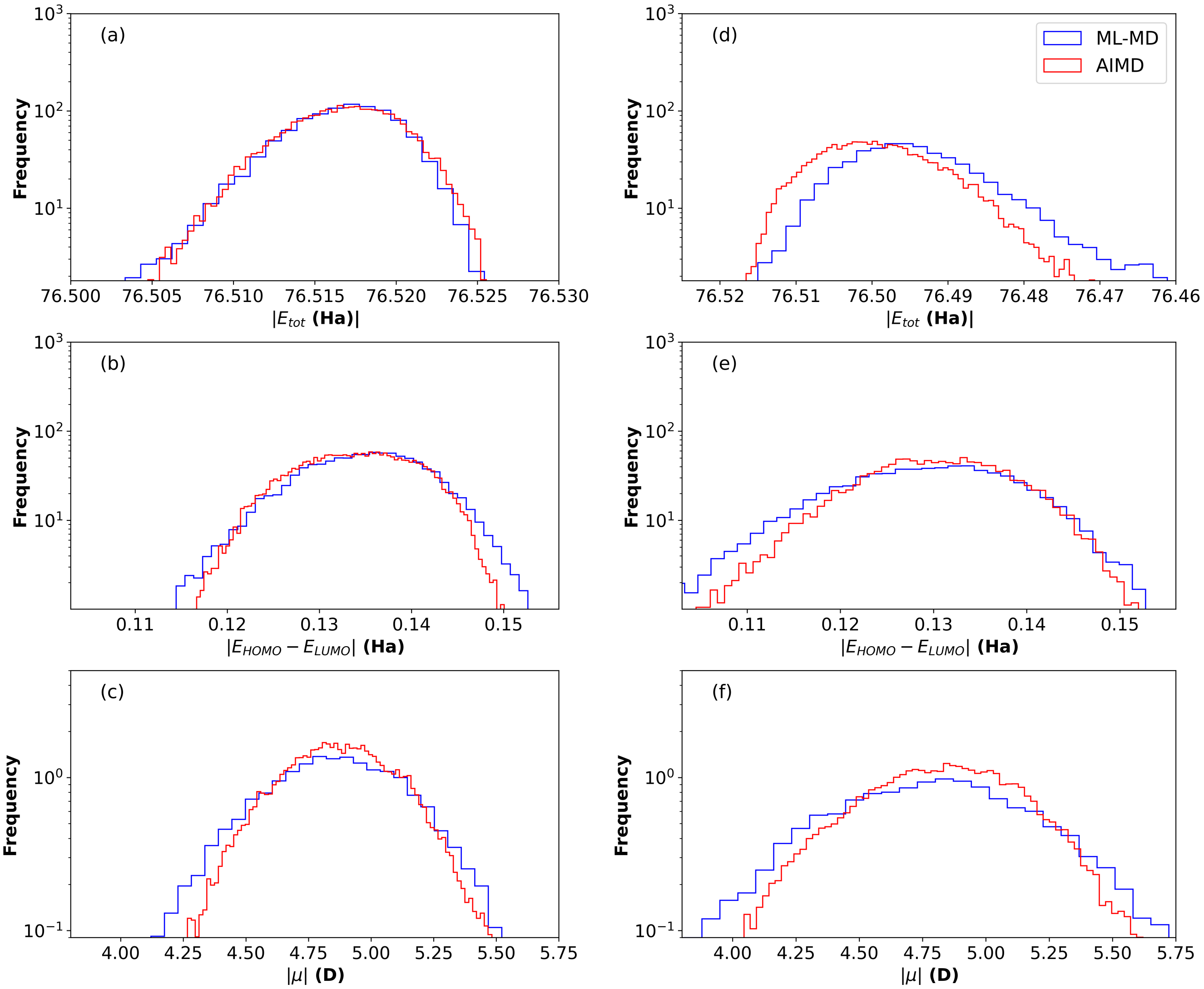

Figure 4.16: Histogram of various observables computed over the different molecular dynamics trajectories for the Uracil molecule. Panels (a) and (d) show the distribution for the total ground state energies (E_{tot}) at 100 K and 200 K, respectively. Panels (b) and (e) display the $|HOMO - LUMO|$ energy gaps over the trajectories at 100 K and 200 K respectively. Finally, panels (c) and (d) depict the thermal distribution of the total dipole moment magnitude ($|\mu|$) for two temperatures 100 K and 200 K. In all these graphs, the x-axis is for the values of the observables, while the y-axis represents the associated frequencies.

All these panels demonstrate the expected behaviour since throughout this study we have seen for MD, our ML DM approach is capable of producing DFT quality atomic forces to conduct the simulations. Thus the ML approach and the full self-consistent DFT MD runs yield similar results in this case as well. From the histogram in panel (a) the average energy is computed at -76.5165 (Ha) from the DFT run, and -76.5158 Ha from the ML MD simulation. Similarly, in panel (b) the average

$|HOMO - LUMO|$ gap is calculated from the DFT-run at 0.1340 Ha, from the ML simulations it is 0.1346 Ha. Finally, in panel (c) the average value of the dipole moments is 4.8695 (D) for DFT, while for the ML DM trajectory, it is calculated at 4.8519 (D). Furthermore, panels (d), (e), and (f) also show the distributions for the same properties: total energies, HOMO-LUMO gap, and the magnitude of dipole moments, respectively, from the 200 K trajectories obtained through AIMD and non-self-consistent MD simulations. In panel (d), the average ground state energy obtained from the AIMD run is -76.4984 (Ha), while ML MD yields -76.4928 Ha. Panel (e) shows the mean HOMO-LUMO gap at 0.1302 Ha, provided through $ab - initio$ simulations and 0.1282 Ha obtained from ML MD run. Finally, for the dipole moments [panel (f)], the average magnitude is 4.8247 D for the full-self-consistent MD and 4.7833 D for the ML DM-based trajectory. Here for the 100 K runs, the difference for average energy between AIMD and ML MD run is $\sim 10^{-4}$ Ha, whereas, for 200 K it is $\sim 10^{-2}$ Ha. The difference in average $|HOMO - LUMO|$ gap between the same is also $\sim 10^{-4}$ Ha for 100 K simulations, and it is $\sim 10^{-3}$ Ha for 200 K. Finally, the difference between AIMD and ML MD schemes for average $|\mu|$ is 0.0176 D for 100 K trajectories, and the value is 0.0414 D for 200 K runs. Here one can see the average differences in various properties over the MD trajectory for 200 K are slightly higher than the 100 K one, which means, as the temperature increases our ML MD workflow accumulates more errors in predicting atomic forces and system dynamics. However, to address this issue we have discussed a few possibilities in the conclusion chapter. Considering the challenges of ML MD at higher temperatures, these results indicate that our ML-based MD scheme not only accurately describes structures but also various properties across the MD trajectory.

5 | Conclusions and Future Work

5.1 Conclusions on NN predicted ML DM Scheme

We have demonstrated that the neural networks (NN) model can be trained to predict the density matrix (DM) constructed over atomic orbital basis sets to accelerate the SCF scheme for density functional calculations. These machine-learned DMs provide high-quality initial guess orbitals to obtain real-space electron density for Kohn–Sham (KS) DFT calculations, resulting in significantly reducing the number of self-consistent iterations needed for convergence, compared to other commonly used initial guesses of DMs. Notably, these DMs yield highly accurate energies and forces even without full self-consistency, enabling cost-effective molecular dynamics simulations at a computational expense comparable to that of machine learning force fields. For this deep learning-based approach we have studied and discussed the results for three different molecules, H_2O, S_2O and $[Fe(H_2O)_6]^{2+}$, concerning both the DFT and ML-driven approach. For instance, this ML DM scheme can be applied together with wave function-based quantum-chemistry methods such as the Hartree-Fock method.

5.2 Conclusion on ML-density Driven DM

In our second proposed method, we present a novel approach to perform DFT calculations and molecular dynamics simulations using the initial guess of ML DM, generated from ML-predicted real space electron density, which is then applied to the KS-Hamiltonian in order to determine DM. These ML DMs are sufficient entities to achieve the fully converged DFT quality results for energy, atomic forces, or other properties while considering only a single self-consistent cycle. We have shown the results for a $C_4H_4N_2O_2$ molecule using the density functional study as the reference data, where the molecular structures are represented in Cartesian coordinates. In conclusion, it is evident that the proposed machine learning-driven approach which provides ML-based DM guesses, offers better initial estimates for the DM, compared to any other commonly used alternatives. Furthermore, employing the JLCDM model for self-consistent density prediction and thereby recalculating the KS-Hamiltonian, eliminates the need for implementing expensive deep learning models (213) to accelerate the SCF scheme.

5.2.1 Limitations and Remedies of ML-Density Driven Method

The limitation of our scheme is that, it was not successful in conducting the non-self-consistent MD simulation at 300 K. As a result, the trajectory often encountered non-physical Uracil structures, where the changes in total ground state energies, and orbital gradients were much higher than the normal structures at the first SCF iteration, and also inconsistent HOMO-LUMO energy gaps were appeared, throughout the 300 K MD run. However, this problematic occurrence suggests that those complicated electron densities are out of the training periphery for the JL-charge density model. It is worth mentioning that in the case of fully converged AIMD simulation at 300 K, sometimes trajectory yields structures where HOMO and LUMO energy levels are the same. To address this issue for our ML-density-based non-self-consistent approach, we applied full convergence once the trajectory experienced inconsistent structures to maintain consistency in the simulation, as the DFT correction. This resulted in a shift in the ground-sate energy scale between the regular trajectory and the one causing HOMO-LUMO inconsistencies. Finally, those shifts produce more unnecessary noises in the entire trajectory [see Fig.-4.12] and also in the spectral region.

To address these complex situations, the following approaches can be applied:: 1) increasing the diversity in the training set by extending training trajectories for the charge density model (JLCDM) to a few additional femtoseconds with varied initial velocities, ensuring greater randomness is incorporated; 2) train the charge density model at a higher temperature using a range of trajectories then set the ML-driven non-self-consistent workflow for low-temperature MD run, e.g, for 300 K trajectory model can be trained at 600 K or even higher, same as the charge density model we trained for 200 K simulation; 3) two distinct charge density models (JLCDM) can be trained with two different sets of training data: one set of data containing charge densities from the normal trajectories and the remaining one containing those complex electron densities. A separate classifier or a preprocessing layer will then determine the appropriate model based on the inputs received by the ML model; 4) by following the strategy outlined in (3), a hybrid or an ensemble-based approach can also be worth exploring. This would involve combining the real space density predictions from both models, using an average-weighted approach.

5.2.2 Optimization Challenges and Workflow Overview

It is worth explaining that our ML strategy generated DM from the KS-Hamiltonian yields advanced and competitive outcomes when compared to other initial guess methods. However, the workflow depends on several key factors. In summary, primarily it follows three steps: 1) prediction of real space electron density using the JL-charge density model (JLCDM) where atomic structures serve as the inputs to the ML model; 2) construction of KS-Hamiltonian from the predicted density, using our PySCF-integrated module, then followed by diagonalization of the KS-Hamiltonian to obtain the density matrix; 3) implementation of ML DM with one SCF cycle to compute atomic forces, enabling MD simulation and determination of new molecular structures for subsequent density predictions. As the system size increases and incorporates various atomic species, computational expenses will escalate. For the JL charge density model, the involvement of varieties of atoms increases the number of descriptors needed for charge density prediction, leading to an expensive computation. Furthermore, the method we designed for constructing the KS-Hamiltonian employs an atomic-grid-based algorithm, which becomes more resource-intensive as the grid size increases. To ensure efficient execution, it is essential to carefully optimize all of these parameters before initiating any specific computational approach. The overall performance of this ML-based density matrix approach offers an alternative method for conducting MD simulation while preserving a fully ab initio accuracy. However, for MD simulation at 300 K, inaccurate real space density predictions in our non-self-consistent workflow led to the generation of a few infeasible structures along the trajectory. This issue is not inherent to our ML DM approach but rather depends on the choice of basis set, DFT functional, and the system type. For this study, we applied AIMD and our ML-driven method to a uracil molecule with LDA DFT functional and observed $HOMO = LUMO$ inconsistency for orbital energies during a room-temperature AIMD simulation.

5.3 Future Directions

In this section, we will discuss future perspectives that can be studied using this ML-DM workflow. This planning will primarily follow two aspects: one is the methodological advancement and another one is various applications, either for molecules or for periodic systems.

Higher orbitals for Coulomb integrals

As we discussed in the result and discussion chapter, we implemented the scheme that computes the Coulomb matrix from the electron density over the Gaussian-type orbitals for a given molecule. This approach includes the calculation of up to p-type orbitals for the GTOs and, in the future, additional orbitals (e.g. d, f etc.) can be added to enhance the functionality of this approach, in order to reconstruct the KS-Hamiltonian from real space charge density.

All Electron Calculations

We introduced here the electron density operated ML-MD workflow for a uracil molecule, which is investigated through the ECP (153, 154)-based basis function as implemented in PySCF. As a future research plan, we will try to train the JL-charge density model to predict and utilize the all-electron-based real-space charge density and to build KS-Hamiltonian to compute ML-DM for accelerating the DFT calculations, respectively.

High Temperature MD Simulation

Throughout this thesis, we studied MD trajectories for four molecules to observe the performance of our ML-DM approach, utilizing either the deep learning model or the linear Ridge regression model. For these investigations, we performed MD simulations where the temperature range varied from 100 K to 300 K. However as we have seen in the result section, the 300 K ML-MD simulation was not able to achieve a satisfactory level to make a comparison with the full ab initio calculation. As a future goal, we will also try to implement our ML-MD workflow to conduct higher-temperature MD (300 K or more) runs to validate our methodological advancement and some application perspectives with the possible directions mentioned in the conclusion section.

ML DM guess for GGA type DFT functional

Here we used local density approximation (LDA) (17) DFT functional for predicted real space electron density to construct KS-Hamiltonian and to obtain the DM. As one of the future works, we will consider some density-derivative-based GGA functional (eg. BLYP (26, 27), PBE (25)) to construct the JL-charge density model. This advancement demands the inclusion of analytical derivatives of the JL-descriptor (derivative for the analytical form of used polynomials) for the JLCDM scheme.

Periodic Systems and Magnetic Molecules

Extending this machine-learning approach to predict electron density for unit cells with boundary conditions can also be designed as one of the future plans. It demands modification of the ML-DM workflow accountable for the reciprocal space (k-points) in the Kohn-Sham Hamiltonian construction.

Implementation of this ML MD for magnetic molecules, the spin-polarized densities have spin-up and spin-down channels. Primarily it demands the construction of two different JLCDM schemes for up and down spin densities. We also need to obtain KS-Hamiltonian to include spin terms, ensuring compatibility with the spin-polarized workflow.

Applicability for Biomolecules

This ML workflow can also be extended for biomolecular systems by leveraging machine learning ML models trained on complex biomolecules to predict the real-space electron density efficiently. This ML DM approach will enable the fast ML-DFT calculations of large biomolecules, such as proteins and nucleic acids, with DFT-level accuracy. Furthermore, integrating this workflow with the atomic force obtaining scheme can facilitate hybrid simulations, providing insights into biomolecular dynamics and interactions critical for understanding biological processes.

Bibliography

[1] Qiming Sun, Xing Zhang, et al. Recent developments in the pyscf program package. J. Chem. Phys, 153:024109, 2020. DOI: 10.1063/5.0006074.

[2] Qiming Sun, T. C. Berkelbach, et al. Pyscf: the python-based simulations of chemistry framework. WIREs Comput. Mol. Sci, 8:e1340, 2018. DOI: 10.1002/wcms.1340.

[3] K. Korsell et al. Principles for a direct scf approach to lcao–mo ab-initio calculations. journal of computational chemistry. Journal of Computational Chemistry, 3(3):385–399, 1982.

[4] J. H. Van Lenthe et al. Starting scf calculations by superposition of atomic densities. journal of computational chemistry. Journal of Computational Chemistry, 27(8):926–932, 2006.

[5] S. Lehtola. Assessment of initial guesses for self-consistent field calculations. superposition of atomic potentials: simple yet efficient. J. Chem. Theory Comput., 15:1593, 2019. doi:10.1021/acs.jctc.8b01089.

[6] S. Lehtola. Fully numerical calculations on atoms with fractional occupations and range-separated exchange functionals. Phys. Rev. A, 101:012516, 2020. doi:10.1103/PhysRevA.101.012516.

[7] A. P. Thompson et al. Lammps - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. Comp Phys Comm, 271:10817, 2022. doi: 10.1016/j.cpc.2021.108171.

[8] Dario A. Estrin, Luca Paglieri, and Giorgina Corongiu. A density functional study of tautomerism of uracil and cytosine. J. Chem. Phys, 98:5653–5660, 1994. DOI: 10.1021/j100073a014.

[9] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. Phys. Rev., 136: 864B–B871, 1964.

[10] E.R. Davidson H. Hsu and R.M. Pitzer. Scf method for hole states. J. Chem. Phys., 65:609–613, 1964.

[11] P.M.W. Gill B.G. Johnson and J.A. Pople. The performance of a family of density functional methods. J. Chem. Phys., 98:5612–5626, 1993.

[12] G. Karlstrom. Dynamical damping based on energy minimization for use in ab initio molecular-orbital scf calculations. Chem. Phys. Lett., 67:348–350, 1979.

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

[14] R.G. Parr and W. Yang. Density-functional theory of atoms and molecules. Oxford University Press, New York, 1964.

[15] J. J. Sakurai and Jim Napolitano. Modern Quantum Mechanics. Addison-Wesley, 2 edition, 2011.

[16] R. Shankar. Principles of Quantum Mechanics. Springer, 2 edition, 1994.

[17] S. H. Vosko, L. Wilk, and M. Nusair. Accurate spin-dependent electron liquid correlation energies for local spin density calculations: A critical analysis. Canadian Journal of Physics, 58(8):1200–1211, 1980. doi: 10.1139/p80-159.

[18] Perdew JP et al. Atoms, molecules,solids, and surfaces—applications of the generalizedgradient approximation for exchange and correlation. Phys Rev B, 46: 6671–6687, 1992.

[19] Perdew JP et al. Generalized gradi-ent approximation made simple. Phys Rev Lett, 77:3865–3868., 1996.

[20] Perdew JP et al. Exact kohn–sham scheme basedon perturbation theory. Phys Rev A, 77:196–204, 1994.

[21] Wei Hu, Xinming Qin, et al. High performance computing of dgdft for tens of thousands of atoms using millions of cores on sunway taihulight. Science Bulletin, 66:111–119, 2021.

[22] M. Sharma and M. Sierka. Efficient implementation of density functional theory based embedding for molecular and periodic systems using gaussian basis functions. Journal of Chemical Theory and Computation, 18:6892–6904, 2022. DOI: https://doi.org/10.1021/acs.jctc.2c00380.

[23] Sherrill et al. The configuration interaction method: Advances in highly correlated approaches. San Diego: Academic Press, 34, 1999. doi: 10.1016/S0065-3276(08)60532-8.

[24] Rodney J. Bartlett and Monika Musiał. Coupled-cluster theory in quantum chemistry. Rev. Mod. Phys., 79:291–352, 2007. doi: 10.1103/RevModPhys.79.291.

[25] J. P. Perdew, K. Burke, and M. Ernzerhof. Generalized gradient approximation made simple. Phys. Rev. Lett., 77:3865–3868, 1996. doi: 10.1103/PhysRevLett.77.3865.

[26] A. D. Becke. Density-functional exchange-energy approximation with correct asymptotic behavior. Phys. Rev. A, 38:3098, 1988. doi:10.1103/PhysRevA.38.3098.

[27] Chengteh Lee, Weitao Yang, and Robert G. Parr. Development of the colle-salvetti correlation-energy formula into a functional of the electron density. Phys. Rev. B, 37:785, 1988. doi:10.1103/PhysRevB.37.785.

[28] A. D. Becke. Density-functional thermochemistry. iii. the role of exact exchange. J. Chem. Phys., 98:5648, 1993. doi:10.1063/1.464913.

[29] C. Lee, W. Yang, and R. G. Parr. Development of the colle-salvetti correlation-energy formula into a functional of the electron density. Phys. Rev. B, 37:785, 1988. doi:10.1103/PhysRevB.37.785.

[30] P. J. Stephens et al. Ab initio calculation of vibrational absorption and circular dichroism spectra using density functional force fields. J. Phys. Chem., 98: 11623–11627, 1994. doi:10.1021/j100096a001.

[31] J. Sun et al. Strongly constrained and appropriately normed semilocal density functional. Phys. Rev. Lett., 115:036402, 2015. doi:doi.org/10.1103/physrevlett.115.036402.

[32] Narbe Mardirossian and Martin Head-Gordon. ωb97m-v: A combinatorially optimized, range-separated hybrid, meta-gga density functional with vv10 non-local correlation. J. Chem. Phys, 144:214110, 2016. doi: https://doi.org/10.1063/1.4952647.

[33] John P. Perdew; Karla Schmidt. Jacob's ladder of density functional approximations for the exchange-correlation energy. AIP Conf. Proc., 577:1–20, 2001. doi: 10.1063/1.1390175.

[34] V. I. Anisimov, F. Aryasetiawan, and A. I. Lichtenstein. First-principles calculations of the electronic structure and spectra of strongly correlated systems: The lda+u method. J. Phys.: Condens. Matter, 9(4):767, 1997. doi: 10.1088/0953-8984/9/4/002.

[35] B. Himmetoglu, A. Floris, S. de Gironcoli, and M. Cococcioni. Hubbard-corrected dft energy functionals: The lda+u description of correlated systems. Int. J. Quantum Chem., 114(1):14–49, 2014. doi: 10.1002/qua.24521.

[36] Antoine Georges et al. Dynamical mean-field theory of strongly correlated

fermion systems and the limit of infinite dimensions. Reviews of Modern Physics, 68:13–125, 1996. doi: https://doi.org/10.1103/RevModPhys.68.13.

[37] G. Kotlia et al. Electronic structure calculations with dynamical mean-field theory. Rev. Mod. Phys., 78:865–951, 2006. doi: http://link.aps.org/doi/10.1103/RevModPhys.78.865.

[38] L Hedin. New method for calculating the one-particle green's function with application to the electron-gas problem. Phys. Rev., 139, 1965. doi: http://link.aps.org/doi/10.1103/RevModPhys.78.865.

[39] M. J. van Setten et al. The gw-method for quantum chemistry applications: Theory and implementation. Journal of Chemical Theory and Computation, 9, 2013. doi: 10.1021/ct300648t.

[40] G. Kresse et al. Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. Computational Materials Science, 6:15–50, 1996. doi:10.1016/0927-0256(96)00008-0.

[41] P Giannozzi et al. Advanced capabilities for materials modelling with quantum espresso. J.Phys.:Condens.Matter, 29:465901, 2017. doi: 10.1088/1361-648X/aa8f79.

[42] Stefano Curtarolo et al. Aflow: An automatic framework for high-throughput materials discovery. Comnputational Materials Science, 58:218–226, 2012. doi: doi:10.1016/j.commatsci.2012.02.005.

[43] Markus Scheidgen et al. Nomad: A distributed web-based platform for managing materials science research data. Journal of Open Source Software, 8:5388, 2023. doi: doi:10.1016/j.commatsci.2012.02.005.

[44] P. Pulay. Convergence acceleration of iterative sequences - the case of scf iteration. Chem. Phys. Lett., 73:393–398, 1980.

[45] P. Pulay. Improved scf convergence acceleration. J. Comp. Chem., 3:556–560, 1982.

[46] K.N. Kudin et al. A black-box self-consistent field convergence algorithm: One step closer. J. Chem. Phys, 116:8255–8261, 2002.

[47] E.C and C. Le Bris. Can we outperform the diis approach for electronic structure calculations? Int. J. Quantum Chem., 79:82–90, 2000.

[48] V.R. Saunders and I.H. Hillier. Level-shifting method for converging closed shell hartree-fock wave-functions. Int. J. Quant. Chem., 7:699–705, 1973.

[49] S.P. Bhattacharyya. Accelerated convergence in scf calculations and level shifting technique. Chem. Phys. Lett., 56:395–398, 1978.

[50] Qiming Sun. Co-iterative augmented hessian method for orbital optimization. arXiv preprint, 1978. arXiv:1610.08423.

[51] G.B. Bacskay. A quadratically convergent hartree-fock (qc-scf) method - application to closed-shell systems. Chem. Phys., 61:385–404, 1981.

[52] Qiming Sun et al. A general second-order complete active space self-consistent-field solver for large-scale systems. Chem. Phys. Lett., 683:291–299, 2017.

[53] Y.A. Wang and E.A. Orbital-free kinetic-energy density functional theory. in theoretical methods in condensed phase chemistry. S.D. Schwartz, ed. (Springer, Dordrecht), page 117–184, 2002.

[54] H. Chen and A. Zhou. Orbital-free density functional theory for molecular structure calculations. Numer. Math. Theor. Meth. Appl., 1, 2008.

[55] T.A. Wesolowski and Y.A. Wang. Recent progress in orbital-free density functional theory. World Scientific (Singapore), 2013.

[56] V.V. Karasiev, D. Chakraborty, and S.B. Trickey. Progress on new approaches to old ideas: Orbital-free density functionals. in many- electron approaches in

physics, chemistry and mathematics. V. Bach and L. Delle Site, eds. (Springer, Cham, Switzerland), page 113–134, 2014.

[57] W. C. Witt et al. Orbital-free density functional theory for materials research. Journal of Materials Research, 33:777–795, 2018.

[58] Seeger et al. Self-consistent molecular orbital methods. xvi. numerically stable direct energy minimization procedures for solution of hartree-fock equations. Journal of Chemical Physics, 65(1):265–271, 1976. doi: DOI:10.1063/1.432764.

[59] Susi Lehtola et al. Efficient implementation of the superposition of atomic potentials initial guess for electronic structure calculations in gaussian basis sets. J. Chem. Phys, 152:14410, 2020. doi: DOI:10.1063/5.0004046.

[60] David A. Mazziotti. Towards idempotent reduced density matrices via particle-hole duality: Mcweeny's purification and beyond. Phys. Rev. E, 68:066701, 2003. doi: doi.org/10.1103/PhysRevE.68.066701.

[61] Adam H. R. Palser and David E. Manolopoulos. Canonical purification of the density matrix in electronic-structure theory. Phys. Rev. B, 58:12704, 1998. doi: doi.org/10.1103/PhysRevB.58.12704.

[62] Ryong-Gyu Lee and Yong-Hoon Kim. Convolutional network learning of self-consistent electron density via grid-projected atomic fingerprints. npj Computational Materials, 10:248, 2024. doi: doi.org/10.1038/s41524-024-01433-0.

[63] Batta Mahesh. Machine learning algorithms -a review. International Journal of Science and Research, pages ISSN: 2319–7064, 2018. doi: doi:10.21275/ART20203995.

[64] Raffaele Pugliese et al. Machine learning-based approach: global trends, research directions, and regulatory standpoints. Data Science and Management, 4:19–29, 2021. doi: https://doi.org/10.1016/j.dsm.2021.12.002.

[65] Sebastian Ruder. An overview of gradient descent optimization algorithms. arxiv, 2017. doi: https://doi.org/10.48550/arXiv.1609.04747.

[66] Juan M. Rivas et al. Gradient descent algorithm for the optimization of fixed priorities in real-time systems. Journal of Systems Architecture, 153:103198, 2024. doi: https://doi.org/10.1016/j.sysarc.2024.103198.

[67] Arian Maleki and Tom Do. Review of probability theory. Stanford University.

[68] Amir Emad Marvasti et al. Maximum probability theorem: A framework for probabilistic machine learning. IEEE Transactions on Artificial Intelligence, 2: 214 – 227, 2021. doi: 10.1109/TAI.2021.3086046.

[69] George Obaido et al. Supervised machine learning in drug discovery and development: Algorithms, applications, challenges, and prospects. Machine Learning with Applications, 17:100576, 2024. doi: doi:https://doi.org/10.1016/j.mlwa.2024.100576.

[70] David S. Watson. On the philosophy of unsupervised learning. Philosophy & Technology, 36 (28), 2023. doi: https://doi.org/10.1007/s13347-023-00635-6.

[71] Ashish Kumar Shakya et al. Reinforcement learning algorithms: A brief survey. Expert Systems with Applications, 231:120495, 2023. doi: https://doi.org/10.1016/j.eswa.2023.120495.

[72] K. Gaurav et al. Human disease prediction using machine learning techniques and real-life parameters. International Journal of Engineering, 6:1092–1098, 2023. doi: doi:10.5829/ije.2023.36.06c.07.

[73] Bowman, Joel M, et al. Δ-machine learned potential energy surfaces and force fields. J. Chem. Theory Comput, 19:1–17, 2023. doi:10.1021/acs.jctc.2c01034.

[74] X. Guan et al. Using machine learning to go beyond potential energy surface benchmarking for chemical reactivity. Nature Computational Science, 3:965–974, 2023. doi:10.1038/s43588-023-00549-5.

[75] He Li et al. Deep-learning density functional theory hamiltonian for efficient ab initio electronic-structure calculation. Nature Computational Science, 2:367–377, 2022. doi:10.1038/s43588-022-00265-6.

[76] A. Kumar et al. Deep learning of ab initio hessians for transition state optimization. arxiv, 2024. 10.48550/arXiv.2405.02247.

[77] J. Blumberger. Recent advances in the theory and molecular simulation of biological electron transfer reactions. Chem. Rev., 115:0009–2665, 2015. DOI:10.1021/acs.chemrev.5b00298.

[78] Madhu Puttegowda et al. Artificial intelligence and machine learning in mechanical engineering: Current trends and future prospects. Engineering Applications of Artificial Intelligence, 142:109910, 2025. doi: https://doi.org/10.1016/j.engappai.2024.109910.

[79] L. Waikhom and R Patgiri. A survey of graph neural networks in various learning paradigms: methods, applications, and challenges. Artif. Intell. Rev., pages 1–70, 2022.

[80] J. Pilault et al. Improving transfer learning in nlp using fewer parameters & less data. ArXiv, 2023.

[81] R. van de Schoot et al. Bayesian statistics and modelling. Nat. Rev. Method. Primer., 1(1):1, 2021.

[82] N. Peppes et al. Performance of machine learning-based multi-model voting ensemble methods for network threat detection in agriculture 4.0. Sensors, 21(22):7475, 2021.

[83] S. Ghosh et al. A study on support vector machine based linear and nonlinear pattern classification. International Conference on Intelligent Sustainable Systems (ICISS), IEEE, pages 24–28, 2019.

[84] C. Lv et al. Machine learning: an advanced platform for materials development and state prediction in lithium-ion batteries. Adv. Mater., 34(25):2101474, 2022.

[85] Z.M. Fadlullah et al. State-of-the-art deep learning: evolving machine intelligence toward tomorrow's intelligent network traffic control systems. IEEE Commun. Surv. Tutor., 19(4):2432–2455, 2017.

[86] I.H. Sarker. Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. SN Comput. Sci., 2(6):420, 2021.

[87] K.J. DeMille and A.D. Spear. Convolutional neural networks for expediting the determination of minimum volume requirements for studies of microstructurally small cracks, part i: model implementation and predictions. Comput. Mater. Sci, 207:111290, 2022.

[88] Y. Wang and A.D. others. Mining structure–property relationships in polymer nanocomposites using data driven finite element analysis and multi-task convolutional neural networks. Mol. Syst. Des. Eng, 5(5):962–975, 2020.

[89] A. Benayad and A.D. others. High-throughput experimentation and computational freeway lanes for accelerated battery electrolyte and interface development research. Adv. Energy Mater, 12(17):2102678, 2022.

[90] A. Benayad and A.D. others. Determining favourable process parameters in computer numerically controlled polishing of metal surfaces. Int. J. Manuf. Res., 17(3):310–325, 2022.

[91] J.C.V. Gastelum. Doctoral dissertation. Purdue University Graduate School, 2023.

[92] L. Chen et al. Generative models for inverse design of inorganic solid materials. J. Mater. Inform, 4:1, 2021.

[93] M.R. Dobbelaere et al. Machine learning in chemical engineering: strengths, weaknesses, opportunities, and threats. Engineering, 7(9):1201–1211, 2021.

[94] V. Venkatasubramanian. The promise of artificial intelligence in chemical engineering: is it here, finally? AIChE J, 65(2):466–478, 2019.

[95] T. Zhou et al. Big data creates new opportunities for materials research: a review on methods and applications of machine learning for materials design. Engineering, 5(6):1017–1026, 2019.

[96] W. Chen et al. Data centric design: a new approach to design of microstructural material systems. Engineering, 10:89–98, 2022.

[97] A. Thebelt et al. Maximizing information from chemical engineering data sets: applications to machine learning. Chem Eng Sci, 252:117469, 2022.

[98] Zdeborova and Lenka. Machine learning: New tool in the box. Nat. Phys., 13: 420, 2017.

[99] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborova. Machine learning and the physical sciences. Rev. Mod. Phys., 91:045002, 2019.

[100] Roger Guimera, Ignasi Reichardt, Antoni Aguilar-Mogas, Francesco A. Massucci, Manuel Miranda, Jordi Pallares, and Marta Sales-Pardo. Decoding the development of scientific knowledge. Sci. Adv., 6:eaav6971, 2020.

[101] Elena Agliari, Adriano Barra, Peter Sollich, and Lenka Zdeborova. The physics of learning: A field theory perspective. J. Phys. A: Math. Theor., 53:500401, 2020.

[102] Georgia Karagiorgi, Gregor Kasieczka, Scott Kravitz, Benjamin Nachman, and David Shih. Machine learning in the physical sciences. Nat. Rev. Phys., 4:399, 2022.

[103] Anuj Karpatne, Imme Ebert-Uphoff, Sai Ravela, Hassan Ali Babaie, and Vipin Kumar. Theory-guided data science: A new paradigm for scientific discovery from data. IEEE Trans. Knowl. Data Eng., 31:1544, 2018.

[104] Nicola Jones. How machine learning could help to improve weather forecasts. Nature, 548:379, 2017.

[105] Prasanna V. Balachandran. Adaptive strategies for materials design using machine learning. Comput. Mater. Sci., 164:82, 2019.

[106] Keith T. Butler, Daniel W. Davies, Hugh Cartwright, Olexandr Isayev, and Aron Walsh. Machine learning for molecular and materials science. Nature, 559:547, 2018.

[107] Zoltan Haiman. The promise of machine learning in astronomy. Nat. Astron., 3: 18, 2019.

[108] Silviu-Marian Udrescu and Max Tegmark. Ai feynman: A physics-inspired method for symbolic regression. Sci. Adv., 6:eaay2631, 2020.

[109] Ziming Liu and Max Tegmark. Machine learning conservation laws from trajectories. Phys. Rev. Lett., 126:180604, 2021.

[110] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc. Natl. Acad. Sci. U.S.A., 113:3932, 2016.

[111] He Li et al. Deep-learning density functional perturbation theory. Phys. Rev. Lett., 132:096401, 2024. doi:10.1103/PhysRevLett.132.096401.

[112] A. Bhowmik et al. A. equivariant graph neural networks for fast electron density estimation of molecules, liquids, and solids. npj Comput Mater, 8:183, 2022. doi: 10.1038/s41524-022-00863-y.

[113] B. Focassio et al. Linear jacobi-legendre expansion of the charge density for machine learning-accelerated electronic structure calculations. npj Comput Mater, 9:87, 2023. doi: 10.1038/s41524-023-01053-0.

[114] Liwei Zhang et al. A symmetry-preserving and transferable representation for learning the kohn-sham density matrix. arXiv, 2025. doi: https://doi.org/10.48550/arXiv.2503.08400.

[115] Xuecheng Shao et al. Machine learning electronic structure methods based on the one-electron reduced density matrix. Nat. Commun., 14:6281, 2023. doi: https://doi.org/10.1038/s41467-023-41953-9.

[116] Jiang Wu et al. Construct exchange-correlation functional via machine learning. J. Chem. Phys., 159:090901, 2023.

[117] Yun-Pei Liu et al. Catflow: An automated workflow for training machine learning potentials to compute free energies in dynamic catalysis. The Journal of Physical Chemistry C, 129:1089–1102, 2024.

[118] J.Behle et al. Generalized neural-network representation of high-dimensional potential-energy surfaces. Phys. Rev. Lett., 98:146401, 2007.

[119] Sascha Klawohn et al. Gaussian approximation potentials: Theory, software implementation and application examples. J. Chem. Phys, 159:174108, 2023.

[120] Albert P. Bartok et al. On representing chemical environments. Phys. Rev. B, 96: 019902, 2017.

[121] Kristof T. Schuttand others. A continuous-filter convolutional neural network for modeling quantum interactions. arXiv, 2017.

[122] Jinzhe Zeng et al. Deepmd-kit v2: A software package for deep potential models. J. Chem. Phys, 159:054801, 2023. doi: https://doi.org/10.1063/5.0155600.

[123] M. Pavanello et al. Machine learning electronic structure methods based on the one-electron reduced density matrix. Nature Communications, 14:6281, 2023. doi:https: 10.1038/s41467-023-41953-9.

[124] Felix Brockherde et al. Bypassing the kohn-sham equations with machine learning. Nat. Commun., 8:872, 2017. doi: https://doi.org/10.1038/s41467-023-41953-9.

[125] Beatriz G. del Rio et al. A deep learning framework to emulate density functional theory. npj Comput Mater, 9:158, 2023. doi: https://doi.org/10.1038/s41524-023-01115-3.

[126] Walter Hugo Lopez Pinaya et al. Chapter 10 - convolutional neural networks. Machine Learning, pages 173–191, 2020. doi: https://doi.org/10.1016/B978-0-12-815739-8.00010-9.

[127] Yibin Wu et al. Simple and efficient equivariant message-passing neural network model for non-local potential energy surfaces. J. Phys. Chem. A, 128:11061–11067, 2024. doi: https://doi.org/10.1021/acs.jpca.4c06669.

[128] Anders S. Christensen et al. Orbnet denali: A machine learning potential for biological and organic chemistry with semi-empirical cost and dft accuracy. arXiv, 2021. doi: https://doi.org/10.1063/5.0061990.

[129] Ralf Meyer et al. Machine learning approaches toward orbital-free density functional theory: Simultaneous training on the kinetic energy density functional and its functional derivative. Journal of Chemical Theory and Computation, 16:5685–5694, 2020. doi: https://doi.org/10.1021/acs.jctc.0c00580.

[130] Alexander Ryabov et al. Neural network interpolation of exchange-correlation functional. Sci Rep, 10:8000, 2020. doi: https://doi.org/10.1038/s41598-020-64619-8.

[131] Zhandos Moldabekov et al. Ab initio static exchange–correlation kernel across jacob's ladder without functional derivatives. J. Chem. Theory Comput., 19:1286–1299, 2023. doi: https://doi.org/10.1021/acs.jctc.2c01180.

[132] Liang Sun et al. Machine learning based nonlocal kinetic energy density functional for simple metals and alloys. Phys. Rev. B, 109:115135, 2024. doi: DOI: https://doi.org/10.1103/PhysRevB.109.115135.

[133] Kanun Pokharel et al. Exact constraints and appropriate norms in machine-learned exchange-correlation functionals. J. Chem. Phys., 157:174106, 2022. doi: https://doi.org/10.1063/5.0111183.

[134] Sijia S. Dong et al. Machine learning dielectric screening for the simulation of excited state properties of molecules and materials. Chem. Sci., 12:4970–4980, 2021. doi: https://doi.org/10.1039/D1SC00503K.

[135] Giovanni Pizzi, Andrea Cepellotti, Riccardo Sabatini, Nicola Marzari, and Boris Kozinsky. Aiida: automated interactive infrastructure and database for computational science. Computational Materials Science, 111:218–230, 2016. doi: 10.1016/j.commatsci.2015.09.013.

[136] Leopold Talirz, Snehal Kumbhar, Elsa Passaro, Aliaksandr V. Yakutovich, Valeria Granata, Fernando Gargiulo, Marco Borelli, Martin Uhrin, Sebastiaan P. Huber, Spyros Zoupanos, Carl S. Adorf, Casper W. Andersen, Ole Schütt, Carlo A. Pignedoli, Daniele Passerone, Joost VandeVondele, Thomas C. Schulthess, Berend Smit, Giovanni Pizzi, and Nicola Marzari. Materials cloud, a platform for open computational science. Scientific Data, 7:299, 2020. doi: 10.1038/s41597-020-00637-5.

[137] David Montes de Oca Zapiain et al. Training data selection for accuracy and transferability of interatomic potentials. npj Computational Materials, 8:189, 2022. doi: https://doi.org/10.1038/s41524-022-00872-x.

[138] Stephen L. Mayo, Barry D. Olafson, and A. William. Dreiding: A generic force field for molecular simulations. J. Phys. Chem, 94:8897–8909, 1990.

[139] Christopher M. Baker. Polarizable force fields for molecular dynamics simula-

tions of biomolecules. WIREs Computational Molecular Science, 5:241–254, 2015. DOI: 10.1038/s41524-024-01427-y.

[140] G. Kresse and J. Hafner. Ab initio molecular dynamics for liquid metals. Phys. Rev. B, 47:1–1, 1993. DOI: 10.1103/PhysRevB.47.558.

[141] I-Chun Lin, Ari P. Seitsonen, I. Tavernelli, and U. Rothlisberger. Structure and dynamics of liquid water from ab initio molecular dynamics—comparison of blyp, pbe, and revpbe density functionals with and without van der waals corrections. J. Chem. Theory Comput, 8:3902–3910, 2012. doi: /10.1021/ct3001848.

[142] S. M. Blinder. Chapter 1 - introduction to the hartree-fock method. Developments in Physical & Theoretical Chemistry, Mathematical Physics in Theoretical Chemistry, pages 1–30, 2019. doi: 10.1016/B978-0-12-813651-5.00001-2.

[143] P. Echenique and J. L. Alonso. A mathematical and computational review of hartree–fock scf methods in quantum chemistry. Molecular Physics, 105(23-24): 3057–3098, 2007. doi: 10.1080/00268970701757875.

[144] A. Szabo and N. S. Ostlund. Modern quantum chemistry: Introduction to advanced electronic structure theory. Dover Publications, 1996.

[145] T Helgaker et al. Molecular electronic-structure theory. Wiley, 2000.

[146] I. Shavitt and R. J. Bartlett. Many-body methods in chemistry and physics: Mbpt and coupled-cluster theory. Cambridge University Press, 2009.

[147] Christoph Freysoldt et al. First-principles calculations for point defects in solids. Rev. Mod. Phys., 86:253–305, 2014. doi: DOI:https://doi.org/10.1103/RevModPhys.86.253.

[148] J.K.N et al. Density functional theory in surface science and heterogeneous catalysis. Density Functional Theory in Materials Research (Cambridge University Press), 31:669–674, 2006. doi: DOI:https://doi.org/10.1557/mrs2006.175.

[149] Qiu He et al. Density functional theory for battery materials. Energy Environ. Mater., 2:264–279, 2019. doi: https://doi.org/10.1002/eem2.12056.

[150] A Zunger. Bridging the gap between density functional theory and quantum materials. Nat Comput Sci., 2:529–532, 2022. doi: https://doi.org/10.1038/s43588-022-00323-z.

[151] Susi Lehtola et al. An overview of self-consistent field calculations within finite basis sets. Molecules, 2020. doi:10.3390/molecules25051218.

[152] M. J. T. Oliveira et al. Libxc: A library of exchange and correlation functionals for density functional theory. Comput. Phys. Commun., 256:107647, 2020. doi:10.1016/j.cpc.2020.107647.

[153] Luis R. Kahn, Paul Baybutt, and Donald G. Truhlar. Ab initio effective core potentials: Reduction of all-electron molecular structure calculations to calculations involving only valence electron. J. Chem. Phys, 65:3826–3853, 1976. doi: 10.1063/1.432900.

[154] Brett M. Bode and Mark S. Gordon. Brett m. bode, mark s. gordon; fast computation of analytical second derivatives with effective core potentials: Application si_8c_{12}, ge_8c_{12}, and sn_8c_{12}. J. Chem. Phys, 111:8778–8784, 1999. doi: 10.1063/1.480225.

[155] Larry E McMurchie and Ernest R Davidson. Calculation of integrals over ab initio pseudopotentials. Journal of Computational Physics, 44:289–301, 1981. doi: 10.1016/0021-9991(81)90053-X.

[156] R Flores-Moreno et al. Half-numerical evaluation of pseudopotential integrals. J. Comput. Chem., 27:0192–8651, 2006. doi: 10.1002/jcc.20410.

[157] Chris-Kriton Skylaris et al. An efficient method for calculating effective core potential integrals which involve projection operators. Chemical Physics Letters, 296:445–451, 1998. doi: 10.1016/S0009-2614(98)01077-X.

[158] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors, volume 323. Nature Publishing Group, 1986. doi: 10.1038/323533a0.

[159] T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, 2nd edition, 2009. ISBN 978-0-387-84857-0.

[160] S. Sanvito M. Domina, U. Patil. Cluster expansion constructed over jacobi-legendre polynomials for accurate force fields. Phys. Rev. B, 108:094102, 2023. DOI:10.1103/PhysRevB.108.094102.

[161] M. Domina. The jacobi-legendre framework for machine learning in materials investigation and discovery. Trinity College Dublin, School of Physics, Physics, 2024.

[162] Andrews. G et al. Special functions. cambridge university press. Special Functions. Cambridge University Press, 1999.

[163] F. Pedregosa et al. Scikit-learn: Machine learning in python. J. Mach. Learn. Res, 12:2825–2830, 2011.

[164] C.C.J. Roothan. New developments in molecular orbital theory. Rev. Mod. Phys., 23:69–89, 1951.

[165] Nathaniel Thomas et al. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds. arXiv, 2018. doi:10.48550/arXiv.1802.08219.

[166] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. International Conference on Learning Representations (ICLR), 2015. arXiv:1412.6980.

[167] Thom H. Dunning and Jr. Gaussian basis sets for use in correlated molecular

calculations. i. the atoms boron through neon and hydrogen. J. Chem. Phys., 90: 1007–1023, 1989. doi:10.1063/1.456153.

[168] D. J. Evans and B. L. Holian. The nose–hoover thermostat. J. Chem. Phys., 83: 4069–4074, 1985. doi: 10.1063/1.449071.

[169] A. Droghetti et al. Assessment of density functional theory for iron(ii) molecules across the spin-crossover transition. Journal of Chemical Physics, 137(12):124303, 2012. doi:10.1063/1.4752411.

[170] Alex Domingo et al. Spin crossover in fe(ii) complexes: An ab initio study of ligand σ-donation. International Journal of Quantum Chemistry, 110(2):331–337, 2010. doi: 10.1002/qua.22105.

[171] J. Harris. Simplified method for calculating the energy of weakly interacting fragments. Physical Review B, 31(4):1770–1779, 1985. doi: 10.1103/PhysRevB.31.1770.

[172] F. W. Averill and G. S. Painter. Harris functional and related methods for calculating total energies in density-functional theory. Physical Review B, 41(15): 10344–10363, 1990. doi: 10.1103/PhysRevB.41.10344.

[173] Ralf Drautz. Atomic cluster expansion for accurate and transferable interatomic potentials. Phys. Rev. B, 99:014104, 2019. doi:10.1103/PhysRevB.99.014104.

[174] Alexander V. Shapeev. Moment tensor potentials: A class of systematically improvable interatomic potentials. Multiscale Model. Simul., 14:1153– 1173, 2016. doi:10.1137/15M1054183.

[175] NIST Standard Reference Database and Number Release: 18. Nist computational chemistry comparisons and benchmark database. Johnson, R. D. Ed, NIST: Gaithersburg, MD, 2016.

[176] S Lehtola. Assessment of initial guesses for self-consistent field calculations. superposition of atomic potentials: Simple yet efficient. Journal of Chemical Theory and Computation, 15(3):1593–1604, 2016. doi:10.1021/acs.jctc.8b01089.

[177] S. Hazra et al. Predicting the one-particle density matrix with machine learning. Journal of Chemical Theory and Computation, 20:4569–4578, 2024. doi: 10.1021/acs.jctc.4c00042.

[178] Yuping He et al. Metallic metal-organic frameworks predicted by the combination of machine learning methods and ab initio calculations. The Journal of Physical Chemistry Letters, pages 4562–4569, 2018. 10.1021/acs.jpclett.8b01707.

[179] Sam Schoot, Uday Chopra, et al. Polaron spin dynamics in high-mobility polymeric semiconductors. Nat. Phys, 15:814–822, 2019. DOI:0.1038/s41567-019-0538-0.

[180] V. Bapst, T. Keck, et al. Unveiling the predictive power of static structure in glassy systems. Nat. Phys, 16:448–454, 2020. DOI:10.1038/s41567-020-0842-8.

[181] van Gunsteren and Wilfred F et al. Computer simulation of molecular dynamics: Methodology, applications, and perspectives in chemistry. Angew. Chem. Int. Ed. Engl, 29:992–1023, 1990. DOI:10.1002/anie.199009921.

[182] E. Guardia, I. Skarmoutsos, and M. Marco. Hydrogen bonding and related properties in liquid water: A car–parrinello molecular dynamics simulation study. J. Phys. Chem. B, 119:8926–8938, 2015. DOI:10.1021/jp507196q.

[183] Hoi Ling Luk, Johannes Feist, J. Jussi Toppari, and Gerrit Groenhof. Multiscale molecular dynamics simulations of polaritonic chemistry. J. Chem. Theory Comput, 13:4324–4335, 2017. DOI:10.1021/acs.jctc.7b00388.

[184] B. Ghosh and N. Sengupta. The protein hydration layer in high glucose concentration: Dynamical responses in folded and intrinsically disordered dimeric states. Biochemical and Biophysical Research Communications, 577:124–129, 2021. DOI:10.1016/j.bbrc.2021.09.005.

[185] M. Karplus and G. Petsko. Molecular dynamics simulations in biology. Nature, 347:631–639, 1990. DOI:10.1038/347631a0.

[186] A. Ganesan, Michelle L. Coote, and K. Barakat. Molecular dynamics-driven drug discovery: leaping forward with confidence. Drug Discovery Today, 22:249–269, 2017. DOI:10.1016/j.drudis.2016.11.001.

[187] M. Vivo, M Masetti, G Bottegoni, and A. Cavalli. Role of molecular dynamics and related methods in drug discovery. J. Med. Chem, 59:4035–4061, 2016. DOI:10.1021/acs.jmedchem.5b01684.

[188] T. Mueller, A. Hernandez, and C. Wang. Machine learning for interatomic potential models. J. Chem. Phys, 152:050902, 2020. DOI:10.1063/1.5126336.

[189] Fabian L Thiemann and V. K et al. Introduction to machine learning potentials for atomistic simulations. arXiv, 2024. DOI:10.48550/arXiv.2410.00626.

[190] A. Kabylda and V. V. Galindo et al. Efficient interatomic descriptors for accurate machine learning force fields of extended molecules. Nat. Commun, 14:3562, 2022. DOI:10.1038/s41467-023-39214-w.

[191] A. Musaelian and S. Batzner et al. Learning local equivariant representations for large-scale atomistic dynamic. Nat. Commun, 14:579, 2023. DOI:10.1038/s41467-023-36329-y.

[192] R. M. Barrios and E. N. Coneydo et al. An overview about neural networks potentials in molecular dynamics simulation. Int. J. Quantum Chem, 124:0020–7608, 2024. DOI:10.1002/qua.27389.

[193] J. M. soler et al. The siesta method for ab initio order-n materials simulation. J. Phys.: Condens. Matter, 14:2745, 2002. doi: 10.1088/0953-8984/14/11/302.

[194] G. Kresse et al. Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. Computational Materials Science, 6 (1):15–50, 1996. doi: 10.1016/0927-0256(96)00008-0.

[195] G. Kresse et al. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. Phys. Rev. B, 54 (16):11169–11186, 1996. doi: 10.1103/PhysRevB.54.11169.

[196] Dmitri N. Laikov. Fast evaluation of density functional exchange-correlation terms using the expansion of the electron density in auxiliary basis sets. Chemical Physics Letters, 281:151–156, 1997. doi:10.1016/S0009-2614(97)01206-2.

[197] L. Gagliardi et al. On the resolution of identity coulomb energy approximation in density functional theory. Journal of Molecular Structure: THEOCHEM, 501–502:229–239, 2000. doi:10.1016/S0166-1280(99)00434-0.

[198] Edward F. Valeev and Curtis L. Janssen. Libint library for computing molecular integrals over gaussian functions. `https://github.com/evaleev/libint`, 2004. Reference for Libint integral evaluation library; available via libint documentation.

[199] Daniel G. A. Smith, Lori A. Burns, Andrew C. Simmonett, Robert M. Parrish, Michael C. Schieber, Raimondas Galvelis, Peter Kraus, Holger Kruse, Roberto Di Remigio, Abdullah Alenaizan, Andrew M. James, Susi Lehtola, Jonathan P. Misiewicz, Maximilian Scheurer, Robert A. Shaw, Jeffrey B. Schriber, Yu Xie, Zachary L. Glick, Dominic A. Sirianni, James S. O'Brien, Jonathan M. Waldrop, Ashutosh Kumar, Edward G. Hohenstein, Benjamin P. Pritchard, Bernard R. Brooks, Henry F. Schaefer, Alexander Yu. Sokolov, Konrad Patkowski, A. Eugene DePrince, Uğur Bozkaya, Rollin A. King, Francesco A. Evangelista, T. Daniel Crawford, Cesare Pisani, and C. David Sherrill. Psi4 1.4: Open-source software for high-throughput quantum chemistry. Journal of Chemical Physics, 152(18): 184108, 2020. doi: 10.1063/5.0006002.

[200] Michael W. Schmidt, Kim K. Baldridge, Jerry A. Boatz, Steven T. Elbert, Mark S. Gordon, Jan H. Jensen, Shiro Koseki, Nikita Matsunaga, Kiet A. Nguyen, Shujun Su, Theresa L. Windus, Michel Dupuis, and John A. Montgomery. General

atomic and molecular electronic structure system. Journal of Computational Chemistry, 14(11):1347–1363, 1993. doi: 10.1002/jcc.540141112.

[201] Qiming Sun. Libcint: An efficient general integral library for gaussian basis functions. Journal of Computational Chemistry, 36(22):1664–1671, 2015. doi: 10.1002/jcc.23981.

[202] J. L. Whitten. Coulombic potential energy integrals and approximations. Journal of Chemical Physics, 58:4496, 1973. doi: 10.1063/1.1679012.

[203] F. Weigend, A. Köhn, and C. Hättig. Efficient use of the correlation consistent basis sets in resolution of the identity mp2 calculations. Journal of Chemical Physics, 116:3175, 2002. doi: 10.1063/1.1445115.

[204] Justin T. Fermann and Edward F. Valeev. Fundamentals of molecular integrals evaluation. arXiv, 2020. DOI: arXiv:2007.12057.

[205] Pauli Virtanen et al. Scipy 1.0: Fundamental algorithms for scientific computing in python. nature methods. Nature Methods, 17:261–272, 2020. doi:10.1038/s41592-019-0686-2.

[206] Timothy Clark et al. Efficient diffuse function-augmented basis sets for anion calculations. iii. the 3-21+g basis set for first-row elements, li–f. Journal of Computational Chemistry, 4(3):294–301, 1983. doi:10.1002/jcc.540040303.

[207] P. C. Hariharan and J. A. Pople. The influence of polarization functions on molecular orbital hydrogenation energies. Theoretical Chemistry Accounts, 28(3):213–222, 1973. doi:10.1007/BF00533485.

[208] Marie-Pierre Gaigeot and Michiel Sprik. Ab initio molecular dynamics computation of the infrared spectrum of aqueous uracil. J. Phys. Chem. B, 107 (38): 10344–10358, 2003. doi:doi.org/10.1021/jp034788u.

[209] Peter H. Berens and Kent R. Wilson. Molecular dynamics and spectra. i. diatomic rotation and vibration. J. Chern. Phys., 74(9:4872–4882, 1981. doi: 10.1063/1.441739.

[210] A. D. becke. Density-functional exchange-energy approximation with correct asymptotic behavior. Phys. Rev. A, 38:3098, 1981. doi: 10.1063/1.441739.

[211] Fiona Sim et al. Gaussian density functional calculations on the allyl and polyene radicals: c_3h_5 to $c_{11}h_{13}$. J. Chem. Phys., 95:4317–4326, 1991. doi:/10.1063/1.461755.

[212] Willis B Person and Giuseppe Zerbi. Vibrational intensities in infrared and raman spectroscopy. Amsterdam, 1982.

[213] RG. Lee and YH. Kim. Convolutional network learning of self-consistent electron density via grid-projected atomic fingerprints. npj Comput Mater, 10:248, 2024. DOI: 10.1038/s41524-024-01433-0.

A1 | Appendix

A1.1 Coulomb Matrix Integral

From Eq.(40) we get

$$\begin{aligned} I'(\mathbf{r}') &= \int_0^\infty \frac{s^{L-2L'} s^{-1/2} e^{-\frac{\gamma s|\mathbf{r}'-\mathbf{P}|^2}{(\gamma+s)}}}{(\gamma+s)^{L-L'}(\gamma+s)^{3/2}} ds = \\ &= \int_0^\infty \left(\frac{s}{\gamma+s}\right)^{L-L^{'}} \frac{s^{-L^{'}} s^{-1/2}}{(\gamma+s)^{3/2}} e^{-\frac{\gamma s|\mathbf{r}'-\mathbf{P}|^2}{(\gamma+s)^{3/2}}} ds \end{aligned} \tag{1}$$

Now by using $t^2 = \frac{s}{(s+\gamma)}$, we get $ds = (\gamma+s)^{3/2} s^{1/2} \times \frac{2}{\gamma} dt$, hence integration limit will change to 0 to 1, and the expression for s goes to $s = \frac{\gamma t^2}{(1-t^2)}$, then we can write from Eq.(1)

$$I^{'} = \frac{2}{\gamma^{(L^{'}+1)}} \int_0^1 t^{2(L-2L^{'})} (1-t^2)^{L^{'}} e^{\gamma t^2 (\mathbf{r}^{'}-\mathbf{P}_r)^2} dt\ . \tag{2}$$

Again, by using the binomial expansion, the term $(1-t^2)^{L^{'}}$ can be expanded as

$$(1-t^2)^{L^{'}} = \sum_{h=0}^{L^{'}} (-1)^h C(L^{'}, h) t^{2h}\ . \tag{3}$$

Hence

$$
\begin{aligned}
I^{'} &= \sum_{h=0}^{L^{'}} \frac{2(-1)^h}{\gamma^{L^{'}+1}} C(L^{'},h) \int_0^1 t^{2(L-2L^{'}+h)} e^{-\gamma t^2 (r^{'}-P_r)^2} dt \\
&= \sum_{h=0}^{L^{'}} \frac{2(-1)^h}{\gamma^{L^{'}+1}} C(L^{'},h) \left[\frac{\int_0^{\gamma(r^{'}-P_r)^2} u^{[(L-2L^{'}+h)+\frac{1}{2}]-1} e^{-u} du}{2\gamma^{(L-2L^{'}+h+\frac{1}{2})} |r^{'} - P_r|^{2(L-2L^{'}+h)+1}} \right] \\
&= \sum_{h=0}^{L^{'}} \frac{(-1)^h}{\gamma^{L-L^{'}+h+1.5}} \times C(L^{'},h) \times \\
&\quad \frac{\Gamma\left[L-2L^{'}+h+0.5\right] \Gamma_{inc}\left[(L-2L^{'}+h+0.5), \gamma(r^{'}-P_r)^2\right]}{|r^{'}-P_r|^{2(L-2L^{'}+h)+1}}
\end{aligned}
\tag{4}
$$

Here we made a substitution $\gamma t^2(\mathbf{r}^{'} - \mathbf{P}_r)^2 = u$ and hence integration limit has been changed between 0 to $\gamma(r^{'} - P_r)^2$. Thereafter the standard integral form of incomplete gamma used in Eq.(4), where the incomplete gamma function Γ_{inc} (205) can be defined as

$$
\Gamma_{inc}(a,x) = \frac{1}{\Gamma(a)} \int_0^x t^{a-1} e^{-t} dt \tag{5}
$$